



DOCUMENTAÇÃO DO PROJETO
CONTROLE DE ESTOQUES MULTIUSO

Gabriel Gasperi Casagrande

LAVRAS - MG
2020

SUMÁRIO

1.Introdução.....	3
2.Levantamento dos requisitos.....	4
2.1 Requisitos.....	4
2.1 Requisitos Funcionais.....	4
2.1.1 Requisitos de Usuário.....	5
2.1.2 Requisitos de Banco de Dados.....	5
2.2 Requisitos Não Funcionais.....	5
2.2.1 Eficiência do Sistema.....	5
2.2.2 Brechas do Sistema.....	5
2.2.3 Confiabilidade do Sistema.....	5
2.2.4 Portabilidade do Sistema.....	5
3.Estratégias de codificação.....	6
4.Resultados alcançados.....	9
5.Conclusão.....	10

1. INTRODUÇÃO

A pandemia ocasionada pelo coronavírus motivou modificações bruscas no cenário mundial, o que consequentemente afetou os diversos setores mercadológicos, como as empresas, que tiveram em grande medida suas fragilidades expostas e novos desafios para serem enfrentados.

Nesse sentido, o objetivo deste projeto é entregar aos gestores de pequenas empresas a alternativa de ter o seu próprio controle de estoques, considerando que ter um planejamento prévio pode ser um facilitador para que a crise seja superada tendo menores prejuízos, uma vez que com o controle de estoque bem estabelecido, amplia-se o manejo estratégico, reduz-se os desperdícios e possibilita que a gestão tenha maior conhecimento sobre a real demanda da empresa, atuando desde o controle da produção até ao investimento financeiro, a fim de garantir o seu futuro.

Assim, o controle de estoques que será apresentado tem como potencialidade a capacidade de se adaptar ao perfil da empresa e as suas necessidades, a praticidade para lançar e retirar os dados, necessitando de pequenos ajustes para que seja possível configurar o estoque mínimo e o estoque máximo dos produtos ou vinculá-los a um fornecedor, por exemplo.

2. LEVANTAMENTO DE REQUISITOS

A princípio foi pensado nas partes indispensáveis do sistema para que ele pudesse ser simples bem como funcional, mas também extensível, para que futuras integrações e adaptações não afetem o sistema base.

O que foi concluído é que o essencial para um estoque seria uma listagem dos produtos que estão disponíveis no mesmo, contendo dados relevantes sobre cada um desses produtos, seja esse estoque contido em um espaço físico acessível ou não. Com isso nos deparamos com o outro elemento chave do sistema que é cada produto e seus atributos específicos.

O sistema é independente e totalmente auto-contido, fornecendo funcionalidades para uma pessoa que pode cadastrar um produto para então visualizá-lo em seu estoque, bem como manipular algum produto específico do estoque a fim de manter o controle e os dados consistentes com a realidade.

Com esses pontos em mente foi iniciado o desenvolvimento do projeto que será melhor detalhado a seguir.

2.1 Requisitos

Os requisitos são uma maneira pelo qual o sistema oferecido deve fazer algo, ou um condicionamento no desenvolvimento do sistema. Os requisitos funcionais fazem parte da arquitetura do aplicativo de um sistema onde há a materialização de uma necessidade ou solicitação realizada, enquanto os requisitos não funcionais denotam a arquitetura técnica de um sistema.

2.1.1 Requisitos Funcionais

- **[RF01] Cadastrar Produto**

O sistema deverá possibilitar a opção de cadastrar um novo produto fornecendo seu Nome, Código de Série, Quantidade em estoque e Valor unitário.

- **[RF02] Remover Produto**

O sistema deverá possibilitar a opção de remover um produto do estoque excluindo seu registro permanentemente.

- **[RF03] Editar Produto**

O sistema deverá possibilitar a opção de editar um produto do estoque, podendo alterar sua quantidade, valor e nome.

- **[RF04] Visualizar produtos em estoque**

O sistema deverá possibilitar a visualização do estoque de produtos em uma tabela podendo ser ordenada por cada atributo.

- **[RF05] Bloquear Cadastro**

O sistema só deverá permitir ao usuário cadastrar um produto quando todos campos forem inseridos.

2.1.2 Requisitos de Banco de Dados

- **[RF05] CRUD de Produto**

O sistema deverá possibilitar as opções de criar, remover, atualizar e ler os dados de cada produto existente no banco de dados.

2.2 Requisitos Não Funcionais

Este tópico descreve os requisitos não funcionais associados à eficiência, uso de recursos e tempo de resposta do sistema.

- **[RNF01] Eficiência do Sistema**

O sistema não deve apresentar travamentos durante a execução.

- **[RNF02] Brechas do Sistema**

O sistema não deve apresentar brechas para ataques SQL injection ao banco de dados.

- **[RNF03] Portabilidade do sistema**

O sistema deve ser portátil executando em qualquer navegador e permitindo fácil extensão.

3. ESTRATÉGIAS DE CODIFICAÇÃO

Feito o levantamento dos requisitos, foi iniciado o desenvolvimento partindo da prototipagem, onde é criado um modelo estrutural do sistema, com o objetivo de transferir as ideias para algo mais concreto e também trabalhar na questão visual do sistema de forma mais prática e rápida, revelando possíveis problemas e adequações.

Para tal, foi utilizado a ferramenta *Adobe XD* tendo como resultado uma boa base para desenvolvimento do sistema, bem como referência de cores e tamanhos para estruturar a página.

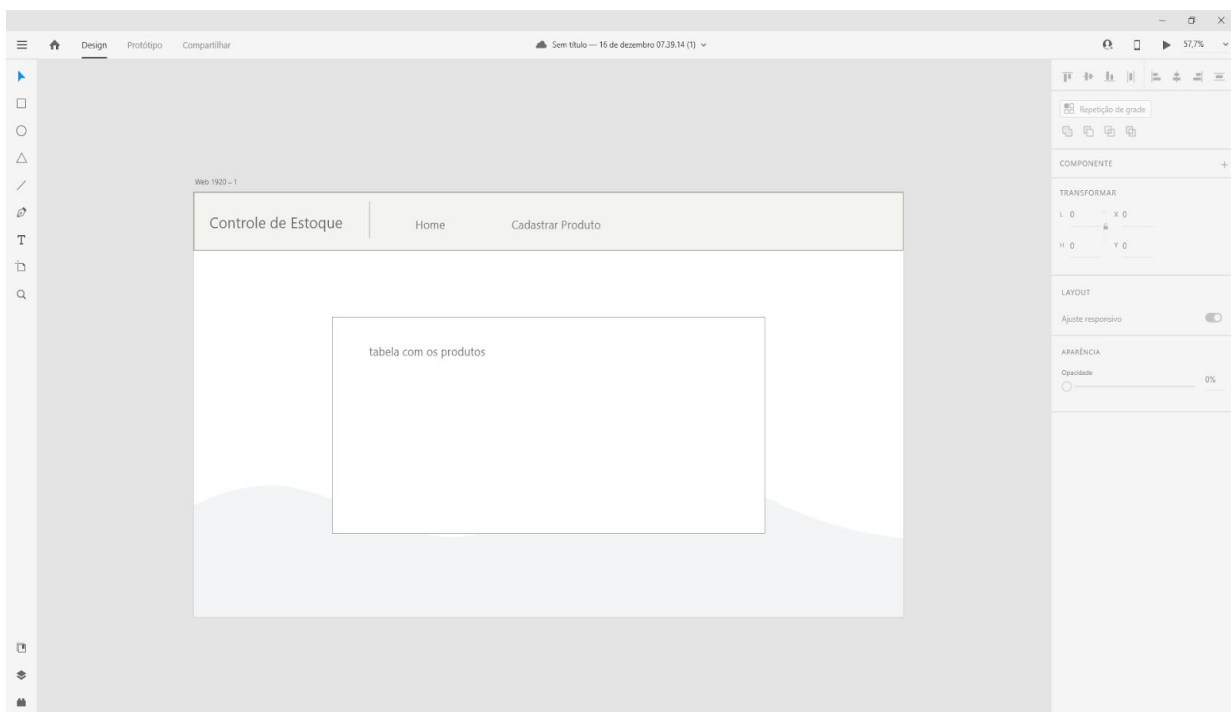


Figura: Protótipo feito no Adobe XD

Com o protótipo definido, foi iniciado o frontend do sistema WEB estruturando as páginas e para isso foi utilizado como ferramenta o framework *JavaScript* chamado **Vue JS**, que é muito popular atualmente por possuir uma sintaxe clara, flexibilidade e performance. Aplicações que utilizam Vue são constituídas de componentes criados com a sintaxe *HTML*, *CSS* e *JavaScript* em um único arquivo *.vue*, que facilitam o isolamento e a manutenção das funcionalidades, além de promover a reusabilidade.

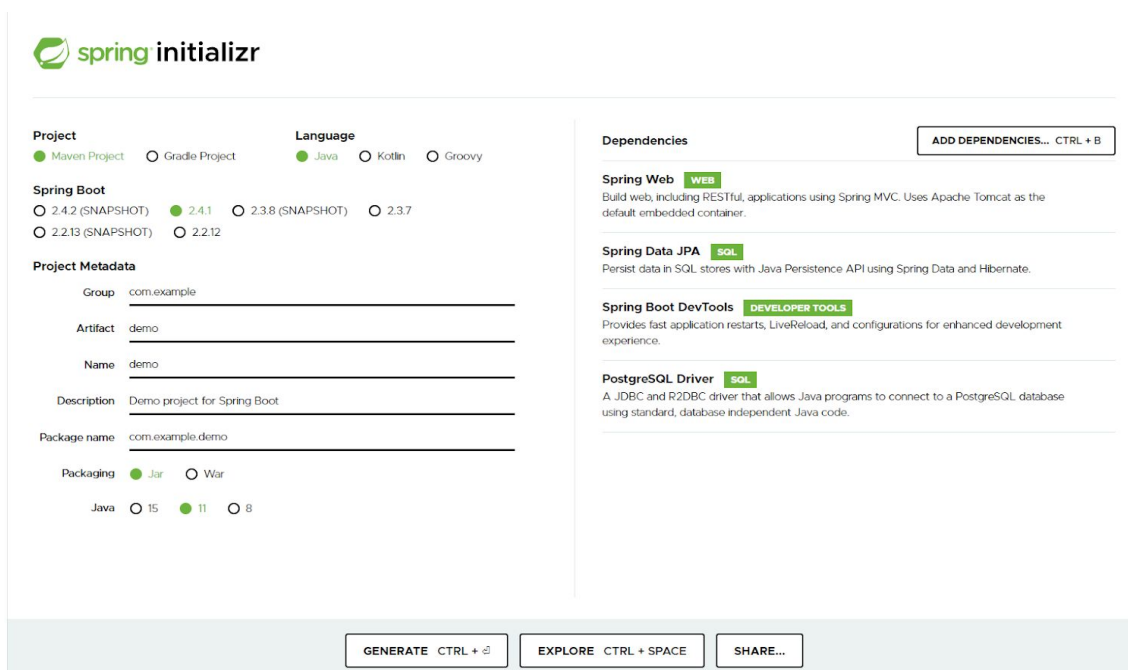
O protótipo ajuda também a visualizar quais serão esses componentes reutilizáveis e que podem ser modularizados, que no caso foram a barra de navegação superior, o

background e o container com o conteúdo de cada tela, divididas na tela principal ou *Home* e na tela de Cadastro de Produto.

Finalizado a estruturação do frontend do sistema, iniciou-se o desenvolvimento do backend visando uma aplicação robusta e modularizada foi utilizado os conceitos de API Rest para o desenvolvimento do mesmo, onde a classe que controla as requisições (*controller*) utiliza dos significados semânticos de cada verbo HTTP para redirecionar as requisições e comunicam os dados geralmente no formato JSON, onde cada aplicação que consumir esses dados será capaz de tratar esse objeto e exibir de forma adequada.

Para o backend, utilizou-se da linguagem Java e da plataforma do Spring Boot que tem como objetivo facilitar o processo de configuração e publicação de nossas aplicações. Ele consegue isso favorecendo a convenção sobre a configuração dos módulos, onde você escolhe os que deseja utilizar e ele vai reconhecer e configurar.

Para essa configuração, o ecossistema Spring possui outra ferramenta chamada Spring Initializr (em *start.spring.io*) onde você seleciona opções como linguagem e suas versões, metadados do projeto e os módulos que deseja utilizar e é gerado um projeto com tudo pronto para ser executado mesmo sem ter desenvolvido.



The screenshot displays the Spring Initializr web interface. On the left, under 'Project', 'Maven Project' is selected. Under 'Language', 'Java' is selected. For 'Spring Boot', version '2.4.1' is chosen. The 'Project Metadata' section includes fields for Group (com.example), Artifact (demo), Name (demo), Description (Demo project for Spring Boot), and Package name (com.example.demo). Packaging is set to 'Jar' and Java version to '11'. On the right, the 'Dependencies' section lists 'Spring Web' (WEB), 'Spring Data JPA' (SQL), 'Spring Boot DevTools' (DEVELOPER TOOLS), and 'PostgreSQL Driver' (SQL). At the bottom, there are buttons for 'GENERATE', 'EXPLORE', and 'SHARE'.

Figura: Configuração inicial do backend do projeto

Os módulos utilizados foram : *Spring Web* que provê configurações para se construir uma aplicação RESTful já com um servidor tomcat embutido que hospeda a aplicação; *Spring Data JPA* que provê os mecanismos de mapeamento objeto-relacional como o *Hibernate*, é ele que fará a comunicação com a base de dados, não sendo necessário utilizar comandos SQL; *Spring Boot DevTools* provê ferramentas para o

processo de desenvolvimento como fast-reloads entre outras; e o driver para conexão com a nossa base de dados que em questão foi utilizado um banco *PostgreSQL* por ser open source e ter um bom desempenho.

Com o projeto inicial pronto, foi desenvolvido o backend partindo da criação da entidade principal, o Produto, bem como suas interfaces de acesso ao banco, os repositórios, e a controladora das requisições que irão expor os recursos da aplicação em caminhos seguindo o padrão de URI para serem acessados externamente.

Com o backend pronto, foi criada a base de dados inicial da aplicação para poder ser validado o funcionamento da aplicação utilizando a ferramenta *Insomnia* que simula as requisições que devem ser feitas para a aplicação e exibem os dados no formato de JSON, no caso.

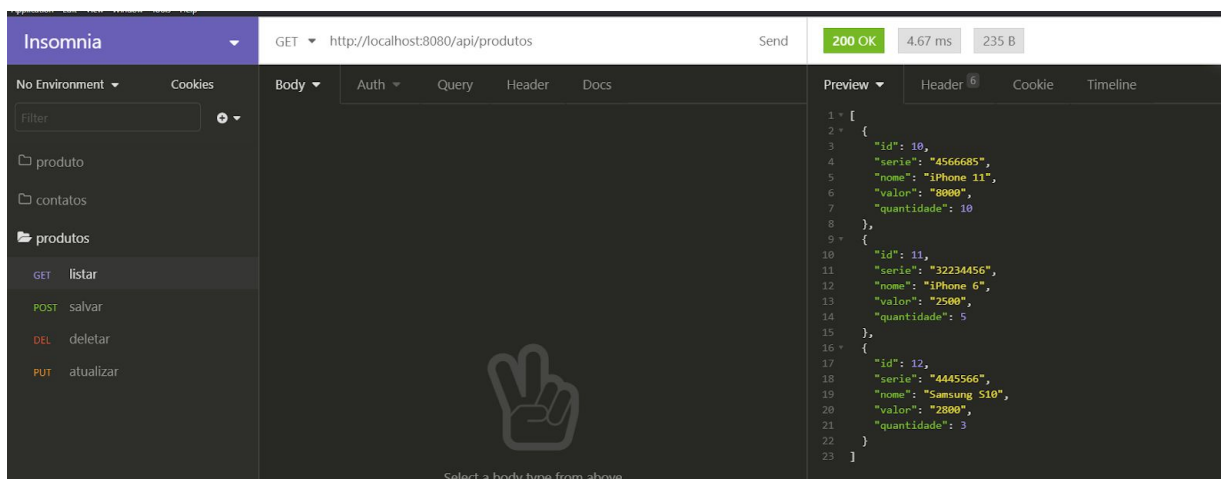


Figura: Validando funcionamento da aplicação pela simulação das requisições pelo Insomnia

Validado o funcionamento das requisições e suas devidas manipulações na base de dados, foi finalizado o frontend realizando a integração com os serviços expostos nas respectivas funcionalidades do sistema, e com isso temos um sistema WEB consumindo uma API RESTful.

Após isso, foram realizados alguns testes básicos na controladora das requisições REST para verificar seu comportamento diante de alguns casos de uso, abrangendo assim todo o conteúdo estudado na disciplina.

4. RESULTADOS ALCANÇADOS

Como resultados alcançados na elaboração deste projeto temos: um sistema Web com interface simples e funcional que traz as informações relevantes ao usuário, como uma listagem de todos produtos em estoque, bem como informações específicas de cada um (*quantidade, valor, número de série*).


Além disso, cada produto pode ter suas informações alteradas a qualquer momento pela modal que é aberta ao clicar sobre sua linha na listagem, tendo suas alterações persistidas na base de dados e atualizadas na página.

Um novo produto pode ser cadastrado pela tela específica de cadastro, situada na barra de navegação que fica na parte superior da página onde o usuário só consegue efetuar o cadastro após todos dados necessários serem inseridos. Para isso foi necessário trabalhar com o mapeamento das rotas pela aplicação para que ela saiba quais telas devem aparecer em determinado caminho.

Testes foram realizados para validar o funcionamento de algumas requisições, porém tive dificuldades em realizar todos eles. Outro ponto é que o sistema pode facilmente ser evoluído pela arquitetura de desenvolvimento possibilitada pelas tecnologias utilizadas.

☰ Controle de Estoque Home Cadastrar Produto				
Esses são seus produtos				
Nº Série	Nome	Quantidade	Valor Un.	
4566685	iPhone 11	10	8000	
4445566	Samsung S12	3	3000	
123123	Motorola Z2	3	2000	
123132	Teste	2	2300	

Figura: Tela principal do sistema

 **Controle de Estoque**

HomeCadastrar Produto

Cadastro de novos produtos

Nome

Nome do produto

Nº de Série

xxxxxxxx

Valor

R\$ Digite o valor unitário

Quantidade

0

✕ Limpar

+ Cadastrar

Figura: Tela de cadastro do sistema

5. CONCLUSÃO

Apesar das fragilidades apresentadas pelo presente Controle de Estoque Multiuso, tem-se a garantia de sua aplicabilidade prática, o que se caracteriza como uma iniciativa para criações mais robustas visto que com os aprendizados construídos durante a disciplina e tendo maior conhecimento das interações mercadológicas pode intervir de forma mais específica dentro dos contextos que forem solicitados.

Porém, o fato de contar com itens básicos, que por sua vez devem estar presentes em todo controle de estoque, possibilita que aos gestores que iniciem o controle de estoque com um menor investimento e gerencie os seus investimentos dominando e tendo confiabilidade para lidar com os passos que são iniciais e subsidiadores para instaurar o uso de controle de estoque mais complexos.