

Projeto A3 Gestão e Qualidade de Software

Alunos: Diogo Galdino **RA:** 122214901

Fernando Hermes **RA:** 122213295

Sistema de Gerenciamento de Inventário

Objetivo do Projeto:

Desenvolver e implementar um sistema de gerenciamento de inventário para a empresa com o objetivo de otimizar o controle de estoque, reduzir custos operacionais e melhorar a eficiência na gestão de produtos.

Escopo:

1. Levantamento de Requisitos:

- Entrevistas com stakeholders para entender as necessidades do sistema.
- Análise de processos atuais de gerenciamento de inventário.
- Identificação de funcionalidades essenciais do sistema.

2. Design e Desenvolvimento:

- Criação de um banco de dados para armazenar informações de inventário.
- Desenvolvimento de uma interface de usuário intuitiva e amigável.
- Implementação de funcionalidades como cadastro de produtos, registro de entradas e saídas, controle de estoque mínimo, relatórios de inventário, entre outros.

3. Testes:

- Realização de testes de unidade, integração e sistema para garantir a qualidade do software.
- Feedback contínuo dos usuários para identificar e corrigir possíveis falhas.

4. Implementação:

- Implantação do sistema em ambiente de produção.
- Treinamento dos usuários finais para uso adequado da ferramenta.
- Monitoramento inicial do desempenho do sistema e ajustes conforme necessário.

5. Suporte e Manutenção:

- Prestação de suporte técnico contínuo após a implantação.
- Atualizações regulares do sistema para adicionar novas funcionalidades e corrigir bugs.

Critérios de Aceitação:

1. A interface de usuário deve ser acessível e fácil de usar para todos os funcionários designados.
2. O tempo médio de resposta do sistema não deve exceder 2 segundos em condições normais de uso.

Restrições do Projeto:

1. Orçamento limitado de \$50.000 para desenvolvimento e implementação.
2. Utilização de tecnologias de código aberto para minimizar custos de licenciamento.

Histórias:

Como gerente de estoque, eu quero poder adicionar novos itens ao inventário com facilidade, incluindo detalhes como nome do item, quantidade inicial, categoria e preço de compra, para manter o controle preciso do estoque da minha loja.

Como um funcionário de vendas, eu quero poder verificar rapidamente a disponibilidade de um item no estoque, inserindo seu código de identificação ou pesquisando pelo nome, para atender aos clientes de forma eficiente e precisa.

Como gerente de compras, eu quero receber notificações automáticas quando a quantidade de um item atingir um nível mínimo pré definido, para que eu possa reabastecer o estoque a tempo e evitar interrupções nas operações da loja.

Prototipação de telas:**Definição de testes:**

Garantir a qualidade de um software envolve a realização de vários tipos de testes para identificar e corrigir defeitos antes do lançamento. Aqui estão alguns dos testes básicos de qualidade de software que podem ser realizados:

Testes Funcionais

Testes de Unidade: Verifica se cada unidade individual do código funciona corretamente. focaremos em funções e métodos específicos. Será utilizado o **JUnit** para este teste.

Testes de Integração: Avalia a interação entre diferentes módulos ou componentes do software para garantir que eles funcionem juntos conforme o esperado. Será utilizado o **JUnit** para este teste.

Testes Não Funcionais

Testes de Desempenho: Avalia a velocidade, capacidade de resposta e estabilidade do software sob diferentes condições de carga. Será utilizado o **JMeter** para realizar os testes.

Metodologia

Scrum comparado à Metodologia Cascata:

Flexibilidade

Scrum: Permite mudanças frequentes e ajustes com base no feedback contínuo do cliente, permitindo que o produto evolua conforme as necessidades mudam.

Cascata: Segue uma sequência linear de fases (requisitos, design, implementação, testes, manutenção) e não é fácil fazer mudanças depois que uma fase é concluída.

Entrega Contínua:

Scrum: Entrega incrementos de produto em curtos períodos (Sprints), proporcionando valor contínuo ao cliente e permitindo a validação de hipóteses de negócio mais rapidamente.

Gerenciamento de Riscos:

Scrum: Identifica e resolve problemas rapidamente graças a ciclos curtos de feedback e a possibilidade de adaptação imediata.

Cascata: Pode resultar em grandes problemas no final do ciclo de desenvolvimento, uma vez que problemas identificados tardiamente são mais difíceis e caros de corrigir.

Análise de Riscos:

Riscos de Segurança

Acesso Não Autorizado: Hackers ou usuários não autorizados podem obter acesso ao sistema e comprometer dados sensíveis.

Perda de Dados: Ataques cibernéticos, falhas de hardware ou software, funcionários com má fé, contribuem para a perda de dados críticos.

Plano B:

Implementar uma monitoração contínua, usando sistemas de detecção de intrusão para monitorar atividades suspeitas em tempo real.

Manter backups de dados em locais seguros para restauração rápida em caso de possível comprometimento dos dados.

Riscos de Desempenho

Desempenho Lento: O software pode se tornar lento ou ineficiente à medida que o volume de dados ou o número de usuários aumentam.

Plano B:

Utilizar servidores adicionais para distribuir a carga em caso de aumento de dados e usuários.

Riscos de Dados

Qualidade dos Dados: Dados imprecisos ou desatualizados podem levar a decisões incorretas sobre o estoque.

Integração de Dados: Problemas ao integrar o software com outros sistemas de gerenciamento de dados podem resultar em inconsistências e erros.

Plano B:

Validação de Dados: Implementar processos de validação e limpeza de dados para garantir a precisão e a atualidade.

Recuperação de Dados: Implemente um sistema de backup e recuperação que permita a restauração rápida dos dados perdidos.

