Travel Green
VEÍCULOS SUSTENTÁVEIS

Relatório Final de Testes

Data	Versão	Descrição	Autor
11/04/2023	1.0	Avaliação de Testes do projeto Travel Green - Release Público Inicial	Matheus Rayol Ferreira

Índice

1. Objetivo

Este Relatório de Avaliação de Testes descreve os resultados obtidos nos testes da aplicação Travel Green no seu Release 1.0 em termos de cobertura de testes (tanto baseada em requisitos quanto baseada em código) e análise de defeitos (densidade). Estes testes foram conduzidos num período de 8 semanas (4 Sprints antes do Release público inicial).

2. Escopo

Este Relatório de Avaliação de Testes é aplicado ao Release 1.0 da aplicação Web Travel Green. Os testes conduzidos estão descritos no Plano de Testes [5]. Este Relatório de Avaliação deve ser utilizado para o seguinte:

- Avaliar a capacidade de aceitação e apropriação do(s) comportamento(s) de desempenho do sistema Travel Green;
- Avaliar a capacidade de aceitação dos testes;
- Identificar aprimoramentos para aumentar a cobertura de testes e/ou a sua qualidade.

3. Referências

As referências aplicáveis são:

- Base2 Tecnologia (https://www.base2.com.br)
- Disciplina de Testing I, Digital House Brasil

4. Introdução

Os casos de teste definidos no Conjunto de Testes foram executados seguindo a estratégia de testes definida no Plano de Teste [5]. Os defeitos de testes foram registrados na plataforma disponibilizada e todos os defeitos de prioridade média, alta ou crítica estão atualmente corrigidos.

A cobertura dos testes em termos de cobertura de casos de uso e requisitos definidos no Plano de Testes foi 100% concluída.

A cobertura de código foi medida utilizando a ferramenta de execução de testes com cobertura do IntelliJ IDEA e está descrita na seção [6].

A análise dos defeitos (conforme exibido na seção [7] a seguir) indica que a maioria dos defeitos encontrados foram erros relacionados à responsividade e/ou usabilidade do sistema.

5. Cobertura de Testes

Todos os casos de teste, conforme definidos no Conjunto de Testes, foram executados. Do total de testes executados no relatório final, não houveram falhas no conjunto de testes exploratórios.

Taxa de Casos de Teste Executados: 98%
Taxa de Casos de Teste com Êxito: 84,5%

A área de testes com a maior taxa de falha durante o projeto foi:

- Testes de responsividade da aplicação;
- Testes de usabilidade.

6. Cobertura do Código

O IntelliJ IDEA foi utilizado para medir a cobertura de código dos testes unitários.

Sprint	Pacote	Classes	Métodos	Linhas
Sprint 1	N/A	0%	0%	0%
Sprint 2	Service.impl	100%	96,6%	98,6%
Sprint 3	Service.impl	100%	100%	100%
Sprint 4	Service.impl	100%	100%	100%

Tabela 1: Percentual de cobertura de código por Sprint - Backend

100% do código da implementação dos serviços está coberto nos testes, ultrapassando o mínimo de 90% estabelecido.

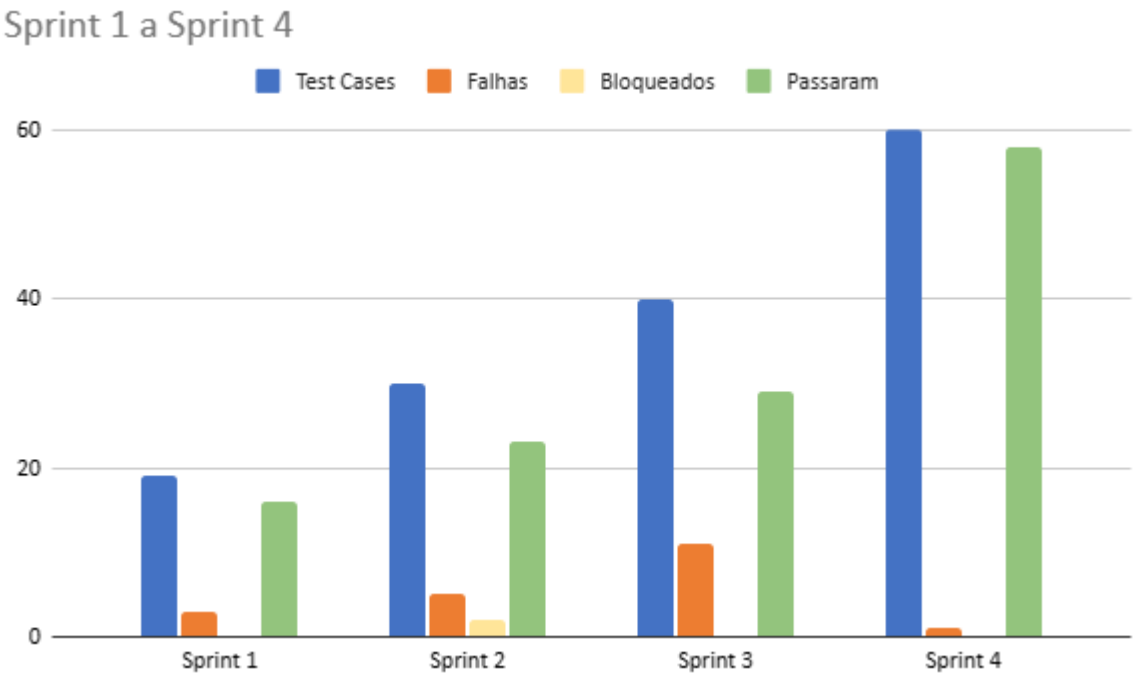
7. Análise de Defeitos

Esta seção resume os resultados da análise de defeitos gerada com base nos resultados obtidos da Sprint 1 à Sprint 4. A seção 8 lista as ações tomadas para correção e mitigação de defeitos durante a execução do projeto.

7.1. Densidade de defeitos

Foram gerados dados sobre a densidade de defeitos utilizando os dados extraídos das Sprints do projeto.

	Sprint 1	Sprint 2	Sprint 3	Sprint 4	Média
Test Cases	19	30	40	60	37,25
Falhas	3	5	11	1	5
Bloqueados	0	2	0	0	0,5
Passaram	16	23	29	58	32
% Falhas	15,79%	16,67%	27,50%	1,67%	14,09%
% Pass	84,21%	76,67%	72,50%	96,67%	84,56%



No contexto dos dados fornecidos, podemos observar uma alta densidade de defeitos no Front-End durante a Sprint 3, com 11 falhas em 40 testes manuais realizados. Por outro lado, houve uma melhora significativa na densidade de defeitos no Back-End, que não apresentou falhas em seus testes automatizados em todas as sprints.

7.2. Tendência de defeitos

Foram gerados dados sobre a densidade de defeitos utilizando os dados extraídos das Sprints do projeto. Ao observar os dados apresentados nos testes manuais, podemos notar uma tendência de aumento no número de falhas nas últimas sprints. A Sprint 1 apresentou 3 falhas, a Sprint 2 apresentou 5 falhas e 2 bloqueios, a Sprint 3 apresentou 11 falhas e a Sprint 4 apresentou 1 falha.

Essa tendência indica que houve uma complexidade crescente no desenvolvimento do projeto, levando a mais falhas e bloqueios. Foi de extrema importância analisar as causas das falhas para identificar padrões e adotar ações para melhorar a qualidade do produto e reduzir o número de falhas em sprints futuras.

7.3. Idade dos defeitos

Durante a execução do projeto, os defeitos reportados tiveram um tempo médio para sua resolução de 1,5 dias. A rapidez na implementação de correções ao código do projeto foi essencial para que pudéssemos focar em melhorias e reestruturações no código.

8. Ações sugeridas e conclusão

Com base nas informações apresentadas, realizaremos algumas ações no futuro para melhorar o ambiente de testes:

- Melhorar a cobertura de testes automatizados no front-end: É importante aumentar a cobertura de testes automatizados para garantir que as funcionalidades do front-end estejam sempre funcionando corretamente. Além disso, isso ajudará a reduzir o tempo gasto com testes manuais e a identificar erros mais rapidamente.
- Realizar testes exploratórios regularmente: Testes exploratórios são uma ótima maneira de encontrar erros e pontos de melhoria que podem ser difíceis de detectar com testes automatizados ou manuais. No nosso projeto, os testes exploratórios são executados semanalmente.
- Monitorar a tendência de defeitos: É importante monitorar a tendência de defeitos ao longo do tempo para identificar áreas que precisam de mais atenção e melhorias. Caso o número de defeitos aumente, é necessário investigar e corrigir as causas.

9. Conclusão

O ambiente de testes do projeto "Travel Green" tem apresentado progresso constante durante as Sprints, principalmente na redução de novos defeitos e reincidências com um monitoramento contínuo do desenvolvimento, melhoria da cobertura de testes automatizados no back-end e na criação de testes automatizados no front-end. No entanto, há espaço para melhorias, principalmente na realização de testes exploratórios e no monitoramento constante da tendência de defeitos.

10. Documentação dos testes

Testing

Test Cases

- Consolidated Test Cases

API Testing

- Postman API Collection

Front-End Testing

- Selenium Framework for Not Cars

Back-End Testing

- Unit Tests - Back End (With Coverage Report)

Test Reports

Test Reports - API

- Postman Automated Test Run - Sprint 1
- Postman Automated Test Run - Sprint 2
- Postman Automated Test Run - Sprint 2

Test Reports - Manual Testing

- Manual Test - Sprint 1 Week 1
- Manual Test - Sprint 1 Week 2
- Manual Test - Sprint 2 Week 1
- Manual Test - Sprint 3 Week 1
- Manual Test Report - Sprint 4 Week 1

Test Reports - Exploratory Testing

- Teste Exploratório - Sprint 2
- Teste Exploratório - Sprint 3
- Teste Exploratório - Sprint 4

Test Reports - Back End

- Code Coverage Report - Sprint 2
- Code Coverage Report - Sprint 3
- Code Coverage Report - Sprint 4