



JOÃO VITOR SOARES REZENDE

**ESTÁGIO NA EMPRESA PROGEN BRASIL TECNOLOGIA:
GERÊNCIA DE PROJETOS E DESENVOLVIMENTO DE
SOFTWARE**

**LAVRAS - MG
2023**

JOÃO VITOR SOARES REZENDE

**ESTÁGIO NA EMPRESA PROGEN BRASIL TECNOLOGIA: GERÊNCIA DE
PROJETOS E DESENVOLVIMENTO DE SOFTWARE**

**INTERNSHIP AT THE COMPANY PROGEN BRASIL TECNOLOGIA: PROJECT
MANAGEMENT AND SOFTWARE DEVELOPMENT**

Relatório de estágio supervisionado apresentado
à Universidade Federal de Lavras, como parte
das exigências do Curso de Sistemas de
Informação, para a obtenção do título de
Bacharel.

APROVADO em 03 de Março de 2023
Prof. Dr. André Grützmann UFLA
Profª. Dra. Renata Teles Moreira UFLA
Prof. Dr. Raphael Winckler De Bettio UFLA

Prof. Dr. André Grützmann
Orientador

**LAVRAS - MG
2023**

JOÃO VITOR SOARES REZENDE

**ESTÁGIO NA EMPRESA PROGEN BRASIL TECNOLOGIA: GERÊNCIA DE
PROJETOS E DESENVOLVIMENTO DE SOFTWARE**

Relatório de estágio supervisionado apresentado à Universidade Federal de Lavras, como parte das exigências do Curso de Sistemas de Informação, para a obtenção do título de Bacharel.

Prof. Dr. André Grützmann
Orientador

**LAVRAS - MG
2023**

RESUMO

Este relatório traz a experiência de um discente da UFLA (Universidade Federal de Lavras), como estagiário na empresa Progen Brasil Tecnologia, mais conhecida pelo seu nome de fantasia ProgenBR. O estágio abrangeu atividades na área de gerência de projetos e desenvolvimento de software. As atividades de gerência de projetos abordaram temas como a supervisão de atividades e planejamento de funcionalidades, utilizando o método Kanban e a ferramenta KanbanFlow e, também, o controle de versão dos projetos utilizando o Git e o Github. Na parte de desenvolvimento envolveu-se atividades relacionadas com projetos feitos em PHP, HTML, CSS e JavaScript, efetuando o *deploy* de alguns desses projetos, por meio de servidores *Web* Apache executados sobre o sistema operacional Linux.

Palavras-chave: Gerência de projetos. Desenvolvimento de software. Pré-Processador de Hipertexto (PHP). Linguagem de Marcação de HiperTexto (HTML). Folha de Estilo em Cascatas (CSS). JavaScript. Git. Github. Linux. Apache. Servidor Web. Kanban.

ABSTRACT

This report brings the experience of a student from UFLA (Federal University of Lavras), as an intern at the company Progen Brasil Tecnologia, better known by its fantasy name ProgenBR. The internship covered activities in the area of project management and software development. Project management activities addressed topics such as activity supervision and functionalities planning, using the Kanban method and the KanbanFlow tool, as well as project version control using Git and Github. The development part involved activities related to projects made in PHP, HTML, CSS and JavaScript, deploying some of these projects through Apache Web servers running on the Linux operating system.

Keywords: Project Management. Software development. Hypertext Preprocessor (PHP). HyperText Markup Language (HTML). Cascading Style Sheet (CSS). JavaScript. Git. Github. Linux. Apache. Web server. Kanban.

LISTA DE FIGURAS

Figura 2.1 - Personalização da mesa de trabalho no Gather	8
Figura 2.2 - Entrada do ambiente virtual da ProgenBR no Gather	8
Figura 3.1 - Organograma da ProgenBR	9
Figura 4.1 - Exemplo de um <i>Pull Request</i> realizado no Gestor	13
Figura 4.2 - Imagem da tela de consulta do projeto XML NFe	15
Figura 4.3 - Imagem do novo template do Progen Store	19
Figura 4.4 - Página inicial do Progen Store	22
Figura 4.5 - Documentação da Versão 33.1.0	23

SUMÁRIO

1. INTRODUÇÃO	1
2. DESENVOLVIMENTO	3
2.1. Referencial teórico	3
2.1.1. Kanban	3
2.1.2. Linguagens de programação	3
2.1.2.1. Linguagem Python	4
2.1.2.2. Linguagem C#	4
2.1.2.3. HyperText Markup Language (HTML)	5
2.1.2.4. Hypertext Preprocessor (PHP)	5
2.1.2.5. Cascading Style Sheets (CSS)	6
2.1.2.6. JavaScript	6
2.1.2.7. Structured Query Language (SQL)	6
2.1.2.8. Git e Github	7
2.1.2.9. Gather	7
3. DESCRIÇÃO DO LOCAL DO ESTÁGIO	9
4. DESENVOLVIMENTO DAS ATIVIDADES	12
4.1. Revisão de Pull Requests no Github	12
4.2. Consulta do XML da NFe na SEFAZ	14
4.3. Publicação da atualização do App One nas lojas de aplicativos	16
4.4. VIEW de consulta dos menus do template	17
4.5. Instalador do PDV	18
4.6. Novo Frontend para o Progen Store	18
4.6.1. Ambiente de Homologação do novo frontend do Progen Store	19
4.7. Importador do Gestor	20
4.8. Melhoria nos componentes do Progen Store	20
4.9. Desenho da tela de atualização em massa e definição do seu funcionamento	21
4.10. Criação de uma nova instância do Progen Store	21
4.11. Atualização do ambiente de produção dos projetos	22
5. CONCLUSÃO	25
REFERÊNCIAS	27

1. INTRODUÇÃO

Realizar um estágio traz oportunidades profissionais únicas para um estudante. Como se trata de sua primeira experiência profissional, após sua vida acadêmica, ele tem a oportunidade de mostrar o seu potencial e conseguir, a partir deste ponto, trilhar a sua caminhada no mercado de trabalho.

De acordo com Almeida e Guido (2023) o mercado de trabalho mundial de TI (Tecnologia da Informação) está muito inflado e neste último ano houve um aumento no número de funcionários da área. Muitas empresas estão procurando por profissionais e, algumas das vezes, não os encontram. Isto faz com que as remunerações sejam mais atrativas, refletindo o efeito da Lei da Oferta e da Procura. Além disso, as ramificações na área de TI são muito extensas, sendo possível atuar em diversas áreas e com diferentes ferramentas.

O estágio aqui relatado foi realizado na empresa Progen Brasil Tecnologia LTDA., conhecida pelo seu nome de fantasia ProgenBR, cuja atividade principal é o desenvolvimento e licenciamento de programas de computador não-customizáveis. As atividades foram realizadas, em formato de *Home Office*, na área de desenvolvimento de *software* e gerência de projetos. Atualmente, a empresa comercializa 4 produtos principais, sendo eles: o *Progen Store*, que é uma solução para lojas virtuais; PDV, um sistema *desktop* de frente de loja; Gestor, um sistema Web de auxílio na gestão de empresas e o *App One*, um aplicativo que realiza funções complementares ao Gestor.

No período em que o estágio foi executado, o referente estagiário esteve responsável por gerir estes 4 projetos, sendo supervisionado por Rafael dos Santos Fratucci, um dos sócios fundadores, gerente de produtos e diretor executivo da empresa, e também por Alex Rosa de Jesus Junior, o outro sócio fundador e diretor de operações e desenvolvimento.

O estagiário atuou no levantamento de requisitos, identificação de erros, planejamento, revisão, gestão do controle de versão e atualização dos sistemas nos devidos servidores. Além disso, de acordo com o surgimento das atividades, realizou-se algumas contribuições no desenvolvimento dos projetos, a fim de agilizar a entrega ou então para aproveitar de momentos em que esteve menos ocupado com as demais tarefas.

A empresa utiliza o método Kanban como técnica de gerência de projetos por meio da ferramenta KanbanFlow¹. Nesta técnica são criados *cards* onde é descrita a tarefa a ser desenvolvida. Quanto mais detalhada for esta descrição, em geral, mais otimizado será o tempo utilizado na criação de uma solução.

¹ <https://kanbanflow.com/>

Além do Kanban, já que toda a equipe trabalha na modalidade *Home Office*, a empresa adotou o Gather² como ferramenta de representação virtual do ambiente de trabalho. O objetivo dessa adoção é ter controle dos funcionários que estão trabalhando num determinado momento e também permitir que os colaboradores possam conversar e fazer reuniões quando necessárias.

A principal linguagem de programação utilizada na empresa é o PHP, mas também são utilizados *Python* e *C#*, juntamente com outras ferramentas de apoio ao desenvolvimento de *softwares*.

As atividades realizadas durante o período do estágio incluem: identificação e solução de problemas, supervisão das tarefas realizadas, planejamento de tarefas, implementação de novas funções para os sistemas e publicação dos projetos no ambiente de produção. Todas estas atividades são descritas com maiores detalhes nas seções posteriores.

Além deste capítulo introdutório, este relatório está organizado da seguinte forma: no capítulo 2 é apresentada a definição e explicação de termos utilizados no desenvolvimento das atividades. A descrição da empresa e as informações sobre a contratação do estagiário encontram-se no capítulo 3. O capítulo 4 relata como foram as atividades exercidas pelo estagiário e ao final do relatório é feita uma conclusão a respeito do estágio.

² Plataforma que fornece um ambiente virtual, onde as pessoas podem se encontrar e realizar reuniões de trabalho, participar de eventos, entre outras situações.

2. DESENVOLVIMENTO

2.1. Referencial teórico

Com o objetivo de apresentar os principais conceitos que foram abordados no período do estágio, nesta seção são mostradas as definições de alguns termos e das ferramentas utilizadas durante o desenvolvimento das atividades.

2.1.1. Kanban

Atualmente várias empresas, principalmente as relacionadas a desenvolvimento de *software*, têm adotado o uso de metodologias ágeis para ajudar na organização dos seus processos. A metodologia ágil pode ser definida da seguinte forma:

Metodologia ágil é uma forma de conduzir projetos que busca dar maior rapidez aos processos e à conclusão de tarefas. Não apenas isso, mas o agile baseia-se em um fluxo de trabalho mais ágil, flexível, sem tantos obstáculos, com total iteratividade. (EQUIPE DA TOTVS, 2021)

O método Kanban pode ser considerado uma metodologia ágil, conforme relatado por Anderson (2011):

Sanjiv Augustine, autor de *Managing Agile Projects* e um dos fundadores da *Agile Project Leadership Network (APLN)*, visitou a Corbis em Seattle e descreveu nosso sistema Kanban como o “primeiro novo método ágil que observei em cinco anos”. (ANDERSON, 2011, p. 8)

O sistema Kanban utiliza-se de *cards* (cartões) que “... representam itens de trabalho” como explica Anderson (2010, p. 29). Estes cartões ficam em paredes ou mural, que por si só não se caracterizam como um sistema Kanban. Este conjunto de cartões e parede de cartões, pode ser melhor definido como um sistema de controle visual, por meio do qual uma equipe pode observar o andamento do trabalho e melhor organizar as suas ações. Além disso, permite que os colaboradores escolham a ordem de execução das tarefas, movendo cada trabalho concluído para uma lista de itens finalizados sem a obrigatoriedade da orientação de um supervisor.

2.1.2. Linguagens de programação

Para desenvolver *softwares* é necessário fazer o uso de alguma linguagem de programação que permitirá a criação do código fonte. Uma linguagem de programação pode ser entendida como um conjunto de instruções que fornecemos ao computador. Esta linguagem pode ser comparada com uma linguagem humana, já que “... permitem que

palavras e frases sejam combinadas de novas maneiras, tornando possível expressar conceitos sempre novos”, conforme apresentado por Marjin (2018).

As empresas fazem uso de determinadas linguagens para o desenvolvimento de *software* de acordo com suas finalidades. Sendo algumas delas o SQL, para a consulta em bancos de dados; PHP, Python, C#, Java, para o desenvolvimento do *backend* de sistemas, tipicamente; HTML, CSS, JavaScript, *JavaFX*, para o *frontend*, dentre outras várias existentes no mercado.

2.1.2.1. Linguagem Python

O *Python* pode ser utilizado no desenvolvimento de APIs (*Application Programming Interface*) e/ou *backend* de projetos. O *Python* é uma linguagem que possui uma sintaxe de rápido aprendizado, por possuir comandos simples e mais próximos da linguagem natural. Isto traz muitas vantagens, como apresentado por Stross-Radschinski.

Python é uma linguagem de programação muito enxuta. Os programas Python são muito mais curtos do que o código escrito em outras linguagens de programação modernas. Como resultado, os tempos de desenvolvimento e os custos de manutenção são drasticamente reduzidos. Menos código significa menos erros, o que significa que o custo de identificação e eliminação desses erros também é reduzido (STROSS-RADSCHINSKI, 2015, p. 3).

Como é informado por Lutz e Ascher (2007, p. 31), *Python* “é uma linguagem de *script* orientada a objetos - uma definição que combina suporte para POO (Programação orientada a objetos) com orientação global voltada para funções de *script*”.

2.1.2.2. Linguagem C#

Outra linguagem que pode ser utilizada para desenvolvimento de aplicações *desktop* é o C#, que é orientada pelos objetos e pode também ser usada para construção de aplicativos *mobiles*, usando o Xamarin Forms. Esta linguagem foi criada pela Microsoft para ser a ferramenta oficial de desenvolvimento de aplicações para o sistema operacional Windows. Trigo e Henriques (2018, p. 20) dizem que “A linguagem C# é considerada uma linguagem de alto nível, ou seja, uma linguagem com um nível de abstração relativamente elevado longe do código de máquina e mais próxima da linguagem humana”. Posteriormente também citam a semelhança do C# com VB (*Visual Basic*), C, C++, *Java* e *Python*.

O C# é executado pela máquina virtual conhecida como *.NET* (*dotNET*) que pode ser definida como:

... uma plataforma de execução de aplicações no Windows. Esta plataforma inclui um sistema de execução virtual designado de CLR (*Common Language Runtime*) e uma extensa biblioteca de classes com funcionalidades que vão desde a apresentação dos dados na tela pelo console, à manipulação de arquivos e acesso a bases de dados. (TRIGO; HENRIQUES, 2018, p. 21)

2.1.2.3. HyperText Markup Language (HTML)

As páginas dos sites que são criados com o PHP estão no formato HTML. Nelas são definidas *tags* de marcação para estruturar a página e definir uma espécie de etiqueta que identifica cada um dos componentes apresentados.

Ducket (2011, p. 28) define HTML da seguinte forma:

HTML significa *HyperText Markup Language*. A parte do Hipertexto refere-se ao fato de que o HTML permite criar links que permitem que os visitantes se movam de uma página para outra de forma rápida e fácil. Uma linguagem de marcação permite anotar texto e essas anotações fornecem significado adicional ao conteúdo de um documento. Se você pensar em uma página da Web, adicionamos código ao texto original que queremos exibir e o navegador usa o código para exibir a página corretamente. Portanto, as tags que adicionamos são a marcação. (DUCKET, 2011, p. 28)

2.1.2.4. Hypertext Preprocessor (PHP)

Diferentemente das duas linguagens anteriores que são utilizadas no *desktop*, o PHP é utilizado na *Web*. O PHP (um acrônimo recursivo para PHP: *Hypertext Preprocessor*) é uma linguagem de *script open source* de uso geral muito utilizada, especialmente adequada para o desenvolvimento *Web* e que pode ser embutida dentro do HTML.

Tatroe, MacIntyre e Lerdorf (2013, p. 2) destacam a flexibilidade do uso do PHP, listando uma variedade de sistemas operacionais e de servidores *Web* que o suportam.

O PHP é executado em todos os principais sistemas operacionais, desde variantes do Unix, incluindo Linux, FreeBSD, Ubuntu, Debian e Solaris até Windows e Mac OS X. Ele pode ser usado com todos os principais servidores da Web, incluindo Apache, Microsoft IIS e servidores Netscape/iPlanet. [...] Um dos recursos mais significativos do PHP é seu amplo suporte para bancos de dados. O PHP suporta todos os principais bancos de dados (incluindo MySQL, PostgreSQL, Oracle, Sybase, MS-SQL, DB2 e bancos de dados compatíveis com ODBC). (TATROE; MACINTYRE; LERDORF, 2013, p. 2)

Estas funcionalidades dão ao PHP funções muito úteis à criação de sistemas que sejam eficientes e eficazes e permitem às páginas do site terem dinamismo no seu conteúdo apresentado.

2.1.2.5. Cascading Style Sheets (CSS)

O CSS (*Cascading Style Sheets*), além de muitas outras coisas, complementa as páginas HTML. Enquanto o HTML define a estrutura da página, o CSS faz a sua estilização. Meyer e Weyl (2018, p. 1) definem CSS como:

...uma ferramenta poderosa que transforma a apresentação de um documento ou uma coleção de documentos, e se espalhou para quase todos os cantos da *Web* e em muitos ambientes ostensivamente não-*Web*. Por exemplo, navegadores baseados em Gecko usam CSS para afetar a apresentação do próprio cromo do navegador, muitos clientes RSS permitem que você aplique CSS a feeds e entradas de feed, e alguns clientes de mensagens instantâneas usam CSS para formatar janelas de bate-papo. (MEYER; WEYL, 2018, p. 1)

2.1.2.6. JavaScript

Complementando o trio de ferramentas comumente utilizadas para criar uma página *Web* temos o JavaScript, definido por Haverbeke (2018) como mostrado a seguir.

O JavaScript foi introduzido em 1995 como uma forma de adicionar programas a páginas da *Web* no navegador Netscape Navigator. Desde então, a linguagem foi adotada por todos os outros principais navegadores gráficos. Ele tornou possíveis os aplicativos da *Web* modernos — aplicativos com os quais você pode interagir diretamente sem precisar recarregar a página para cada ação. O JavaScript também é usado em sites mais tradicionais para fornecer várias formas de interatividade e inteligência. (HAVERBEKE, 2018)

2.1.2.7. Structured Query Language (SQL)

Na grande maioria dos projetos que são criados no mercado, é necessário que se tenha algum repositório para armazenar as informações manipuladas pelos *softwares*. Para isso, são utilizados SGBDs (Sistema de gerenciamento de banco de dados) que, em alguns casos, utilizam a linguagem SQL para manipular os dados gravados.

Para entender o SQL é necessário entender um pouco mais sobre bancos de dados, que Taylor (2019, p. 9) define “como uma coleção autodescritiva de registros. E sim, isso implica tecnologia de computador, completa com linguagens de programação como SQL”.

SQL significa *Structured Query Language* (Linguagem de Consulta Estruturada) e não é uma linguagem procedural como são Python, Java, C, C++ e C#. SQL refere-se, portanto, a uma linguagem não procedural, onde para cada solicitação feita ao SGBD, realiza a ação requisitada, da melhor forma possível, sem que seja necessário dizer como deve ser feito (ALLEN, 2019).

2.1.2.8. Git e Github

É altamente recomendável que se faça um controle das versões de um código fonte. Isto se faz necessário por algumas razões, principalmente, para poder restaurá-lo em um determinado ponto, de maneira rápida, quando necessário. Git é um sistema de controle de versão e, de acordo com Tsitoara (2020, p. 3):

“... trata do gerenciamento de várias versões de um projeto. Para gerenciar uma versão, cada alteração (adição, edição ou remoção) nos arquivos de um projeto deve ser rastreada. O Controle de versão registra cada alteração feita em um arquivo (ou grupo de arquivos) e oferece uma maneira de desfazer ou reverter cada alteração”. (TSITOARA, 2020, p.3)

Github é uma ferramenta que se utiliza do Git e, segundo Tsitoara (2020, p. 3), “... é uma plataforma de desenvolvimento inspirada na maneira como você trabalha. Do código aberto aos negócios, você pode hospedar e revisar códigos, gerenciar projetos e criar softwares ao lado de 36 milhões de desenvolvedores”.

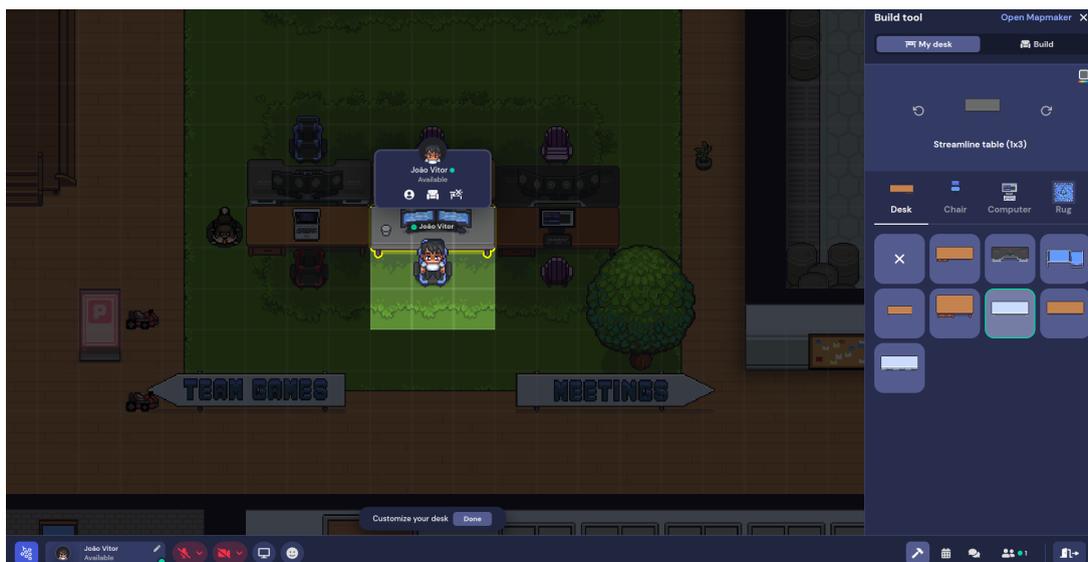
2.1.2.9. Gather

Como definido no site da empresa responsável pelo desenvolvimento, Gather é uma ferramenta de comunicação *online* que permite aos usuários realizar bate-papos com vídeo em tempo real, num ambiente virtual que simula a interação social de um espaço físico compartilhado. A ferramenta é projetada para criar uma experiência de comunicação mais imersiva, colaborativa e envolvente, permitindo que os usuários interajam com outros participantes como se estivessem em um ambiente físico real (GATHER PRESENCE, c2022).

Além disso, a ferramenta Gather também oferece recursos de compartilhamento de tela e áudio, bem como opções de customização de fundo e de efeitos visuais para melhorar a experiência de comunicação. É possível utilizar esta ferramenta em diferentes contextos, como reuniões de trabalho, eventos sociais, aulas *online*, entre outros.

A interface do Gather apresenta um ambiente gráfico em 2D, com avatares representando os usuários e a possibilidade de personalizar o espaço virtual com objetos e decorações, como pode ser visto na Figura 2.1. Os usuários podem se movimentar livremente pelo espaço virtual, interagir com outros avatares e participar de diferentes salas de bate-papo com vídeo, tendo cada uma sua própria temática e propósito.

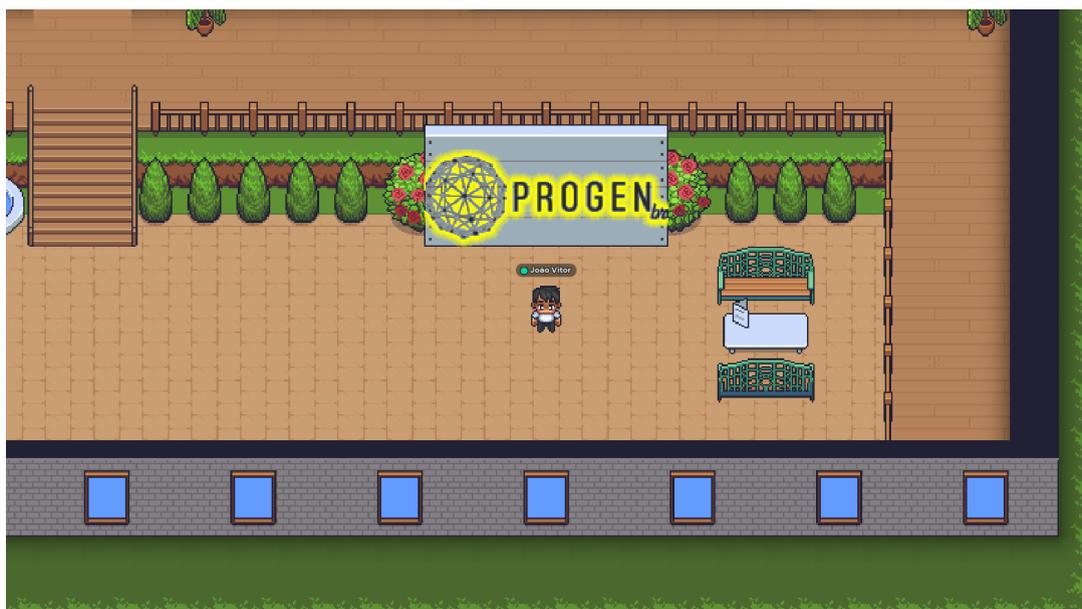
Figura 2.1 - Personalização da mesa de trabalho no Gather.



Fonte: Gather.

A customização do ambiente virtual do Gather permite que empresas personalizem o espaço virtual de acordo com a sua identidade visual e necessidades, tornando a experiência do usuário ainda mais envolvente e personalizada. Essa funcionalidade permite que as empresas possam promover sua marca, como tem sido feito pela ProgenBR e exibido na Figura 2.2, enquanto os usuários podem aproveitar a experiência do Gather de forma personalizada.

Figura 2.2 - Entrada do ambiente virtual da ProgenBR no Gather.



Fonte: Adaptado de Gather

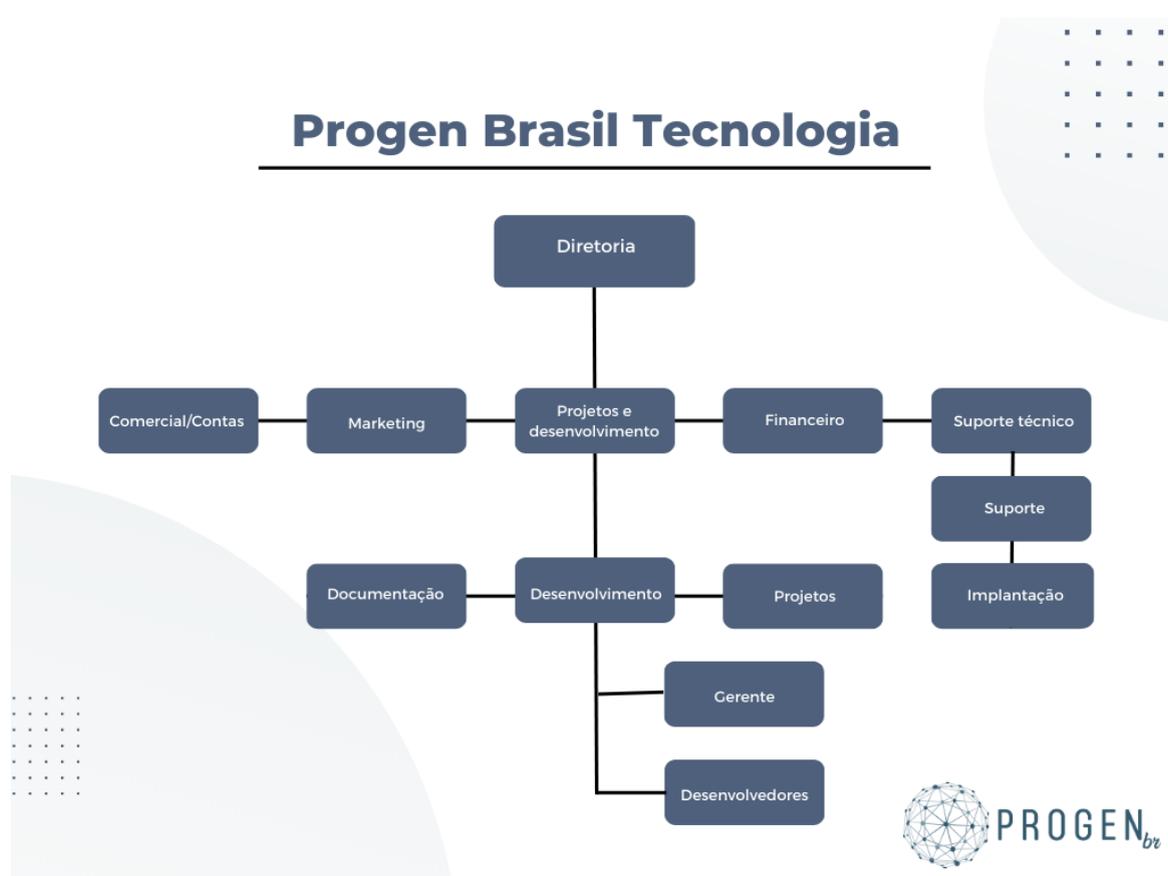
3. DESCRIÇÃO DO LOCAL DO ESTÁGIO

A empresa Progen Brasil Tecnologia Ltda., inscrita no CNPJ sob número 37.856.916/0001-58, conhecida pelo seu nome fantasia ProgenBR, pode ser categorizada como uma *software house*, já que tem o desenvolvimento e o licenciamento de programas de computador não-customizáveis como sua principal atividade.

Ela também possui algumas atividades secundárias como: comércio varejista especializado em equipamentos e suprimentos de informática; consultoria em tecnologia da informação (TI) e suporte técnico; manutenção e outros serviços em TI.

A diretoria da ProgenBR é composta, atualmente, somente pelos sócios-fundadores da empresa. Como pode ser visto no organograma da empresa, mostrado na Figura 3.1, abaixo desta diretoria, a empresa possui as áreas de Comercial/Contas, Marketing, Projetos e Desenvolvimento, Financeiro e Suporte Técnico.

Figura 3.1 - Organograma da ProgenBR.



Fonte: Progen Brasil Tecnologia LTDA

O estagiário foi inserido na área de Projetos e Desenvolvimento, sendo responsável pela gestão dos projetos da empresa, criação das documentações e auxílio aos outros programadores no cumprimento das suas tarefas.

A principal área de atuação da empresa é a automação comercial, pois fornece soluções para auxiliar e tornar os processos gerenciais, realizados por outras empresas, principalmente varejistas, mais eficazes e eficientes.

A comercialização dos seus produtos é feita pelo processo de licenciamento, que define os serviços e recursos que poderão ser utilizados, além de seus respectivos períodos de uso. Dentre os principais produtos da empresa estão: Gestor, *App One*, PDV e Progen Store.

O Gestor é um sistema que auxilia no gerenciamento da empresa-cliente. Ele é estruturado por módulos como: cotação, gerenciamento de pedidos de venda, gerenciamento de pedidos de compra, gerenciamento de contas a receber, gerenciamento de contas a pagar, dentre outras funcionalidades. O sistema foi desenvolvido utilizando o PHP com o *framework* CodeIgniter para as funções de *backend* e HTML, CSS e JavaScript com o *framework* jQuery para o *frontend*.

Já o PDV é um sistema de venda, muito utilizado nos caixas dos comércios. PDV significa Ponto de Venda e apresenta algumas funções como controle de estoque, venda de produtos, identificação de cliente, relatórios de vendas, entre outras funcionalidades. Sua arquitetura é desacoplada, ou seja, o *backend* é um projeto à parte do *frontend*. O *backend* utiliza a linguagem Python, enquanto o *frontend* foi desenvolvido em JavaFX.

O *App One* é um aplicativo complementar ao Gestor. Ele possui funções de gerenciamento de recebimentos, gerenciamento de inventários, consulta de produtos, *dashboards* de vendas, dentre outras funções. Sua arquitetura também é desacoplada, possuindo o seu *backend* em PHP e o *frontend* em C#.

O Progen Store é uma loja virtual, que tem a conexão contínua ao sistema de gerenciamento como sua principal vantagem. Ele também possui uma arquitetura desacoplada, utilizando o PHP tanto no *backend* quanto no *frontend*.

A ProgenBR possui 5 anos de existência e sua sede está localizada na cidade de Ribeirão Preto, São Paulo. Ela conta com 5 colaboradores, que cumprem com as metas e exigências definidas pela empresa. Todos os colaboradores trabalham no modelo *Home Office* e interagem por meio de uma sala virtual no Gather, conforme apresentado na figura 2.2 da seção 2.1.2.8 deste documento.

A empresa possui um programa para contratação de estagiários, que é realizado por meio de um processo seletivo. Este processo é composto por uma entrevista e um teste de capacitação, feito de acordo com a área de atuação desejada pelo candidato.

Para o estagiário deste relatório, os trâmites para sua contratação seguiram um fluxo diferente do descrito anteriormente. Como já havia prestado alguns serviços para a empresa de maneira terceirizada, estava periodicamente em contato com os responsáveis pela ProgenBR. Sendo assim, o estagiário não precisou passar pelo processo seletivo supramencionado, já que suas capacidades técnicas eram conhecidas pelos responsáveis da empresa.

Com o objetivo de auxiliar na adaptação do colaborador recém-ingresso na empresa, além de capacitá-lo para desempenhar suas atividades de forma mais eficiente e obter melhores resultados, a ProgenBR oferece cursos profissionalizantes em diversas áreas para os funcionários interessados.

4. DESENVOLVIMENTO DAS ATIVIDADES

O estagiário ficou responsável pela supervisão de todos os projetos da ProgenBR, sendo supervisionado por um dos sócios da empresa. De forma geral, as tarefas realizadas podem ser categorizadas em duas grandes áreas: gerenciamento de projetos e desenvolvimento de *software*.

No que se refere ao gerenciamento de projetos, foi efetuado o planejamento das funcionalidades, identificação e solução de problemas e supervisão no desenvolvimento do projeto. Em relação ao desenvolvimento, houve a produção de código e publicação dos aplicativos.

Para as atividades de desenvolvimento da API de consulta do XML (*eXtensible Markup Language*) da NFe (Nota Fiscal Eletrônica), do instalador do PDV, do novo *frontend* do Progen Store, do importador do Gestor e da atualização em massa de produtos, realizou-se o planejamento das funcionalidades utilizando-se de documentos descritivos, *cards* no KanbanFlow com o detalhamento do que deveria ser feito e de como fazer, além de esboços de telas, prazos e metas.

Durante a execução das tarefas, os problemas identificados foram resolvidos usando as soluções que serão descritas mais adiante neste documento. Estes problemas se referem às atividades que envolvem revisões de código e ao desenvolvimento da API de XML, que apresentava uma vulnerabilidade em sua estrutura.

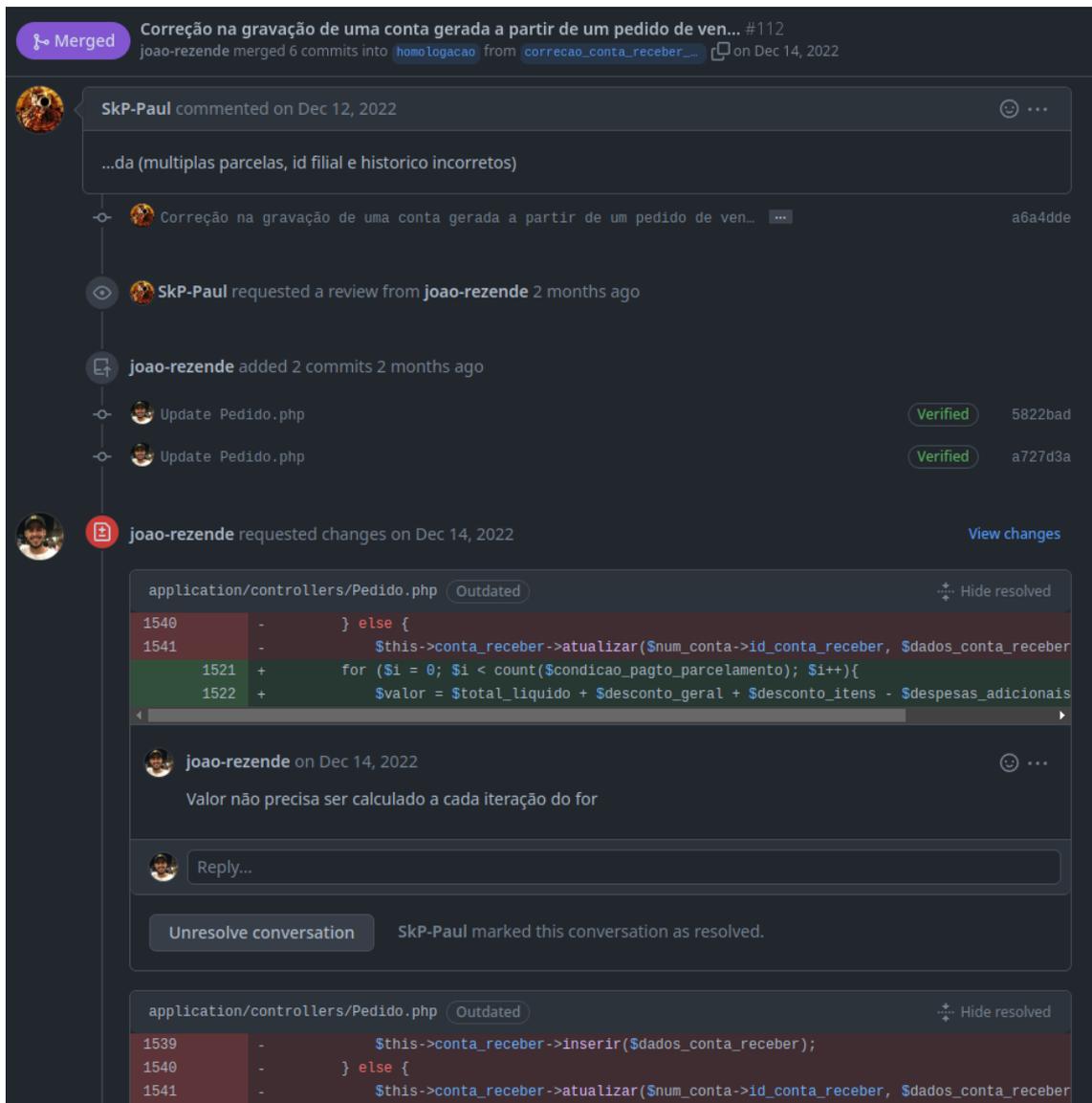
Nas tarefas de revisões e de planejamento, exigiu-se a supervisão do desenvolvimento das atividades, a fim de garantir que fossem entregues dentro prazo combinado e com a maior qualidade possível.

As demais atividades realizadas envolveram o desenvolvimento de *software*. Estas atividades serão detalhadas nas próximas seções e, resumidamente, são compostas pela criação de consultas SQL, tabelas e *views* em banco de dados, pela migração de código para o *microframework* Lumen, pela publicação do *App One* nas lojas de aplicativos, pela escrita de código e pelo *deploy* para o Progen Store. Além disso, o estagiário trabalhou no controle de versão dos projetos.

4.1. Revisão de *Pull Requests* no Github

As primeiras atividades que foram designadas para serem executadas estavam relacionadas à revisão de códigos nos PR (*Pull Requests*) do Github, conforme pode ser visto na Figura 4.1.

Figura 4.1 - Exemplo de um *Pull Request* aprovado realizado no Gestor.



Fonte: Github

A ProgenBR tem um fluxo de desenvolvimento muito semelhante com aquele que foi mostrado na disciplina de Desenvolvimento de Software Livre (GCC259) do curso de Sistemas de Informação, onde os desenvolvedores resolvem as suas tarefas que estão no Kanban em uma *branch* separada da principal e depois envia um PR no repositório do projeto com as suas mudanças. Em seguida o mantenedor, que neste caso é o estagiário deste relatório, faz uma revisão do código e adiciona comentários com sugestões ou requisições de correções no código que está sendo revisado. Depois disto, se estiver tudo correto com o PR,

o mantenedor aprova o código e, posteriormente, faz um *merge* das alterações na *branch* principal.

A leitura de um código escrito por outra pessoa e a potencial existência de diversas formas de se implementar soluções para um mesmo problema, tornam esta atividade complexa. Estes fatores fazem com que sejam necessárias paciência e atenção na leitura do código.

Além disto, é necessário entender bem a lógica usada por dois motivos:

- Evitar que algum erro passe despercebido e chegue ao ambiente de produção, causando problemas que, dependendo da gravidade, possam acarretar danos maiores;
- Permitir a proposição de melhorias que tornem o sistema mais eficiente e eficaz, oferecendo melhor qualidade de experiência (*Quality of Experience - QoE*) ao usuário final.

A empresa possui alguns guias de convenções de código que norteiam tanto o trabalho dos desenvolvedores quanto dos revisores. Como mostrado nas disciplinas de Gerência de Projetos de Software (GCC135) e de Modelagem e Implementação de Software (GCC132), tais convenções tornam a qualidade de um código melhor, já que seu uso produz um nível maior de padronização da estrutura do sistema, gerando, conseqüentemente, um software com melhor legibilidade e manutenibilidade. Estas convenções contribuem, de forma significativa, no rápido entendimento de um novo código, dada a similaridade estrutural com outros já desenvolvidos pela empresa.

A maioria das revisões de PR foram feitas no projeto Gestor, que possui um volume de código bem maior que os demais. Somente no mês de dezembro, totalizou-se 47 PRs, sendo 27 do Gestor, 18 do Progen Store e 2 de um projeto de XML de NFe. Nem todos estes PRs foram aceitos e alguns deles foram fechados sem que o código tenha sido mesclado com a *branch* principal.

4.2. Consulta do XML da NFe na SEFAZ

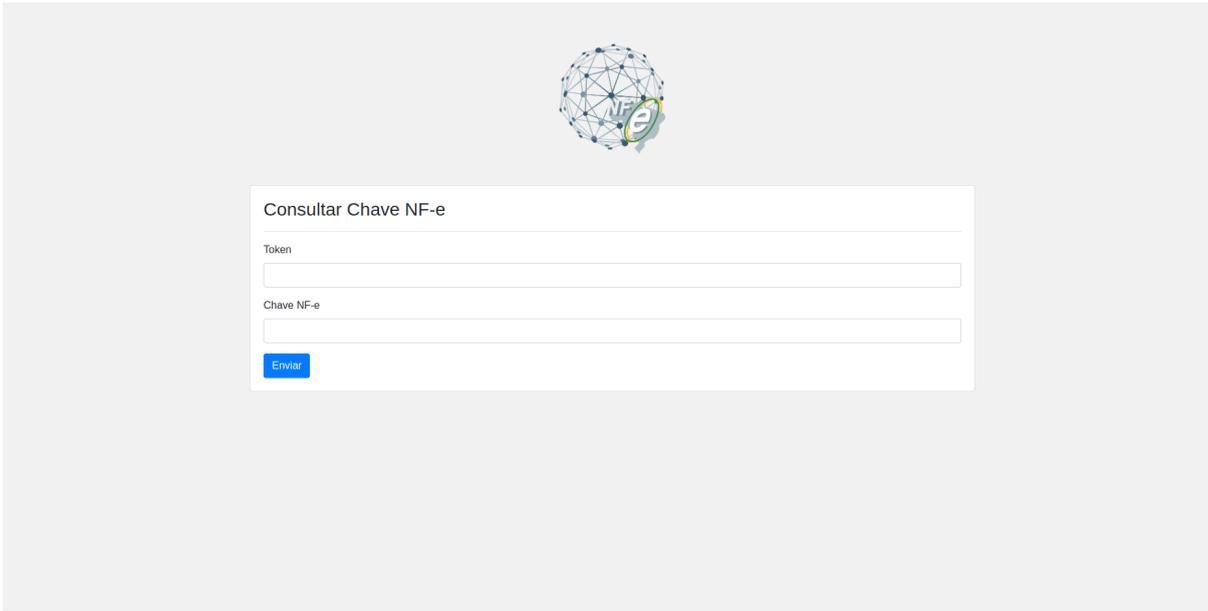
Durante o estágio foi requisitado o desenvolvimento de uma API para consultar o SEFAZ (Secretaria de Fazenda) e salvar os dados do XML de uma NFe (Nota Fiscal Eletrônica) no banco de dados da empresa. Ao fazer o planejamento do projeto, a equipe optou pela terceirização do desenvolvimento.

Após cerca de 3 semanas, uma solução foi entregue à ProgenBR. Devido a falta de detalhamento de algumas funcionalidades nas especificações que foram enviadas ao

desenvolvedor, esta API não foi entregue da forma esperada, sendo, então, repassada para que o estagiário deste relatório fizesse os ajustes necessários. Logo de início foi percebido que alguns conceitos utilizados já haviam sido vistos na disciplina de Sistemas Distribuídos (GCC129), como o Modelo cliente-servidor, Sistemas distribuídos baseados na Web e API de *Socket*.

A primeira etapa executada neste projeto foi a padronização do código para a forma como era realizada internamente na ProgenBR. Após a conclusão desta etapa, foram implementadas consultas a servidores, tratamento de erros, envio de resposta utilizando os códigos HTTP (*HyperText Transfer Protocol*) e a utilização de requisições com diferentes métodos HTTP. O resultado final foi a criação de uma API e de uma interface por meio das quais fosse possível a busca do XML desejado. A Figura 4.2 mostra a tela de consulta criada para este projeto.

Figura 4.2 - Imagem da tela de consulta do projeto XML NFe



Consultar Chave NF-e

Token

Chave NF-e

Enviar

Fonte: Progen Brasil Tecnologia LTDA

O passo seguinte foi a realização de uma adaptação no *App One*, que consistia da criação de uma função que consultava a API mencionada anteriormente, fazendo o tratamento dos resultados, de forma que as informações exibidas fossem uma mescla dos dados obtidos via interface com o usuário combinadas com a resposta da API.

Esta função foi concluída em 2 semanas e encaminhada para alguns testes. Contudo, durante seu desenvolvimento, percebeu-se a existência de uma pequena vulnerabilidade na

API. Tal vulnerabilidade já havia sido citada durante os estudos da disciplina de Segurança, Auditoria e Avaliação de Sistemas de Informação (GCC242).

Após detectar esta falha e perceber a necessidade da criação de um mecanismo de autenticação de requisições, decidiu-se alterar o projeto. Inicialmente, fez-se uma análise de viabilidade para o uso de um *framework* e, considerando-se a fase inicial em que o projeto se encontrava, optou-se por usar o *microframework* Lumen. Após esta análise, iniciou-se a migração do código para o Lumen com as correções das vulnerabilidades e a implementação da nova funcionalidade de autenticação das requisições.

Algumas das implementações efetuadas na versão anterior do projeto, nomeadamente o *middleware* e o roteamento pelos métodos HTTP, foram consideradas como possuindo qualidade insuficiente. Em virtude disso, aproveitou-se a disponibilidade de tais recursos no Lumen, para substituí-las pelas funcionalidades nativas do referido *microframework*. O restante do código sofreu apenas pequenas modificações e foi realizado com êxito o processo de migração, com a contribuição do ChatGPT³.

4.3. Publicação da atualização do *App One* nas lojas de aplicativos

O desenvolvimento do *App One*, demandou uma sequência de testes, realizados pelo sócios da empresa, com posteriores ajustes, até que fosse considerado pronto para uso. Em seguida, deu-se início ao processo de atualização nas lojas de aplicativos *App Store* e *Play Store*.

Em geral, a *App Store* costuma apresentar um pouco mais de burocracia que sua concorrente para a atualização dos aplicativos. Porém, desta vez, foi a *Play Store* quem apresentou maiores dificuldades.

A principal dificuldade estava relacionada à exibição de uma mensagem na plataforma de *upload* do aplicativo da *Play Store*, na qual se solicitava uma justificativa para o uso de determinada permissão por meio do envio de um vídeo demonstrando o seu uso. A permissão em questão, dá liberdade para que um aplicativo gere e acesse todo o armazenamento externo disponível no dispositivo.

O *App One* foi inicialmente desenvolvido por um colaborador terceirizado. Por esta razão, era necessário descobrir o motivo e o local de utilização de tal permissão. Esta atividade não foi tão simples de ser realizada.

³ O ChatGPT é um modelo de linguagem natural desenvolvido pela OpenAI capaz de gerar respostas coerentes e humanizadas a partir de uma ampla gama de tópicos.

De acordo com informações encontradas na Internet a respeito do funcionamento da permissão em questão nos *smartphones*, criou-se um vídeo, conforme solicitado pela equipe do Google. Contudo, tal vídeo não foi aceito como uma justificativa plausível, num primeiro momento.

Diante deste impasse, decidiu-se por gerar uma nova versão do aplicativo que não usasse a permissão mencionada. Esta versão foi gerada e encaminhada para os responsáveis pelos testes. Após a validação desta versão, o aplicativo foi enviado para a *Play Store* e, desta vez, aceito com sucesso.

4.4. *VIEW* de consulta dos menus do template

Em seu processo natural de aprimoramento, o sistema Gestor teve um novo template incorporado, por meio do qual se definia a visibilidade dos menus em função do perfil do usuário. A criação de menus flexíveis que substituíram a exibição fixa, até então usada pelo sistema, foi outra tarefa desenvolvida por este estagiário. O resultado final trouxe uma otimização do tempo de carregamento da tela do aplicativo.

Baseado nos conteúdos aprendidos em Sistemas Gerenciadores de Banco de Dados (GCC175), criou-se tabelas para o armazenamento dos menus e seus respectivos relacionamentos, assim como consultas SQL capazes de realizar as operações adequadas para o funcionamento do sistema. Estas consultas foram estruturadas de modo que, cada linha do resultado, apresentasse os dados do menu, juntamente com um JSON de seus respectivos submenus.

A primeira versão da consulta foi feita usando somente o JOIN e uma função nativa do PostgreSQL, chamada de *row_to_json*. Entretanto, não foi o bastante, já que um submenu poderia ter seus submenus também. Desta forma, decidiu-se pelo uso de *subquery* na consulta.

Uma segunda versão da consulta SQL foi gerada, usando a função *JSON_AGG* e *JSON_OBJECT*, além dos atributos *ORDER BY* e *GROUP BY*. Com esta nova consulta, os dados foram gerados, porém ainda com um problema na ordem esperada, devido à maneira como se usou o *GROUP BY*. A solução definitiva foi o uso de uma *subquery* dentro da outra.

Após estas alterações, utilizou-se o conceito de CTE (*Common Table Expression* ou Expressão de Tabela Comum) para a melhoria da legibilidade da consulta.

O código do sistema, no entanto, ainda não estava com boa legibilidade, já que uma quantidade significativa de código SQL era usada como uma string de parâmetro da função de construção de *query* da biblioteca do *framework*. Assim sendo, desenvolveu-se uma *view*,

com a mesma consulta SQL descrita anteriormente, para melhorar a legibilidade do código fonte do sistema Gestor.

4.5. Instalador do PDV

Em atendimento à demanda dos clientes da empresa, uma nova versão do instalador do PDV foi desenvolvida, agora, para o ambiente Linux. Esta atividade, realizada em paralelo com a anterior, ficou a cargo de outro colaborador, porém, com a supervisão deste estagiário.

A partir das necessidades apontadas, foram definidas as funcionalidades requeridas para o projeto, sendo criado um *card* no KanbanFlow com essas informações. Feito isso, deu-se início à construção do instalador. Durante o desenvolvimento desta atividade, utilizou-se a técnica de programação em par.

Terminada a etapa de programação, este instalador foi encaminhado para a revisão e algumas melhorias foram apresentadas, apenas, pela leitura do código fonte.

Uma vez que o projeto tenha passado em todos os testes, foi então encaminhado para sua homologação. Nesta etapa, outras pessoas fariam testes adicionais, em ambientes controlados, para que, posteriormente, fosse enviado para o ambiente de produção.

4.6. Novo Frontend para o Progen Store

Após algumas análises das lojas virtuais existentes no mercado, tecnologias que estavam sendo utilizadas em *frontends Web* e a defasagem do framework Yii2, optou-se por iniciar um novo modelo de *frontend* para o Progen Store. Desta vez utilizando o React, *framework* JavaScript.

Nesta tarefa, a programação também foi feita por outro colaborador, sendo supervisionado pelo estagiário deste relatório. Primeiro realizou-se um planejamento prévio do que seria importante, alinhado com o fluxo, normalmente utilizado em *e-commerce*, no ato de uma compra. Neste caso, o planejamento foi feito pelo desenvolvedor em conjunto com o *stakeholder* e foi convertido em *cards* no KanbanFlow.

Algumas destas tarefas descritas nos *cards* necessitavam de maior clareza no que diz respeito ao *design* do software. Desta forma, foram feitos alguns desenhos de telas, com o auxílio da ferramenta Pencil, que foi liberado para o desenvolvedor, após a aprovação dos *stakeholders*.

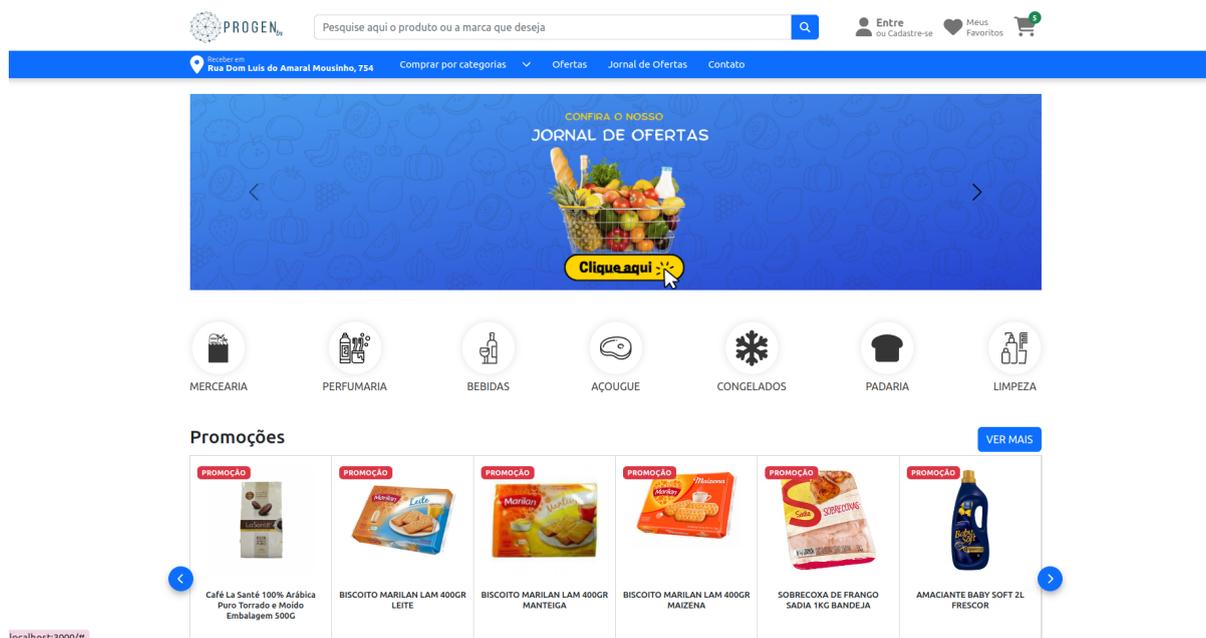
4.6.1. Ambiente de Homologação do novo frontend do Progen Store

Depois de algum tempo, quando finalizada uma certa quantidade de funcionalidades no sistema, decidiu-se fazer o *deploy* no servidor para iniciar sua homologação. Foi utilizado um servidor *web* Apache, na versão 2.4, para o *deploy*. O processo foi relativamente simples, bastando criar o arquivo de configuração do Apache, com a pasta que ele abriria quando fosse feita alguma requisição no servidor pelo domínio do Progen Store, fazer o *build* do React e, por fim, fazer *upload* para a pasta do projeto.

Certificados SSL (*Secure Sockets Layer*) foram utilizados a fim de habilitar o uso do protocolo HTTPS (*HyperText Transfer Protocol Secure*), que adiciona uma camada que criptografa a conexão por onde os dados são transmitidos. Isto foi feito, utilizando-se o *Certbot*, um robô que cria e renova os certificados SSL.

A versão final deste projeto está disponível a partir da URL <https://storehmg.progenbr.com> e uma de suas interfaces pode ser vista na Figura 4.3.

Figura 4.3 - Novo template do Progen Store



Fonte: Progen Brasil Tecnologia LTDA

Importante destacar que, as configurações utilizadas para este servidor Apache foi tema de uma atividade na disciplina de Redes de Computadores (GCC125) e o uso do protocolo HTTPS também foi explicado naquela disciplina.

4.7. Importador do Gestor

O Gestor possui algumas funcionalidades de sincronização de dados com outros sistemas, por meio da qual os dados recebidos são importados para o banco de dados.

Esta função de importação de dados estava incorporada às operações comuns do sistema e, por isto, estava prejudicando a fluidez do Gestor, já que, algumas vezes, uma grande quantidade era enviada. A solução deste problema, deu-se pela separação das atividades com a criação de um serviço exclusivo para a importação de clientes, fornecedores, produtos e entradas de produtos.

O serviço de importação de produtos foi o primeiro a ser reestruturado, por ser considerado o mais trabalhoso, por apresentar alguns problemas e pela necessidade de atualização há mais tempo. Posteriormente, foi realizado o desenvolvimento dos demais processos de importação.

Durante a execução desta atividade, percebeu-se a necessidade de se criar um ambiente de homologação para poder realizar os testes no projeto. Para a criação deste serviço, utilizou-se o *Python* como linguagem de programação, o módulo WSGI (*Web Server Gateway Interface*) do Servidor *Web Apache* como interface de entrada entre os servidores *Web* e aplicação, além de se utilizar ambientes virtuais para separar tanto as variáveis de ambiente dos outros projetos quanto os módulos do *Python* que foram instalados.

Depois de criados os arquivos de configurações do Apache, do React, do WSGI na aplicação Python, usou-se o Certbot para a criação e manutenção dos certificados SSL. Em seguida, a liberação do projeto foi feita para que os responsáveis pelos testes iniciassem seus respectivos trabalhos no projeto.

4.8. Melhoria nos componentes do Progen Store

Durante o estágio, alguns melhoramentos foram realizados em componentes do Progen Store.

A primeira mudança foi realizada na barra superior do sistema a fim de melhorar a estética da mesma. Ela ocupava espaço excessivo da tela e possuía a mesma cor de fundo. Assim, ao realizar um *scroll* na página, ocorria um efeito estético pouco agradável. Isto foi solucionado por meio da implementação de uma função que analisava se o *scroll* da página ultrapassava determinado valor, de modo que, se isso ocorresse, uma classe seria adicionada a *tag* HTML desta barra.

A segunda melhoria ocorreu no *card* de animação do carregamento da página, que não possuía as mesmas dimensões do *card* do produto. Para solucionar esse problema, foram corrigidos os valores de *padding*, *margin*, *height* e *width*.

4.9. Desenho da tela de atualização em massa e definição do seu funcionamento

Um dos *stakeholders* fez uma solicitação para a criação de uma nova funcionalidade no sistema Gestor, que consistia na atualização em massa de certos dados dos produtos. Foram apresentados os requisitos necessários para a funcionalidade e um esboço preliminar da tela desejada para essa finalidade.

De posse destas informações, a equipe responsável trabalhou em conjunto para implementar a nova funcionalidade, respeitando os padrões de projeto e *layout* previamente estabelecidos.

Dessa forma, foi desenvolvido um desenho detalhado da tela que incluiu todos os conteúdos utilizados em nosso sistema, juntamente com os mecanismos necessários para suprir as demandas do solicitante.

Após a verificação de todos os requisitos solicitados em relação às informações técnicas, gerou-se um *card*, no *KanbanFlow*, que continha a descrição geral da função, juntamente com o desenho da tela para que o desenvolvedor pudesse realizar sua implementação.

4.10. Criação de uma nova instância do Progen Store

Para cada cliente que contrata o direito de uso do sistema Progen Store, é criada uma nova instância deste projeto num servidor da empresa desenvolvedora ou num servidor próprio do cliente. Durante o estágio, um novo cliente passou a usar este produto e, portanto, foi necessário criar uma nova instância do projeto.

Para realizar este procedimento, é utilizado um *script* que foi desenvolvido em *shell* e que executa todas as funções necessárias para fazer uma nova instanciação do projeto e deixá-lo acessível via Internet.

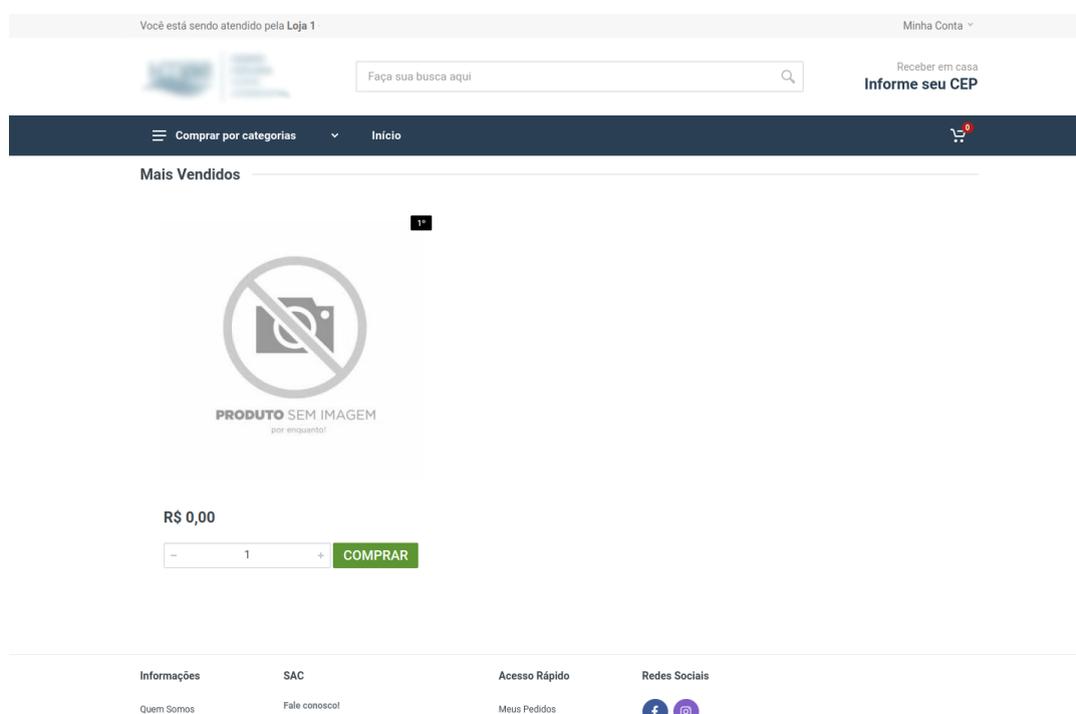
Durante sua execução, informam-se os dados do cliente em um arquivo de configuração, o nome da pasta do projeto, que também é usado no nome do usuário e no banco de dados, informações gerais da empresa, dados do servidor de e-mail e os *tokens* do reCAPTCHA⁴.

⁴ Serviço de verificação de segurança baseado em inteligência artificial que analisa o comportamento do usuário em sites para determinar se são humanos ou bots.

Finalizada esta etapa de parametrização é necessário realizar o tratamento nas imagens do sistema antes de liberar o uso do mesmo. Dentre as imagens estão a logomarca da empresa e o ícone que aparecerá no navegador quando se acessar o site. Para tanto, usou-se a ferramenta GIMP (*GNU Image Manipulation Program*), criou-se um ícone a partir da logomarca e editou-se o tamanho da imagem da página principal (FIGURA 4.4) e dos e-mails.

Após a realização de alguns testes preliminares, verificou-se a necessidade de tratamento das imagens do sistema para uma melhor adaptação. Novos testes foram realizados e, após a confirmação de que tudo estava em conformidade, foi liberada a primeira versão do sistema, cuja tela inicial pode ser vista na Figura 4.4.

Figura 4.4 - Página inicial do Progen Store



Fonte: Progen Brasil Tecnologia LTDA

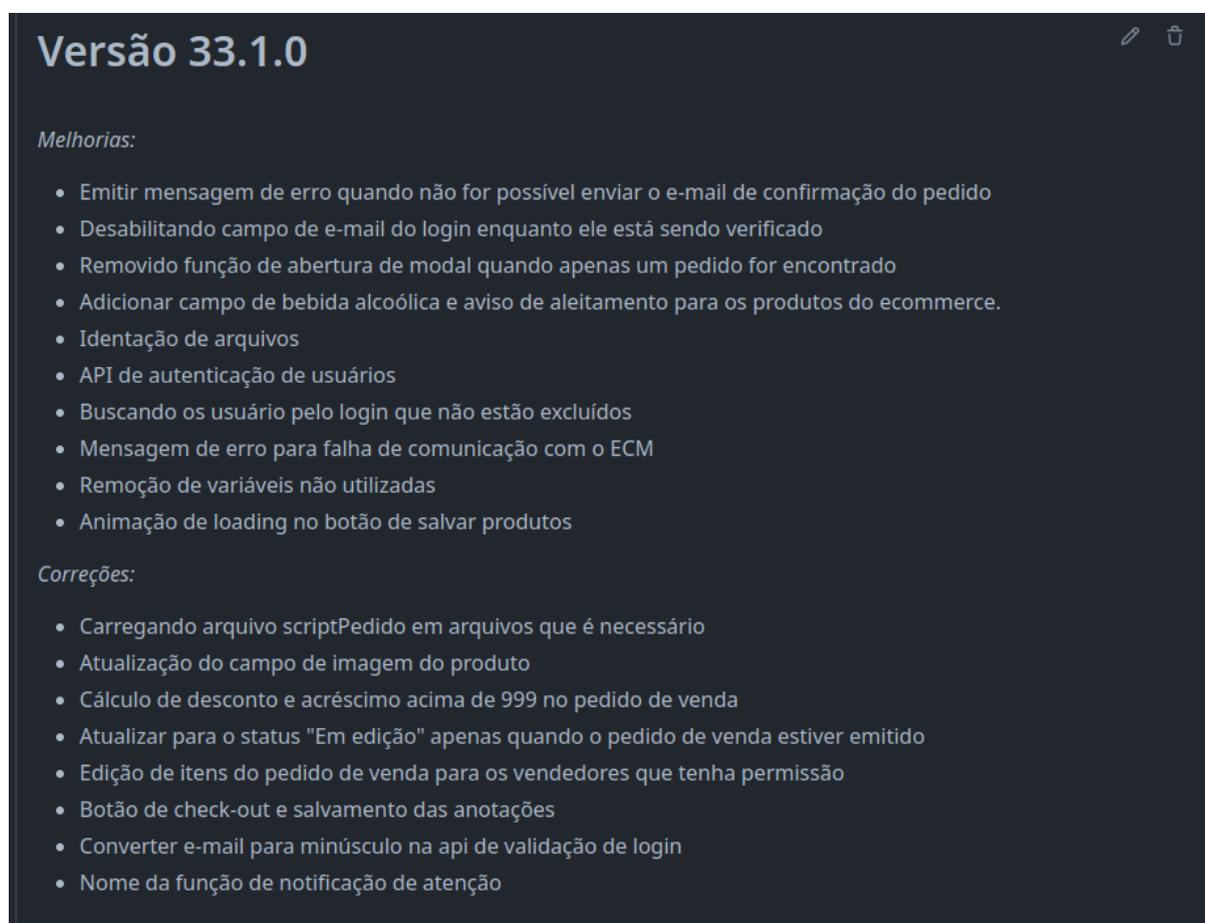
4.11. Atualização do ambiente de produção dos projetos

Os processos de atualização dos projetos mantidos pela ProgenBR diferem uns dos outros. Por exemplo, para atualizar todas as instâncias do Progen Store, um *script* é utilizado e isso simplifica o processo. Já para o Gestor, que utiliza o controle de versão com o Git, é necessário apenas a execução do comando *git pull* para atualizar o código a partir do repositório remoto.

Para atualizar o banco de dados, as *migrations*⁵ são utilizadas. As *migrations* também controlam as versões do banco de dados, garantindo que uma mesma *migration* não seja executada mais de uma vez.

Juntamente com a atualização do código, alguns processos burocráticos devem ser realizados, como a criação de um documento com um identificador de número de versão e uma lista dos itens corrigidos e melhorados naquela versão. A Figura 4.5 mostra um exemplo deste documento. Isso possibilita o mapeamento do que foi adicionado no sistema e sua comunicação aos clientes, já que as funcionalidades implementadas são, em alguns casos, solicitadas pelos mesmos.

Figura 4.5 - Documentação da Versão 33.1.0



Fonte: Github

Para cada versão criada, são utilizadas *tags* do Git para marcar em qual *commit* aquela versão se encontra. Além das *tags*, foi utilizado o recurso de *releases* do Github, por meio do qual um documento descritivo é criado, conforme mencionado anteriormente. Este recurso

⁵ Migrations são arquivos que contêm os códigos SQL necessários para atualizar a estrutura, incluindo tabelas, colunas e funções.

possibilita também o *download* de um arquivo compactado com o código fonte da respectiva versão. Isso proporciona uma organização eficiente de cada versão do projeto.

5. CONCLUSÃO

Após a execução do estágio fica mais evidente o quão importante é esta etapa para a formação profissional. Um estágio pode abrir portas à carreira do estudante, caso opte por ingressar no mercado de trabalho. Além disso, com ele, coloca-se em prática grande parte dos conceitos que são apresentados pelos professores em sala de aula.

A percepção da similaridade entre o que é visto em sala de aula com o que é feito no mercado de trabalho, permite ao estagiário uma maior valorização dos conhecimentos entregues. Percebe-se, de forma mais clara, que a fundamentação teórica, vista durante a graduação, juntamente com exemplos e exercícios práticos, podem auxiliar demasiadamente na resolução de alguns problemas que, possivelmente, encontrarão.

Além do fortalecimento do conhecimento adquirido, o estágio faz com que o estudante ganhe experiências para sua vida profissional e pessoal. No aspecto profissional, de uma forma clara, ganha experiência e maturidade para a realização daquele trabalho desenvolvido durante o estágio. No aspecto pessoal, ganha-se confiança, em situações difíceis, onde se vê obrigado a tomar uma decisão. Isso mostra que, na grande maioria das vezes, é necessário assumir riscos e não ter medo de fazer algo, mesmo que, eventualmente, esteja errado. Além disso, o *networking* proporcionado pelo estágio traz imensuráveis oportunidades futuras.

Como dito na seção 4 (Desenvolvimento das Atividades) foram utilizados alguns conceitos vistos em sala de aula nas disciplinas de Desenvolvimento de Software Livre, Sistemas Distribuídos, Gerência de Projetos de Software, Modelagem e Implementação de Software, Segurança, Auditoria e Avaliação de Sistemas de Informação, Sistemas Gerenciadores de Banco de Dados e Redes de Computadores, principalmente. A sólida fundamentação proporcionada ao estagiário por tais disciplinas, fez com que as atividades por ele desenvolvidas pudessem ser realizadas com um pouco mais de facilidade.

O estágio trouxe, ainda, grandes experiências para o estagiário, tornando-o um profissional e uma pessoa melhor. O conhecimento adquirido foi perceptível, pois o resultado das atividades realizadas hoje são melhores do que aqueles alcançados no início do estágio. É válido destacar que o processo de gerência possui as partes técnicas sobre definir ordens de prioridade, planejar e desenvolver em tempo hábil, porém o gerenciamento dos recursos humanos é algo bem complexo e é preciso ser dada a devida atenção. Isso pôde ser visto com muita clareza no decorrer do estágio.

A ProgenBR ainda é uma microempresa, existindo muitas coisas que podem ser melhoradas. Entre elas destacam-se a implementação de testes automatizados e a definição explícita dos seus processos para que eles possam ser gerenciados e otimizados.

REFERÊNCIAS

ALMEIDA, F., GUIDO, G. Mercado de TI continua contratando, apesar da crise das big techs. Forbes, 31 jan. 2023. Disponível em: <https://forbes.com.br/carreira/2023/01/mercado-tech-continua-aquecido-apesar-das-demissoes/>. Acesso em: 09 mar. 2023.

ANDERSEN, D. J., **Kanban**: Successful Evolutionary Change for Your Technology Business. 1. ed. Washington: Blue Hole Press, 2010.

ANDERSEN, D. J., **Kanban** - Mudança Evolucionária de Sucesso para Seu Negócio de Tecnologia. 1. ed. Washington: Blue Hole Press, 2011.

Caelum, **C# E ORIENTAÇÃO A OBJETOS**. Alura. Disponível em: <https://www.caelum.com.br/apostila/apostila-csharp-orientacao-objetos.pdf>> Acesso em: 24 fev. 2023.

DUCKETT, J., **HTML & CSS - Design and Build Websites**. Indianapolis: John Wiley & Sons, 2011.

EQUIPE TOTVS. **Metodologia ágil**: o que é e como implementar. 12 de jul. de 2021. Disponível em: <https://www.totvs.com/blog/negocios/metodologia-agil>>. Acesso em: 24 de fev. de 2023

Gather Presence, Inc. **GATHER FOR REMOTE WORK** c2022. Disponível em: <https://www.gather.town/remote-work>>. Acesso em: 11 de mar. de 2023.

HAVERBKE, M., **Eloquent JavaScript**. 3. ed. No Starch Press - 2018

LUTZ, M.; ASCHER, D., **Aprendendo Python**. 2. ed. Porto Alegre: Bookman, 2007.

MEYER, E. A.; WEYL, E., **CSS - The Definitive Guide**. 4. ed. Sebastopol: O'Reilly, 2018.

STROSS-RADSCHINSKI, A. C. **Python**: A programming language changes the world, v. 1, 2. ed. – 31, março, 2015. Disponível em: <https://brochure.getpython.info/media/releases/psf-python-brochure-vol.-i-final-download.pdf/view>>. Acesso em: 24 fev. 2023.

TATROE, K.; MACINTYRE, PETER.; LERDORF, R., **Programming PHP**. 3. ed. Sebastopol: O'Reilly, 2013.

TAYLOR, A. G., **SQL For Dummies**. 9. ed. Hoboken: John Wiley & Sons, 2019.

THE PHP GROUP. **O que é o PHP?** c2001-2023. Disponível em: https://www.php.net/manual/pt_BR/intro-what-is.php>. Acesso em: 24 de fev. de 2023.

TSITOARA, M., **Beginning Git and GitHub**. 1. ed. Antananarivo: Apress, 2020.