

Plano de teste

Projeto de Controle de Estoque

Aluno: Guilherme Scholl

Introdução

O controle de estoque é uma das tarefas mais importantes para qualquer empresa que trabalhe com produtos físicos. Uma gestão eficiente desse processo pode trazer inúmeros benefícios para o negócio, como evitar perdas financeiras com produtos parados no estoque ou a falta de produtos para atender a demanda dos clientes.

Pensando nisso, será desenvolvido um software que tem como objetivo oferecer um controle de estoque eficiente e automatizado para o usuário. Com essa ferramenta, o usuário pode cadastrar os produtos em seu estoque, definir quantidades disponíveis e acompanhar as vendas realizadas.

O software permitirá ao usuário manipular produtos com descrição, características e valores, além de realizar operações de CRUD (criação, leitura, atualização e exclusão) dos produtos cadastrados. O controle de estoque também é gerenciado automaticamente conforme as vendas são realizadas, mantendo a quantidade de produtos sempre atualizada.

Para acessar o sistema, o usuário poderá fazer login e senha e se cadastrar, garantindo assim a segurança dos dados e informações gerenciados pelo software. Os critérios de aceitação do projeto foram definidos para que o usuário possa gerenciar suas vendas e estoque de forma fácil e intuitiva, garantindo um maior controle e eficiência no processo de gestão de estoque.

Equipe

Guilherme Scholl - Gerente de teste, analista de teste e QA.

Responsabilidades

Gerenciar e planejar os testes de software é uma tarefa crucial para garantir que o produto final atinja os objetivos esperados em termos de qualidade. Nesse processo, é necessário estabelecer parâmetros e critérios para avaliar se o software atende às especificações e requisitos definidos.

Para alcançar esse objetivo, o gerenciamento de testes deve considerar vários aspectos, como a definição deste plano de testes detalhado, a escolha das técnicas e ferramentas de

teste mais adequadas, a seleção de casos de teste representativos e a realização de testes de regressão para garantir que as correções não causem novos problemas.

Escopo

Neste plano de testes, o escopo de testes que será incluído será de testes unitários e testes de integração e testes de sistema.

Será excluído deste plano de testes de aceitação, pois seria necessário convocar os clientes para verificar se atendem os requisitos solicitados.

Funcionalidades a serem testadas

As novas funcionalidades a serem testadas serão: as manipulações do banco de dados, se a integridade dos dados irá ser mantida e se atende o que o cliente procura

Critérios de Aceitação

Para garantir que o software de controle de estoque desenvolvido atenda às expectativas dos usuários e cumpra sua finalidade de forma eficiente, foram definidos critérios de aceitação do projeto. Esses critérios foram estabelecidos para garantir que o software atenda a requisitos específicos e proporcione uma experiência satisfatória para o usuário.

Dentre os critérios de aceitação definidos para o software, podemos destacar a capacidade de gerenciamento de vendas e manipulação de produtos com descrição, características e valores. O usuário deve ser capaz de cadastrar e gerenciar produtos de forma fácil e intuitiva, bem como acompanhar as vendas realizadas, garantindo assim o controle do estoque.

Além disso, o software deve permitir que o usuário realize operações de CRUD (criação, leitura, atualização e exclusão) dos produtos cadastrados, possibilitando uma gestão mais eficiente do estoque. É importante que o software permita o cadastro de múltiplos produtos, e que possibilite a definição de quantidades disponíveis em estoque.

Outro critério importante é a segurança do software. O usuário deve ser capaz de fazer login e senha e se cadastrar no sistema, garantindo a segurança dos dados gerenciados pelo software. É essencial que o sistema tenha medidas de segurança robustas para garantir a integridade e a privacidade das informações gerenciadas pelo software.

Por fim, é importante que o software seja de fácil utilização, oferecendo uma interface amigável e intuitiva para o usuário. O software deve ser capaz de proporcionar uma experiência de usuário agradável e satisfatória, permitindo que o usuário gerencie seu estoque de forma eficiente e sem complicações.

Esses são os principais critérios de aceitação do projeto para garantir que o software de controle de estoque atenda às necessidades do usuário de forma eficiente e satisfatória.

Ambiente de testes

O ambiente de testes será no sistema operacional windows 10, em um hardware de processador intel core i5, 8gb de ram e com SSD de 240gb.

A versão do java será a jdk 11.0.12.

Premissas

Dentre as premissas do sistema, destacam-se a eficiência e a automação do controle de estoque, permitindo ao usuário gerenciar seu estoque de forma fácil e intuitiva. O software deve permitir o cadastramento de produtos com descrição, características e valores, bem como o controle das quantidades disponíveis em estoque, mantendo-o atualizado automaticamente conforme as vendas são realizadas.

Além disso, o sistema deve garantir a segurança dos dados gerenciados pelo software, por meio de medidas de segurança robustas como login e senha e cadastro de usuário. A interface do sistema deve ser amigável e intuitiva, facilitando a navegação do usuário e possibilitando uma gestão mais eficiente do estoque.

Outra premissa importante é a escalabilidade do sistema, permitindo que o usuário possa cadastrar e gerenciar um grande volume de produtos sem comprometer o desempenho do software. O sistema deve ser capaz de lidar com um grande fluxo de informações sem comprometer a performance do sistema.

Essas premissas foram estabelecidas para garantir que o sistema de controle de estoque seja capaz de atender às necessidades do usuário de forma eficiente e satisfatória, oferecendo uma gestão de estoque automatizada e de fácil utilização, com segurança e escalabilidade.

Riscos

O não cumprimento dessas premissas podem acarretar em bugs, insatisfação do cliente com a plataforma ou até mesmo uma possível invasão do sistema através de brechas que podem ocorrer ao não testar alguns cenários comuns.