Planejamentos de testes

1. Introdução

Buger Eats é um Sistema de gestão de entregas, o sistema foi criado com o propósito de facilitar na hora da entrega e para garantir um renda extra.Conta com funcionalidade simples de cadastro.

1. Arquitetura

O framework utilizado para a implementação do testes automatizados do Buger Eats foi o Cypress.

Para o armazenamento, consulta e alteração de dados da aplicação, será usada uma API que disponibiliza um conjunto de serviços REST.

1. Funcionalidade

|  |  |  |  |
| --- | --- | --- | --- |
| **Funcionalidade** | ID | **Passos** | **Resultado Esperado** |
| Home | 1 | Acessar a página home do Site | Página Home será exibida |
|  | 2 | Clicar em cadastrar-se na página home | Página de cadastro será exibida |

|  |  |  |  |
| --- | --- | --- | --- |
| **Funcionalidade** |  | **Comportamento esperado** | **Verificações** |
| Cadastro |  | Ao digitar nome completo ,CPF,  e-mail,whatsapp,endereço, selecionar a opção de entrega,e a foto da CNH,o cadastro será registrado com sucesso. | * Verificar se é colocado uma máscara para todos os campos requeridos. * Verificar se o campo CPF aceita formatos inválidos. * Alerta com sugestão para formato de e-mail incompleto. * Verificar se o campo de upload da foto só permite formato de imagem. * Verificar se não permite selecionar 2 métodos de entrega. * Verificar se o campo Whatsapp mostra mensagem obrigatória. * Verificar se existe tratativas para Cpf inválido. * Verificar se existe tratativas para DDD inválido * Verificar se existe tratativas para Whatsapp inválido. * Verificar se campo de whatsapp aceita qualquer valor * Verificar se permite email inexistente. * Verificar se campo consegue ignorar pontos e traços. * Verificar se CEP aceita CEP inválido . * Verificar se permite finalizar o cadastro sem Whatsapp. * Verificar se permite finalizar o cadastro sem selecionar a opção de entrega. |

4-Estratégia de Teste

* Escopo de Testes

O plano de testes abrange todas as funcionalidades descritas na tabela acima.Serão executados testes conforme a descrição abaixo.

Testes Automatizados: Serão realizados testes end-to-end na funcionalidade de Cadastro.

Testes Manuais: Todas as funcionalidades serão testadas manualmente pelo time de qualidade.seguindo a documentação de Cenários de teste.

Testes de Integração: Serão executados testes de integração em todos endpoints e esses testes serão de responsabilidade do time de qualidade.

* Ambiente e ferramentas

Os testes serão feitos do ambiente de homologação, e contém as mesmas configurações do ambiente de produção com uma massa de dados gerada previamente pelo time de Qualidade.

As seguintes ferramentas serão ultilizadas nos testes:

|  |  |  |
| --- | --- | --- |
| Ferramenta | Time | Descrição |
| Cypress | Qualidade | Ferramenta para realização de testes de API |
| Cypress | Qualidade | Framework utilizada para testes |
| Cypress | Qualidade | Ferramenta para testes end-to-end |

5-Classificação de Bugs

|  |  |  |
| --- | --- | --- |
| ID | Nivel de Severidade | Descrição |
| 1 | Grave | * Funcionalidade não funciona como o esperado * Input incomum causa efeitos irreversíveis |
| 2 | Moderada | * Funcionalidade não atinge certos critérios de aceitação, mas sua funcionalidade em geral não é afetada * Mensagem de erro ou sucesso não é exibida |
| 3 | Pequena | * Quase nenhum impacto na funcionalidade porém atrapalha a experiência * Erro ortográfico * Pequenos erros de UI |

1. Definição de pronto

Será considerada pronta as funcionalidades que passarem pelas verificações e testes descritas, não apresentarem bugs com a severidade acima de pequena, e passarem por uma validação de negócio de responsabilidade do time de produto.