Universidade Federal do Ceará
Centro de Ciências
Departamento de Computação
Verificação, Validação e Teste de Software (CK0241)

TB02 - Plano de V&V

GoShop

Italo Sousa de Alencar - 495638
José Portela Soares Neto - 377181
Luís Antonio da Silva Maia - 493458

Histórico de versões

Versão	Data	Autor	Descrição
1.0	16/09/2023	Luís	Criação da versão inicial do plano de V&V
1.1	16/09/2023	Italo	Adição das primeiras ideias do que seriam os requisitos não funcionais e requisitos funcionais que não serão testados bem como seus motivos
1.2	16/09/2023	Portela	Inclusão da seção 2 e 2.1 - Adição da Introdução e Aplicação; Adição dos Riscos e atualização do Glossário
1.3	16/09/2023	Luís	Inclusão da seção 5.1
2.0	19/09/2023	Portela	Revisão do documento: inclusão das Seções 1.5, 7 e Glossário; melhoria na descrição das seções;
2.1	22/09/2023	Italo	Atualização das tabelas de requisitos funcionais e não funcionais: Adição das estratégias de teste
2.2	22/09/2023	Luís	Cronograma
2.3	22/09/2023	Portela	Inclusão da seção 6 - Produtos de V&V
2.4	23/09/2023	Italo	Atualização ortográfica da sessão de conclusão; Formatação das referências bibliográficas
2.5	23/09/2023	Luís	Adição das ferramentas, seção 4.1, bem como suas referências. Atualização da seções 2.1 e 10
2.6	23/09/2023	Portela	Atualização das seções 2 e 2.1; Inclusão da seção 3
3.0	23/09/2023	Portela, Luís e Italo	Revisão final do documento.

Sumário

1. Identificador do Plano de Teste

PT01

2. Introdução

Este documento tem como finalidade fornecer uma visão detalhada da aplicação que está sujeita a testes, incluindo sua abrangência, estratégia de teste, requisitos a serem avaliados e ainda apresentar informações sobre a equipe de teste, as ferramentas utilizadas, os resultados esperados, o planejamento de execução dos testes e uma análise dos riscos que podem impactar as atividades de teste.

2.1. A aplicação

O sistema *GoShop [1]* trata-se de uma pequena plataforma de e-commerce eletrônico desenvolvida para atender as necessidades de uma loja online tais como gestão de compras e venda de produtos. O sistema conta um módulos de criação de produtos para que o *administrador* possa gerir seus produtos e o módulo de compras para que o *cliente* possa adicionar itens a um carrinho de compras e comprar os produtos dentre outros.

A infraestrutura do *backend* da aplicação foi desenvolvido utilizando *Node.js [5] e Express [3]*. No que diz respeito a interface do usuário (*frontend)*, foram utilizados *JavaScript [6]*, *HTML [12]*, *CSS [13]* e *ReactJs [4]*. Quanto ao armazenamento de dados,o sistema faz uso do banco de dados *MySQL [11]*. O sistema está disponível através do link [1]. Além disso, uma cópia do código fonte do sistema foi replicada em um repositório local [2] e foi executada com sucesso.

3. Escopo e Estratégia de Teste

No projeto, serão empregadas algumas abordagens técnicas de Verificação e Validação. Para a verificação, haverá uso da análise estática de código com a ajuda da ferramenta SonarQube[10]. Já para a Validação, serão adotadas abordagens de caixa preta e caixa branca. No âmbito dos testes de caixa preta, o foco será a realização de testes funcionais, cujo objetivo é verificar se as funcionalidades previamente descritas estão funcionando conforme especificado. Além disso, também ocorrerão testes exploratórios para identificar possíveis áreas de melhoria. No que diz respeito aos testes de caixa branca, o foco será em testes unitários, que visam avaliar as menores unidades do sistema. Durante o período de teste, também será mantido um registro de bugs em um relatório, a fim de documentar quaisquer problemas identificados pela equipe de teste.

3.1. Requisitos Funcionais

Requisito	Descrição	Tipo de estratégia de teste
RF001	O usuário deve ser capaz de fazer login no sistema. Para acessar a aplicação deve ser fornecido um e-mail e senha válidos.	Teste de caixa-preta (funcional)
RF002	O administrador pode criar, listar, editar e deletar os produtos no sistema.	Teste de caixa-branca, teste de caixa-preta (funcional)
RF003	O usuário pode adicionar e excluir produtos da sua própria lista de desejos.	Teste de caixa-preta (funcional)
RF004	O usuário pode ver, guardar ou comprar produtos.	Teste de caixa-preta (funcional)
RF005	O usuário pode adicionar e deletar produtos do seu carrinho de compras.	Teste de caixa-preta (funcional)
RF006	O usuário pode filtrar produto por: ● Preço ● Categoria	Teste ad-hoc
RF007	O usuário pode ver histórico de pedidos	Teste ad-hoc
RF008	O usuário deve ser capaz de ver e editar os dados do perfil.	Teste de caixa-branca, teste de caixa-preta (funcional)
RF009	O cliente pode criar uma conta.	Teste de caixa-branca

3.1.1 Requisitos funcionais que não serão testados

Requisito	Descrição	Motivo
RF010	O cliente finaliza o pagamento do produto.	Falta de recursos para com os sistemas de pagamento integrados

3.2. Requisitos Não Funcionais

Requisito não funcional	Descrição	Meta	Tipo de teste
RNF001	Segurança	O usuário deve logar no sistema com segurança.	Teste de autenticação e autorização
RNF002	Usabilidade	heurísticas de Nielsen	Teste de Usuário Moderado

4. Ambiente

O ambiente necessário para a realização dos testes devem incluir: Computadores com acesso à Internet, editor de código, linguagem de programação Javascript, Node.js e um banco de dados MySQL.

4.1. Ferramentas

- Node.js [5]. Open Source. O Node.js é um ambiente de execução do código JavaScript [6].
- JavaScript [6]. Open Source, Linguagem de programação que será utilizada para criação dos testes (funcionais e unitários) utilizando Cypress [9] e Jest [8]
- Editor de código (eg. Visual Studio Code) [7]. Ambiente de desenvolvimento integrado Open Source, será utilizado para **escrever, editar e gerenciar os códigos** que serão desenvolvidos, especialmente nas fases de testes.
- Jest [8]. Framework Open Source que possibilita a escrita de testes unitários em JavaScript [6]. Será utilizado para fazer Testes Unitários.

- Cypress [9]. Framework Open Source que possibilita a escrita de testes automatizados em JavaScript [6]. Será utilizado para fazer os Testes Funcionais.
- Acesso a internet
- SonarQube [10] Open source, utilizado para análise estática.
- Computador com Windows 8+, com processador Intel I3 ou superior.

5. Recursos Humanos

5.1. Papéis e Responsabilidades

Papel	Responsável	Responsabilidade	Horas/semana
Analista de teste	Luís	Especificação e criação de casos de testes, automatização dos testes funcionais, realização de testes não funcionais, geração do relatório de bugs.	4h
Testador	Italo	Testes unitários e testes manuais	4h
Analista de teste	Portela	Especificação e criação de casos de testes, automatização dos testes funcionais, realização de testes não funcionais, geração do relatório de bugs.	4h

5.2. Treinamento

Tipo de Treinamento/Conhecimento	Metas	Integrantes
Testes unitários	Aprender a sintaxe do framework Jest	Italo
Testes Funcionais	Aprender a sintaxe do framework Cypress	Luís
Análise estática	Aprender como funciona a ferramenta SonarQube	Luís e Italo

6. Produtos de V&V

Serão desenvolvidos e entregues diversos artefatos utilizados para organização, especificação e acompanhamento, sendo eles:

- **Plano de teste:** um registro que engloba os procedimentos e o calendário de execução.
- **Casos de teste**: um documento que contém todos os cenários de avaliação a serem implementados.
- **Relatório de testes:** uma documentação gerada ao término do ciclo de V&V, detalhando todos os defeitos identificados e sugestões de aprimoramento do aplicativo.
- **Código dos testes automatizados:** conjunto de códigos desenvolvidos durante o processo para realização dos testes automatizados.

7. Cronograma

Atividades	Artefato/ Milestone	Data de início	Data de fim	Responsável
Elaboração do plano de teste de V&V	Plano de V&V	16/09/ 2023	22/09/2023	Italo, Portela e Luís
Estudo das ferramentas de teste	N.A	23/09/ 2023	21/10/2023	Italo, Portela e Luís
Especificar os testes	Especificação de testes	29/09/ 2023	07/10/2023	Italo, Portela e Luís

Execução dos testes unitários	Scripts	22/10/2023	29/10/2023	Italo
Execução dos testes funcionais	Scripts	26/10/2023	03/11/2023	Luís
Análise dos testes feitos	Relatório de bugs	05/11/2023	12/11/2023	Italo, Portela e Luís

8. Riscos

Categoria	**Risco**	**Impacto***	**Probabilidade****	**Estratégia de Mitigação**
Técnico	Dificuldades para entender as ferramentas de testes	Médio	Média	Ajustar o modo como estão sendo estudadas as ferramentas.
Técnico	Hardwares com defeitos durante o processo	Alto	Baixa	Nossos hardwares estão bem preservados
Técnico	Problemas de ambiente de testes	Médio	Baixa	Nos certificar de que temos o ambiente de testes devidamente configurado e estável
Organizacional	Conflitos de agendamento com outros projetos	Médio	Alto	Tentar adiantar os documentos para que não seja impactado por outras disciplinas

*Impacto no projeto: Alto, Médio e Baixo

**Probabilidade de ocorrência: Alta, Média, Baixa

9. Conclusão

Neste documento, foi elaborado o plano de verificação e validação, onde identificou-se os requisitos funcionais e não funcionais que serão submetidos a testes dentro do período determinado. Além disso, o cronograma é apresentado com a distribuição de tarefas e responsabilidades. Foi realizada uma análise prévia dos possíveis riscos que podem surgir durante o processo e assim possibilitando delinear estratégias para mitigá-los. O documento inclui um quadro de horários para otimizar o uso do tempo e atribui responsabilidades para garantir a execução eficaz de todas as atividades planejadas.

10. Referências

1. E-commerce GoShop, 2023. Disponível em: https://ecommerce-goshop.onrender.com/. Acesso em: 16/09/2023.
2. Github. GoShop Testing, 2023. Disponível em: https://github.com/luismaia-git/goshop-testing. Acesso em: 16/09/2023.
3. ExpressJS. Express: Node.js web application framework, 2017. Disponível em: https://expressjs.com/. Acesso em: 19/09/2023.
4. React.dev. Built-in React Hooks, 2023. Disponível em: https://react.dev/reference/react. Acesso em: 19/09/2023.
5. NodeJS. About Node.js. Disponível em https://nodejs.org/en. Acesso em 19/09/2023.
6. JavaScript.com. Learn JavaScript Online, 2023. Disponível em: https://www.javascript.com/. Acesso em 19/09/2023.
7. Visual Studio Code. Visual Studio Code: Code editing. Redefined, 2023. Disponível em: https://code.visualstudio.com/. Acesso em: 23/09/2023.
8. Jest. Delightful JavaScript Testing, 2023. Disponível em: https://jestjs.io/pt-BR/. Acesso em: 23/09/2023.
9. Cypress. Cypress: Test. Automate. Accelerate. Disponível em: https://www.cypress.io/. Acesso em 23/09/2023.
10. SonarQube. Code Quality, Security & Static Analysis Tool with SonarQube, 2023. Disponível em: https://www.sonarsource.com/products/sonarqube/. Acesso em: 23/09/2023.
11. MySQL, 2023. Disponível em: https://www.mysql.com/. Acesso em: 23/09/2023.
12. HTML.com. HTML For Beginners The Easy Way: Start Learning HTML & CSS Today, 2023. Disponível em: https://html.com/. Acesso em: 23/09/2023.
13. MDN Web Docs. CSS: Cascading Style Sheets, 2023. Disponível em: https://developer.mozilla.org/en-US/docs/Web/CSS. Acesso em: 23/09/2023.

Glossário

Termo	Definição
Caixa Branca	Teste direto no código fonte do software.
Caixa Preta	São validações que permitem obter um conjunto de condições de entrada que verificam todos os requisitos funcionais de um software.
Cypress	Framework de testes, de código aberto e de fácil configuração.
Framework	Termo que se refere a estratégias e ações que visam solucionar um tipo de problema.
Heurísticas de Nielsen	são uma forma de avaliar o design de interface, identificando falhas e erros para corrigi-los e otimizar a user experience, ou experiência do usuário
Intel Core i3	Família de processadores da Intel, destinado a desktops x86-64.
Jest	Framework de testes unitários para Javascript.
MySQL	Sistema de gerenciamento de banco de dados.
Node.js	Ambiente de execução que permite executar comandos Javascript.
Open Source	Software de código aberto é o software de computador que tem seu código fonte disponibilizado e licenciado com uma licença de código aberto no qual o direito autoral fornece o direito de estudar,
ReactJS	Biblioteca front-end JavaScript de código aberto com foco em criar interfaces de usuário em páginas web.
SonarQube	Software de análise estática.
V&V	Verificação e Validação.