

**CENTRO UNIVERSITÁRIO ESTÁCIO DE GOIÁS**  
**CAMPUS ESTAÇÃO**

## **Sistema Gerenciador de Produtos Escolares**

**Nome do discente**

Jean Carlos Penha da Conceição

**Matrícula**

202308342665

**Nome do(a) professor(a) orientador**

Leomir Aires Silva de Lima

**2024**  
**Goiânia/GO**

## Sumário

1. DIAGNÓSTICO E TEORIZAÇÃO.....	3
1.1.Identificação das partes interessadas e parceiros.....	3
1.2.Problemática e/ou problemas identificados .....	3
1.3.Justificativa .....	4
1.4.Objetivos/resultados/efeitos a serem alcançados (em relação ao problema identificado e sob a perspectiva dos públicos envolvidos) .....	4
1.5.Referencial teórico (subsídio teórico para propositura de ações da extensão) .....	5
2. PLANEJAMENTO E DESENVOLVIMENTO DO PROJETO .....	8
2.1. Plano de trabalho (usando ferramenta acordada com o docente).....	8
2.2. Descrição da forma de envolvimento do público participante na formulação do projeto, seu desenvolvimento e avaliação, bem como as estratégias pelo grupo para mobilizá-los. ....	9
2.3. Grupo de trabalho .....	10
2.4. Metas, critérios ou indicadores de avaliação do projeto .....	10
2.5. Recursos previstos .....	12
2.6. Detalhamento técnico do projeto .....	13
3. ENCERRAMENTO DO PROJETO .....	13
3.1. Relato Coletivo:.....	13
3.1.1. Avaliação de reação da parte interessada.....	15
3.2. Relato de Experiência Individual (Pontuação específica para o relato individual) .....	15
3.2.1. CONTEXTUALIZAÇÃO .....	15
3.2.2. METODOLOGIA .....	16
3.2.3. RESULTADOS E DISCUSSÃO: .....	16
3.2.4. REFLEXÃO APROFUNDADA .....	17
3.2.5. CONSIDERAÇÕES FINAIS .....	17
4. CÓDIGO FONTE DO PROJETO.....	18
4.1. Código JSP .....	18
4.2. Código Java .....	22
4.3. Código CSS .....	30
4.4. Código XML .....	32
4.5. Código SQL .....	35
5. FOTOS.....	36

## 1. DIAGNÓSTICO E TEORIZAÇÃO

### 1.1. Identificação das partes interessadas e parceiros

**Parte Interessada:** O membro discente Jean Carlos Penha Da Conceição, estudante do curso de Ciências da Computação pela Universidade Estácio de Sá. Esse estudante está interessado no desenvolvimento do projeto proposto pela Universidade, conforme estabelecido no acordo de cooperação mencionado neste documento.

**Parceiro:** Empresa de produtos escolares Aprender LTDA, representada por Guilherme Paulo Tavares, que concordou e nos ouviu, gostou da ideia e autorizou para o desenvolvimento do projeto de extensão, disponibilizou os dados dos produtos.

### 1.2. Problemática e/ou problemas identificados

Através de reuniões com a comunidade parceira, foram identificadas diversas problemáticas relacionadas à gestão de produtos escolares para instituição de ensino. As principais questões incluem a falta de um sistema de gerenciamento desses produtos para a análise e controle dos dados de estoque, o que dificulta a tomada de decisões estratégicas.

A empresa enfrentava desafios em monitorar e gerenciar adequadamente os níveis de estoque e prever a demanda. Além disso, a ausência de visualizações de um site web dos dados históricos limitava a capacidade da equipe de identificar tendências e padrões importantes, resultando em uma gestão de estoque menos eficiente e potencialmente mais custosa.

Essa demanda sócio comunitária foi identificada através de encontros e escuta ativa da equipe da empresa, que expressou a necessidade de soluções para seus problemas identificados. Dessa forma, o projeto de extensão foi desenvolvido com o objetivo de criar uma aplicação web que apresente o estoque de produtos e seja possível interações como o cadastro de novos produtos, exclusão e alteração, proporcionando uma ferramenta eficaz para a gestão de estoque da empresa.

### 1.3. Justificativa

Após acompanhar os processos da empresa com encontros presenciais realizando entrevistas com a gestão responsável, através deles recebemos a escuta e os relatos sobre a falta de um gerenciamento mais preciso e eficiente, causando pouca análise e acompanhamento dos produtos do estoque e por sua vez, as análises financeiras não condiziam como margem de lucro real da empresa.

### 1.4. Objetivos/resultados/efeitos a serem alcançados (em relação ao problema identificado e sob a perspectiva dos públicos envolvidos)

**Objetivo Principal:** Facilitar o gerenciamento e um análise precisa, confiável e periódica do estoque da empresa.

#### **Resultados Esperados:**

- Produção de uma tabela de fácil entendimento que auxilia a gestão na tomada de decisão.
- Melhoria na eficiência da equipe responsável pela gestão de estoque.
- Redução do tempo gasto na contabilização manual de vendas.

#### **Efeitos Esperados:**

- Melhor aproveitamento da equipe responsável no gerenciamento do estoque
- Garantia de que todos os produtos estejam adequados às necessidades reais do estoque.
- Redução de atrasos e erros no gerenciamento de produtos.

Esses objetivos e resultados esperados visam solucionar os problemas atuais de ineficiência no processo de gerenciamento de estoque, beneficiando tanto a gestão quanto os funcionários envolvidos.

## 1.5. Referencial teórico (subsídio teórico para propositura de ações da extensão)

### 1.1.1. Java

O Java é uma das plataformas mais utilizadas para o desenvolvimento de aplicações ao redor do mundo. Além da plataforma realizada pelo StackOverflow, o Java, como plataforma de programação, nasceu no ano de 1995 dentro dos laboratórios da empresa Sun Microsystems como resultado de uma extensa pesquisa científica e tecnológica. A plataforma Java entrega um ambiente completo para o desenvolvimento e execução de programas.

No Java, os programas são escritos em um arquivo com a extensão .java, que em um processo posterior serão compilados para arquivos com a extensão .class. Esses, por sua vez, contêm os códigos a serem executados na máquina virtual, os bytecodes.

A plataforma Java é estruturada em um portfólio de produtos para desenvolvimento e execução de aplicações, idealizando que um mesmo programa possa funcionar em diferentes sistemas operacionais e dispositivos. Atualmente a plataforma está dividida em duas grandes áreas:

#### **1 - Java Standard Edition ou JavaSE**

Componente padrão do Java que fornece um ambiente para o desenvolvimento de aplicações de pequeno e médio porte, além de um conjunto de APIs base da plataforma e a JVM padrão.

#### **2 - Java Enterprise Edition ou JavaEE**

Componente baseado no JavaSE, é focado no desenvolvimento de aplicações empresariais multicamadas de grande porte e provê serviços adicionais, ferramentas e APIs para simplificar a criação de aplicações complexas.

#### **Orientação a Objetos**

O Java é uma linguagem orientada a objetos, que é uma forma de programar que possibilita trabalharmos abstração, encapsulamento de código e herança, o que lhe confere também uma curva de aprendizagem bem suave.

**Referência:** Alura. Java: o que é, linguagem e um Guia para iniciar na tecnologia. Disponível em: <https://www.alura.com.br/artigos/java>

Acesso em: 09/11/2024.

### 1.1.2. NetBeans IDE

O NetBeans IDE é um ambiente de desenvolvimento integrado de código aberto e gratuito para desenvolvimento de aplicações nos sistemas operacionais Windows, Mac, Linux e Solaris.

O IDE simplifica o desenvolvimento de aplicações web, corporativas, de desktop e móveis que usam as plataformas Java e HTML5. O IDE também oferece suporte para o desenvolvimento de aplicações PHP e C/C ++.

O NetBeans IDE oferece ferramentas de primeira classe para o desenvolvimento de aplicações Java para web, corporativas, desktops e dispositivos móveis. É sempre o primeiro IDE a suportar as versões mais recentes de JDK, Java EE e JavaFX. Ele fornece visões gerais inteligentes para ajudar você a entender e gerenciar suas aplicações, incluindo suporte imediato para tecnologias populares, como Maven.

Com seus recursos completos de desenvolvimento de aplicações, aprimorando constantemente o Java Editor e aprimoramentos contínuos de velocidade e desempenho, o NetBeans IDE define o padrão para o desenvolvimento de aplicações com tecnologias de ponta prontas para uso.

**Referência:** ORACLE. NetBeans IDE | Oracle Brasil

Disponível: <https://www.oracle.com/br/tools/technologies/netbeans-ide.html>

Acesso em: 09/11/2024.

### 1.1.3. MySQL

O MySQL é o banco de dados de código aberto mais conhecido no mundo. De acordo com o DB-Engines, o MySQL é o segundo banco de dados mais popular, ficando atrás do Oracle Database. O MySQL alimenta muitas das aplicações mais acessadas, como Facebook, Twitter, Netflix, Uber, Airbnb, Shopify e Booking.com. Portanto, é muito provável que sua aplicação ou linguagem de programação favorita seja compatível com o MySQL Database.

Os bancos de dados são o repositório de dados essencial para todas as aplicações de software. Por exemplo, sempre que alguém realiza uma pesquisa na Web, faz login em uma conta ou conclui uma transação, um sistema de banco de dados armazena as informações para que elas possam ser acessadas no futuro.

Um banco de dados relacional armazena dados em tabelas separadas, em vez de colocar todos os dados em um grande almoxarifado. A estrutura do banco de dados é organizada em arquivos físicos [otimizados](#) para velocidade. O modelo de dados lógico, com objetos como tabelas de dados, views, linhas e colunas, oferece um ambiente de programação flexível.

**Referência:** ORACLE. O que é o MySQL | Oracle Brasil

Disponível: <https://www.oracle.com/br/mysql/what-is-mysql/>

Acesso em: 09/11/2024.

#### 1.1.4. Apache Tomcat

TomCat é um projeto da Apache Software Foundation e foi baixado mais de 10 milhões de vezes. É leve, flexível, de código aberto e tem uma penetração extremamente diversa em toda a infraestrutura de TI. TomCat é um servlet Java — um tipo de programa em JavaScript que define como essas solicitações de servidor são tratadas. Ele também atua como um ambiente de servidor HTTP para hospedar páginas estáticas, embora ele seja menos eficaz que o Apache ou o NGINX para essa função. O TomCat é de código aberto, ou seja, ele é mantido por uma comunidade de desenvolvedores que incrementam as características e eliminam falhas para torná-lo cada vez mais funcional.

##### **O que é o Apache TomCat?**

A primeira referência de servlet de código aberto foi criada em 1995, pelo arquiteto da Sun Microsystems, James Duncan Davidson. Mas o projeto de código aberto do TomCat foi lançado pela primeira vez em 1999.

O TomCat é um projeto de nível superior, já foi baixado mais de 10 milhões de vezes e pode ser usado de várias formas em infraestruturas de TI. Ele vem com o mecanismo Coyote, um servidor da Web que permite aos desenvolvedores conectarem uma variedade de aplicativos e recursos Java. O TomCat também cria um ambiente de servidor da Web HTTP no qual o código Java pode ser executado.

##### **Quais os diferenciais do Apache TomCat?**

- É leve
- É de código aberto
- É flexível
- Estabiliza o servidor
- Incrementa a segurança
- É maduro
- É bem documentado

##### **Qual a diferença entre o Apache e o TomCat?**

O TomCat fornece a funcionalidade JSP, não disponível por meio do servidor Apache HTTPS, que é mais adequado para lidar com conteúdo estático e dinâmico — geralmente baseado em PHP —, mas não tem a capacidade de gerenciar Servlets Java e JSP. Na verdade, o TomCat foi originalmente desenvolvido como um meio de fornecer a funcionalidade JSP que faltava ao Apache.

**Referência:** Rock Content. Saiba o que é Apache Tomcat e por que ele é usado

Disponível: <https://rockcontent.com/br/blog/tomcat/>

Acesso em: 09/11/2024.

## 2. PLANEJAMENTO E DESENVOLVIMENTO DO PROJETO

### 2.1. Plano de trabalho (usando ferramenta acordada com o docente)

1. Coleta dos relatórios dos dados.
2. Elaboração das Ideias do Projeto.
3. Desenvolvimento da aplicação Web.
4. Integração das Análises na aplicação.
5. Revisão, realização dos testes e a entrega.

Etapa	Atividade	Prazo	Responsável	Recursos	Formas de Acompanhamento
Coleta dos relatórios dos dados.	Coleta dos dados e relatórios	1 - 2 Semanas	Jean Carlos	Documento txt dos dados	Reuniões Presenciais e virtuais
Elaboração das Ideias do Projeto.	Análise das conversas realizadas	1 semana	Jean Carlos	App de mensagem: Whatsapp	Conversas virtuais
Desenvolvimento da aplicação Web.	Design e desenvolvimento da aplicação Web	3-4 semanas	Jean Carlos	IDE de desenvolvimento: NetBeans	Revisão através de feedbacks da empresa com a metodologia Scrum
Integração das Análises na aplicação.	Inserção dos produtos na aplicação	2- 3 semanas	Jean Carlos	MySQL WorkBench	Ferramenta de consulta de dados
Revisão, realização dos testes e a entrega.	Revisão final e preparação para entrega do projeto	1 semana	Jean Carlos	Internet e Código Java	Ferramentas de desenvolvimento



**Formas de Acompanhamento dos Resultados:** Reuniões Presenciais: Para verificar o progresso de cada etapa e resolver possíveis bloqueios. Feedback Contínuo: Sessões de feedback entre os membros da equipe e com o docente responsável. Testes Funcionais e Revisões: Testes regulares da interface e revisões de código para garantir a qualidade do trabalho.

**Recursos:**

**Ferramenta de desenvolvimento:** NetBeans - Java (Bibliotecas: JavaSQL e tecnologia JSP).

**Linguagens de Desenvolvimento Web:** HTML e CSS.

**Servidor Web:** Tomcat, banco de dados MYSQL.

**Comunicação e Colaboração:** WhatsApp e Teams.

**Equipamento, Materiais e Tecnologias:** Notebook e internet.

2.2. [Descrição da forma de envolvimento do público participante na formulação do projeto, seu desenvolvimento e avaliação, bem como as estratégias pelo grupo para mobilizá-los.](#)

Durante a formulação do projeto, realizamos encontros iniciais com os representantes da empresa para compreender suas necessidades e expectativas. Utilizamos reuniões presenciais e conversas pelos meios de contato da empresa para o levantamento de requisitos.

No desenvolvimento, com a orientação do professor e feedback contínuo da empresa, criamos análises e aplicações baseadas nos dados fornecidos. Iteramos o design e testamos a usabilidade da aplicação web, registrando capturas de tela.

Para a avaliação, apresentamos o projeto final aos representantes da empresa, coletando feedback detalhado através de demonstrações. Posteriormente, utilizamos questionários e entrevistas para avaliar o impacto do projeto e identificar melhorias, documentando-as.

As estratégias de mobilização incluíram comunicação constante via mensagens e ações de sensibilização sobre a importância da colaboração e a criação de um cronograma de atividades compartilhadas, garantindo o envolvimento contínuo dos participantes.

### 2.3. Grupo de trabalho

O discente do projeto foi responsável pela análise de dados do estoque, observando-se que funções específicas foram atribuídas:

- Jean Carlos Penha da Conceição: foi responsável pela coleta dos dados de estoque e no desenvolvimento da aplicação, realizou testes e fez a entrega do projeto.

### 2.4. Metas, critérios ou indicadores de avaliação do projeto

#### 2.4.1. Metas

##### **Produção da Aplicação Web de Fácil Entendimento:**

- **Objetivo:** Criar uma Aplicação Web clara e intuitiva que representem tabelas interativas e realização de análises do estoque.
- **Critério de Sucesso:** Feedback positivo dado pela gerente da equipe de gestão de estoque sobre a clareza das informações e dos dados.
- **Indicador:** Número de dados coletados e aprovados após testes.

##### **Melhoria na Eficiência da Equipe de Gestão de Estoque:**

- **Objetivo:** Aumentar a eficiência da equipe de gestão no processo de análise de estoque.
- **Critério de Sucesso:** Redução do tempo necessário para a análise de dados e tomada de decisão.
- **Indicador:** Comparação do tempo gasto pela equipe antes e depois da implementação da aplicação.

#### 2.4.2. Critérios e Indicadores

##### **Precisão das Análises:**

- **Critério:** Acuracidade das análises geradas a partir dos dados de estoque.
- **Indicador:** Comparação entre os dados analisados e os resultados reais de estoque.

**Confiabilidade das Informações:**

- **Critério:** Consistência das informações apresentadas ao longo do tempo.
- **Indicador:** Número de erros ou discrepâncias identificadas nas análises mensais.

**Periodicidade das Análises:**

- **Critério:** Regularidade na atualização e apresentação das análises.
- **Indicador:** Frequência de atualização dos dados e sua adesão ao cronograma estabelecido.

### 2.4.3. Etapas para Atingir os Objetivos

**Coleta de Dados:**

**Ação:** Recolher dados fornecidos pela empresa.

**Indicador:** Completação da coleta de dados dentro do prazo definido.

**Análise de Dados:**

**Ação:** Realizar análises estatísticas e interpretar os dados coletados.

**Indicador:** Conclusão das análises iniciais e elaboração do banco de dados.

**Desenvolvimento da aplicação Web:**

**Ação:** Criar uma aplicação Web que seja intuitiva e fácil de usar, interagindo com os dados.

**Indicador:** Finalização da aplicação Web e aprovação em testes de usabilidade com usuários.

**Testes e Ajustes:**

**Ação:** Realizar testes com a equipe de gestão de estoque para ajustar e refinar os dados e a interface.

**Indicador:** Feedback da equipe e número de ajustes necessários após os testes.

**Monitoramento e Avaliação contínua:**

**Ação:** Monitorar continuamente o uso da interface e a efetividade dos dados, fazendo ajustes conforme necessário.

**Indicador:** Obtenção de Relatórios de uso e eficácia.

## 2.5. Recursos previstos

### **Materiais:**

**Computadores:** Utilizados para realizar as análises de dados e desenvolver a aplicação Web. Esses computadores são os mesmos utilizados nas atividades regulares dos alunos, portanto, não geram custo adicional.

**Software de Análise de Dados:** Utilizei a ferramenta MySQL Workbench para manipulação de dados. **Ferramenta de desenvolvimento:** NetBeans IDE para Java (com as bibliotecas JavaSQL e tecnologia JSP).

**Servidor Web:** Tomcat, banco de dados MYSQL.

**Linguagens de Desenvolvimento Web:** HTML e CSS para gerar e desenvolver a aplicação Web. Todos os softwares citados estão disponíveis gratuitamente e as bibliotecas são de código aberto.

**Institucionais:** Infraestrutura da Universidade: Utilização dos laboratórios de informática e suas respectivas instalações (internet, energia elétrica), já disponíveis para as atividades regulares dos alunos.

### **Humanos:**

**Estudante:** Integrante da disciplina de extensão que participou ativamente do desenvolvimento do projeto como parte das suas atividades curriculares. Não houve necessidade de remuneração adicional, pois a participação foi integrada ao programa de ensino.

**Professor Orientador:** Professor que auxiliou no projeto, oferecendo orientação e suporte técnicos sobre a linguagem java. A atuação dele está dentro das atribuições normais de sua função.

**Consultores Externos:** Embora não tenha sido necessário contratar consultores externos, mantivemos contato com profissionais da empresa para alinhar as necessidades e validar os resultados das análises.

**Estratégias de Minimização de Custos:** Utilização de banco de dados local através da ferramenta XAMPP, uso de Recursos Gratuitos e a escolha por softwares gratuitos e de código aberto foi estratégica para reduzir ao máximo os custos financeiros do projeto.

**Aproveitamento de Infraestrutura Existente:** Todo o trabalho foi realizado utilizando os recursos já disponíveis na instituição e na empresa selecionada, evitando gastos adicionais.

**Integração com Atividades Curriculares:** A participação do aluno no projeto foi integrada às suas atividades curriculares, eliminando a necessidade de remuneração extra.

## 2.6. Detalhamento técnico do projeto

**Análise dos Dados:** Nessa etapa, coletamos e analisamos dados históricos. Identificamos o desempenho de produtos e outras variáveis relevantes para a empresa. A análise focou em entender as tendências e áreas que poderiam beneficiar de melhorias, como gerenciamento de estoque como o cadastro de novos produtos, remoção e alteração.

**Desenvolvimento:** Com base nas análises realizadas, desenvolvemos tabelas que representam visualmente os dados coletados. Utilizamos ferramentas de codificação para o desenvolvimento, facilitando a compreensão dos resultados. Em seguida, integramos uma interface web intuitiva, permitindo que os usuários naveguem facilmente pelas informações e insights apresentados, além de implementar uma função que permite aos usuários adicionarem, excluïrem e alterarem novos produtos.

**Testes:** Nesta etapa, realizamos testes rigorosos na interface web e no banco de dados desenvolvido. Testamos a usabilidade da interface para garantir que os usuários pudessem interagir de maneira intuitiva e eficiente com as visualizações. Também verificamos a precisão dos dados apresentados nas tabelas, corrigindo quaisquer discrepâncias encontradas. Os testes incluíram feedback da parte interessada para identificar e resolver possíveis problemas de usabilidade e desempenho.

**Implementação:** Com os testes concluídos e todos os problemas resolvidos, auxiliamos os funcionários da empresa no uso da interface, garantindo que pudessem extrair o máximo proveito das análises apresentadas. O projeto visou proporcionar uma ferramenta eficiente para a tomada de decisões baseada nos dados, melhorando a gestão de estoque.

## 3. ENCERRAMENTO DO PROJETO

### 3.1. Relato Coletivo:

Eu, Jean, acredito que o projeto atingiu com sucesso os objetivos socio comunitários estabelecidos. Durante o desenvolvimento e implementação da interface web para a parte interessada, consegui promover um impacto positivo tanto na organização interna da empresa quanto na comunidade atendida por ele. A seguir, detalhe de minhas considerações:

#### 1. Melhoria na Gestão de Estoques:

A análise detalhada dos dados históricos e a visualização clara proporcionada permitiram a empresa uma melhor compreensão das necessidades de estoque. Isso resultou em uma gestão mais eficiente, reduzindo desperdícios e garantindo que os produtos estivessem

disponíveis conforme a demanda dos clientes. A melhoria na gestão de estoque também contribuiu para uma oferta mais consistente de produtos à comunidade, evitando rupturas de estoque.

## **2. Capacitação e Empoderamento dos Funcionários:**

A implementação da interface web permite à empresa interessada acompanhar as sessões de treinamento dos funcionários da empresa. Esse processo não apenas os capacita tecnicamente, mas também os empodera, fornecendo-lhes ferramentas e conhecimentos para tomar decisões mais informadas e baseadas em dados. A capacidade de entender e utilizar dados para decisões estratégicas é um valor agregado que beneficia, a longo prazo, tanto os funcionários quanto a organização.

## **3. Sustentabilidade e Redução de Desperdícios:**

Com uma gestão de estoque mais precisa e informada, a empresa pôde adotar práticas mais sustentáveis, diminuindo o desperdício de produtos. Esse aspecto contribui diretamente para a sustentabilidade ambiental e econômica da comunidade, alinhando-se aos objetivos socio comunitários de promover práticas responsáveis e conscientes.

## **4. Fortalecimento da Parceria Universidade-Comunidade:**

O projeto serviu como um elo de fortalecimento entre a universidade e a comunidade local. A colaboração permitiu que os conhecimentos acadêmicos fossem aplicados em um contexto real, trazendo benefícios tangíveis para a empresa e, por extensão, para seus clientes. Essa parceria reforça a importância do trabalho extensionista e da contribuição mútua entre instituições de ensino e a comunidade.

## **5. Feedback Positivo e Abertura para Novos Projetos:**

Recebemos feedback positivo dos gestores e funcionários da empresa quanto à utilidade e eficácia da interface web. Esse retorno confirma que atingimos nossos objetivos de forma satisfatória e abre portas para futuras colaborações e projetos. A possibilidade de continuidade e expansão do projeto representa um avanço significativo para ambos os lados.

## **6. Conclusão:**

Em resumo, o projeto alcançou seus objetivos sócio comunitários, promovendo uma gestão mais eficiente e sustentável da empresa, capacitando os funcionários, fortalecendo a relação universidade-comunidade e contribuindo para a sustentabilidade local. Eu, Jean, estou satisfeito com os resultados obtidos e ansioso para explorar novas oportunidades de colaboração que continuem a gerar impactos positivos na comunidade.

### 3.1.1. Avaliação de reação da parte interessada

No processo de avaliação da reação das partes interessadas, buscamos obter feedback direto sobre a eficácia e relevância de nossos esforços na empresa com o proprietário, Guilherme Paulo Tavares. Convidei ele a compartilhar suas percepções e opiniões sobre os insights extraídos das análises de dados realizadas e a facilidade de uso da interface web.

#### **Segue o feedback fornecido por ele:**

“Gostaria de expressar minha profunda gratidão pelo excelente trabalho na análise de dados. As análises detalhadas revelaram uma discrepância significativa entre nosso estoque físico e o estoque virtual criado, o que nos permitiu identificar e corrigir erros críticos no processo. Devido às informações obtidas, melhoramos nosso sistema de verificação de estoque, reduzindo perdas. O projeto desenvolvido transformou dados de produtos em insights valiosos, otimizando nossa gestão de inventário, além de permitir que novas tabelas de estoque pudessem ser incluídas facilmente na interface, o que me deixou muito satisfeito.”

### 3.2. Relato de Experiência Individual (Pontuação específica para o relato individual)

#### **- Jean Carlos Penha da Conceição**

Minha função principal no projeto extensionista para a Empresa de produtos escolares Aprender LTDA foi conseguir os relatórios e colaborar na análise dos dados, desenvolver a interface web que apresentaria os resultados dessas análises de forma limpa e precisa e a criação do banco de dados local e implementação na aplicação.

#### 3.2.1. CONTEXTUALIZAÇÃO

Uma experiência bastante interessante, tanto com a empresa quanto com a instituição. Nas conversas debatidas, enfrentamos muitos desafios e problemas, sendo eles como a questão da demora de conseguir os dados de produtos da empresa para dar início ao trabalho, gerando certas indecisões de como iríamos iniciar esse projeto. Minha participação foi dar ideias de como seria, através de pesquisas, eu ajudando nas análises dos registros e insights sobre o desenvolvimento da aplicação.

### 3.2.2. METODOLOGIA

A experiência ocorreu em colaboração com a empresa local, que nos forneceu os dados. O projeto foi realizado ao longo de um semestre, com as seguintes etapas:

**1. Acesso, coleta e análise dos dados:** Eu fui responsável por obter os relatórios da empresa e colaborei na análise desses dados.

**2. Elaboração de ideias e planejamento do projeto:** Eu auxiliei na elaboração das ideias iniciais e no planejamento das atividades.

**3. Desenvolvimento da interface web:** Eu fui o principal responsável pelo desenvolvimento da interface web que apresentaria os resultados das análises.

### 3.2.3. RESULTADOS E DISCUSSÃO:

Inicialmente, esperei que a integração dos dados em uma interface web fosse um processo relativamente simples. No entanto, enfrentei desafios técnicos significativos, como a limpeza e a formatação dos dados, além da criação da interface informativa e visual atrativa.

Durante o projeto, observei a importância de uma análise de dados detalhada e precisa. A interface web que desenvolvi resultou em uma ferramenta intuitiva que facilitou a compreensão dos dados pela empresa. Pessoalmente, senti uma grande satisfação ao ver a interface final funcionando, pois ela agregava valor real ao cliente.

Apreendi a importância da colaboração e comunicação efetiva com os membros da empresa. As habilidades técnicas que desenvolvi na criação de interfaces web e na análise de dados foram valiosas.

As principais facilidades incluíram a divisão clara das responsabilidades entre a parte interessada e eu. No entanto, a limpeza dos dados foi uma dificuldade significativa, exigindo muita atenção e trabalho minucioso.

Para futuros projetos, recomendaria um planejamento ainda mais detalhado na fase inicial, especialmente em relação ao tratamento e organização dos dados.



### 3.2.4. REFLEXÃO APROFUNDADA

Experimentei diversas frustrações, medos, conflitos, indecisões e principalmente na resolução de erros ao desenvolver em java e alguns momentos, felicidade, sensação de missão cumprida, trazendo mais motivação.

Com relação ao relato coletivo na área de psicologia e sociologia através de palestras, vídeos de apoio para tais situações e problemas emocionais.

### 3.2.5. CONSIDERAÇÕES FINAIS

Eu pretendo trabalhar com a parte interessada como coleta de feedbacks, necessidades para melhoria sobre algumas situações, como a pouca quantidade de clientes que frequentam a empresa.

**Perspectivas de trabalho futuros para a extensão:** – Como melhoria do projeto, desenvolvimento de outros projetos que vão facilitar tanto com os clientes quanto com os funcionários.

**Para a pesquisa:** – Pesquisar e estudar formas para desenvolver análises no futuro, se isso for da necessidade da parte interessada. Utilização de ferramentas, tanto físicas quanto as virtuais, para comunicações, soluções de certos problemas.

## 4. CÓDIGO FONTE DO PROJETO

Disponibilizarei o código fonte do projeto nos tópicos seguintes. Porém, caso esteja interessado em executar a aplicação, segue abaixo o repositório no GitHub relacionado ao projeto:

<https://github.com/JeanPTBR/GerenciadorProdutosEscolares>

### 4.1. Código JSP

#### 4.1.1. index.jsp -> Página inicial da aplicação

```
4. <%@page contentType="text/html" pageEncoding="UTF-8"%>
5.
6. <%@ page import="java.util.List" %>
7. <%@ page import="model.Produto" %>
8. <%@ page import="dao.ProdutoDAO" %>
9. <!DOCTYPE html>
10. <html lang="pt">
11. <head>
12.     <meta charset="UTF-8">
13.     <title>Lista de Produtos</title>
14.     <link rel="stylesheet" type="text/css" href="css/style.css">
15. </head>
16. <body>
17.     <h1>Gerenciador de Produtos Escolares</h1>
18.
19.     <table border="1">
20.         <tr>
21.             <th>ID</th>
22.             <th>Nome</th>
23.             <th>Descrição</th>
24.             <th>Preço</th>
25.             <th>Quantidade</th>
26.             <th>Ações</th>
27.         </tr>
28.         <%
29.             ProdutoDAO dao = new ProdutoDAO();
30.             List<Produto> produtos = dao.listarProdutos();
31.             for (Produto produto : produtos) {
32.                 %>
33.                 <tr>
```

```

34.         <td><%= produto.getId() %></td>
35.         <td><%= produto.getName() %></td>
36.         <td><%= produto.getDescription() %></td>
37.         <td><%= produto.getPrice() %></td>
38.         <td><%= produto.getQuantity() %></td>
39.
40.         <td>
41.             <a href="editar.jsp?id=<%= produto.getId() %>">Editar</a>
42.             <a href="excluir.jsp?id=<%= produto.getId() %>">Excluir</a>
43.         </td>
44.     </tr>
45.     <% } %>
46. </table>
47. <br>
48. <center><a href="adicionar.jsp">Adicionar Novo Produto</a></center>
49. <br><br><br>
50. </body>
51. </html>
52.

```

#### 4.1.2. adicionar.jsp -> Página de adicionar novo produto

```

<%@page contentType="text/html" pageEncoding="UTF-8"%>

<%@ page import="dao.ProdutoDAO" %>
<%@ page import="model.Produto" %>
<!DOCTYPE html>
<html lang="pt">
<head>
    <meta charset="UTF-8">
    <title>Adicionar Produto</title>
    <link rel="stylesheet" type="text/css" href="css/style.css">
</head>
<body>
    <h1>Adicionar Produto</h1>
    <form action="salvarProduto.jsp" method="post">
        Nome: <input type="text" name="name" required><br>
        Descrição: <input type="text" name="description" required><br>
        Preço: <input type="number" name="price" step="0.01" required><br>
        Quantidade: <input type="number" name="quantity" required><br>
        <input type="submit" value="Salvar">
    </form>
    <br>
    <center><a href="index.jsp">Voltar à Lista de Produtos</a></center>

```

```
</body>
</html>
```

#### 4.1.3. editar.jsp -> Página de edição de acordo com o ID do produto

```
4. <%@page contentType="text/html" pageEncoding="UTF-8"%>
5.
6. <%@ page import="dao.ProdutoDAO" %>
7. <%@ page import="model.Produto" %>
8. <%
9.     int id = Integer.parseInt(request.getParameter("id"));
10.    ProdutoDAO dao = new ProdutoDAO();
11.    Produto produto = dao.buscarProdutoPorId(id); // Método adicionado no
    ProdutoDAO
12. %>
13. <!DOCTYPE html>
14. <html lang="pt">
15. <head>
16.     <meta charset="UTF-8">
17.     <title>Editar Produto</title>
18.     <link rel="stylesheet" type="text/css" href="css/style.css">
19. </head>
20. <body>
21.     <h1>Editar Produto</h1>
22.     <form action="atualizarProduto.jsp" method="post">
23.         <input type="hidden" name="id" value="<%= produto.getId() %>">
24.         Nome: <input type="text" name="name" value="<%= produto.getName()
    %>" required><br>
25.         Descrição: <input type="text" name="description" value="<%=
    produto.getDescription() %>" required><br>
26.         Preço: <input type="number" name="price" step="0.01" value="<%=
    produto.getPrice() %>" required><br>
27.         Quantidade: <input type="number" name="quantity" value="<%=
    produto.getQuantity() %>" required><br>
28.         <input type="submit" value="Atualizar">
29.     </form>
30.     <br>
31.     <center><a href="index.jsp">Voltar à Lista de Produtos</a></center>
32. </body>
33. </html>
34.
```

4.1.4. salvarProduto.jsp -> recebe os dados do formulário de adicionar.jsp e salva o novo produto

```
4. <%@page contentType="text/html" pageEncoding="UTF-8"%>
5.
6. <%@ page import="dao.ProdutoDAO" %>
7. <%@ page import="model.Produto" %>
8. <%
9.     String nome = request.getParameter("name");
10.    String descricao = request.getParameter("description");
11.    double preco = Double.parseDouble(request.getParameter("price"));
12.    int quantidade = Integer.parseInt(request.getParameter("quantity"));
13.
14.    Produto novoProduto = new Produto(0, nome, descricao, preco,
    quantidade);
15.    ProdutoDAO dao = new ProdutoDAO();
16.    dao.adicionarProduto(novoProduto);
17.
18.    response.sendRedirect("index.jsp");
19. %>
```

4.1.5. atualizarProduto.jsp -> recebe os dados do formulário de editar.jsp e atualiza o produto no banco de dados

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>

<%@ page import="dao.ProdutoDAO" %>
<%@ page import="model.Produto" %>
<%
    int id = Integer.parseInt(request.getParameter("id"));
    String nome = request.getParameter("name");
    String descricao = request.getParameter("description");
    double preco = Double.parseDouble(request.getParameter("price"));
    int quantidade = Integer.parseInt(request.getParameter("quantity"));

    Produto produto = new Produto(id, nome, descricao, preco, quantidade);
    ProdutoDAO dao = new ProdutoDAO();
```

```
dao.atualizarProduto(produto);

response.sendRedirect("index.jsp");
%>
```

#### 4.1.6. excluir.jsp -> Excluir um produto com base no seu ID

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>

<%@ page import="dao.ProdutoDAO" %>
<%
    int id = Integer.parseInt(request.getParameter("id"));
    ProdutoDAO dao = new ProdutoDAO();
    dao.deletarProduto(id);

    response.sendRedirect("index.jsp");
%>
```

### 4.2. Código Java

#### 4.2.1. Produto.java -> Classe que representa o modelo dos dados de cada produto

```
5. package model;
6.
7. /**
8.  *
9.  * @author Jean
10. */
11.
12. public class Produto {
13.     private int id;
14.     private String name;
15.     private String description;
16.     private double price;
17.     private int quantity;
18.
19.     // Construtor padrão
20.     public Produto() {}
21.
22.     // Construtor com parâmetros
23.     public Produto(int id, String name, String description, double price,
        int quantity) {
```

```
24.     this.id = id;
25.     this.name = name;
26.     this.description = description;
27.     this.price = price;
28.     this.quantity = quantity;
29. }
30.
31.
32. // Getters e Setters
33. public int getId() {
34.     return id;
35. }
36.
37. public void setId(int id) {
38.     this.id = id;
39. }
40.
41. public String getName() {
42.     return name;
43. }
44.
45.
46.
47.
48. public void setName(String name) {
49.     this.name = name;
50. }
51.
52. public String getDescription() {
53.     return description;
54. }
55.
56. public void setDescription(String description) {
57.     this.description = description;
58. }
59.
60. public double getPrice() {
61.     return price;
62. }
63.
64. public void setPrice(double price) {
65.     this.price = price;
66. }
67.
68. public int getQuantity() {
```

```

69.         return quantity;
70.     }
71.
72.     public void setQuantity(int quantity) {
73.         this.quantity = quantity;
74.     }
75. }

```

#### 4.2.2. Database.java -> Gerenciar a conexão com o banco de dados MySQL

```

package dao;

/**
 *
 * @author Jean
 */

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;

public class Database {
    private static final String URL =
"jdbc:mysql://localhost:3306/product_manager?useUnicode=true&characterEncoding=
=UTF-8";
    private static final String USER = "root";
    private static final String PASSWORD = "";

    public static Connection getConnection() throws SQLException {
        try {
            Class.forName("com.mysql.cj.jdbc.Driver");
            return DriverManager.getConnection(URL, USER, PASSWORD);
        } catch (ClassNotFoundException e) {
            throw new SQLException("Driver MySQL não encontrado!", e);
        }
    }

    // Testando a conexão com o banco de dados (execute esse arquivo para
    teste)
    public static void main(String[] args) {
        try (Connection connection = Database.getConnection()) {

```



```

        System.out.println("Conexão com o banco de dados estabelecida com
sucesso!");
    } catch (SQLException e) {
        System.err.println("Erro ao conectar com o banco de dados: " +
e.getMessage());
    }
}
}
}

```

#### 4.2.3. ProdutoDAO.java -> Métodos CRUD para executar comandos SQL

```

package dao;

/**
 *
 * @author Jean
 */

import model.Produto;
import java.sql.*;
import java.util.ArrayList;
import java.util.List;

public class ProdutoDAO {

    // Método para adicionar um novo produto ao banco de dados
    public void adicionarProduto(Produto produto) {
        String sql = "INSERT INTO products (name, description, price,
quantity) VALUES (?, ?, ?, ?)";

        try (Connection conn = Database.getConnection();
            PreparedStatement stmt = conn.prepareStatement(sql)) {
            stmt.setString(1, produto.getName());
            stmt.setString(2, produto.getDescription());
            stmt.setDouble(3, produto.getPrice());
            stmt.setInt(4, produto.getQuantity());

            stmt.executeUpdate();
            System.out.println("Produto adicionado com sucesso!");
        } catch (SQLException e) {

```

```

        System.err.println("Erro ao adicionar produto: " +
e.getMessage());
    }
}

// Método para obter todos os produtos do banco de dados
public List<Produto> listarProdutos() {
    List<Produto> produtos = new ArrayList<>();
    String sql = "SELECT * FROM products";

    try (Connection conn = Database.getConnection();
        Statement stmt = conn.createStatement();
        ResultSet rs = stmt.executeQuery(sql)) {

        while (rs.next()) {
            Produto produto = new Produto();
            produto.setId(rs.getInt("id"));
            produto.setName(rs.getString("name"));
            produto.setDescription(rs.getString("description"));
            produto.setPrice(rs.getDouble("price"));
            produto.setQuantity(rs.getInt("quantity"));
            produtos.add(produto);
        }
    } catch (SQLException e) {
        System.err.println("Erro ao listar produtos: " + e.getMessage());
    }

    return produtos;
}

// Método para atualizar um produto no banco de dados
public void atualizarProduto(Produto produto) {
    String sql = "UPDATE products SET name = ?, description = ?, price =
?, quantity = ? WHERE id = ?";

    try (Connection conn = Database.getConnection();
        PreparedStatement stmt = conn.prepareStatement(sql)) {
        stmt.setString(1, produto.getName());
        stmt.setString(2, produto.getDescription());
        stmt.setDouble(3, produto.getPrice());
        stmt.setInt(4, produto.getQuantity());
        stmt.setInt(5, produto.getId());

        stmt.executeUpdate();
        System.out.println("Produto atualizado com sucesso!");
    } catch (SQLException e) {
        System.err.println("Erro ao atualizar produto: " +
e.getMessage());
    }
}

```

```

    }
}

// Método para excluir um produto pelo ID
public void deletarProduto(int id) {
    String sql = "DELETE FROM products WHERE id = ?";

    try (Connection conn = Database.getConnection();
        PreparedStatement stmt = conn.prepareStatement(sql)) {
        stmt.setInt(1, id);

        stmt.executeUpdate();
        System.out.println("Produto deletado com sucesso!");
    } catch (SQLException e) {
        System.err.println("Erro ao deletar produto: " + e.getMessage());
    }
}

// Método para buscar produto pelo ID para edição
public Produto buscarProdutoPorId(int id) {
    Produto produto = null;
    String sql = "SELECT * FROM products WHERE id = ?";

    try (Connection conn = Database.getConnection();
        PreparedStatement stmt = conn.prepareStatement(sql)) {
        stmt.setInt(1, id);

        try (ResultSet rs = stmt.executeQuery()) {
            if (rs.next()) {
                produto = new Produto();
                produto.setId(rs.getInt("id"));
                produto.setName(rs.getString("name"));
                produto.setDescription(rs.getString("description"));
                produto.setPrice(rs.getDouble("price"));
                produto.setQuantity(rs.getInt("quantity"));
            }
        }
    } catch (SQLException e) {
        System.err.println("Erro ao buscar produto: " + e.getMessage());
    }
    return produto;
}
}

```

#### 4.2.4. ProdutoDAOTest.java -> Verifica se as operações CRUD estão funcionando conforme esperado

```
5.
6. package dao;
7.
8. /**
9.  *
10.  * @author Jean
11.  */
12.
13. import model.Produto;
14.
15. public class ProdutoDAOTest {
16.     public static void main(String[] args) {
17.         ProdutoDAO dao = new ProdutoDAO();
18.
19.         // Teste de Adição
20.         Produto novoProduto = new Produto(0, "Produto Teste", "Descrição do
Produto Teste", 10.99, 100);
21.         dao.adicionarProduto(novoProduto);
22.
23.         // Teste de Listagem
24.         System.out.println("Lista de Produtos:");
25.         for (Produto p : dao.listarProdutos()) {
26.             System.out.println(p.getName() + " - " + p.getDescription());
27.         }
28.
29.         // Teste de Atualização
30.         novoProduto.setName("Produto Atualizado");
31.         dao.atualizarProduto(novoProduto);
32.
33.         // Teste de Exclusão
34.         dao.deletarProduto(novoProduto.getId());
35.     }
36. }
```

#### 4.2.5. JavaEE8Resource.java -> Dependências do JavaEE

```
package com.mycompany.productmanagerapp.resources;

import javax.ws.rs.GET;
import javax.ws.rs.Path;
import javax.ws.rs.core.Response;

/**
 *
 * @author
 */
@Path("/javaee8")
public class JavaEE8Resource {

    @GET
    public Response ping(){
        return Response
            .ok("ping")
            .build();
    }
}
```

#### 4.2.6. JAXRSConfiguration.java -> Dependências do JavaEE

```
4. package com.mycompany.productmanagerapp;
5.
6. import javax.ws.rs.ApplicationPath;
7. import javax.ws.rs.core.Application;
8.
9. /**
10.  * Configures JAX-RS for the application.
11.  * @author Juneau
12.  */
13. @ApplicationPath("/resources")
14. public class JAXRSConfiguration extends Application {
15.
16. }
```

### 4.3. Código CSS

#### 4.3.1. style.css -> Estilização geral de todas as páginas

```
/* Tema Escuro */
body {
    background-color: #2c2f37;
    color: #e0e0e0;
    font-family: Arial, sans-serif;
    margin: 0;
    padding: 0;
}

h1 {
    color: #f5a623;
    text-align: center;
    margin-top: 30px;
}

table {
    width: 80%;
    margin: 20px auto;
    border-collapse: collapse;
}

th, td {
    padding: 10px;
    text-align: center;
    border: 1px solid #444;
}

th {
    background-color: #f5a623;
    color: #fff;
}

tr:nth-child(even) {
    background-color: #444;
}

a {
```

```
    color: #f5a623;
    text-decoration: none;
}

a:hover {
    text-decoration: underline;
}

/* Estilo do Formulário */
form {
    width: 40%;
    margin: 20px auto;
    padding: 20px;
    background-color: #3a3f47;
    border-radius: 8px;
}

form input[type="text"],
form input[type="number"],
form input[type="submit"] {
    width: 100%;
    padding: 10px;
    margin: 10px 0;
    border: 1px solid #666;
    border-radius: 4px;
    background-color: #444;
    color: #fff;
}

form input[type="submit"] {
    background-color: #f5a623;
    border: none;
    cursor: pointer;
}

form input[type="submit"]:hover {
    background-color: #f39c12;
}

/* Links */
a {
    color: #f5a623;
}

a:hover {
```

```
text-decoration: underline;
}
```

#### 4.4. Código XML

##### 4.4.1. pom.xml -> Dependências do projeto

```
6. <project xmlns="http://maven.apache.org/POM/4.0.0"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
7.     xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
   http://maven.apache.org/xsd/maven-4.0.0.xsd">
8.     <modelVersion>4.0.0</modelVersion>
9.     <groupId>com.mycompany</groupId>
10.    <artifactId>ProductManagerApp</artifactId>
11.    <version>1.0-SNAPSHOT</version>
12.    <packaging>war</packaging>
13.    <name>ProductManagerApp-1.0-SNAPSHOT</name>
14.
15.    <properties>
16.        <maven.compiler.source>1.8</maven.compiler.source>
17.        <maven.compiler.target>1.8</maven.compiler.target>
18.        <endorsed.dir>${project.build.directory}/endorsed</endorsed.dir>
19.        <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
20.        <failOnMissingWebXml>false</failOnMissingWebXml>
21.        <jakartaee>8.0</jakartaee>
22.    </properties>
23.
24.    <dependencies>
25.        <dependency>
26.            <groupId>javax</groupId>
27.            <artifactId>javaee-api</artifactId>
28.            <version>${jakartaee}</version>
29.            <scope>provided</scope>
30.        </dependency>
31.
32.        <!-- Dependência do MySQL JDBC Driver -->
33.        <dependency>
34.            <groupId>mysql</groupId>
35.            <artifactId>mysql-connector-java</artifactId>
36.            <version>8.0.33</version>
37.        </dependency>
38.
```



```
39.     </dependencies>
40.
41.     <build>
42.         <plugins>
43.             <plugin>
44.                 <groupId>org.apache.maven.plugins</groupId>
45.                 <artifactId>maven-compiler-plugin</artifactId>
46.                 <version>3.1</version>
47.                 <configuration>
48.                     <source>1.8</source>
49.                     <target>1.8</target>
50.                     <compilerArguments>
51.                         <endorseddirs>${endorsed.dir}</endorseddirs>
52.                     </compilerArguments>
53.                 </configuration>
54.             </plugin>
55.             <plugin>
56.                 <groupId>org.apache.maven.plugins</groupId>
57.                 <artifactId>maven-war-plugin</artifactId>
58.                 <version>3.3.1</version>
59.                 <configuration>
60.                     <failOnMissingWebXml>>false</failOnMissingWebXml>
61.                 </configuration>
62.             </plugin>
63.             <plugin>
64.                 <groupId>org.apache.maven.plugins</groupId>
65.                 <artifactId>maven-dependency-plugin</artifactId>
66.                 <version>2.6</version>
67.                 <executions>
68.                     <execution>
69.                         <phase>validate</phase>
70.                         <goals>
71.                             <goal>copy</goal>
72.                         </goals>
73.                     </execution>
74.                 </executions>
75.             </plugin>
76.             <plugin>
77.                 <groupId>org.apache.maven.plugins</groupId>
78.                 <artifactId>maven-jar-plugin</artifactId>
79.                 <version>3.1.1</version>
80.                 <configuration>
81.                     <outputDirectory>${endorsed.dir}</outputDirectory>
82.                     <silent>>true</silent>
83.                     <artifactItems>
84.                         <artifactItem>
85.                             <groupId>javax</groupId>
86.                             <artifactId>javaee-api</artifactId>
87.                             <version>${jakartaee}</version>
88.                             <type>jar</type>
89.                         </artifactItem>
90.                     </artifactItems>
91.                 </configuration>
92.             </plugin>
93.         </plugins>
94.     </build>
95. </project>
```

```

83.             </artifactItems>
84.         </configuration>
85.     </execution>
86. </executions>
87. </plugin>
88. </plugins>
89. </build>
90. </project>

```

#### 4.4.2. web.xml -> Dependências Web do projeto

```

<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns="http://xmlns.jcp.org/xml/ns/javaee"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee
http://xmlns.jcp.org/xml/ns/javaee/web-app 4.0.xsd"
  version="4.0">
  <session-config>
    <session-timeout>
      30
    </session-timeout>
  </session-config>

  <filter>
    <filter-name>encodingFilter</filter-name>
    <filter-
class>org.apache.catalina.filters.SetCharacterEncodingFilter</filter-class>
    <init-param>
      <param-name>encoding</param-name>
      <param-value>UTF-8</param-value>
    </init-param>
    <init-param>
      <param-name>forceEncoding</param-name>
      <param-value>true</param-value>
    </init-param>
  </filter>
  <filter-mapping>
    <filter-name>encodingFilter</filter-name>
    <url-pattern>/*</url-pattern>
  </filter-mapping>

</web-app>

```

## 4.5. Código SQL

### 4.5.1. Banco de dados product\_manager

```
CREATE DATABASE product_manager;
```

### 4.5.2. Tabela products

```
CREATE TABLE `products` (  
  `id` int(11) NOT NULL,  
  `name` varchar(255) NOT NULL,  
  `description` text DEFAULT NULL,  
  `price` double DEFAULT NULL,  
  `quantity` int(11) DEFAULT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;  
  
ALTER TABLE `products`  
  ADD PRIMARY KEY (`id`);  
  
ALTER TABLE `products`  
  MODIFY `id` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=56;  
COMMIT;
```

## 5. FOTOS

### 5.1. Aplicação Web

Gerenciador de Produtos Escolares					
ID	Nome	Descrição	Preço	Quantidade	Ações
6	Caneta esferográfica	Caneta esferográfica azul, ponta 0.7mm	2.49	500	<a href="#">Editar</a>   <a href="#">Excluir</a>
7	Lápis Preto	Lápis preto HB, com borracha	0.99	300	<a href="#">Editar</a>   <a href="#">Excluir</a>
8	Apontador	Apontador com depósito, colorido	3.49	200	<a href="#">Editar</a>   <a href="#">Excluir</a>
9	Borracha	Borracha branca para lápis	1.49	450	<a href="#">Editar</a>   <a href="#">Excluir</a>
10	Marcador de Texto	Marcador de texto em várias cores	4.99	350	<a href="#">Editar</a>   <a href="#">Excluir</a>
11	Fichário	Fichário com 2 argolas, tamanho A4	12.99	180	<a href="#">Editar</a>   <a href="#">Excluir</a>
12	Mochila Escolar	Mochila com compartimento para notebook e várias divisórias	89.99	50	<a href="#">Editar</a>   <a href="#">Excluir</a>
13	Estojo	Estojo de lona com zíper, para canetas e lápis	7.99	150	<a href="#">Editar</a>   <a href="#">Excluir</a>
14	Pasta de Arquivo	Pasta de arquivo em plástico transparente, com elástico	6.99	200	<a href="#">Editar</a>   <a href="#">Excluir</a>
15	Papel Sulfite A4	Papel sulfite branco, gramatura 75g/m², pacote com 500 folhas	13.99	100	<a href="#">Editar</a>   <a href="#">Excluir</a>
16	Regua de Plástico	Régua de plástico transparente, 50cm	2.49	250	<a href="#">Editar</a>   <a href="#">Excluir</a>
17	Tesoura Escolar	Tesoura com ponta arredondada, ideal para crianças	5.99	100	<a href="#">Editar</a>   <a href="#">Excluir</a>
18	Cola Branca	Cola branca escolar, 100g	4.99	300	<a href="#">Editar</a>   <a href="#">Excluir</a>
19	Canetinha Hidrocor	Canetinha hidrocor em 12 cores	10.99	180	<a href="#">Editar</a>   <a href="#">Excluir</a>
<a href="#">Adicionar Novo Produto</a>					

### Adicionar Produto

Nome:

Descrição:

Preço:

Quantidade:

Salvar

[Voltar à Lista de Produtos](#)

## Editar Produto

Nome:

Caneta esferográfica

Descrição:

Caneta esferográfica azul, ponta 0.7mm

Preço:

2,49

Quantidade:

500

Atualizar

[Voltar à Lista de Produtos](#)

### 5.2. Foto do discente

