



**GUSTAVO REIS SILVA**

**RELATÓRIO DE ESTÁGIO:  
DESENVOLVIMENTO WEB NA CREDITAS**

**LAVRAS – MG**

**2020**



**GUSTAVO REIS SILVA**

**RELATÓRIO DE ESTÁGIO:  
DESENVOLVIMENTO WEB NA CREDITAS**

Relatório de Estágio 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. Maurício Ronny de Almeida Souza  
Orientador

**LAVRAS – MG**

**2020**

**GUSTAVO REIS SILVA**

**RELATÓRIO DE ESTÁGIO: DESENVOLVIMENTO WEB NA CREDITAS**  
**INTERNSHIP REPORT: WEB DEVELOPMENT AT CREDITAS**

Relatório de Estágio 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 25 de Agosto de 2020.

Prof. DSc. Antônio Maria Pereira de Resende UFLA  
Vinicius Alves Oyama Credits

  
Prof. DSc. Maurício Ronny de Almeida Souza  
Orientador

**LAVRAS – MG**  
**2020**

*Dedico este trabalho a minha mãe, por tudo.*



## **AGRADECIMENTOS**

Agradeço principalmente meu pai, por ter me dado forças quando eu mais queria desistir.

Agradeço a minha irmã, por ser minha inspiração.

Agradeço aos meus amigos, em especial: Jean Roberto, Bruno Amaral, Leonardo Campelo e Guilherme Soares, por serem minha casa em Lavras.

À Comp Júnior, que me ensinou a ir além e fazer acontecer.

Agradeço a Universidade Federal de Lavras, pelas oportunidades.

Por fim, agradeço meu professor orientador Maurício Ronny por me manter em curso durante a escrita deste trabalho.



## RESUMO

Este documento apresenta as principais atividades realizadas durante o processo de estágio em desenvolvimento de software, centrado em web, na Credits. Será descrita uma introdução à organização, as tecnologias e conceitos estudados pelo estagiário, e os trabalhos executados, que sumariamente consistem em um treinamento e no desenvolvimento de artefatos de software para um produto da companhia, em uma das equipes de desenvolvimento. O intuito deste trabalho é apresentar as atividades realizadas durante o estágio na empresa Credits tratando temas do escopo do Bacharelado em Ciência da Computação.

**Palavras-chave:** Credits. Web. Back-end.



## **ABSTRACT**

This document presents the main activities done during the software engineering internship process, focused on web development, at Creditas. It will describe an introduction to the organization, the technologies and concepts studied by the intern, and the tasks fulfilled, that summarily consists on a training and at the development of software artifacts for one of the company products, with one of the development teams. The goal of this paper is to show the carried out during the internship at Creditas, addressing topics within the scope of the Bachelor of Computer Science.

**Keywords:** Creditas. Web. Back-end.



## LISTA DE FIGURAS

Figura 2.1 – Modelo Spotify . . . . .	20
Figura 4.1 – Representação do Projeto “Website Estático” . . . . .	30
Figura 4.2 – Representação do Projeto “Calculadora para o Navegador” . .	31
Figura 4.3 – Representação do Tutorial “Sistema de Rastreamento de Peso”	33
Figura 4.4 – Representação do Projeto “MVC” . . . . .	34



## SUMÁRIO

<b>1</b>	<b>Introdução</b>	15
<b>1.1</b>	<b>Objetivos do estágio</b>	15
<b>1.2</b>	<b>Organização do trabalho</b>	16
<b>2</b>	<b>Sobre a organização</b>	17
<b>2.1</b>	<b>A Credits</b>	17
<b>2.2</b>	<b>O processo organizacional</b>	18
<b>2.2.1</b>	<b>Modelo Spotify</b>	18
<b>3</b>	<b>Tecnologias</b>	21
<b>3.1</b>	<b>Desenvolvimento WEB</b>	21
<b>3.1.1</b>	<b>Tecnologias para o <i>front-end</i></b>	21
<b>3.1.2</b>	<b>Tecnologias para o back-end</b>	22
<b>3.2</b>	<b>Gerência de Configuração</b>	23
<b>3.3</b>	<b>Metodologias e Tecnicas de Desenvolvimento</b>	24
<b>3.3.1</b>	<b>Padrões de Projeto</b>	24
<b>3.3.2</b>	<b>SOLID</b>	24
<b>3.3.3</b>	<b>Domain Driven Design</b>	25
<b>3.3.4</b>	<b>Testes</b>	25
<b>3.3.5</b>	<b>Test Driven Development (TDD)</b>	26
<b>3.4</b>	<b>Banco de Dados</b>	26
<b>4</b>	<b>Atividades Desenvolvidas</b>	29
<b>4.1</b>	<b>Treinamento</b>	29
<b>4.1.1</b>	<b>Módulo “Web Básico”</b>	29
<b>4.1.1.1</b>	<b>Projeto “Website Estático”</b>	30
<b>4.1.1.2</b>	<b>Projeto “Calculadora para o Navegador”</b>	30
<b>4.1.2</b>	<b>Módulo “Fullstack”</b>	31
<b>4.1.2.1</b>	<b>Projeto “Sistema de Funcionários”</b>	32
<b>4.1.2.2</b>	<b>Tutorial “Sistema de Rastreamento de Peso”</b>	32

<b>4.1.2.3</b>	<b>Projeto “MVC”</b>	<b>33</b>
<b>4.1.3</b>	<b>Módulo ”Especialização Backend“</b>	<b>34</b>
<b>4.2</b>	<b>Crédito Consignado Privado</b>	<b>35</b>
<b>4.3</b>	<b>Disciplinas do Bacharelado</b>	<b>37</b>
<b>4.4</b>	<b>Análise e Discussões</b>	<b>37</b>
<b>5</b>	<b>Conclusão</b>	<b>39</b>
	<b>REFERÊNCIAS</b>	<b>43</b>

## 1 INTRODUÇÃO

O curso de bacharelado em Ciência da Computação consiste em capacitar os alunos em habilidades referentes ao desenvolvimento de software, e como trabalho de conclusão de curso possibilita que o estudante transite por um estágio relacionado às disciplinas cursadas para complementar a grade curricular e disponibilizar ao aluno experiências que não seriam concebíveis dentro de uma sala de aula.

Assim, este documento escreve um relatório de estágio realizado pelo autor, de janeiro/2020 à julho/2020. O contexto do estágio englobou o trabalho com desenvolvimento de software, área fundamental do bacharelado em ciência da computação, com destaque em desenvolvimento *web*, que possibilitou ao estagiário trabalhar em projetos com uma complexidade maior do que o que era possível entregar nas disciplinas ofertadas durante o curso. Além disso, o estágio possui um caráter mais especialista, acarretando no aprofundamento dos conhecimentos da área pelo estagiário.

Portanto este documento propõe exibir as atividades realizadas durante o estágio em desenvolvimento de software na empresa Creditas, *fintech* atuante no mercado de crédito privado, que possuiu carga horária de trinta horas semanais, durante sete meses.

### 1.1 Objetivos do estágio

O objetivo do estágio foi preparar o estagiário para ingressar no mercado de trabalho de desenvolvimento de software e validar o conteúdo apresentado no bacharelado. Para cumprir estes critérios, foram realizados alguns projetos durante o estágio que podem ser divididos em duas etapas: (i) treinamento e (ii) produto de crédito consignado privado. Na primeira etapa, o estagiário executou individualmente seis projetos distintos para consolidar as habilidades necessárias para a atuação na empresa. Na segunda etapa, o estagiário ingressou em uma equipe

de desenvolvimento da empresa para trabalhar em um projeto para um dos produtos da empresa (consignado privado) e participar dos processos de software da empresa.

Já o objetivo deste texto é apresentar os resultados e conclusões do estágio de modo a sintetizar o que foi aprendido e fazer um paralelo com o curso de Ciência da Computação, visando abordar os pontos que podem trazer uma comunicação maior entre o bacharelado e o ingresso no mercado de trabalho.

## **1.2 Organização do trabalho**

Além deste capítulo introdutório, este trabalho foi organizado em quatro capítulos de modo a abranger os principais pontos do estágio:

- Capítulo 2 Organização: tem o objetivo de descrever a organização que o estagiário trabalhou.
- Capítulo 3 Tecnologias: possui o intuito de contextualizar o leitor quanto as tecnologias utilizadas no estágio.
- Capítulo 4 Atividades Desempenhadas: descreve as atividades desempenhadas pelo estagiário durante o curso do estágio.
- Capítulo 5 Conclusão: mira traçar as conclusões através de um paralelo entre o estágio e o bacharelado de Ciência da Computação.

## 2 SOBRE A ORGANIZAÇÃO

Este capítulo tem o intuito de descrever a organização em que o estágio foi efetuado, uma breve contextualização de sua história, assim como a área de atuação e objetivo. É contexto deste capítulo também descrever os produtos de software desenvolvidos pela companhia e a estruturação do setor responsável por software.

### 2.1 A Creditas

A Creditas (CREDITAS, ) é uma fintech de empréstimo com garantia brasileira, tem como objetivo "viabilizar suas novas conquistas, multiplicando o potencial de seus bens" e possui escritórios no Brasil, em São Paulo, Porto Alegre e Recife, além de um escritório na Espanha e um no México. Atua hoje no mercado de crédito, tendo como carro chefe três produtos principais:

- Crédito com garantia de imóvel (HOME, ): uma estratégia de empréstimo onde o cliente deixa um imóvel como garantia para conseguir taxas de juros mais baratas, por diminuir o risco da operação. Também é conhecido como *home equity* ou refinanciamento imobiliário;
- Crédito com garantia de veículo (AUTO, ): segue a mesma premissa do crédito com garantia de imóvel, porém a garantia do produto financeiro é um veículo. Também é conhecido como refinanciamento de veículo.
- Crédito consignado privado (PAYROLL, ): um modelo de empréstimo pessoal onde as parcelas são descontadas diretamente da folha de pagamento. Para ter acesso a este produto, a empresa em que a pessoa trabalha deve ser parceira da Creditas.

Fundada em 2012 sob o nome de BankFacil (CREDITAS, ), tinha como missão facilitar o acesso do brasileiro a melhores taxas e condições de crédito, era

uma empresa compradora de produtos financeiros, não atuando de ponta a ponta nos produtos servidos. Em 2013 fez a primeira indicação de crédito com garantia de imóvel para um dos *bancarizadores* parceiros e teve seu primeiro investimento captado. Em 2014 o projeto de crédito com garantia de veículo foi iniciado, através da indicação de novos clientes às instituições parceiras.

Em 2016, a organização passou a se responsabilizar por todo o processo de empréstimo (CREDITAS, ), de ponta a ponta e focar em apenas crédito com garantia, para mudar seu nome em 2017 e se consolidar como a principal fintech de crédito com garantia do país. Foi então que em 2019, comprou a Creditoo, empresa especializada em crédito consignado, e incluiu este produto em seu portfólio.

## **2.2 O processo organizacional**

A companhia é dividida em três grandes setores: Product-Technology, Business e Customer Journey. O setor Product-Technology, onde o estagiário atuava.

Product-technology é a área responsável por manter os produtos da empresa e pelo desenvolvimento de novos produtos. Por se tratar de uma empresa digital, os produtos são artefatos de software, como por exemplo, um aplicativo para acesso dos clientes ou uma calculadora para simulação de crédito, até produtos com menos visibilidade para o consumidor final, como um sistema de integração bancária ou um sistema interno para gerência dos clientes.

O modo como a área organiza as equipes de tecnologia segue o modelo Spotify (KNIBERG, 2014).

### **2.2.1 Modelo Spotify**

O modelo Spotify (KNIBERG, 2014), mostrado na figura 2.1, se caracteriza pelas equipes serem autônomas, isto é, cada equipe é responsável por um ou mais módulos do projeto, que podem ser construídos de maneira independente. Es-

tas equipes, conhecidas como *Squads*, são grupos multidisciplinares, geralmente com até 8 pessoas que é capaz de se organizar independentemente.

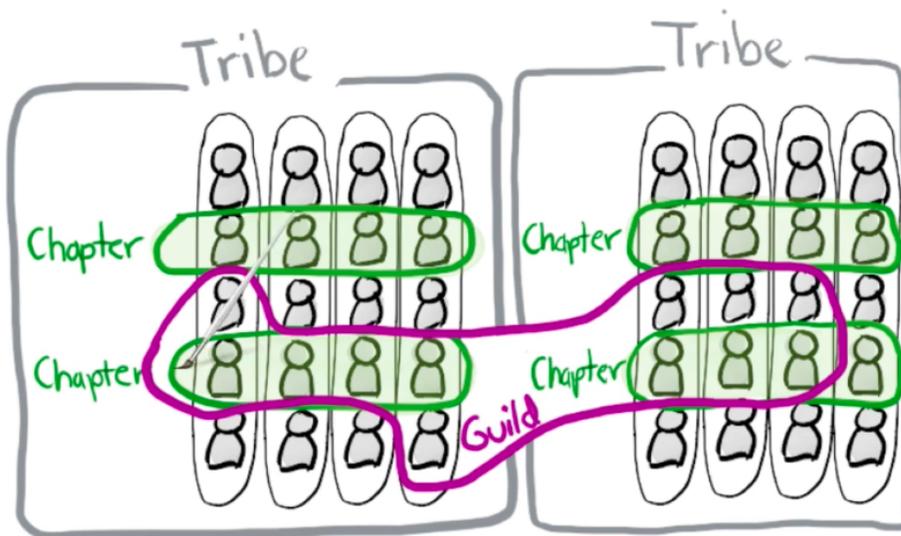
Um conjunto de *squads* que trabalha no mesmo produto é uma *Tribe*, a soma de todos os módulos desenvolvidos pelos *squads* se concretiza como o produto final da *Tribe*.

Além de fazer parte de um *squad*, um integrante da área também possui um *chapter* associado. O *chapter* é separação, dentro de cada *squad*, da área de atuação de um funcionário, exemplos de área de atuação seriam: *backend*, *frontend* ou *design* de produto. O ponto interessante desta separação por *chapter* se dá porque o gestor direto é um gestor do mesmo *chapter*, o que possibilita a mudança de *squad*, sem que seja necessário a mudança de gestor.

A última classificação deste modelo seriam as *Guilds*, que não possuem responsabilidade de negócio, podendo ser categorizado como um grupo de profissionais que se estende além das *Tribes* e tem por objetivo a troca de experiência e boas práticas com um tema em específico.

Vale ressaltar que este modelo segue uma abordagem de gestão ágil, mas não se limita a um *framework* em específico, ou seja, cada *squad* pode ter sua própria configuração e ferramentas de gestão e desenvolvimento. Porém em muitos casos, estas ferramentas podem se disseminar entre *squads* diferentes por se provarem boas soluções.

Figura 2.1 – Modelo Spotify



Fonte: Spotify engineering culture (part 1)(KNIBERG, 2014)

### 3 TECNOLOGIAS

Este capítulo tem como objetivo contextualizar as tecnologias e conceitos utilizados pelo estagiário. Somado a isto, também estão descritos processos utilizados no desenvolvimento de software aplicados durante o estágio.

#### 3.1 Desenvolvimento WEB

O desenvolvimento web é caracterizado pela criação de um software que é capaz de se conectar a uma rede de computadores, como por exemplo, a internet. Comumente é separado em três grandes áreas de atuação, sendo elas:

- *Fron-tend*: é classificado como a parte do software com que o usuário consegue interagir, isto inclui as telas exibidas no navegador e toda a interação provocada pelo contato da aplicação com o usuário;
- *Back-end*: é responsável por fazer a conexão entre o banco de dados e a aplicação front-end, mantendo a consistência dos dados e aplicando qualquer tipo de processamento necessário para manter a coerência do software em relação às regras de negócio.
- *Full-stack*: é especificado pela atuação nas duas áreas, *back-end* e *front-end*, simultaneamente.

Nas subseções a seguir serão detalhadas as tecnologias utilizadas no *front-end* e *back-end*.

##### 3.1.1 Tecnologias para o *front-end*

Para a estruturação do conteúdo das páginas desenvolvidas durante o estágio foi utilizada a linguagem de marcação HTML (HyperText Markup Language ou Linguagem de Marcação de Hipertexto), que é interpretada pelos navegadores, e caracterizada pelos comandos serem marcadores (tags) entre chevrons (<>),

sendo o final de uma tag representado por uma barra seguido de um chevron (`/>`) (W3, ).

Para estilizar as páginas, foi utilizada a linguagem Cascading Style Sheets ou Folhas de Estilo em Cascata (CSS), que é interpretada pelo navegador que renderiza os componentes da página de acordo com a descrição criada nos arquivos CSS do software (W3, ). Além disso, foi utilizada uma biblioteca de componentes visuais, o Bootstrap<sup>1</sup>, para simplificar o processo de criação das telas e aumentar a velocidade do desenvolvedor.

A linguagem de programação Javascript(CROCKFORD, 2008) foi utilizada para ter maior controle dos dados e ter a capacidade de manipulá-los no navegador possibilitando uma interação mais fluída entre a aplicação front-end e o usuário.

Visando a criação de páginas dinâmicas, que podem possuir conteúdo diferente dependendo dos dados atribuídos a ela, foi utilizada uma ferramenta de criação de templates com javascript, chamada de EJS<sup>2</sup>, capaz de embutir javascript em documentos HTML.

### 3.1.2 Tecnologias para o back-end

Durante o treinamento, para o desenvolvimento de aplicações back-end do módulo fullstack, e evitar a necessidade de o estagiário ter de aprender duas linguagens de programação ao mesmo tempo, foi utilizado o interpretador de Javascript Node.js<sup>3</sup>. Esta ferramenta é conhecida por ser voltada à programação assíncrona, tratando problemas de concorrência sem a necessidade de gerenciamento de threads.

---

<sup>1</sup> <https://getbootstrap.com>

<sup>2</sup> <https://ejs.co>

<sup>3</sup> <https://nodejs.org/en/>

Uma das ferramentas utilizadas foi o *framework* Hapi.js<sup>4</sup>, desenvolvido em Node.js, é capaz de criar aplicações web para o servidor e possui ferramentas para auxiliar o desenvolvedor nas atividades mais comuns ao se trabalhar com um servidor web, como gerenciamento das requisições HTTP e roteamento. A ferramenta possui um recurso chamado de plugins, utilizado para aumentar as funcionalidades da mesma através de código customizado pelo usuário do *framework*.

Para a criação de uma api REST (*Representational State Transfer* ou *Transferência Representacional de Estado*) foi utilizada a linguagem de programação Kotlin(JEMEROV; ISAKOVA, 2017) e o *framework* Spring<sup>5</sup> e em um dos projetos da empresa a linguagem utilizada foi PHP, com o *framework* Laravel<sup>6</sup>.

### 3.2 Gerência de Configuração

Gerência de configuração é uma área de estudos da engenharia de software caracterizada por estabelecer ferramentas de controle de mudanças e auditoria durante o processo de desenvolvimento (BOURQUE; FAIRLEY et al., 2014).

No decorrer do processo de estágio foram utilizadas duas ferramentas referentes à gerência citada:

- Git<sup>7</sup>: software livre de controle de versão de arquivos. O sistema funciona de maneira descentralizada, isto é, podem existir cópias diferentes de um mesmo repositório em diferentes computadores, o que permite que um grupo trabalhe em um mesmo software simultaneamente;
- Gitlab<sup>8</sup>: é uma plataforma para compartilhamento e hospedagem de projetos que possibilita a utilização da ferramenta git de maneira remota, sim-

---

<sup>4</sup> <https://hapi.dev>

<sup>5</sup> <https://spring.io>

<sup>6</sup> <https://laravel.com>

<sup>7</sup> <https://git-scm.com>

<sup>8</sup> <https://gitlab.com>

plificando a gestão dos projetos e facilitando o trabalho coletivo sobre um mesmo software.

### **3.3 Metodologias e Técnicas de Desenvolvimento**

Esta sessão propõe descrever sobre padrões de projeto, além de processos de desenvolvimento de software utilizados durante a realização do estágio.

#### **3.3.1 Padrões de Projeto**

Um padrão de projeto é uma solução de software para um problema recorrente e específico no desenvolvimento de software. Os padrões de projeto são considerados soluções fundamentadas por terem sido testadas em situações reais e entregarem resultados consistentes mesmo após uso extenso (FREEMAN et al., 2008).

O estagiário entrou em contato com alguns padrões para elucidar situações encontradas no desenvolvimento dos projetos:

- Facade: padrão de projeto que tem o intuito de simplificar a complexidade de um sistema em uma interface;
- Factory Method: padrão responsável por encapsular a lógica de instanciação de objetos em um método específico;
- Adapter: tem o objetivo de fazer a adaptação de um modelo de dados para outro, possibilitando a comunicação entre sistemas com uma modelagem diferente.

#### **3.3.2 SOLID**

SOLID é um conjunto composto de cinco princípios de desenvolvimento de software em orientação a objetos que propõe estratégias de para desenvolver código de qualidade, resistente e reutilizável (MARTIN, 2002). Os princípios são:

- Single Responsibility Principle (Princípio da Responsabilidade Única)
- Open/Closed Principle (Princípio do Aberto/Fechado)
- Liskov Substitution Principle (Princípio da Substituição de Liskov)
- Interface Segregation Principle (Princípio da Segregação de Interfaces)
- Dependency Inversion Principle (Princípio da Inversão de Dependências)

### 3.3.3 Domain Driven Design

Domain Driven Design consiste em um conjunto de técnicas de desenvolvimento de software que busca incorporar ao software o conhecimento do domínio específico da aplicação em questão. Todo o desenvolvimento da aplicação gira em torno de conseguir entender o modelo de negócio que o software deve implementar e criar uma linguagem uniforme para comunicação entre os especialistas do negócio (domínio) e os especialistas técnicos (desenvolvedores).

### 3.3.4 Testes

Dentro do conceito deste trabalho podemos interpretar "Testes" como os testes automatizados criados no desenvolvimento de um sistema. São os artefatos, código escrito em alguma linguagem de programação, criados pelo desenvolvedor para validar que as funcionalidades do sistema estão corretas. Na organização é importante que um teste seja idempotente, isto é, quando executado em momentos diferentes mas com as mesmas entradas, o resultado deve ser sempre o mesmo e pode ser segmentado em três etapas diferentes:

- Preparação: a primeira etapa de um teste, onde tudo aquilo necessário para a execução do teste é instanciado.
- Ação: nesta etapa, a funcionalidade que se deseja testar é executada.

- Asserção: a última etapa de um teste é composta por um conjunto de comparações

### 3.3.5 Test Driven Development (TDD)

Para encaixar o desenvolvimento de testes nos projetos, o estagiário utilizou um processo de desenvolvimento de software conhecido como *Test Driven Development* (BECK, 2003), um processo cíclico com três etapas:

- Vermelho: é sempre o primeiro passo do processo. Consiste em construir um teste que respeite a especificação atribuída à ele onde se espera que falhe por ainda não existir implementação da funcionalidade.
- Verde: nesta etapa do processo espera-se deve ser construída a implementação que faça com que o teste passe.
- Refatoração: nesta etapa são aplicadas as boas práticas de programação e design de software à funcionalidade construída, caso o teste falhe, voltar ao primeiro passo do processo.

## 3.4 Banco de Dados

Bancos de dados são conjuntos de arquivos correlacionados responsáveis por garantir a persintência das informações relevantes em um sistema (SILBERSCHATZ et al., 1997). São gerenciados por softwares conhecidos por SGBDs (sistemas gerenciadores de banco de dados) e, os mais tradicionais, utilizam de uma linguagem declarativa conhecida como SQL (*Structured Query Language*).

O estagiário trabalhou com três destes sistemas: PostgreSQL<sup>9</sup>, MySQL<sup>10</sup> e SQLite<sup>11</sup>. Todos utilizavam de SQL para possibilitar o programador de interagir

---

<sup>9</sup> <https://www.postgresql.org>

<sup>10</sup> <https://www.mysql.com>

<sup>11</sup> <https://www.sqlite.org/index.html>

com o sistema. Esta interação era possível por pela utilização de ORMs. Somado ao Kotlin e ao Spring, a ORM utilizada no desenvolvido back-end com foi a especificação JPA.



## 4 ATIVIDADES DESENVOLVIDAS

Este capítulo é destinado à descrever as atividades realizadas pelo estagiário, correlacionadas ao bacharelado de Ciência da Computação, durante o período estagiado na companhia.

### 4.1 Treinamento

Esta seção é dedicada a explicar a etapa de treinamento do processo de estágio, o método utilizado para organizá-lo, os conceitos aprendidos e atividades realizadas.

O treinamento teve duração de cinco meses com o objetivo de desenvolver as habilidades referentes ao desenvolvimento web full stack, traduzido em três módulos progressivos: módulo “Web Básico”, “Fullstack” e “Especialização”. No módulo de "Especialização", o estagiário tem a opção de escolher entre *frontend* e *backend*.

#### 4.1.1 Módulo “Web Básico”

A primeira fase do treinamento teve o objetivo ensinar os conceitos base de HTML, CSS, Javascript, versionamento e boas práticas de código .

Para a fixação dos conceitos, foram realizados dois projetos: o desenvolvimento de um website estático, referenciado na imagem 4.1, e de uma calculadora para o navegador web, na imagem 4.2, descritos abaixo, ambos utilizando a ferramenta Git para controle de versão e a plataforma Gitlab para armazenamento remoto e compartilhamento.

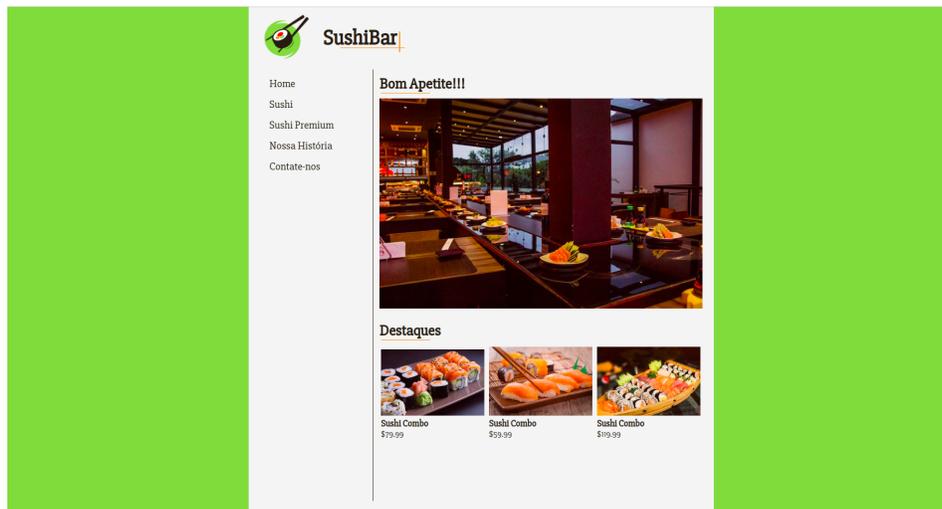
Além dos projetos, também foi realizada a leitura dos primeiros capítulos do livro “Código Limpo” (MARTIN, 2019) com o intuito de desenvolver habilidades para produção de código com maior legibilidade.

#### 4.1.1.1 Projeto “Website Estático”

A abordagem de construir diferentes layouts para a mesma aplicação foi utilizada para que o estagiário tivesse experiência com abordagens de layout diferentes, sendo elas:

- Fixa, onde os componentes visuais possuem tamanho absoluto;
- Fluida, onde o tamanho dos componentes se adapta ao tamanho do navegador;
- Mista, onde alguns componentes são adaptáveis e outros possuem tamanho absoluto.

Figura 4.1 – Representação do Projeto “Website Estático”

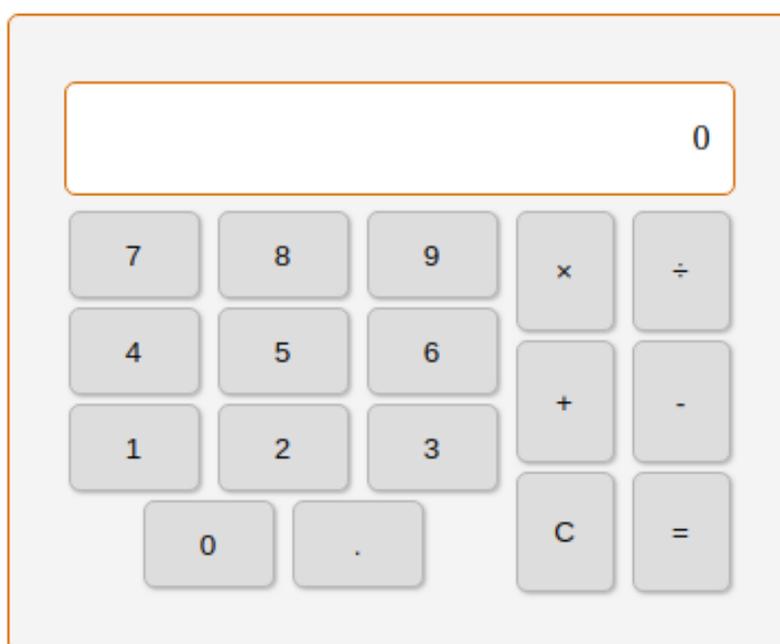


#### 4.1.1.2 Projeto “Calculadora para o Navegador”

Com o propósito de aumentar a complexidade e possibilitar a adição de itens lógicos aos projetos realizados, esta etapa do módulo web básico consistiu em construir uma calculadora para o navegador. A calculadora deveria ser capaz de realizar as quatro operações básicas: adição, subtração, multiplicação e divisão.

Os conceitos aprendidos na realização do projeto website estático deveriam ser aplicadas à calculadora, e a adição da linguagem de programação JavaScript permitiu que fosse possível atualizar o conteúdo do navegador em tempo real, assim ocasionando maior interação entre o usuário e a aplicação.

Figura 4.2 – Representação do Projeto “Calculadora para o Navegador”



#### 4.1.2 Módulo “Fullstack”

O objetivo desta fase foi ampliar o conhecimento do estagiário sobre o ecossistema de desenvolvimento web, incluindo o conceito de backend ao treinamento.

Para atingir ao propósito desta etapa, três projetos foram realizados: (i) um sistema de controle de funcionários, (ii) um sistema de rastreamento de peso, e (iii) uma evolução do projeto de controle de funcionários.

O primeiro (i) foi um projeto simples, apenas com a lógica do modelo de negócio, sem persistência de dados. No segundo (ii) foram utilizadas a ferramenta Node.js e o *framework* Hapi.js para o desenvolvimento, e houve treinamento na persistência de dados utilizando o banco de dados PostgreSQL. Neste projeto também, foram introduzidos conceitos de arquitetura cliente-servidor, HTTP e orientação a objetos. O último projeto (iii), teve o objetivo de apresentar ao estagiário a prática no modelo MVC (Model-View-Controller) e apresentar o conceito de testes de unidade.

#### **4.1.2.1 Projeto “Sistema de Funcionários”**

O projeto possibilita ao usuário cadastrar, editar, remover e buscar usuários de uma empresa através de uma interface web. Como não era um requisito que o sistema fosse capaz de persistir os dados cadastrados, ao recarregar a página, todos a base era apagada.

Como o foco deste projeto não era a interface, foi utilizada a biblioteca de componentes front-end Bootstrap para facilitar a produção dos layouts.

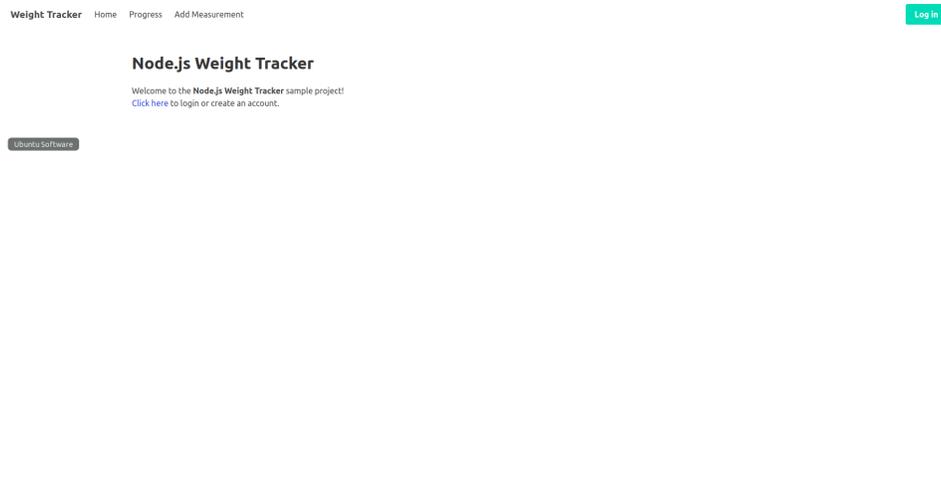
#### **4.1.2.2 Tutorial “Sistema de Rastreamento de Peso”**

Esta etapa do treinamento foi desenvolver um sistema de rastreamento de peso, demonstrado na imagem 4.3, seguindo o tutorial (NEAL, ). Foi possível ao estagiário realizar um treinamento prático com a ferramenta Node.js, o *framework* Hapi.js e o banco de dados PostgreSQL, em um container Docker. Neste projeto também foi utilizada a técnica de templates para HTML através da linguagem de templates EJS, além de conseguir praticar os conceitos da arquitetura cliente-servidor e do padrão MVC.

O objetivo deste tutorial foi ensinar as tecnologias e conceitos básicos que seriam usados no Projeto “MVC”, que foi realizado em sequência e aumentar a

proficiência com a linguagem Javascript, essencial no desenvolvimento de aplicações web.

Figura 4.3 – Representação do Tutorial “Sistema de Rastreamento de Peso”



#### 4.1.2.3 Projeto “MVC”

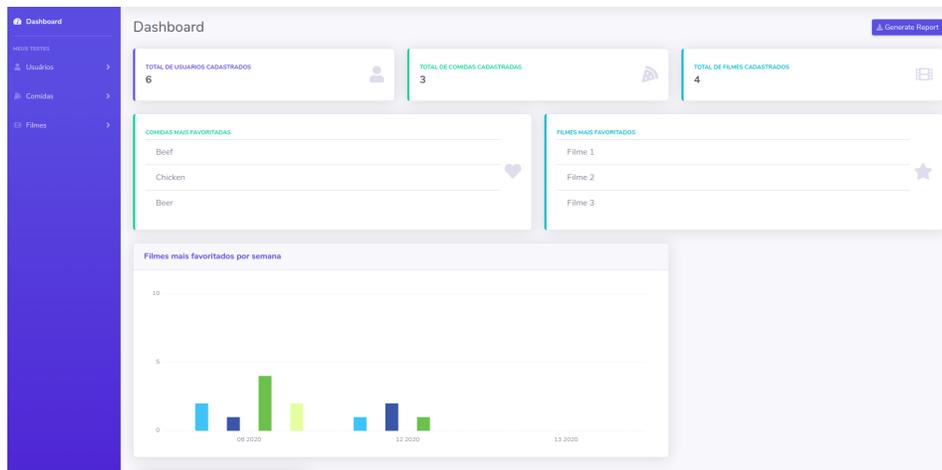
Este projeto, ilustrado na imagem 4.4, foi planejado para ser uma evolução ao Projeto “Sistema de Funcionários”. Além do CRUD dos usuários, o estagiário desenvolveu as funcionalidades:

- CRUD de comidas;
- CRUD de filmes;
- Usuário pode conter uma comida favorita;
- Usuário pode ter filmes favoritos;
- Relatórios do sistema: quantos usuário, comidas e filmes foram cadastrados, quais as comidas e filmes mais favoritas e os filmes mais favoritados por semana no sistema;

Teve duração de duas semanas e foi desenvolvido individualmente, possibilitou que o estagiário fosse capaz de aumentar seus conhecimentos em SQL, ao ter como requisito relações entre as tabelas, e conhecer o primeiro padrão arquitetural do treinamento, MVC.

Este projeto tinha também como requisito a criação de testes para as funcionalidades implementadas, portanto foi utilizada a biblioteca Jest para a criação dos testes do sistemas. Os testes realizados foram testes de requisição, ou seja, os outputs esperados eram para validar se uma requisição ao servidor foi executada corretamente.

Figura 4.4 – Representação do Projeto “MVC”



### 4.1.3 Módulo "Especialização Backend"

O módulo de especialização tinha como objetivo apresentar as tecnologias utilizadas na Credits e aumentar a experiência do estagiário no desenvolvimento backend. A etapa foi composta pelo desenvolvimento de apenas um projeto que foi apresentado de maneira evolutiva para simular o crescimento de um produto real da empresa, isto é, a cada iteração (que possuía duração média de uma semana) o projeto era atualizado com novos requisitos. O escopo do projeto foi uma API

capaz de cadastrar mensagens de texto e organizá-las por categorias específicas do sistema.

A primeira iteração foi o desenvolvimento da lógica central do projeto, uma estrutura de mensagem capaz de ser categorizada, e apresentação da linguagem de programação Kotlin. Durante esta etapa, não era esperada nenhuma forma de persistência dos dados nem de conexão à rede. Os requisitos técnicos foram a utilização do Kotlin e que o software deveria ter testes automatizados. A segunda iteração esperava a integração do projeto com a rede, criando uma aplicação *web*. Para isso, foi utilizado o *framework* Spring Boot para levantar a aplicação, que foi apenas testada em um ambiente local de desenvolvimento.

A terceira iteração esperava que os dados inseridos no sistema pudessem ser persistidos. Para isso, foi utilizado o banco de dados PostgreSQL e utilizando a especificação JPA, já integrada ao *framework*. A quarta, e última iteração, do projeto acrescentava novas regras de classificação ao projeto e a paginação dos dados buscados no banco de dados, além da funcionalidade de filtrar a busca por uma ou mais classificações.

Parte da metodologia desta etapa do treinamento foi ensinar o estagiário a procurar informação de maneira autônoma, portanto foi implementada uma estratégia onde primeiro era dado um problema, seguido de um tempo para o estagiário resolvê-lo e só então, durante a revisão da abordagem adotada pelo estagiário que o conteúdo era passado de maneira estruturada.

## 4.2 Crédito Consignado Privado

Esta seção visa apresentar o trabalho realizado pelo estagiário no produto de crédito consignado privado da empresa. Para poder ser uma continuidade do treinamento, o estagiário entrou no *squad Disbursement & Assignment* para trabalhar com desenvolvimento *back-end*, desenvolvendo APIs e sistemas voltado para suportar o modelo de negócios da companhia.

O *squad* era responsável pela implementação do software incumbido à suportar o desembolso do crédito ao cliente final. Para concluir as tarefas designadas ao estagiário foram criadas ou refatoradas funcionalidades em um sistema já existente, escrito na linguagem de programação PHP e utilizando o *framework* Laravel.

Um dos pontos mais marcantes em relação à sessão Módulo "Especialização Backend" e a etapa descrita nesta seção foi a complexidade do sistema que o estagiário entrou em contato, ela foi maior que qualquer um dos sistemas desenvolvidos durante o treinamento. Isto ocorreu por consequência da complexidade das regras do sistema para o qual o estagiário foi designado a trabalhar.

O segundo ponto de divergência entre as etapas anteriores e esta está no fato de a qualidade de código cobrada se tornar mais incisiva. Era esperado do estagiário a escrita de código limpo, legível, além de boas práticas de desenvolvimento de software, como os princípios de SOLID. Além da aplicação de alguns padrões de projeto em etapas específicas, como por exemplo o padrão *facade*, para simplificar a interface das camadas mais internas do projeto.

O último grande ponto de evolução desta etapa do estágio se dá nos processos de planejamento da arquitetura do sistema da qual o estagiário fez parte. O *squad* se organizava antes da execução das tarefas para planejar quais eram os artefatos existentes no sistema, quais as entradas e saídas de cada método, qual seria o fluxo do sistema e qual a quais padrões de projeto seriam utilizados. Esta estratégia de planejamento permitiu à equipe reconhecer problemas antes de começar a implementação, consequentemente sendo capaz de resolvê-los com menor esforço, além de possibilitar que as regras do sistema que não eram claras em um momento anterior ao planejamento fossem levantadas em momento recente do processo de desenvolvimento de software, possibilitando que resoluções fossem menos dispendiosas.

### 4.3 Disciplinas do Bacharelado

Esta seção tem o intuito de especificar algumas das disciplinas do curso de Bacharelado em Ciência da Computação que foram essenciais para o desenvolvimento do estagiário para ser inserido no mercado de trabalho.

Elas estão listadas a seguir pelo nome, seguido pelo código de identificação e uma breve descrição do porquê a foram importantes:

- Introdução aos Algoritmos (GCC224): por ser a matéria introdutória à programação, trazendo conceitos básicos importantes ao aluno.
- Estrutura de Dados (GCC216): ensina as estruturas de dados mais comuns, que podem ser encontradas em diversos contextos.
- Paradigmas de Linguagens de Programação (GCC198): dá ao aluno o contexto básico de vários paradigmas de programação, como por exemplo o paradigma funcional e o paradigma orientado a objetos.
- Introdução a Sistemas de Banco de Dados (GCC214): dá ao aluno o contexto básico de como trabalhar com bancos de dados, com foco em SQL.
- Sistemas Distribuídos (GCC129): por permitir ao estudante entender a teoria de sistemas distribuídos e diversas estratégias utilizadas para contornar os problemas encontrados neste tipo de arquitetura.
- Programação WEB (GCC209): por solidificar os conceitos básicos de web, que foram essenciais no estágio em questão.

### 4.4 Análise e Discussões

Esta sessão tem a pretensão de fazer uma análise crítica do estágio em relação às atividades desempenhadas e a correlação com o curso de Bacharelado em Ciência da Computação.

As divergências do mercado com o bacharelado podem ser vistas em três pilares:

- Tecnologias utilizadas: onde a experiência do estagiário indicou que o mercado utiliza de tecnologias e ferramentas mais modernas, como por exemplo a linguagem de programação Kotlin, enquanto a faculdade se mantém equipada com tecnologias mais tradicionais, como a linguagem de programação Java.
- Processos: não é do escopo do bacharelado ter grande foco em processos, porém o mercado trabalha e exigiu que o estagiário aprendesse sobre processos de desenvolvimento, engenharia de software e gestão de projetos.
- Qualidade de código: boas práticas para desenvolvimento de software foram consideradas uma das primeiras habilidades para serem trabalhadas durante o estágio. Em contrapartida a universidade, pela experiência do aluno, não trouxe oportunidades sólidas para aprender sobre o tema na grade obrigatória do curso.

A empresa trouxe uma maneira inovadora de inserir o aluno no mercado através do treinamento intensivo. Isto possibilitou ao estagiário uma base sólida nas ferramentas, processos e tecnologias que a empresa utilizava no trabalho cotidiano. Outro ponto positivo do estágio foi a evolução constante do programa para se adaptar à evolução do estagiário, acarretando na possibilidade do estudante de se desenvolver muito em um curto espaço de tempo.

Em contrapartida, seria válido caso o estagiário tivesse mais contexto e contato com as áreas da empresa durante o treinamento, para que tivesse a oportunidade de entender melhor o negócio e os processos específicos das equipes com mais antecedência e com mais solidez.

## 5 CONCLUSÃO

Este capítulo busca recapitular resumidamente a obra e traçar um paralelo entre o estágio e a experiência acadêmica vivenciada pelo estagiário, além de tecer as conclusões que prezam por conectar as disciplinas acadêmicas e o mercado de trabalho.

O estágio teve duração de 7 meses. Neste tempo houveram duas grandes etapas realizadas pelo estagiário: o Treinamento e o produto de Crédito Consignado Privado. O objetivo da primeira etapa era preparar o estagiário para conhecer as tecnologias e processos de desenvolvimento de software utilizados na Creditas para o desenvolvimento dos produtos de softwares da empresa e permitir que o estagiário adquirisse experiência de programação. Para isso, o estagiário realizou diversos projetos individuais e recebeu acesso à um portfólio para embasamento teórico composto de livros, artigos e seminários realizados por membros mais experientes da companhia.

Na segunda etapa do estágio, esperava-se que o estagiário possuísse os conhecimentos necessários para poder participar ativamente de uma equipe em projeto efetivo. No desenrolar das atividades, o estagiário foi apresentado a uma nova linguagem de programação e framework para atuar em um dos sistemas da empresa, fora tecnologias já conhecidas durante a primeira etapa. O estagiário também teve contato com processos organizacionais para o desenvolvimento de software além de novos padrões e boas práticas de programação.

O curso de Bacharelado em Ciência da Computação da Universidade Federal de Lavras provê uma base em estruturas de dados que se provou útil durante o estágio, auxiliando o estagiário a entender assuntos como índices em banco de dados até a escolher uma estrutura de dados mais bem planejada para lidar com um problema específico. Somado a isso, o embasamento teórico em linguagens de programação contribuiu no entendimento rápido de Kotlin, uma linguagem com a qual o estagiário nunca havia tido contato.

Se provou benéfico o estudo dos assuntos de redes de computadores e sistemas distribuídos, porque os sistemas que o estagiário trabalhou são definidos como sistemas *web*, e entender o funcionamento de protocolos de troca de mensagem e as especificações de um sistema distribuído em rede foram facilitadores para o desenvolvimento dos projetos durante o curso do estágio.

O estágio se caracterizou como validador ao curso de Ciência da Computação, considerando que o estagiário teve contato com temas análogos na graduação e no estágio. Além disso, o estágio também foi complementar ao curso, na medida em que aprofundava em conceitos já vistos na graduação, mas não foram possíveis de serem estudados durante as disciplinas. Isso ocorreu porque a complexidade dos projetos da empresa não poderiam ser replicados no tempo decorrido por uma disciplina.

Ademais, o estágio valoriza a realização de atividades em equipe, devido à complexidade dos produtos da empresa, acarretando na utilização de habilidades comunicativas e de organização com maior frequência que o bacharelado, porém, a realização de trabalhos em grupos e apresentações formais auxiliaram o estagiário na adaptação ao mercado de trabalho.

A maior divergência do bacharelado em contraste com o estágio são as tecnologias utilizadas. O curso de Ciência da Computação poderia buscar maneiras de possibilitar aos alunos terem contato com tecnologias utilizadas no mercado. Isto poderia acontecer com esforços para ampliar grupos de estudos externos à classe voltados para estudar tecnologias utilizadas no mercado de trabalho e tecnologias mais modernas.

Já as disciplinas poderiam ter um caráter mais prático em que o aluno possuía flexibilidade para definir quais ferramentas ele considera as melhores para resolver os projetos. Deve-se levar em conta que esta estratégia seria mais benéfica durante o ciclo avançado do curso, porque neste ponto o aluno já possui experiência para tomar escolhas deste calibre. O ciclo básico de ensino deveria

ter um caráter mais rígido na escolha das tecnologias para garantir a qualidade do aprendizado.

O segundo ponto em que existe uma divergência significativa entre o a formação universitária e o estágio se encontra na cobrança alta na qualidade do software exigido no segundo, buscando utilizar processos de desenvolvimento de software, como *Domain Driven Design*, ou padrões de projeto de orientação a objeto. Faz sentido para a universidade tratar questões como código limpo e arquitetura básica de software como pontos a serem ensinados durante o ciclo de ensino porque isto pode se provar benéfico para os alunos, que teriam maior facilidade para desenvolver em equipe e também para os professores, que encontrariam softwares melhor escritos.

Com o mundo tornando-se cada vez mais digital e conectado, espera-se que os softwares modernos estejam cada vez mais conectados à internet, isto pode ser visto com todos os projetos que o estagiário desenvolveu terem contato com o ambiente *web*. O curso de Ciência da Computação poderia dar mais visibilidade para esta tendência, considerando que as matérias obrigatórias não visam esta área da computação, e pode ser confirmado quando a matéria "Desenvolvimento Web", que deveria ser uma das fundações desta área de conhecimento, é eletiva na graduação.

Finalmente, o curso de Ciência da Computação provê uma base sólida para o estágio, provendo ao estudante bagagem suficiente em linguagens de programação, desenvolvimento de sistemas e banco de dados para conseguir alavancar a carreira no mercado de trabalho.



## REFERÊNCIAS

- AUTO, C. **Creditas - Auto**. Accessed: 2020-06-08. Disponível em: <<https://www.creditas.com/emprestimo/garantia-veiculo>>.
- BECK, K. **Test-driven development: by example**. [S.l.]: Addison-Wesley Professional, 2003.
- BOURQUE, P.; FAIRLEY, R. E. et al. **Guide to the software engineering body of knowledge (SWEBOK (R)): Version 3.0**. [S.l.]: IEEE Computer Society Press, 2014.
- CREDITAS. **Creditas - Quem Somos**. Accessed: 2020-08-06. Disponível em: <<https://www.creditas.com/quem-somos>>.
- CROCKFORD, D. **JavaScript: The Good Parts**. [S.l.]: "O'Reilly Media, Inc.", 2008.
- FREEMAN, E. et al. **Head first design patterns**. [S.l.]: "O'Reilly Media, Inc.", 2008.
- HOME, C. **Creditas - Home**. Accessed: 2020-06-08. Disponível em: <<https://www.creditas.com/emprestimo/garantia-imovel>>.
- JEMEROV, D.; ISAKOVA, S. **Kotlin in Action**. Manning Publications, 2017. Accessed: 2020-06-04. ISBN 9781617293290. Disponível em: <<https://books.google.com.br/books?id=qtclAEACAAJ>>.
- KNIBERG, H. **Spotify engineering culture (part 1)**. 2014. Accessed: 2020-04-08. Disponível em: <<https://labs.spotify.com/2014/03/27/spotify-engineering-culture-part-1/>>.
- MARTIN, R. C. **Agile software development: principles, patterns, and practices**. [S.l.]: Prentice Hall, 2002.
- MARTIN, R. C. **Código Limpo**. [S.l.]: Alta Books, 2019.
- NEAL, D. **Build a Weight Tracker App with Node.js and PostgreSQL**. <<https://scotch.io/tutorials/build-a-weight-tracker-app-with-nodejs-and-postgresql>>. Accessed: 2020-05-04.
- PAYROLL, C. **Creditas - Consignado**. Accessed: 2020-06-08. Disponível em: <<https://www.creditas.com/emprestimo/consignado-privado>>.
- SILBERSCHATZ, A. et al. **Database system concepts**. [S.l.]: McGraw-Hill New York, 1997. v. 5.
- W3. **Standards HTML CSS**. Accessed: 2020-08-06. Disponível em: <<https://www.w3.org/standards/webdesign/htmlcss>>.