FACULDADE DA INDÚSTRIA DE SÃO JOSÉ DOS PINHAIS

ANDRÉ FELIPE CASTRO DOS SANTOS
ARTHUR DAVI DOS SANTOS
GABRIELLE DE LIMA NUNES
MAYANE APARECIDA GALDINO DE LIMA
RHUAN FABRÍCIO DIAS DOS SANTOS
RÔMULO DORIGAN

ROADMAP DE INOVAÇÕES

SÃO JOSÉ DOS PINHAIS

2022

ANDRÉ FELIPE CASTRO DOS SANTOS
ARTHUR DAVI DOS SANTOS
GABRIELLE DE LIMA NUNES
MAYANE APARECIDA GALDINO DE LIMA
RHUAN FABRÍCIO DIAS DOS SANTOS
RÔMULO DORIGAN

ROADMAP DE INOVAÇÕES

Trabalho apresentado para a documentação do Projeto Integrador do 5º período do curso Bacharelado em Sistemas de Informação apresentado à Faculdade da Indústria de São José dos Pinhais.

Orientador(a): Maurício Antônio Ferste

SÃO JOSÉ DOS PINHAIS

2022

RESUMO

Este projeto tem como finalidade o desenvolvimento de uma aplicação web para a empresa Volkswagen, voltado para o cadastro de inovações propostas pelos colaboradores, tendo como objetivo transformar o método utilizado atualmente que se trata de uma planilha em excel para um sistema web contendo as funcionalidades já utilizadas na planilha. Transformando o Roadmap em um sistema, verificou-se a facilitação no processo de acesso, utilização, manutenção e compartilhamento de ideias, pois o método atual utiliza um sharepoint para que os funcionários possuam acesso ao documento.

Palavras-chave: aplicação web; inovações; Roadmap.

ABSTRACT

This project aims to develop a web application for the Volkswagen company, aimed at registering innovations proposed by employees, with the objective of transforming the method currently used, which is an excel spreadsheet, to a web system containing the functionalities already used in the spreadsheet. By transforming the Roadmap into a system, the process of accessing, using, maintaining and sharing ideas was facilitated, as the current method uses a sharepoint so that employees have access to the document.

Keywords: web application; innovations; Roadmap.

LISTA DE FIGURAS E TABELAS

Figura 1 – Fábrica de São José dos Pinhais 10
Figura 2– Fluxo Scrum.....16
Figura 3– Ferramenta MySQL workbench.....20
Figura 4– Exemplo de Requisição.....21
Figura 5– Repositório do Projeto.....24
Figura 6– RoadMap Atual.....26
Figura 7– Operações Padrões.....27
Figura 8– Protótipo de Tela.....28
Figura 9– Conteúdo.....29
Figura 10– Conteúdo Tela Principal.....30
Figura 11– Injeção de dependência 31
Figura 12– Classe repositor.....31
Figura 13– Classe controladora 31
Figura 14– Entidade.....32
Figura 15– DTO da inovação 33
Figura 16– Imports MUI.....34
Figura 17– Constantes e Axios.....34
Figura 18– Componentes Restantes.....35
Figura 19– Função onClick e Axios.....36
Figura 20– Bibliotecas Segunda Página.....36
Figura 21– Função “Segundo” 37
Figura 22– Método “handleSubmit” 37
Figura 23– Subpáginas.....38
Figura 24– Login Postman.....39
Figura 25– Conteúdo do Postman.....40
Figura 26– Collections.....41
Figura 27– Collections.....42

Tabela 1– 5W2H.....25

LISTA DE ABREVIATURAS E SIGLAS

API *Application Programming Interface* (Interface de programação de aplicações)

DTO *Data Transfer Object* (Objeto de transferência de dados)

JPA *Java Persistence API* (API de persistência java)

JSF *JavaServer Faces*

NPM *Node Package Manager* (gerenciador de pacotes)

SQL *Structure Query Language*

UI *User Interface (*interface de usuário)

SUMÁRIO

1. INTRODUÇÃO 9

1.1 CONTEXTO EMPRESARIAL 9

1.2 PROBLEMÁTICA 11

1.3 OBJETIVO GERAL 11

1.5 JUSTIFICATIVA 12

2 FUNDAMENTAÇÃO TEÓRICA 13

2.1.1 Inovação 13

2.1.2 RoadMap de Inovação 13

2.2 PROCESSO DE DESENVOLVIMENTO 14

2.3 FUNDAMENTAÇÃO DAS TECNOLOGIAS 16

2.4 *FRONT-END* 16

2.4.1 Javascript, React e MUI 17

2.5 *BACK-END* 18

2.5.1 Java 18

2.5.2 Spring Boot 19

2.5.3 MySQL 19

2.5.4 Postman e Serviços RESTful 20

2.5.5 Node.js e Axios 21

3 METODOLOGIA 22

3.1 SCRUM 22

3.2 TRELLO 22

3.3 GIT E GITHUB 23

3.4 FERRAMENTA 5W2H 24

4. RESULTADOS 26

4.1 IDENTIFICAÇÃO DO PROCESSO ATUAL 26

4.2 PROPOSTA DE PROJETO 27
4.3 EXECUÇÃO E RESULTADO 28
4.4 DESENVOLVIMENTO BACK-END 30
4.5 DESENVOLVIMENTO FRONT-END 33
4.5.1 Primeira Tela 33
4.5.2 Segunda Tela 36
4.5.3 Subpáginas 38
5. TESTES DE API 38
5.1 CRIANDO UMA COLLECTION 40
6. CONCLUSÃO 42

1. INTRODUÇÃO

Atualmente nas organizações existe um desafio no que tange integrar tecnologias atuais com antigas, ou seja, a dificuldade em manipular sistemas legados. Estes sistemas acabam atrapalhando no processo de modernização e aquisição de novas ferramentas, técnicas e novos processos de inovações (COSTA, 2018).

Partindo desse princípio de inovações foi desenvolvido internamente na empresa Volkswagen um RoadMap em forma de planilha aplicada em um excel, onde possibilita a cada setor da empresa listar planos de ações para o desenvolvimento de inovações que ofereçam melhorias no ambiente de trabalho, nos processos de produção e até mesmo no produto oferecido pela fábrica. Esse RoadMap pode ser usando tanto para listar as propostas de inovações, como uma forma de gerenciamento das necessidades de cada setor, já que o plano de ação é proposto por colaboradores que presenciam essa necessidade no dia a dia, utilizando o RoadMap como uma plataforma auxiliar para gerenciamento, permite verificar prazo para aplicação da solução, investimentos necessários, despesas, quantidade de pessoas, e benefícios que a inovação deve fornecer para o setor em si e para a indústria com um todo também.

1.1 CONTEXTO EMPRESARIAL

A Volkswagen é a maior produtora de veículos e a maior exportadora de veículos da história do Brasil, tendo por volta de 68 anos de mercado brasileiro e atualmente com mais de 15 mil funcionários atuando em quatro fábricas, um centro de peças e escritórios regionais espalhados no país (VOLKSWAGEN, 2022).

A empresa foi marcante para a cadeia de fornecedores do Brasil com a fabricação da primeira Kombi com 50% das peças nacionais em 1956, sendo decisiva para a economia brasileira, marcando a história do Brasil assim como em 1959, que Juscelino Kubitschek, desfilando na fábrica de Fusca conversível, entre outros marcos.

Figura 1 – Fábrica de São José dos Pinhais

Fonte: Volkswagen do Brasil (2021)

A unidade presente na cidade de São José dos Pinhais/PR (figura 1), inaugurada no dia 18 de janeiro de 1999, foi a precursora na utilização do robô colaborativo, trabalhando junto aos empregados na área de montagem, aumentando a flexibilidade e segurança do processo. Também houve muitos avanços na área de armação, como novos equipamentos de medição in line, um sistema que pausa a linha de montagem quando apresentam dados fora do especificado. Já na linha de Pintura, foi preparada ao longo de muitos anos para produzir veículos com pinturas bi-tone, acompanhando as tendências no mercado.

A inovação está presente em grande parte das áreas da montadora da Volkswagen, pretendendo evoluir e acrescentar mais profundidade em seus projetos.

1.2 PROBLEMÁTICA

Sendo uma das indústrias que mais crescem no Brasil, a Volkswagen é uma empresa altamente preocupada com inovações, seja com tecnologias para aplicarem no desenvolvimento de seus produtos, nos produtos em si e no processo de trabalho no dia a dia na fábrica.

Visando sempre em melhorias, foi desenvolvido pela organização um RoadMap de inovações, com o objetivo de listar melhorias divididas por setores, onde é possível listar propostas de inovações e estimar coisas como: o responsável pela ideia, departamento, tempo de aplicação da inovação, ganhos previstos, recursos necessários, time de trabalho entre outros.

Atualmente tal RoadMap está sendo utilizado e desenvolvido pela ferramenta Excel e compartilhado com os colaboradores através de um sharepoint, um meio colaborativo de edição de documentos. Devido a aplicação estar em uma ferramenta limitada e que não possui suporte, a utilização do mesmo acaba sendo obsoleta, necessitando o objetivo é transformar esse RoadMap em um sistema intuitivo, moderno, e que facilitaria o acesso, utilização e a manutenção desse sistema.

1.3 OBJETIVO GERAL

O objetivo principal é desenvolver uma aplicação WEB, que apresenta as mesmas funcionalidades presentes no documento Excel, além de outras funções que priorizem a manutenção e usabilidade do sistema.

1.4 OBJETIVOS ESPECÍFICOS

O projeto tem como objetivos específicos:

- Tornar o processo de acesso mais prático.
- Oferecer melhorias no processo de controle de inovações.
- tornar o RoadMap mais agradável visualmente

1.5 JUSTIFICATIVA

Atualmente a empresa Volkswagen utiliza como ferramenta de RoadMap, um arquivo de extensão .xlsm, aberto pelo programa Excel e compartilhado com os colaboradores através de um sharepoint. Devido às limitações do próprio programa da Microsoft, como a falta de manutenção, segurança e organização, tornou-se necessário por parte da empresa a alteração deste sistema.

Tendo em vista os estudos em conjunto com a empresa, visamos criar uma aplicação web para ser a arquitetura candidata, definida para facilitar o fluxo de acesso, utilização e manutenção desse sistema, além de gerar uma identidade visual facilitando a compreensão e o controle das inovações propostas juntamente com seus componentes.

2 FUNDAMENTAÇÃO TEÓRICA

2.1.1 Inovação

O surgimento de novas tecnologias nos setores industriais ocasionou mudanças positivas, aparecendo novas estratégias de mercado e outras alterações de comércio. Tais aparições citadas podem ser chamadas de “Inovações”, segundo Cristina Lemos (2009), porém, podem ser divididas em incrementais ou radicais.

Pode-se entender como inovação radical aquela que traz novos produtos, formas de organização na produção e de processos, ocorrendo uma ruptura entre a estrutura tecnológica anterior, originando novos mercados, setores e indústrias. Em certos casos, Freeman (1988) relata que é possível que haja redução de custos e elevação da qualidade dos produtos oferecidos por este meio.

Inovações incrementais, em sua totalidade, visam a melhoria de um produto, processo ou na organização de uma produção já existente. Freeman também cita que, devido a sua composição de “evolução”, não há alterações em sua estrutura industrial e que, devido a isto, tal inovação acaba sendo imperceptível para o consumidor, mas acaba por afetar positivamente a eficiência técnica, assim como sua qualidade e possíveis ampliações de um produto ou um processo.

2.1.2 RoadMap de Inovação

Segundo Coetes (2001), as organizações que realizam o monitoramento de informações tendem a identificar oportunidades de inovação, oferecendo-lhes vantagens competitivas no sentido de estabelecer planos estratégicos que visam novos produtos e inovações no mercado. Araújo e Chueri (2018) reconhecem a importância que os processos de pesquisas e inovações tecnológicas complementam, e garantem que tais métodos conferem visões aplicadas a problemas reais e a desafios para a criação de soluções.

Os RoadMaps incluem desde pesquisas científicas de ciência e tecnologia, previsão para políticas governamentais ou para determinado setor industrial, até estudos da evolução de produtos ou tecnologias. E como o nome sugere, Roadmap é uma espécie de mapa que aponta como será o produto ou projeto a cada período de sua evolução. O objetivo do mapa é alinhar diferentes visões para responder de forma coordenada a perguntas relacionadas com a evolução da ideia. Além disso, é

uma técnica utilizada para organizar de forma detalhada tais pesquisas e informações.

2.2 PROCESSO DE DESENVOLVIMENTO

O planejamento é um dos principais passos durante o desenvolvimento do projeto. Através dele, é possível acompanhar o progresso do trabalho do início até a sua conclusão. Dessa forma, um cronograma acaba suprindo tal necessidade, justamente por realizar o controle e monitoria de todas as etapas do projeto, exibindo as tarefas que foram concluídas, planejadas, em andamento ou atraso. Também é possível monitorar seus integrantes, e o tempo gasto em cada atividade. Contudo, a organização dos membros para determinadas tarefas, assim como o gerenciamento da estrutura do projeto e reuniões depende de uma técnica, e uma das mais conhecidas chama-se *Scrum*.

Segundo Schawaber (2017), o *Scrum* trata-se de uma metodologia ágil criada para desenvolver produtos complexos, extraindo a mais alta qualidade e agilidade durante o desenvolvimento de um projeto. A principal função do *Scrum* é auxiliar os envolvidos a resolverem problemas complexos de maneira produtiva, criativa e ágil.

O *Scrum* é utilizado desde o início da década de 1990 para desenvolvimento e gerenciamento de grandes projetos. Esta metodologia sugere em como pode-se utilizar várias técnicas e processos para obter excelentes resultados. Este é baseado no empirismo. O empirismo afirma que o conhecimento provém das experiências vivenciadas e tomadas de decisões. Com isso o *Scrum* sugere aperfeiçoamento em previsibilidade e controle de riscos do projeto.

Ainda segundo Schawaber, há três pilares importantes na metodologia *Scrum*, sendo eles: O Time *Scrum* de desenvolvimento, *Product Owner* e o *Scrum Master*. O time *Scrum* trata-se de uma equipe de pessoas nas quais realizarão as tarefas manuais para garantir progresso e resultado ao projeto, integrando em suas rotinas a flexibilidade, auto-organização e multifunção, evitando que o projeto fique estagnado em algum ponto. A pessoa que ocupa a posição de *Product Owner* denomina-se “dona do produto”, responsável por maximizar o valor deste, acompanhar o desenvolvimento e atendimento do *backlog* (requisitos) do produto, gerenciar o time de *Scrum* e demais atividades relacionadas. O *Scrum Master* garante que todos os

envolvidos estejam seguindo a metodologia. Ele também promove e ensina a iterações do *Scrum* ao time.

Há outros três pontos importantes dentro da metodologia, sendo eles: Transparência, Inspeção e Adaptação (SCHAWABER, 2017).

A transparência deve fazer parte dos envolvidos pelos resultados, sendo que a linguagem utilizada entre eles deve ser de fácil compreensão para que haja entendimento do que está sendo apresentado. Os envolvidos devem realizar inspeções constantes em seu projeto, observando possíveis erros, falhas, evoluções e variações.

A adaptação sugere mudanças quando algumas linhas de trabalho se desviam do foco e impactam no resultado do produto. As mudanças devem ser realizadas o mais breve possível para evitar mais adversidades. Todas as questões acima podem ser discutidas em reuniões com os envolvidos, sendo reuniões diárias, planejamento, revisão e retrospectiva.

As *Sprints* dentro da metodologia *Scrum* são como metas, onde são definidos o que será trabalhado, um plano previsto e com flexibilização para guiar a construção do projeto e o produto resultante do trabalho. As *Sprints* são limitadas a um mês, onde neste tempo pode-se ocorrer intervenções e modificações de acordo com o progresso do projeto. (SCHAWABER, K.; SUTHERLAND, J. 2017, p.9). A figura x mostra os fluxos e as repetições necessárias para o atingimento da meta.

Figura 2 - Fluxo *Scrum*

Fonte: O autor (2022).

2.3 FUNDAMENTAÇÃO DAS TECNOLOGIAS

A área tecnológica está evoluindo gradativamente, oferecendo uma vasta variedade de recursos para diversos setores. Portanto, muitas organizações acabam tendo que aderir aos novos meios para aumentar sua produtividade ou evitar erros humanos (REZENDE, 2002).

2.4 *FRONT-END*

O desenvolvimento *Front-End* está inteiramente relacionado ao *client-side*, ou seja, é envolvido com o conteúdo presente no navegador, sendo interface de navegação, design e interação com o usuário (MARIO SOUTO, 2019).

Para desenvolver uma página WEB, é necessário ter conhecimento sobre as tecnologias das quais serão utilizadas, e devido a isto, no presente tópico será relatado quais foram as linguagens, tecnologias e bibliotecas utilizadas para o desenvolvimento do *Front*.

2.4.1 Javascript, React e MUI

Segundo Flanagan (2012) a linguagem de programação Javascript é umas das mais populares no século XXI e vem crescendo como uma grande tendência. Isso se dá pela popularização exponencial da WEB, uma vez que o *core* da linguagem é no *front-end*. Com a criação do node.Js, que leva o JavaScript para as fronteiras do desenvolvimento *Back-End*, muitos programadores têm a possibilidade de trabalhar com um projeto inteiro usando apenas uma linguagem de programação. Elementos dinâmicos como mapas, formulários, cálculos numéricos, animações e várias outras possibilidades podem ser incluídos por meio da linguagem, podendo ser implantado em páginas estáticas. Outro ponto positivo da ferramenta é a construção de aplicativos mobile com *frameworks* javascript, sendo alguns dos mais famosos, *react native.*

React é uma *library* JavaScript *Open Source*, que permite compor interfaces de usuário partindo de códigos chamados de “componentes”. Esta biblioteca consegue renderizar e atualizar os componentes em tempo real, na medida que os dados mudam (REACT, 2022).

O MUI, ou Material-UI, é uma biblioteca de React que oferece um conjunto extenso de ferramentas para uso em interfaces de usuário, fazendo com que seja possível o envio de informações em menos tempo. Os widgets de UI presentes nesta biblioteca são fortemente inspirados pelos princípios do Google, portanto, é fácil para os desenvolvedores criar aplicativos visualmente atraentes. (MUI, 2022)

2.4.2 HTML5 e Css3

Silva (2014) relata que o compartilhamento de informações e pesquisas na década de 90 era precário, havia muitas complicações que acabaram por limitar o uso da WEB. E por meio dessa necessidade, o HTML surgiu, se popularizando entre os desenvolvedores.

O HTML é uma linguagem de marcação utilizada para construção de páginas. Ele possui elementos que proporcionam liberdade para lidar com o código, delimitando ou ajustando os conteúdos inseridos. Portanto, o HTML é um código que estrutura a base do conteúdo de uma página WEB. No ano de 2008, o HTML5

revolucionou o mercado, pois além de dar suporte a áudio e vídeo, ela proporcionou uma semântica atualizada, suporte aprimorado para recursos de aplicativos da WEB e diversos elementos que aprimoraram a experiência para os desenvolvedores (SILVA, 2008, p.26). Atualmente qualquer site que se preze acaba utilizando ao menos uma das ferramentas da “tríade do desenvolvimento web”, onde o HTML5, CSS3 e JavaScript estão presentes.

CSS é uma linguagem auxiliar com propriedades estilísticas, com ela é possível fazer alterações visuais em elementos de linguagens de marcação como HTML. Ele foi desenvolvido em 1996 como uma ferramenta para ajudar os desenvolvedores a substituir as tags de formatação na linguagem HTML. A versão CSS3 entrega aos desenvolvedores novas interfaces e recursos, como módulos, o que acaba tornando a linguagem mais flexível que suas versões anteriores. (SILVA, 2011, p.18).

2.5 *BACK-END*

Diferentemente do *Front-End*, o *Back* é relacionado com o *server-side*, ou seja, ele é responsável para que as funções de um site funcionem corretamente, de forma interna (MARIO SOUTO, 2019).

Como dito anteriormente, para desenvolver uma página WEB, é necessário ter conhecimento sobre as tecnologias das quais serão utilizadas, sendo assim, no presente tópico será relatado quais foram as linguagens e tecnologias utilizadas para o desenvolvimento do *back*end.

2.5.1 Java

Java é uma das linguagens de programação mais populares pelo mundo, sendo utilizada por mais de 12 milhões de usuários espalhados pelo globo. A linguagem está presente em uma infinidade de ambientes, devido à sua característica multi-paradigma e multi-plataforma. Ela é uma linguagem onde é capaz de desenvolver aplicações de gestão*, android* ou até mesmo jogos. Além de sua usabilidade padrão, ainda existem vários frameworks para utilizar em desenvolvimentos, como por exemplo o Spring, Quarkus, Micronauts, JSF (*JavaServer Faces*), dentre outros.

Em 2017, a empresa Oracle anunciou que pretende migrar o Java para uma nova versão a cada seis meses. No dia 22 de março de 2022 foi lançado o Java 18, a versão mais recente da linguagem de programação e plataforma de desenvolvimento. Esta versão oferece diversas novidades relacionadas a desempenho, estabilidade e segurança, além de atualizações e melhorias nas bibliotecas: JEP 400, JEP 408 e entre outras.

2.5.2 Spring Boot

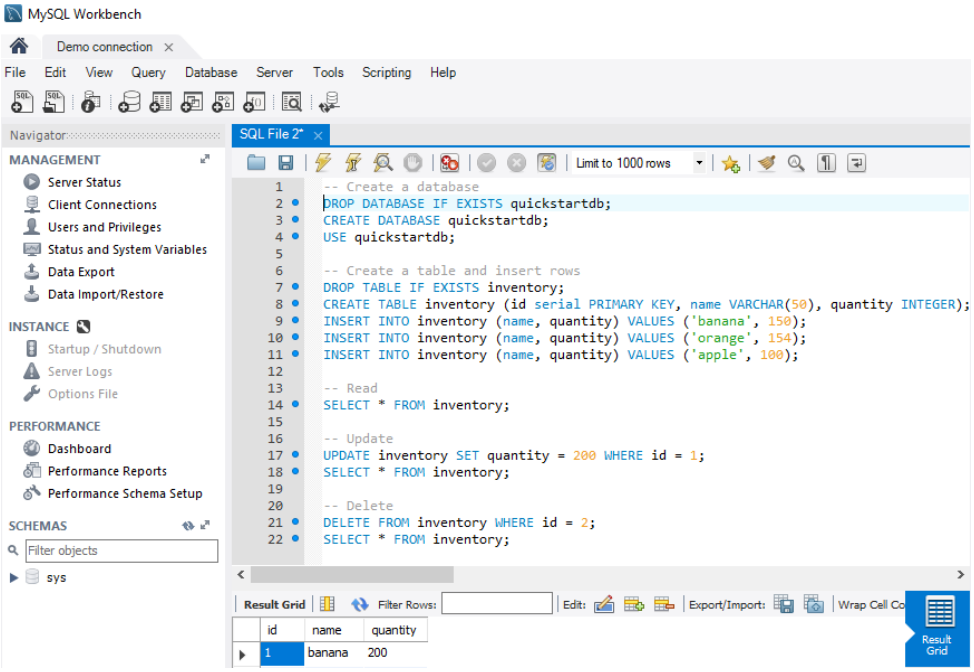
Spring Boot é um *framework* Java de código aberto, que permite a configuração de processos de forma simplificada, configurando apenas o necessário para o funcionamento de um sistema. É possível selecionar os módulos que serão utilizados no projeto, assim como implementar outras configurações posteriormente. (SPRING, 2020)

2.5.3 MySQL

MySQL é um sistema de gerenciamento de banco de dados relacional, com o serviço baseado em linguagem SQL (Structure Query Language), uma linguagem de pesquisa declarativa padrão para banco de dados relacional. De acordo com Heuser (2009), Banco de dados pode ser definido como um conjunto de dados integrados, que tem por objetivo atender um conjunto de sistemas, possuindo modelagem em linhas e colunas numa série de tabelas para que haja uma consulta eficiente dos dados disponíveis

Atualmente o MySQL é amplamente utilizado por empresas que trabalham com grandes volumes de dados, tais como, NASA, Bradesco, HP, Sony entre outras grandes corporações de renome. No entanto, as operações tradicionais de um banco de dados através de linhas de comando podem ser custosas para um desenvolvedor. Devido a isso, surgiram várias ferramentas que apoiam a realização das operações comuns em um banco de dados, como o MySQL *workbench*, de acordo com o exemplo da figura 3.

Figura 3 - Ferramenta MySQL *workbench*



Fonte: O autor (2022).

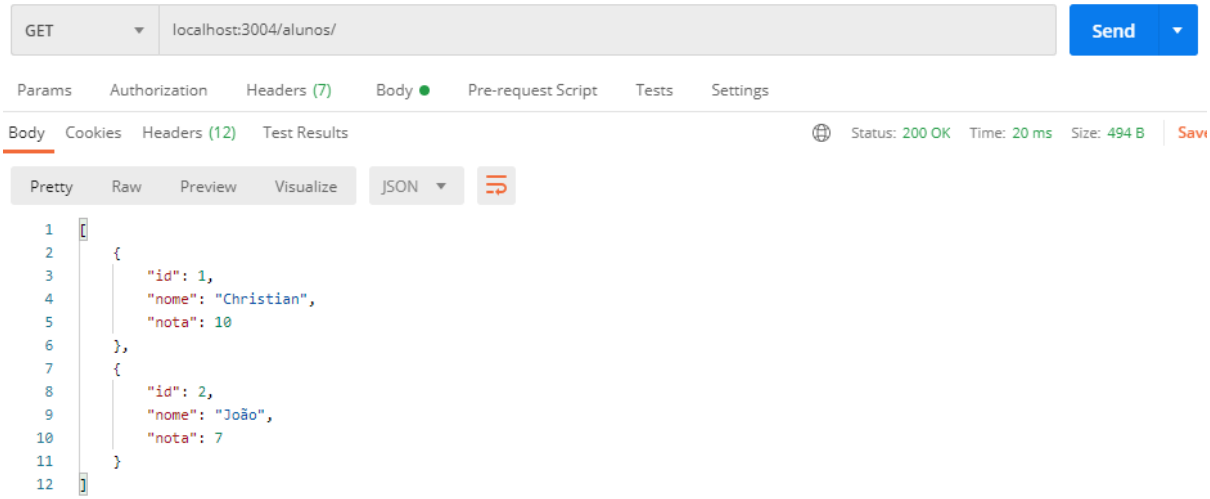
2.5.4 Postman e Serviços RESTful

Postman é uma plataforma de API (*Application Programming Interface*), onde é possível construir, documentar, testar e compartilhar APIs. Essa ferramenta possui como objetivo realizar testes de serviços RESTful, enviando requisições HTTPS, e analisando seu retorno em seguida.

Os sistemas mais recentes que são desenvolvidos baseando-se em serviços, têm se popularizado pela sua flexibilidade, garantindo aos clientes a entrega dos dados sob qualquer aplicação capaz de fazer o consumo do serviço, seja em mobile ou desktop. Segundo Joel Rodrigues (2021), grande parte destes serviços estão sendo desenvolvidos pelo REST, com a troca de mensagens através de requisições HTTP.

No entanto, os serviços HTTP não possuem UI (User Interface), sendo o responsável apenas pelo recebimento e entrega dos dados pelas requisições. Ou seja, para suprir a necessidade que permita realizar requisições HTTP partindo de uma interface, surge o Postman, através dos serviços REST (de acordo com o exemplo da figura 4).

Figura 4 - Exemplo de Requisição



Fonte: O autor (2022).

2.5.5 Node.js e Axios

Node.js é um ambiente de execução JavaScript server-side, ou seja, é possível desenvolver aplicações JavaScript que não seja dependente de um browser para a execução. O NPM (Node Package Manager) é o gerenciador de pacotes do Node.js e é uma subsidiária do GitHub, que fornece hospedagem para desenvolvimento de software e controle de versão com o uso do Git.

Axios é um cliente HTTP baseado em Promises para fazer requisições que pode ser utilizado tanto no Node.js quanto em navegadores mais recentes. Podemos utilizar a mesma biblioteca em vários ambientes: Front End, Back End, Desktop, Mobile, ou seja, Full Stack.

3 METODOLOGIA

Este capítulo descreve as técnicas de elicitação e análise de requisitos envolvidas na elaboração de um sistema de um sistema WEB capaz de suprir a necessidade de um RoadMap.

Metodologia pode ser resumida como estudo das etapas para se chegar a uma determinada finalidade. Possui como objetivo estudar e realizar os melhores métodos praticados em determinada área para a produção do conhecimento, podendo variar amplamente de acordo com as pesquisas.

3.1 SCRUM

A metodologia utilizada pela equipe no desenvolver da aplicação WEB é o *Scrum*, que possibilita oferecer uma gestão dinâmica de projetos; Após a escolha da metodologia, foi definido os papéis de cada integrante do projeto, montando assim o *Scrum* Team.

O *Scrum* tem alguns fluxos básicos e através deles a *Scrum Master* do projeto definiu como a equipe deverá se comportar durante o processo de desenvolvimento do projeto.

- **Reuniões semanais:** Reuniões objetivas que serão realizadas toda terça e sexta-feira a partir das 19:00 através da ferramenta de comunicação: *Discord*.
- **Atividades definidas**: As tarefas foram definidas, para que nenhum membro do desenvolvimento seja sobrecarregado, afetando assim de forma negativa o projeto. Estas atividades foram organizadas e dispostas para visualização da equipe através do site e ferramenta *Trello*.

3.2 TRELLO

Trello é uma ferramenta colaborativa com a função de organizar projetos em quadros, nele é inserido uma lista de tarefas em que um determinado grupo de trabalho deve seguir, cada lista de tarefas recebe um cartão que deve conter a descrição, prazos e objetivos a serem concluídos para realização de um projeto.

O *Trello* tem um sistema de etiquetas com cores que ajuda categorizar as atividades, tornando a filtragem dos cartões visualmente entendíveis. A categorização

das cores é distribuída em dez opções diferentes, algumas delas são vermelhos para urgência, tarefas em andamento podem ser marcadas de amarelo e verde para tarefas já concluídas (TRELLO, 2021).

3.3 GIT E GITHUB

De acordo com Mariot (2020), o *Git* é um sistema de controle de versão distribuída, projetado para lidar de uma melhor forma com grandes projetos, atuando principalmente através das seguintes funcionalidades:

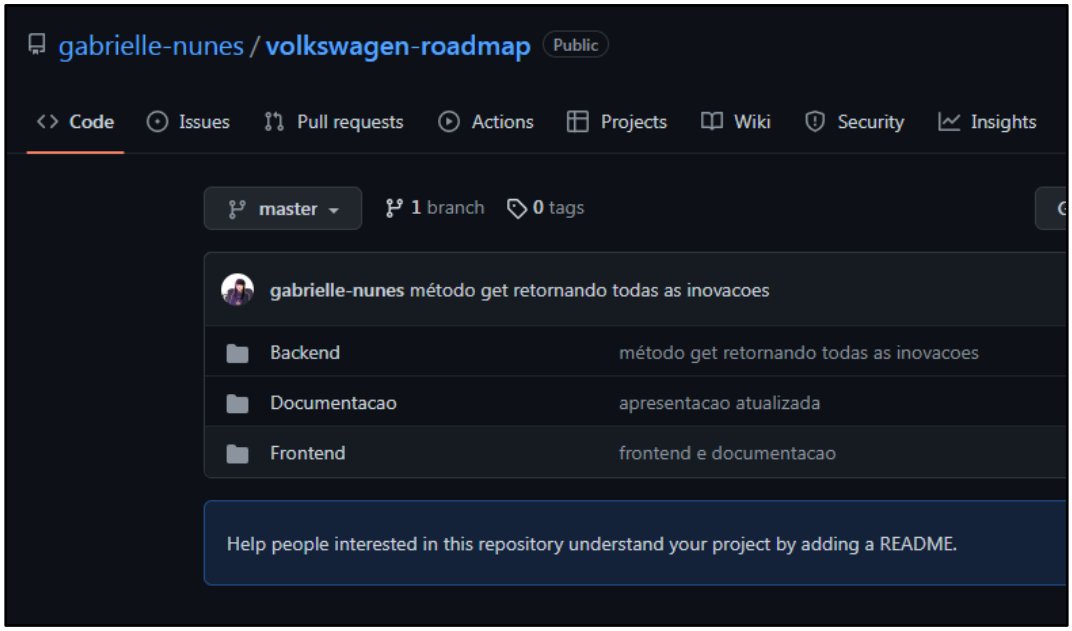
- Revisitar versões anteriores e atualizadas.
- Revisar diferenças entre versões.
- Monitoramento do histórico de versões de um arquivo.
- Marcação de versão para uma rápida referência.

Foi desenvolvido no ano de 2005 por Linus Torvalds, o criador do conhecido *kernel* do sistema operacional *Linux*. Tal sistema é usado para manter um histórico detalhado de alterações nos códigos, que também possibilita a restauração das mesmas para uma versão anterior do aplicativo, se necessário. Além disso, é um modelo usado mundialmente (TSITOARA, 2020).

Diferentemente do Git, que cria repositórios locais, o GitHub é um serviço online e gratuito, considerado por muitos como a “rede social de desenvolvimento de software” devido aos seus aspectos sociais e ferramentas. Ele atua como um servidor, armazenando todas as modificações feitas em um determinado projeto, e unifica diferentes versões do código, que podem ser executadas por outros colaboradores, permitindo que o projeto seja compartilhado entre as equipes de desenvolvimento.

Ter um controle de versionamento é mais que o necessário para o desenvolvimento de um projeto, já que dependendo do tamanho da aplicação e de quantas pessoas estão colaborando com ela, muitas alterações podem entrar em conflito com diferentes versões do código. Como solução, um repositório de nome “volkswagen-roadmap” foi criado dentro da plataforma GitHub, contendo todos os códigos e documentações utilizadas ao longo do projeto (figura 5).

Figura 5 - Repositório do Projeto



Fonte: O autor (2022).

3.4 FERRAMENTA 5W2H

5W2H é uma metodologia que responde a sete perguntas essenciais que determinam atividades específicas. Estas devem ser desenvolvidas com o máximo de clareza por todos os integrantes envolvidos em um projeto.

Essa ferramenta permite uma visualização rápida das necessidades, motivações, atividades a serem realizadas, responsáveis e o que falta para atingir uma meta, pois evidencia as tarefas necessárias até que o objetivo seja alcançado.

De acordo com as necessidades para o desenvolvimento do projeto de RoadMap, as perguntas, que são separadas em: W: What (o que será feito?) – Why (por que será feito?) – Where (onde será feito?) – When (quando?) – Who (por quem será feito?) 2H: How (como será feito?) – How much (quanto vai custar?), foram colocadas dentro de uma planilha com sete colunas, de acordo com a tabela 1.

Tabela 1 - 5W2H

What	Why	Where	When	Who	How	How much
Criação de um sistema WEB	Aumentar a produtividade da criação de inovações	Online	De 15/02/2022 a 21/06/2022	Arthur, Mayane, Rômulo, Gabrielle, André e Rhuan	Desenvolvimento	R$ 0
Capacitação das Tecnologias front-end pela equipe	Obter conhecimento para o desenvolvimento front-end e suas ferramentas	Online	De 15/02/2022 a 21/06/2022	Rômulo, Rhuan e Gabrielle	Treinamentos por vídeos, cursos e artigos diversos	R$ 0
Capacitação das Tecnologias back-end pela equipe	Obter conhecimento para o desenvolvimento back-end	Online	De 15/02/2022 a 21/06/2022	Gabrielle e André	Treinamentos por vídeos, cursos e artigos diversos	R$ 0
Capacitação da Tecnologia Postman	Realizar testes de requisições HTTP	Online	De 15/02/2022 a 21/06/2022	Gabrielle	Treinamento por vídeos e artigos diversos.	R$ 0
Reunião de Revisão	Verificar andamento do desenvolvimento	Online/ Discord	De 15/02/2022 a 21/06/2022	Arthur e Mayane	Reunião com a equipe inteira	R$ 0
Mapeamento do processo atual e proposto	Identificação dos pontos fortes e fracos no processo antigo e novo	Online	De 15/02/2022 a 21/06/2022	Arthur e Mayane	Matriz Swot	R$ 0

Fonte: O autor (2022).

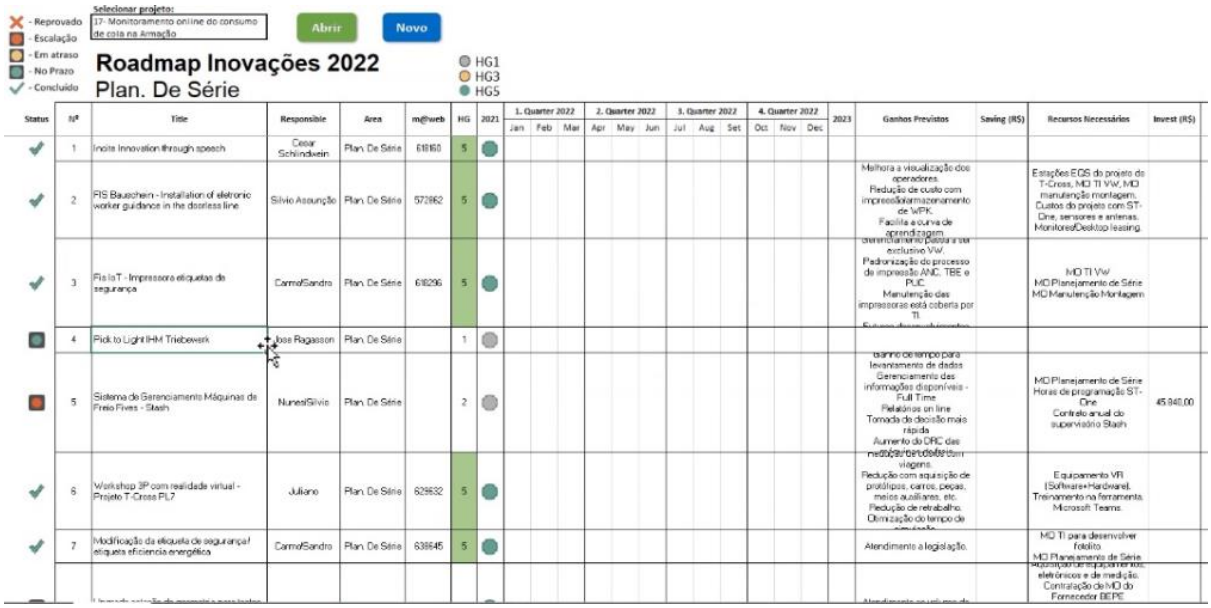
4. RESULTADOS

Neste capítulo serão abordados os resultados obtidos através das tecnologias citadas na fundamentação teórica e dos métodos descritos na metodologia. Assim, permite-se a criação de telas que representam a base desta proposta de desenvolvimento, cobrindo todos os objetivos específicos deste trabalho.

4.1 IDENTIFICAÇÃO DO PROCESSO ATUAL

Na montadora Volkswagen, todas as oportunidades de inovação são descritas estrategicamente no RoadMap, estabelecendo planos para cada tópico, de acordo com a figura 6.

Figura 6 - RoadMap Atual

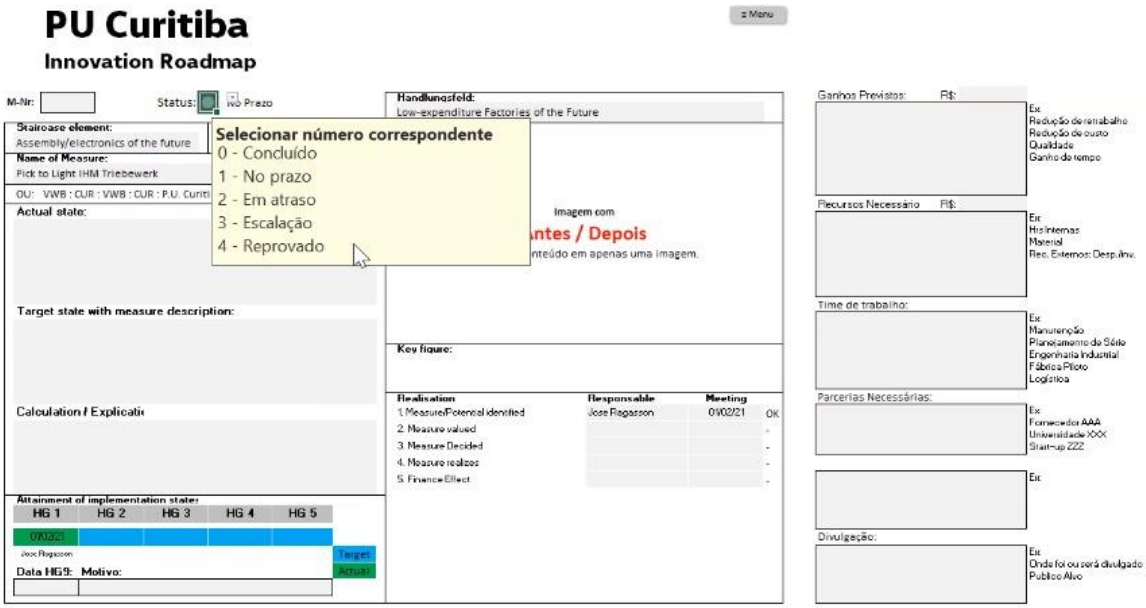


Fonte: Volkswagen (2022).

Nele, existem procedimentos padrões utilizados pelos funcionários, para manipular as colunas de dados, que são: Inserir, Deletar, Editar e Abrir. Ou seja, são operações que em sua totalidade formam um C.R.U.D (Create, Read, Update e Delete), funções básicas de manipulação de dados.

Todas as informações que são contidas para cada inovação, que estão separadas tópicos, são: Título da inovação, Responsável, Área, HG (status atual), Ganhos Previstos, Recursos Necessários, Prazos, Time de Trabalho, Parcerias Necessárias, Pontos de Escalação, Imagens (antes/depois) e Divulgação. Todas de acordo com a figura 7.

Figura 7 - Operações Padrões



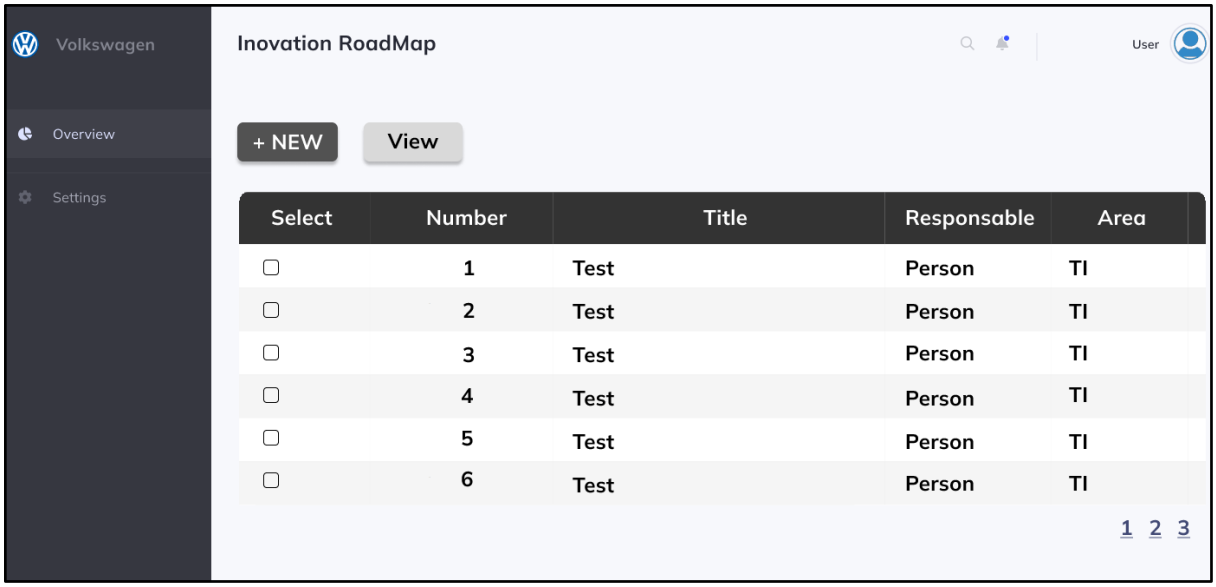
Fonte: Volkswagen (2022).

4.2 PROPOSTA DE PROJETO

Ao identificar os requisitos principais da ferramenta utilizada na empresa, foi formada a proposta de projeto: desenvolver um sistema WEB que apresentasse as mesmas funcionalidades da planilha.

Utilizando a ferramenta de edição gráfica Figma, criamos o protótipo da tela, utilizando como base um monitor de 1440x900 de resolução. O resultado adquirido para o design proposto está de acordo com a figura 8.

Figura 8 - Protótipo de Tela



Fonte: O autor (2022).

As principais funções são: Adicionar, visualizar, editar e por fim deletar (últimos três campos aparecem apenas após selecionar determinada inovação). Também há certas opções genéricas de dashboard, como as Settings (engrenagem presente na barra de navegação), usuário atual, notificações e barra de pesquisa.

Após a definição do design, o próximo passo foi a escolha de quais tópicos presentes na tabela deveriam ser obrigatoriamente preenchidos. Dito isso, optamos por deixar o campo “*Title*”, “*Responsable*”, "*Area*" e “HG” como tópicos obrigatórios.

4.3 EXECUÇÃO E RESULTADO

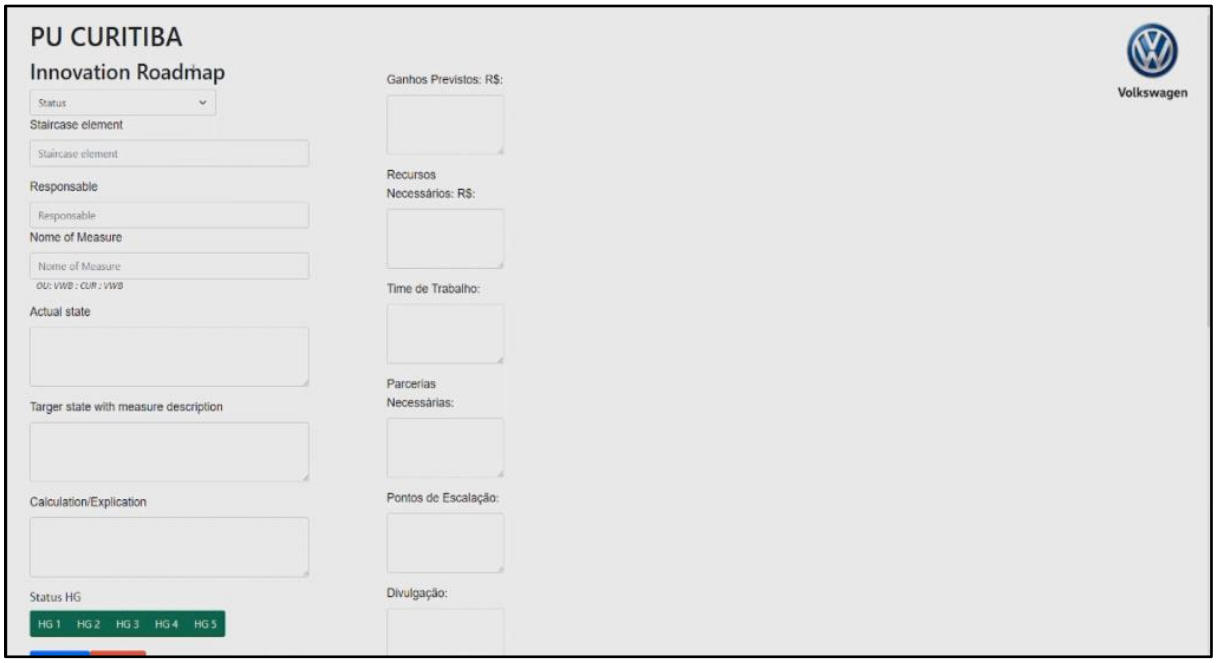
Utilizando o editor de código aberto Visual Studio Code (VS code), o ambiente de desenvolvimento foi criado, baixando as dependências do Node e fazendo a instalação em seguida. Com o editor de código configurado, usamos o NPM (Node Package Manager), que é o gerenciador de pacotes do Node para configurar o servidor local.

Por meio do comando "NPM Start” dentro do console, é possível iniciar o servidor local *(localhost*), possibilitando que o mesmo leia os códigos HTML, JS e CSS, transcrevendo para a tela WEB principal.

Para desenvolver o front-end da página, utilizamos como base o protótipo, com HTML5 para estruturação, CSS3 para o design e algumas bibliotecas como o

bootstrap e MIUI para botões, formulários, tabelas e entre outros, como podemos visualizar na figura 9, que revela a segunda página.

Figura 9 - Conteúdo

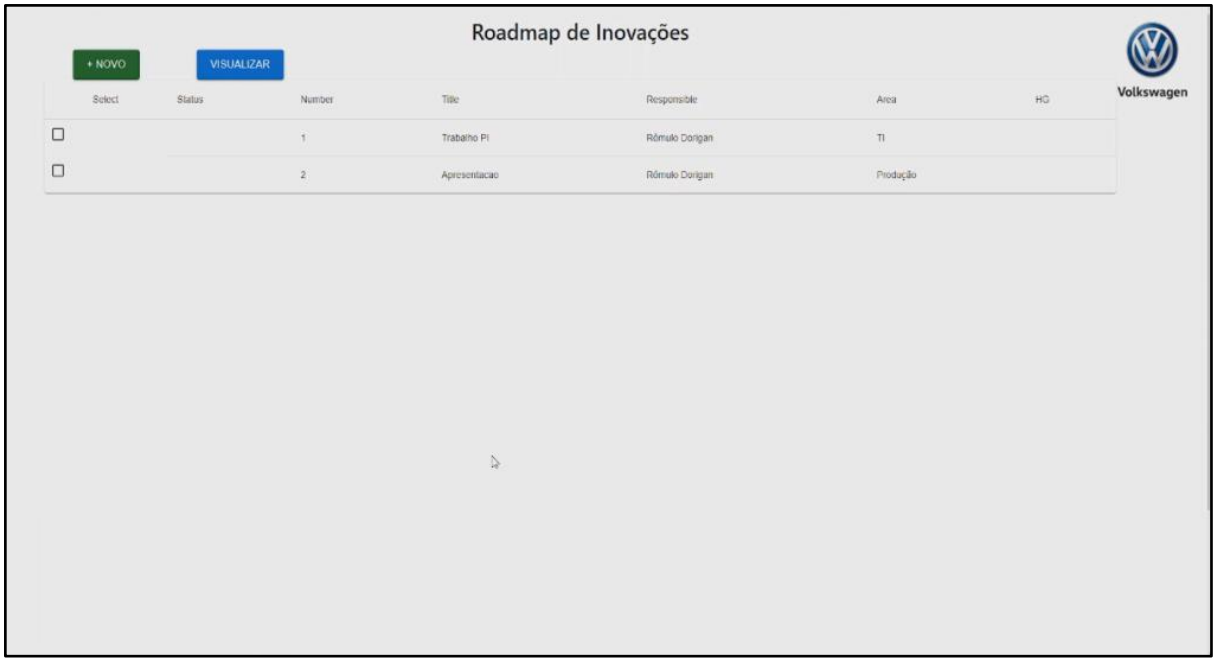


Fonte: O autor (2022).

O back-end, conforme descrito nas tecnologias fundamentadas, foi desenvolvido com Spring Boot e Java (versão 8). Os códigos do backend tem a função de receber dados do usuário, tratá-los e executá-los conforme a solicitação. Criamos métodos para a criação, exclusão, edição e visualização das inovações, utilizando-se dos métodos HTTP (post, put, get e delete).

O cliente Axios foi utilizado como ferramenta que faz a ligação entre *o back-end e front-end*, permitindo a possibilidade de inserção de informações (post, put, get e delete) com o preenchimento dos formulários. Com os dados alterados, as informações presentes na tela principal também se atualizam, de acordo com a figura 10.

Figura 10 - Conteúdo Tela Principal



Fonte: O autor (2022).

Com o uso da biblioteca reactDOM, foi possível configurar as rotas entre cada uma das telas, fazendo a transação correta e permitindo exceções, caso haja incoerências no link.

A ligação das tecnologias back, front e do banco de dados, permite que nós tenhamos como resultado um sistema WEB que permite a inserção de novos dados, edição e exclusão dos mesmos, assim como exceções em determinados casos (campos obrigatórios não preenchidos, por exemplo).

4.4 DESENVOLVIMENTO BACK-END

Para o desenvolvimento do projeto na parte de *back-end* utilizamos um padrão de camadas, dividindo as responsabilidades entre classes, como acesso aos dados através de um DTO – Data *Transfer Object*, camada de serviço (Service) onde os métodos são implementados, auxílio da JPA (*Java Persistence API*) para simplificar métodos CRUD.

A figura 11 demonstra um exemplo de código do que foi implementado e testado para cadastrar uma inovação:

Figura 11 - Injeção de dependência

```
@Service
public class InovacaoService {

        @Autowired
        private InovacaoRepository inovacaoRepository;

        public InovacaoDTO cadastrarInovacao(InovacaoDTO inovacaoDTO) throws Exception {
                Inovacao inovacao = inovacaoRepository.save(Inovacao.convert(inovacaoDTO));
                return InovacaoDTO.convert(inovacao);
        }
```

Fonte: O autor (2022).

A figura 12 trata da Injeção de dependência do *Repository* da Inovação, para que possamos utilizar o método *save*, proveniente da JPA e simplificar o código. A classe *repository,* como mostrado na figura, está estendendo da JPA, e consequentemente nos auxiliando.

Figura 12 - Classe *repository*

```
8   @Repository
9   public interface InovacaoRepository extends JpaRepository<Inovacao, Long>{
10
11  }
```

Fonte: O autor (2022).

O *Controller,* ou classe controladora, presente na figura 13, é a responsável pelas requisições feitas para o endereço do nosso site /inovação/cadastro.

Figura 13 - Classe controladora

```
@RestController
@RequestMapping("/inovacao")
public class InovacaoController {

        @Autowired
        private InovacaoService inovacaoService;

        @PostMapping("/cadastro")
        @Transactional
        public InovacaoDTO cadastrarInovacao(@RequestBody InovacaoDTO inovacaoDTO) throws Exception {
                return inovacaoService.cadastrarInovacao(inovacaoDTO);
        }
```

Fonte: O autor (2022).

A figura 14 demonstra a entidade/tabela onde os dados serão gravados. Gets/Sets omitidos.

Figura 14 - Entidade

```
@Entity
@Table(name = "tbl_inovacao")
public class Inovacao {

        @Id
        @GeneratedValue(strategy = GenerationType.IDENTITY)
        private Long id;
        private String title;
        private String responsible;
        private String area;
        private Integer mweb;
        private String ganhosPrevistos;
        private Integer saving;
        private String recursosNecessarios;
        private Integer invest;
        private String timeTrabalho;
        private String parceriasNecessarias;
        private String pontosEscalacao;
        private String divulgacao;
        private String actualState;
        private String targetState;
        private String calculationExplication;
        private String staircaseElement;
        private String handlungsfeld;
```

Fonte: O autor (2022).

O DTO da inovação, onde são manipulados os dados, evitando-se manipular diretamente na tabela/entidade, para manter as boas práticas (figura 15). Gets/Sets omitidos.

Figura 15 - DTO da inovação

```
public class InovacaoDTO {

        private Long id;
        private String title;
        private String responsible;
        private String area;
        private Integer mweb;
        private String ganhosPrevistos;
        private Integer saving;
        private String recursosNecessarios;
        private Integer invest;
        private String timeTrabalho;
        private String parceriasNecessarias;
        private String pontosEscalacao;
        private String divulgacao;
        private Hg hg;
        private Setor setor;
        private String status;
```

Fonte: O autor (2022).

4.5 DESENVOLVIMENTO FRONT-END

Em sua totalidade, o sistema web contém duas páginas, com a principal incluindo uma tabela com os nomes/informações das novas inovações, enquanto a segunda tela possui um formulário para cadastro de novas inovações.

4.5.1 Primeira Tela

Como citado no tópico 4.3, onde é relatado o uso do MUI para criação dos componentes e dos ícones que compõem a tela, tal biblioteca foi-se utilizado por meio de importações, conforme a figura 16 ilustra.

Figura 16 - Imports MUI

```
import Table from '@mui/material/Table';
import TableBody from '@mui/material/TableBody';
import TableCell from '@mui/material/TableCell';
import TableContainer from '@mui/material/TableContainer';
import TableHead from '@mui/material/TableHead';
import TableRow from '@mui/material/TableRow';
import Paper from '@mui/material/Paper';
//import Checkbox from '@mui/material/Checkbox';
import React, {useEffect, useState} from "react";
import Button from '@mui/material/Button';
import Container from '@mui/material/Container';
import Grid from '@mui/material/Grid';
//import Avatar from '@mui/material/Avatar';
//import { blueGrey} from '@mui/material/colors';
import Stack from '@mui/material/Stack';
import {useNavigate} from 'react-router-dom';
//import Fade from 'react-bootstrap/Fade';
//import Collapse from 'react-bootstrap/Collapse'
//import Segundo from '../PaginaDois/Segundo';
import axios from 'axios';
import "./styles10.css";
import EditIcon from '@mui/icons-material/Edit';
import VisibilityIcon from '@mui/icons-material/Visibility';
import DeleteIcon from '@mui/icons-material/Delete';
```

Fonte: O autor (2022).

A tela principal contém partes de código onde a função “Inicial” possui duas constantes, uma delas chamada de “navegar”, responsável pela navegação entre as duas páginas da aplicação web, já a outra chama-se “posts”, que auxilia no processo de CRUD da página, realizando o GET e o DELETE (figura 17).

Outra biblioteca muito importante é a Axios, pois é ela quem gerencia as requisições HTTPS entre a aplicação web e nosso banco de dados. Percebe-se tal implementação nos métodos axios.get e no axios.delete, presentes na figura 17.

Figura 17 - Constantes e Axios

```
function Inicial() {
    const navegar = useNavigate();
    const [ posts, setPosts ] = useState([])


    useEffect(() => {
        axios.get("http://localhost:8080/inovacao/lista")
        .then((response) => {
            setPosts(response.data)
        })

        .catch(() => {
            console.log("Deu errado")
        })

      }, [])

      function deletePost(id){

        axios.delete(`http://localhost:8080/inovacao/excluir/${id}`)

        setPosts(posts.filter(post => post.id !== id))
      }
```

Fonte: O autor (2022).

O restante do código da página principal está implementado os componentes para montagem da tabela e os ícones/botões para fazer as ações dentro da página, como visualizamos o botão “+ Novo”, presente na figura 18. Realizar uma interação com ele levará para a página de cadastro usado para cadastrar uma nova inovação.

Figura 18 - Componentes Restantes

Fonte: O autor (2022).

Por meio da figura 19, podemos visualizar a função *onClick*, responsável por levar a página inicial para a tela de cadastro, ação realizada conforme já foi citado. Também é possível visualizar, na mesma figura, o Axios em ação com a função {post.xxxx}, que mostra o campo nomeado no banco de dados na aplicação WEB. O {post.map} possui um *get* que existe para mapear os dados já cadastrados no banco, assim como a função *deletePost*, que usa o método *axios.delete* descrito no começo do código.

Figura 19 - Função onClick e Axios

```
        <TableCell align="center">m@web</TableCell>
        <TableCell align="center">HG</TableCell>
        <TableCell align="left"></TableCell>
        <TableCell align="left"></TableCell>
      </TableRow>
    </TableHead>
    <TableBody>
      {posts.map((post, key) => (
        <TableRow>

          <VisibilityIcon className="Icone" align="center" color="action" onClick={() => navegar({pathname: `/visualizar/${post.id}` })} />
          <TableCell align="center">{post.status}</TableCell>
          <TableCell align="center">{post.id}</TableCell>
          <TableCell align="center">{post.title}</TableCell>
          <TableCell align="center">{post.responsible}</TableCell>
          <TableCell align="center">{post.area}</TableCell>
          <TableCell align="center">{post.mweb}</TableCell>
          <TableCell align="center">{}</TableCell>
          <EditIcon className="Icone" align="center" color="info" onClick={() => navegar({pathname: `/edit/${post.id}` })} />
          <TableCell text-align="center">{}</TableCell>
          <DeleteIcon className="Icone" text-align="center" color="error" onClick={() => deletePost(post.id)} />

        </TableRow>
      ))}
    </TableBody>
  </Table>
</TableContainer>
</Container>
</Container>
```

Fonte: O autor (2022).

4.5.2 Segunda Tela

As principais bibliotecas que estão importadas na segunda tela são as do *react* e *bootstrap* para construção dos componentes, conforme podemos visualizar na figura 20.

Figura 20 - Bibliotecas Segunda Página

```
import "./styles.css";
import React from "react";
import Table from "react-bootstrap/Table";
import Form from "react-bootstrap/Form";
import ButtonGroup from 'react-bootstrap/ButtonGroup';
import Button from 'react-bootstrap/Button';
import { useForm } from 'react-hook-form';
import * as yup from "yup";
import { yupResolver } from '@hookform/resolvers/yup';
import {useNavigate} from 'react-router-dom';
import axios from 'axios';
import TextField from '@mui/material/TextField';
```

Fonte: O autor (2022).

A função denominada “Segundo”, conforme a figura 21, contém as constantes para navegação entre páginas e de registro das informações no banco de dados.

Figura 21 - Função “Segundo”

```
function Segundo() {

  const navegar = useNavigate();

  const { register, handleSubmit, formState:{ errors } } = useForm({
      //resolver: yupResolver(schema),
  });

  const addPost = data => axios.post("http://localhost:8080/inovacao/cadastro", data)
  .then(() =>  {
        console.log("Deu certo")
        navegar("/")
})
.catch(() => {
        console.log("Deu errado")

})
```

Fonte: O autor (2022).

O resto do código descreve a criação dos componentes da tela, mas nele é criado um método *“handleSubmit”* (figura 22) para capturar os dados digitados no campo quando o comando POST for recebido, enviando para o banco de dados em sequência. É válido ressaltar que as variáveis pegas pelo *handleSubmit* devem conter o mesmo nome dos campos no banco de dados.

Figura 22 - Método “*handleSubmit*”

```
<main>

  <div className="row">

    <Form onSubmit={handleSubmit(addPost)}>

      <div className="titulo">
        <div className="col-sm-6">
          <h1> PU CURITIBA</h1>
          <h2>Innovation Roadmap</h2>
        </div>
      </div>



      <div className="mweb">
      <TextField id="mweb" label="M-NR:" variant="standard" type="number" name="mweb" {...register("mweb")} />
      </div>

      <div className="btn-status">
        <div className="col-sm-2">
          <h5> Status
          <Form.Select aria-label="Default select example" type="text" name="status" {...register("status")}>
            <option>Escalação</option>
            <option>No prazo</option>
            <option>Em atraso</option>
            <option>Concluído</option>
            <option>Reprovado</option>
          </Form.Select>
          </h5>
        </div>
      </div>
```

Fonte: O autor (2022).

4.5.3 Subpáginas

As subpáginas são compostas pela página de edição (PUT) e de visualização (GET). Todas possuem a mesma estrutura da página de cadastro de inovação (segunda página), mas o que as diferencia são suas respectivas funções: uma é responsável pela edição dos campos da inovação e a outra é apenas para visualizar os dados da mesma.

As mudanças estruturais no código são: a responsável pela edição contém o método GET para retornar os valores do banco para página, para quando o usuário realizar a edição dos campos, os valores pré-cadastrados apareçam. Já o PUT é para reescrever determinado dado em um campo pré-estabelecido, mas para que tal ação funcione, é necessário identificar o ID único do campo dentro do banco de dados, para que quando se atualize a informação, ele consiga fazer a leitura da inovação correta, por isso nos métodos contém o ${id}, para que seja feita essa identificação (figura 23).

Figura 23 - Subpáginas

```
useEffect(() => {
    axios.get(`http://localhost:8080/inovacao/lista/${id}`)
    .then((response) => {
      reset(response.data)
    })
}, [])


const addPut = data => axios.put(`http://localhost:8080/inovacao/editar/${id}`, data)
.then(() =>  {
    console.log("Deu certo")
    navegar("/")
)
catch(() => {
    console.log("Deu errado")
```

Fonte: O autor (2022).

5. TESTES DE API

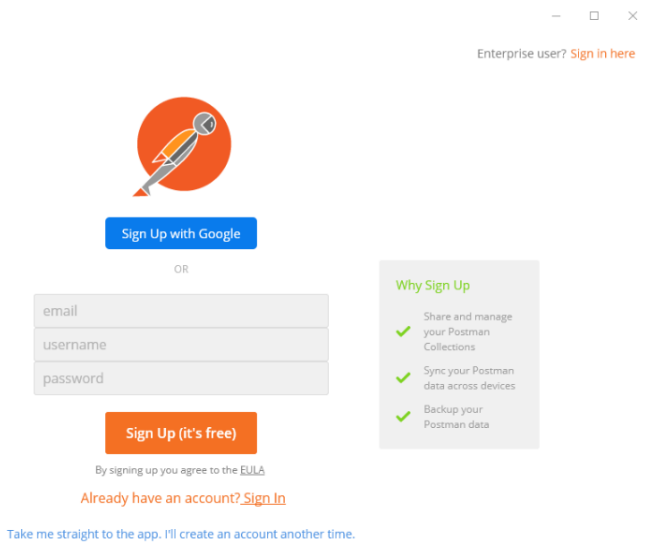
Para cada método criado no *back-end*, foram testados localmente através da ferramenta Postman, simulando uma requisição de um cliente e como o sistema se

comporta. Os códigos de respostas recebidos durante os testes, estão de acordo com o planejado pelos desenvolvedores.

Para utilizar o Postman, é necessário que o aplicativo seja instalado na máquina, mas também há a opção de utilizar a extensão presente na Chrome Web Store.

Acessando o aplicativo pela primeira vez, o programa solicitará, de forma opcional, login e senha. Com a criação de um usuário, as configurações que foram feitas serão mantidas independentemente da máquina que esteja utilizando, desde que utilize a mesma conta.

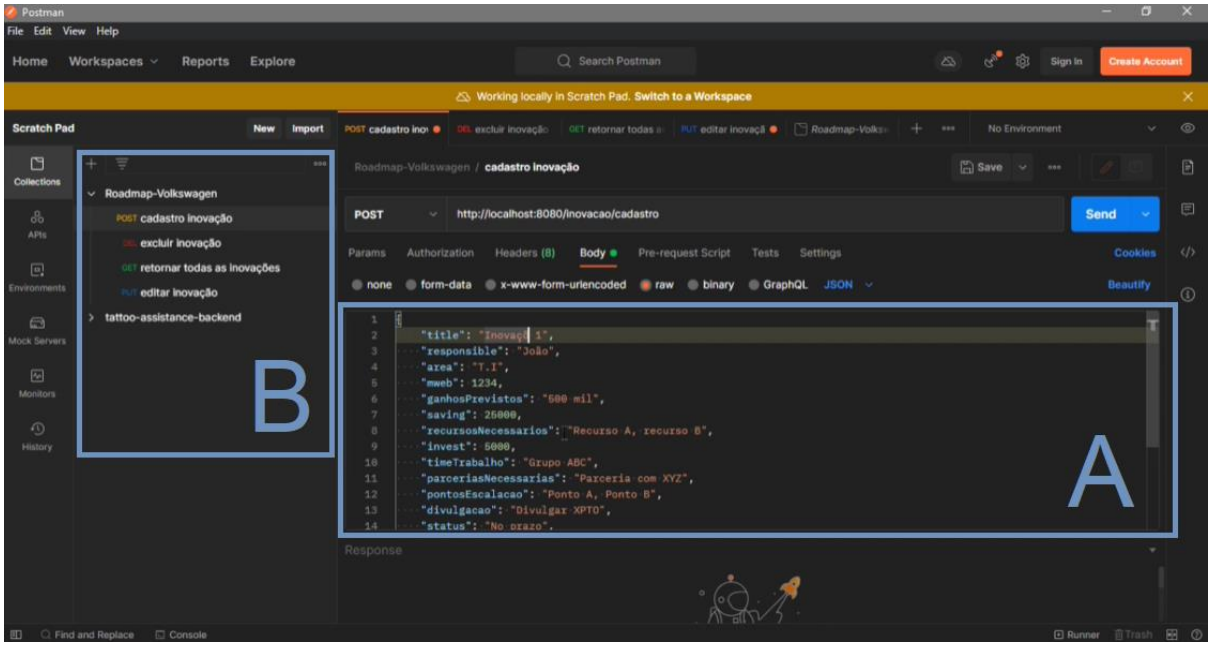
Figura 24 - Login Postman



Fonte: O autor (2022).

A partir do prosseguimento do login, a interface principal do programa vai abrir, com todo conteúdo necessário para configurações primárias, como o ambiente e coleções, como demonstra a figura 25.

Figura 25 - Conteúdo do Postman



Fonte: O autor (2022).

A seleção “A” é o conteúdo das requisições, ou seja, o Body. Aqui é possível alterar ou criar exemplos para a API e validar o conteúdo do POST/GET/PUT.

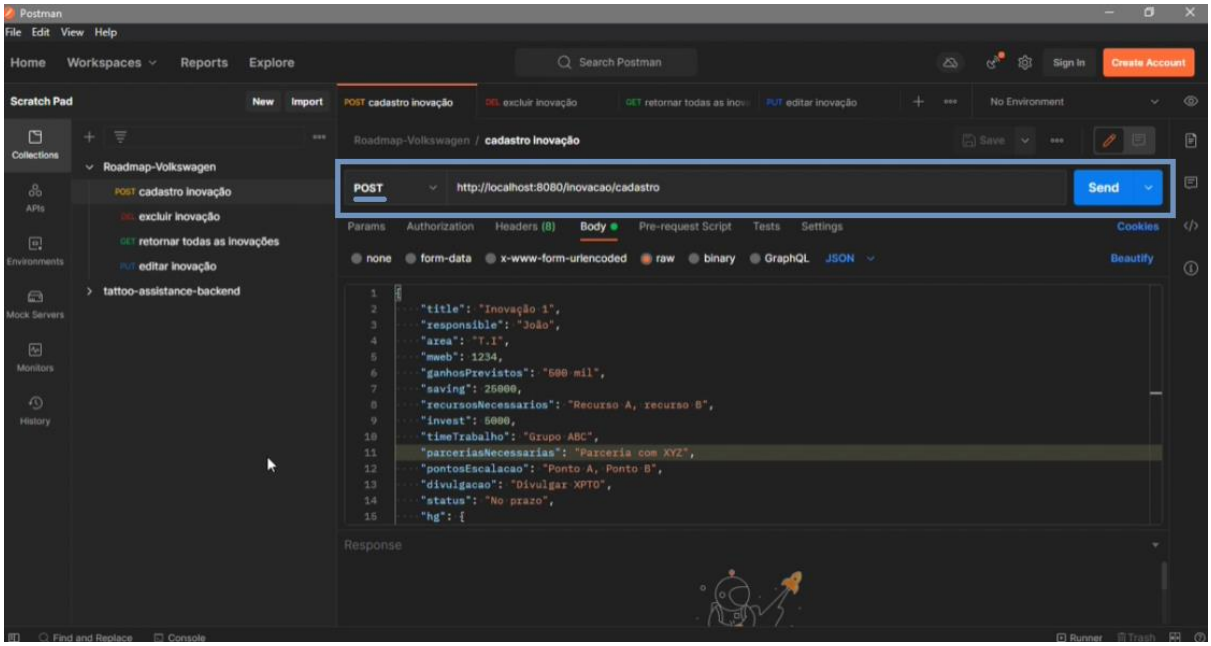
A zona “B” indica o local que contém todos os exemplos e códigos presentes nas Collections. Nesta figura, é possível visualizar que existem quatro tipos de requisição: POST cadastro Inovação, DEL excluir Inovação, GET retornar todas as inovações e PUT editar Inovação.

5.1 CRIANDO UMA COLLECTION

Collection (coleções), como dito anteriormente, é o local onde fica guardado todos os códigos desenvolvidos. Para criar uma nova requisição, é necessário pressionar a opção “New”, sequentemente selecionando o tipo de requisição. O escolhido para nosso projeto é chamado de “HTTP request”.

Dentro do campo “Request URL”, inserimos o diretório correto de cada requisição, todos presentes no localhost. Podemos observar, como exemplo, a figura 26 e 27, que demonstra a funcionalidade.

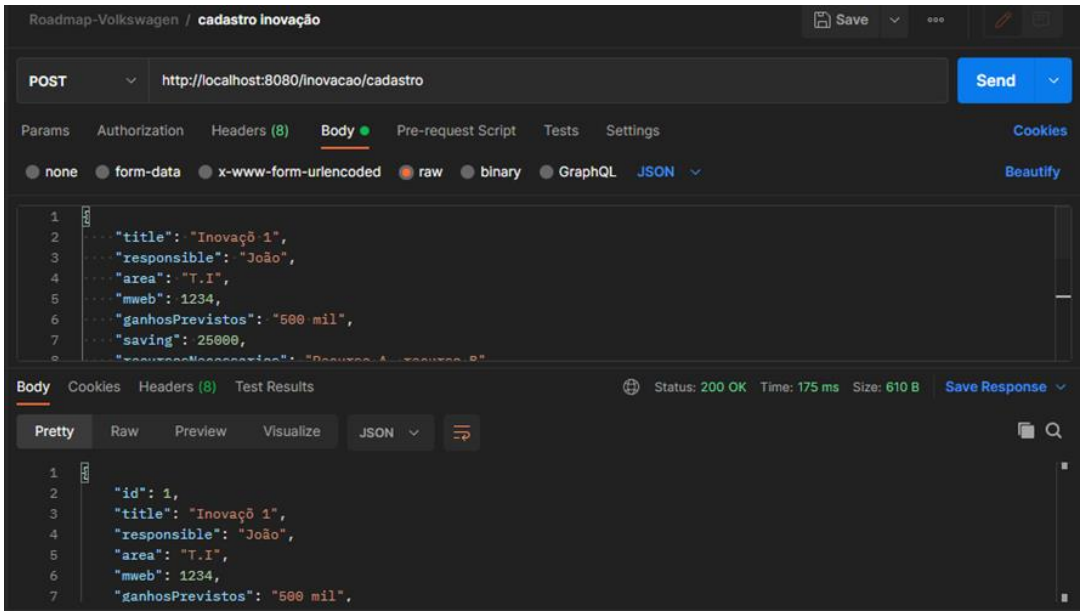
Figura 26 - Collections



Fonte: O autor (2022).

Os testes de Cadastro via Postman acabam retornando os dados persistidos no banco de dados H2, junto de seu ID. O corpo da requisição é um JSON, e consequentemente tendo a resposta da requisição como um JSON, faz o retorno ao banco.

Figura 27 - Collections



Fonte: O autor (2022).

6. CONCLUSÃO

Neste trabalho solidificou-se o desenvolvimento de uma aplicação WEB, capaz de gerenciar os processos de inovação de uma indústria fabricante de veículos em São José dos Pinhais, Paraná. O objetivo, que propõe a produção de um sistema completo que, consiga desempenhar de forma equivalente ou superior a planilha excel foi atingido.

Por meio do uso de diversas tecnologias, a fim de elaborar um RoadMap mais aderente possível, desenvolvemos uma ferramenta moderna e intuitiva, contendo as funcionalidades que já possuíam na planilha e aprimorando sua identidade visual. Portanto conclui-se que os objetivos gerais e específicos determinados previamente pela equipe foram alcançados com sucesso durante o período de criação.

Para melhorias futuras no desenvolvimento do projeto, propomos experimentos utilizando novas tecnologias, realização de testes em condições de fábrica, popularizando o sistema com diversos dados, colaboradores acessando a aplicação simultaneamente para adequarmos o sistema a essas condições, além de melhorias na interface gráfica com a adição de novas funcionalidades.

REFERÊNCIAS

BACK-END: Seu guia sobre back-end: o que é, para que serve e como aprender?. DigitalHouse, 24 out. 2019. Disponível em: digitalhouse.com/br/blog/back-end-o-que-e-para-que-serve-e-como-aprender/. Acesso em: 19 abr. 2022.

COSTA, Fernanda Guimarães. "**Problemas na manutenção de sistemas legados: um estudo de caso**." (2018).

FLANAGAN, David. **JavaScript: O Guia Definitivo**. 6. ed. [S.l]: Bookman, 2012.

FRONT-END: o que é, para que serve e como aprender. Digital House, 1 nov. 2019. Disponível em: https://www.digitalhouse.com/br/blog/front-end-o-que-e-para-que-serve-e-como-aprender/. Acesso em: 19 abr. 2022.

GOMES, André Faria. **Agile: Desenvolvimento de software com entregas frequentes e foco no valor de negócio.** Casa do código, 2018.

JAVA. Java, 2022. Disponível em: https://www.java.com/pt-BR/download/help/whatis_java.html. Acesso em: 12 abr. 2022.

JAVA: O que é tecnologia Java e por que preciso dela?. Java, 2022. Disponível em: https://www.java.com/pt-BR/download/help/whatis_java.html. Acesso em: 20 maio 2022.

JSON, 2021 introduction json. Disponível em:<https://www.json.org/json-en.html>. Acesso

em 11 de mai. de 2022.

LEMOS, Cristina. Inovação na era do conhecimento. **Parcerias estratégicas**, v. 5, n. 8, p. 157-180, 2009.

MUI: Move faster with intuitive React UI tools. [*S. l.*], 2022. Disponível em: https://mui.com/pt/. Acesso em: 19 abr. 2022.

ORACLE anuncia o Java 18. Oracle, 2022. Disponível em: https://www.oracle.com/br/news/announcement/oracle-releases-java-18-2022-03-22/. Acesso em: 1 jun. 2022.

PEREIRA, Adriano; DE OLIVEIRA SIMONETTO, Eugênio. Indústria 4.0: conceitos e perspectivas para o Brasil. **Revista da Universidade Vale do Rio Verde**, v. 16, n. 1, 2018

POSTMAN: O que é o POSTMAN?. Postman, 2022. Disponível em: https://www.postman.com/product/what-is-postman/. Acesso em: 1 jun. 2022.

REACT. React, 2022. Disponível em: https://pt-br.reactjs.org/. Acesso em: 19 abr. 2022.

REZENDE, D. **A evolução da tecnologia da informação nos últimos 45 anos**. Revista FAE Business, 2002, p. 2.

RODRIGUES , Joel. Testando serviços Web API com Postman. Linhadecodigo, 2022. Disponível em: http://www.linhadecodigo.com.br/artigo/3712/testando-servicos-web-api-com-postman.aspx. Acesso em: 25 maio 2022.

SCHAWABER, K.; SUTHERLAND, J. **Scrum Guide.** Ken Schwaber e Jeff Sutherland, 2017.

SILVA, Maurício Samy. CSS3: desenvolva aplicações web profissionais com uso dos poderosos recursos de estilização das CSS3. Novatec Editora, 2019.

SILVA, Mauricio Samy. HTML5: a linguagem de marcação que revolucionou a web. Novatec Editora, 2019.

SOUTO, Mario. O que é front-end e back-end?. Alura, 25 set. 2019. Disponível em: https://www.alura.com.br/artigos/o-que-e-front-end-e-back-end. Acesso em: 26 maio 2022.

SPRING Boot. DevJava, 27 fev. 2020. Disponível em: https://www.devjava.com.br/o-que-e-como-surgiu-e-como-funciona-o-spring-boot/. Acesso em: 19 abr. 2022.

Trello 2021. Disponível em <https://trello.com/>. Acesso em 10 de maio. de 2021.