

**Tópico em desenvolvimentos de sistemas**

**Matheus Silva**

**Controle de estoque**

Salvador

2020

**Matheus Silva**

**Controle de estoque**

Atividade avaliativa: tópico de desenvolvimento de sistemas, universidade UNIRUY Wyden.

Orientador: Prof.: Heleno Filho

Salvador

2020

**SUMÁRIO**

**1. INTRODUÇÃO4**

**2. OBJETIVO DA APLICAÇÃO4**

2.1Modelo de desenvolvimento**5**

**3. LISTA DE EVENTOS6**

3.1REQUISITOS FUNCIONAIS**6**

3.2REQUISITOS NÃO FUNCIONAIS**6**

**3.3 LEVANTAMENTO DE REQUISITOS7**

3.3.1ENTREVISTA**7**

3.3.2DIAGRAMA DE CASOS DE USO**8**

3.4 **ESPECIFICAÇÃO DE PROGRAMA10**

3.4.1REGRAS DE NEGÓCIO10

3.4.2ENTIDADES ENVOLVIDAS**13**

**4. POLITICA DE TESTE 14**

**5. IMPLEMENTAÇÃO 14**

**6. VERSIONAMENTO 14**

**7. APLICAÇÃO PROTÓTIPO 16**

**8. CONCLUSÃO 19**

**1.INTRODUÇÃO**

A proposta do Trabalho é a criação de um software que utilize métodos e técnicas da engenharia de requisitos para implementar uma aplicação protótipo.  
Conforme a solicitação do cliente, busco atender as demandas e problemas com gastos elevados de produtos no ramo de conveniência.

Por meio de conversas inicias, foi proposto a elaboração de um projeto de um sistema de controle de estoque que consiga atender os problemas enfrentados.

Nesse trabalho serão apresentados todo o levantamento no processo de desenvolvimento do software utilizado, seguindo a análise abortando os requisitos funcionais e não-funcionais, especificações do programa políticas de teste, e as telas funcionais.

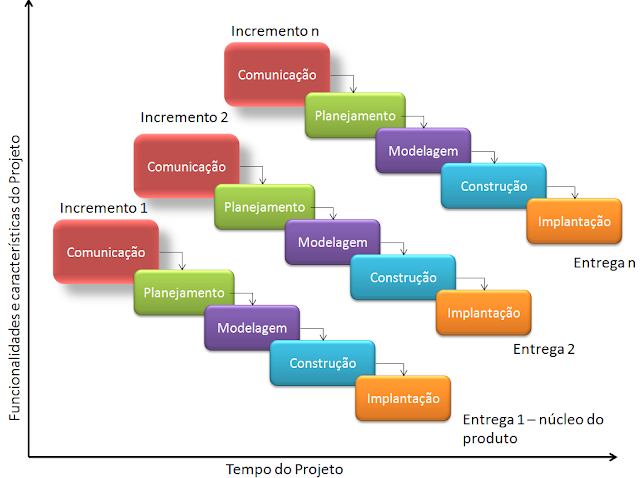
Durante o processo de desenvolvimento do projeto pretendo adquirir conhecimentos similar a uma implementação real de um software utilizando a engenharia para auxiliar em um produto final com qualidade.

**2. OBJETIVO DA APLICAÇÃO**

A modernização tecnológica trouxe benefícios para empreses de várias formas, como otimização e automatização dos meios produtivos visto a dificuldade da empresa em possui um controle de custo bom, será desenvolvido um controle de estoque que por meio dele obterá um controle de custos, otimização, flexibilidade, redução de perdas sobre o estoque. Um modelo de processo de desenvolvimento é utilizado para definir como as etapas pertencentes ao desenvolvimento do software serão conduzidas, para atingir o objetivo do desenvolvimento, que é a obtenção do software.

**2.1** **MODELO DE DESENVOLVIMENTO.**

Um dos modelos evolutivos, o modelo Incremental é a combinação entre os modelos linear e de prototipação sugerido por Barry Boehm. O desenvolvimento é dividido em etapas chamadas de incrementos que por meio dele produzira o sistema, até sua versão final. Cada incremento é realizado todo o ciclo do desenvolvimento do software, do planejamento aos testes, em cada etapa concluída será produzida um sistema totalmente funcional. A escolha desse modelo é devido a forma de trabalho que será abordada, trazendo o cliente mais próximo do projeto com isso podendo esclarece seus requisitos e suas prioridades com tal auxílio possui um melhor gerenciamento de riscos assim podendo confirma o resultado depois de cada versão do sistema tornando o software mais flexível.

Figura 1: Modelo Incremental

Fonte: PRESSMAN (2010)

**3. LISTA DE EVENTOS**

**3.1 REQUISITOS FUNCIONAIS**

Requisitos funcionais é a parte de um projeto onde se descreve as ações que um software deve prover, o software de implementação.

**Tela de Login:**

O sistema terá tela para autentificarão do usuário.

**Cadastro de Produtos:**

O Sistema deve realizar o cadastro de produtos.

**Gerar relatórios:**

Gerar relatórios sobre os produtos cadastrado no sistema.

**Alertas:**

O sistema deve realizar alertas inteligentes para produtos armazenados em estoque.

**3.2 REQUISITOS NÃO FUNCIONAIS**

São as qualidades e restrições que o software deve ter.

**Requisitos do produto:**

* Confiabilidade – Os dados deverão estar sempre atualizados.
* Performance – tempo de resposta não deve passar 2 segs.
* Usabilidade – Rodar em todos os navegadores nas suas ultimas atualizações.

**Requisitos Organizacionais:**

* Requisitos de Padrões - o relatório deve ser gerado com o slogan do mercado.
* Requisitos Implementação – estabelecer comunicação entre o sistema do caixa com o do estoque.

**Requisitos Externos:**

* Requisitos éticos: o sistema não apresentará aos usuários quaisquer dados de cunho privativo.
* Requisitos legais: o sistema deverá atender às normas legais, tais como padrões, leis, etc.

**3.3 LEVANTAMENTO DE REQUISITOS**

**3.3.1 ENTREVISTA**

**Identificação das fontes de informações.**

Para ajudar na implementação do software é necessário a escolha de um grupo de fontes para contribuição no projeto, pensando nisso foi selecionado:

Stakeholders - Estoquista, Gerente e Caixa.

Sistema de caixa da empresa.

**Comunicação**:

A estratégia que usei para o levantamento de requisitos foi por meio de uma entrevista semiaberta e conforme a mesma vai avançando, levantamos espécie de protótipos para que possamos ter uma base solida do objetivo final do cliente.

**Entrevistas:**

**Por que será realizada de forma semiabertas?**

O foco é extrair o máximo de requisitos fundamentais do cliente, utilizando perguntas previamente elaboradas e mantendo a informalidade com o intuito de aumentar o leque de possibilidades que podem sair desta entrevista.

Obs.: conforme vamos avançando na entrevista será realizado desenho de Layout no próprio local para melhor compreensão do que o cliente deseja que o sistema tenha.

**O que motivou a criação deste projeto?**

Foi constatado um problema com as despesas para a manutenção do estoque.

Trata-se de gastos elevados com produtos que não obtiveram lucro esperado através das vendas.

O intuito deste projeto é a elaboração de um sistema de controle de estoque que consiga atender as demandas presentes com uma IA (Inteligência artificial) especial para se atentar a este problema.

**Quem vai usar a solução?**

O Programa será utilizado pelo: Estoquista: afim de fazer a manutenção dos produtos que entram e saída (quando, por exemplo, o produto excede a data de validade)

Gestor: afim de receber os relatórios e fazer avaliação mensal das despesas e receitas.

**Qual será o benefício econômico para uma solução bem-sucedida?**

Há uma ideia que se concentra na IA, nela o programa percebe e notifica quando estamos enchendo demais o estoque de produtos que não são vendidos satisfatoriamente. Ou quando o produto é vendido rapidamente, e tem potencial para haver mais dele no estoque.

**Sobre a efetividade na extração de requisitos?**

Devemos nos fazer essas perguntas:

As questões fechadas são relevantes e englobam o que se procura de modo satisfatório?

A parte da entrevista aberta possui potencial para nortear o entrevistado para conseguir mais informações sobre o projeto a ser desenvolvido?

As questões de modo geral são suficientes?

**3.3.2 Diagrama de caso e uso.**

O diagrama de caso e uso é utilizado para representar os requisitos funcionais que o sistema deve prover associando com os “atores” que irá utilizar cada funcionalidade

**Ator estoquista:**

* Efetuar login: Acesso ao seu perfil.
* Controlar o estoque: Gerencia de entrada e saída de produtos diretamente do estoque do mercado.
* Emitir relatório: Em caso de solicitação da área de gerencia.
* Emitir alertas: receberá alertas sobre funções pré-estabelecidas para ajudar gerencia.

**Ator Gestor:**

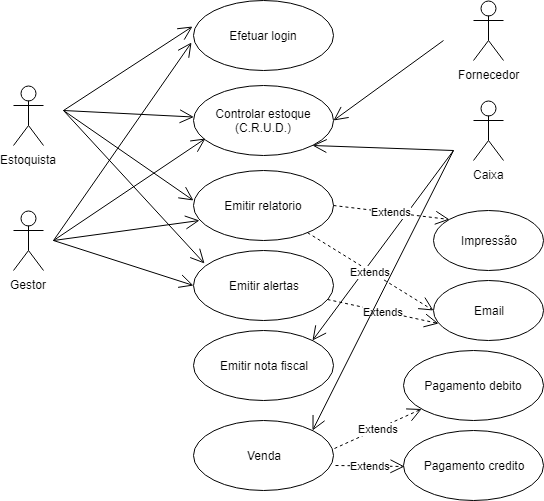
* Efetuar login: acesso ao seu perfil.
* Emitir relatório: Visualização de relatório para tomada de decisão.
* Emitir Alertas: receberá alertas sobre funções pré-estabelecidas.

**Ator Caixa:**

* Efetuar login: acesso ao seu perfil.
* Controlar Estoque: Em caso de venda ou devolução de produto irá modificar a quantidade de itens no estoque.
* Emitir nota fiscal: Geração de nota pós venda.
* Venda: Saída através de venda ou entrada através de devoluções.

**Ator Fornecedor:**

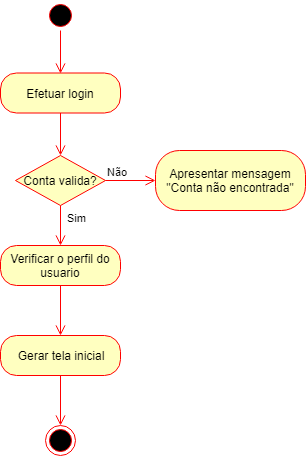
* Controlar estoque: Fornece os produtos para o estoque.



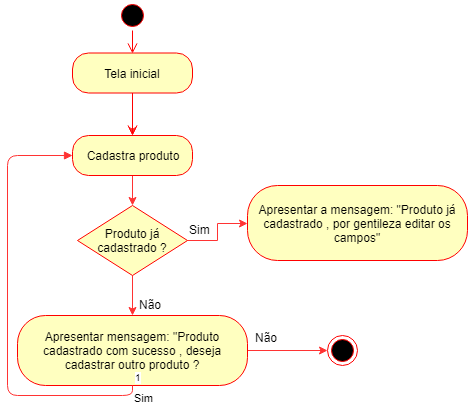
**3.4 ESPECIFICAÇÃO DE PROGRAMA**

**3.4.1 REGRAS DE NEGÓCIO**

**Efetuar login**: Na tela será solicitada os dados de acesso do funcionário, após validação será direcionado para a tela do perfil de cada usuário pois existe funcionalidade que só serão permitida para um perfil do tipo gerente.

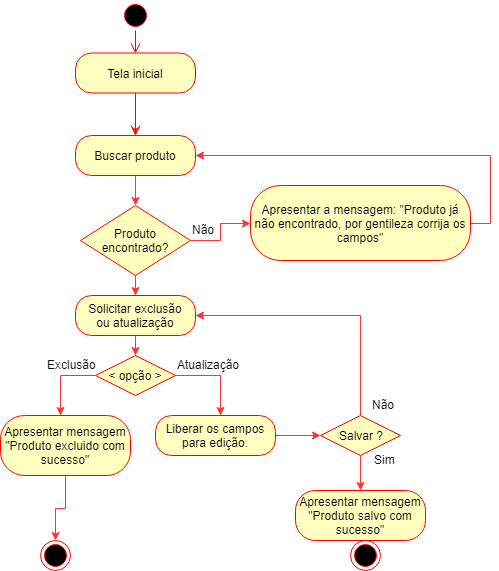
****

**Cadastro do produto**: Uma tela para preencher os dados será gerada, ao clicar em salvar irá ser feita a verificação se o produto já está registrado no banco de dados, caso o produto não constar será permitido o registro no sistema.



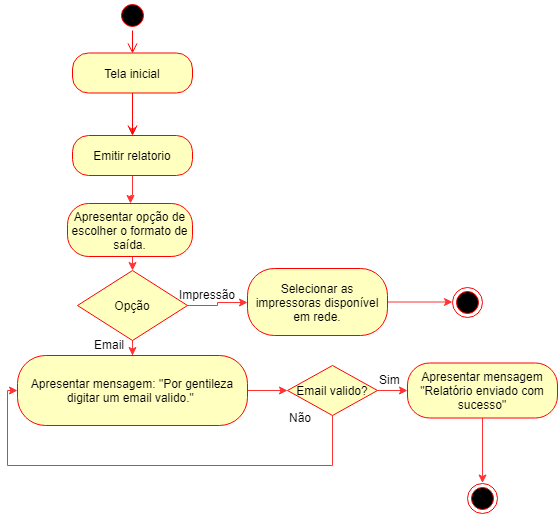
**Cadastro de produto (Buscar)**: O usuário poderá buscar um produto através do código ou nome, ao preencher será feito a busca, caso encontre será liberado a opção de excluir ou editar.

Abaixo da opção de busca, será criado uma tabela para mostrar os item cadastrados.



**Emitir relatório**: após buscar pelo produto, o usuário terá opção de gerar o relatório direto em seu e-mail ou abrir uma nova aba para salvar documento em PDF.

O tipo de relatório é sobre quantidade de produto em estoque, entrada e saída do mês.



**3.4.2 ENTIDADES ENVOLVIDAS**

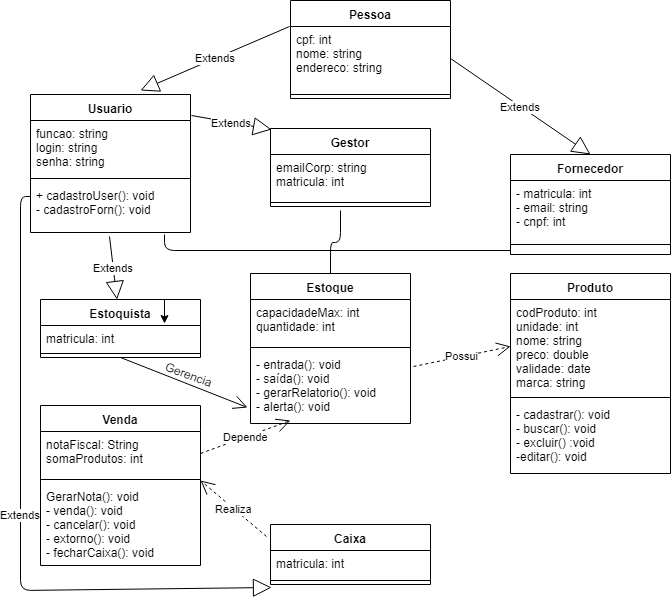
No diagrama abaixo, temos a representação UML para a equipe de desenvolvimento ter uma visão mais clara sobre os requisitos solicitados no software.

**Classe Usuário:** herda os atributos de pessoa e é responsável por duas subclasses estoquista e gestor.

**Classe Estoquista:** irá utilizar os métodos da classe estoque e produto para conseguir gerenciar a entrada e saída de produtos no mercado onde será implementado o software.

**Classe Gestor:** irá utilizar apenas o método “gerarRelatorio” da classe Estoque, pois é a única coisa que interessa a regra de negócio do usuário.

**Classe Caixa:** o usuário com esse perfil irá utilizar os métodos da classe venda para poder controlar saída dos produtos na ação da venda.



**4. POLITICA DE TESTE**

**Teste funcional**

1. Caixa preta: o software possui um CRUD como base de sua aplicação, portanto sua funcionalidade mais importante fica na questão do input e output, portanto o teste de caixa preta será submetido a tela de cadastro para verificar sua saída de dados se estão conforme a solicitação do cliente.
2. A funcionalidade de gerador de relatório, será submetido a teste para verificar a veracidade da informação constada neles.

**5. IMPLEMENTAÇÃO**

Como citado no 3.2 , primeiro passo é entrevistar os Stakeholders para obter o máximo de informação para iniciar o mapeamento da real necessidade do cliente, a equipe desenvolvedora irá implementar o software utilizando php, html , css , javascript e framework bootstrap , com o software pronto irá instalar em um sábado a noite devido ao fechamento do estabelecimento.

Após a implantação, será necessário que os colaboradores da empresa passem por um treinamento, no escopo atual será dividido em 3 horas, 1 hora será feira a apresentação do programa e suas funcionalidades abrindo e o resto do tempo será para tirar duvidas e ensinar na pratica como funciona o novo software.

**6.** **VERSIONAMENTO**

Como auxilio no desenvolvimento do software, foi escolhido o GIT como controle de versão

Com a necessidade de realizar testes para verificação das funcionalidades e integração dos componentes, será adotada a forma distribuída.

No sistema distribuído, o cliente faz a cópia de todo o repositório para poder trabalhar na sua área de trabalho remota, e realizar as mudanças necessárias. Permitindo o trabalho em conjunto com diferentes grupos de pessoas, de diversas maneiras, simultaneamente no mesmo projeto.

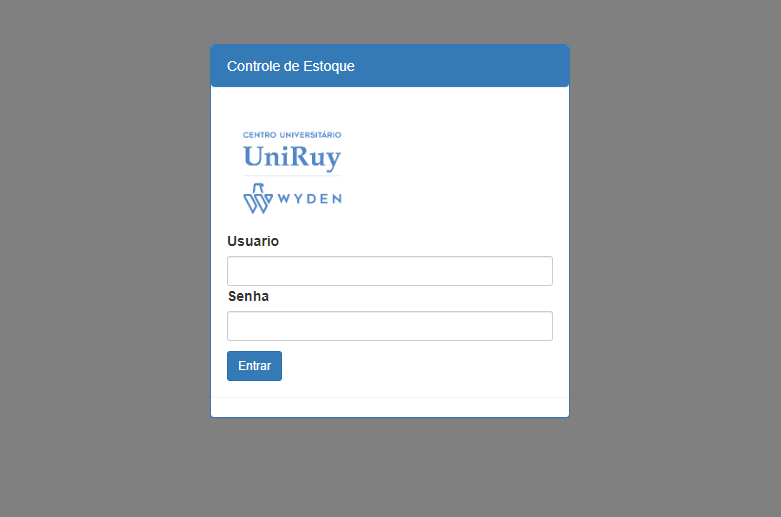




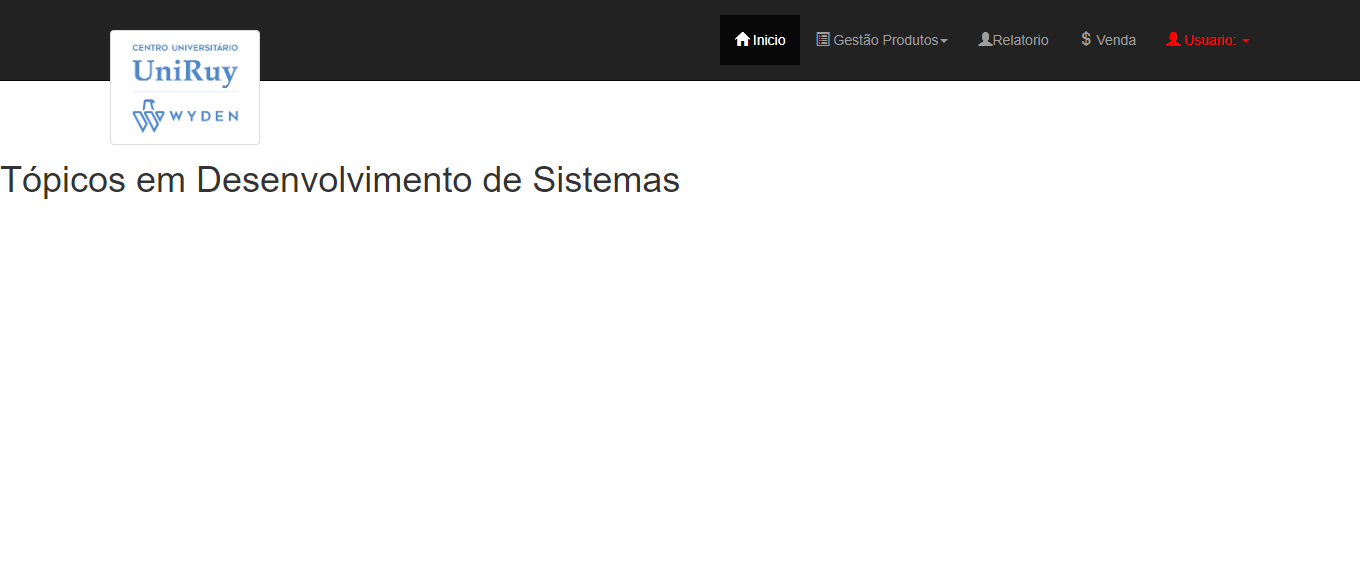
Figura ¹. Diagrama de Controle de Versão Distribuído.

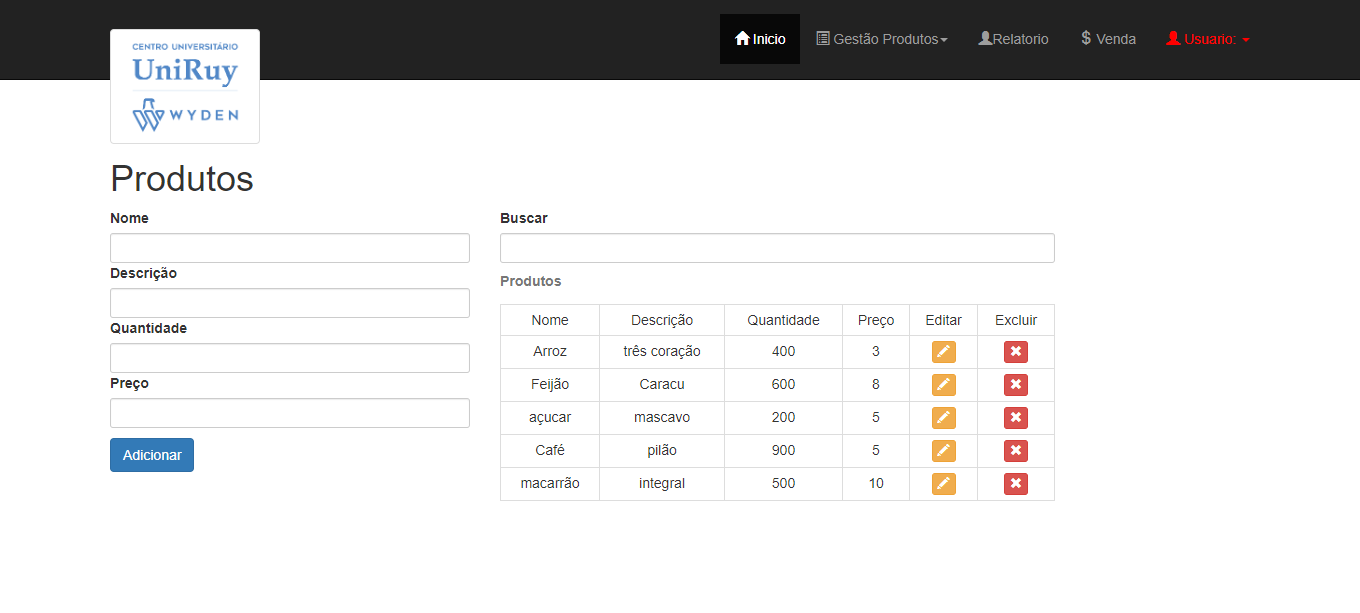
**7. APLICAÇÃO PROTÓTIPO.**

**Tela de autentificarão**: o funcionário irá colocar seus dados para poder acessar o menu da aplicação.

****

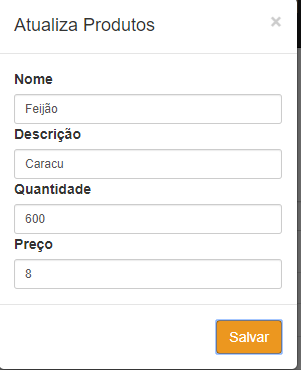
**Menu:** nesta tela, o usuário possui abas de navegações que são redirecionadas para os submenus da aplicação.

* Gestão Produtos: cadastro de novos produtos no estoque.
* ****Relatório: opção será somente exibida no perfil gerente.

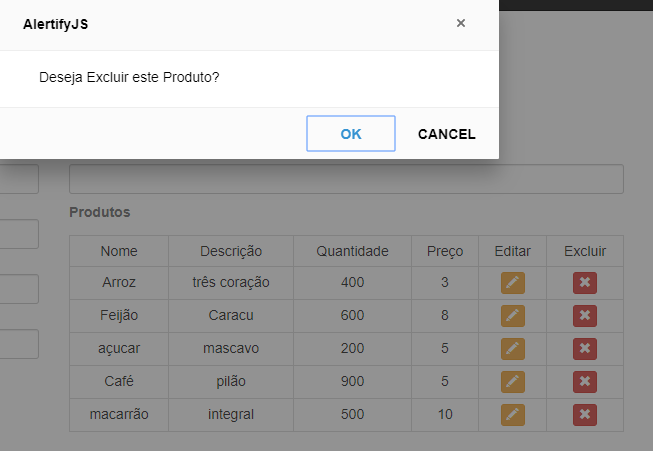
**Submenu:** Tela de gerenciamento de produto, onde o usuário tem um preenchimento de campo para adicionar o produto no estoque.

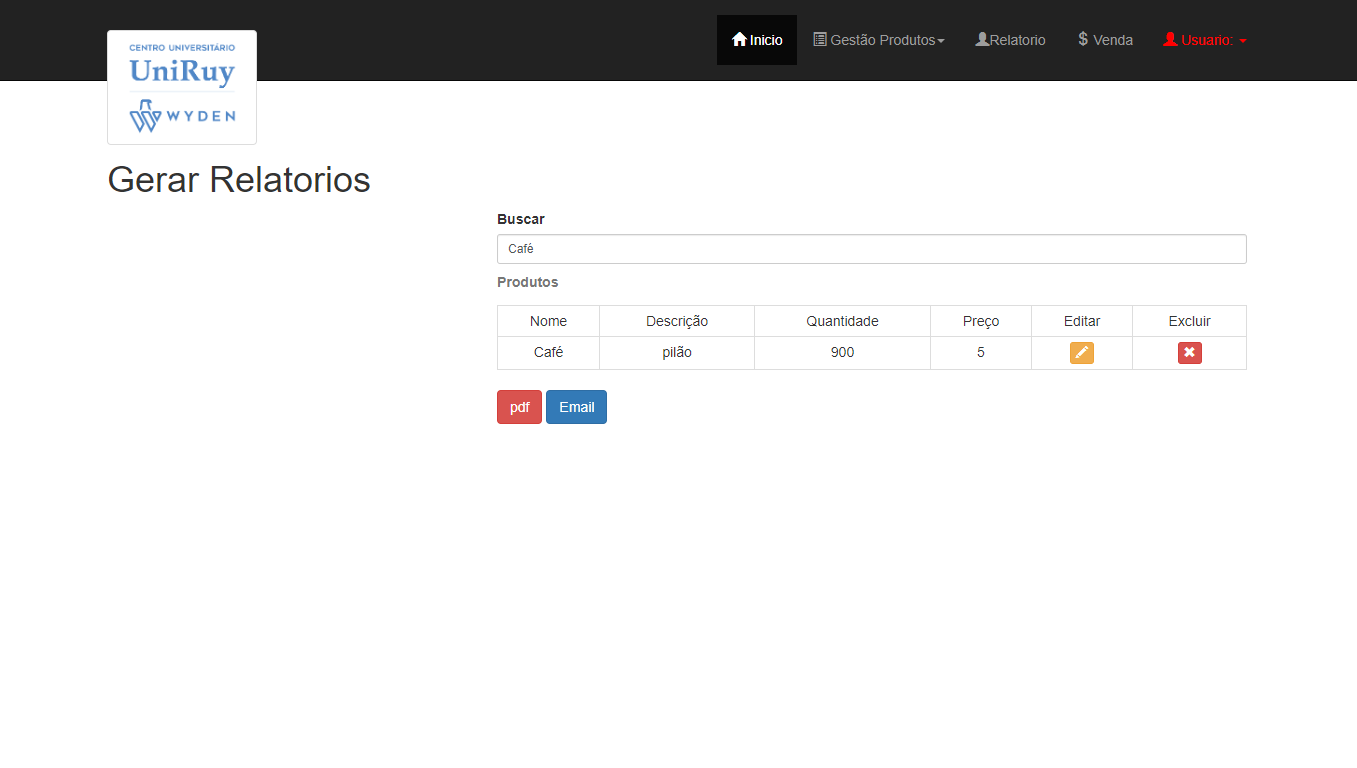
Os itens do estoque são listados nesta pagina com opção de editar e excluir e em caso de uma busca rápida, existe um campo de preenchimento para realizar a ação.

**Sub menu**: Tela gerada ao clicar no ícone “editar” da tela de gerenciamento de produto

****

**Submenu:** ao clicar no excluir terá um alerta de confirmação.

****

**Submenu:** Tela exclusiva de perfil gerente, onde existe uma opção de busca e ao selecionar o campo desejado utiliza-se os botões para gerar relatório de acordo com a necessidade.

**8. Conclusão**

Em vista dos argumentos apresentados, concluímos que durante o processo de desenvolvimento do trabalho, conseguir conciliar os assuntos abordados em sala de aula com o trabalho proposto pelo professor Heleno Filho.

O conhecimento das etapas de um projeto é de extrema importância quando se trata de um desenvolvimento de software pois a “implementação” se trata apenas de uma pequena parte do todo, os maiores fracassos de um projeto é dado pela a má gestão e a despreocupação com os requisitos que o cliente propõe, com esse trabalho é notório a importância da engenharia para evitar uma nova “crise do software”, pois serve de um guia para a entrega de um produto de qualidade.

Com esse aprendizado é possível até mesmo escolher uma área de atuação no qual estava até então, fora do campo de visão, com isso finalizo esse trabalho com uma visão diferente e pronto para colocar todo o aprendizado em pratica.

**REFERÊNCIAS**

Vieira, Rodrigo. **DIAGRAMA DE CASO E USO**, Engenharia de software, 12 de dezembro 2015. Disponível em:

<https://medium.com/operacionalti/uml-diagrama-de-casos-de-uso-29f4358ce4d5>

CHACON, Scott; STRAUB, Ben, **Controle de versões**, Git distributed even if your workflow isnt, 2014 Disponível em:

<https://git-scm.com/book/pt-br/v1/Primeiros-passos-Sobre-Controle-de-Vers%C3%A3o>