



# Processo seletivo - e-Inscrição

| *Tainá Wandermurem Conti*

## Plano de Teste

### Objetivo e escopo

Este plano de teste tem como objetivo garantir a qualidade e funcionalidade do produto, identificando possíveis bugs e problemas de usabilidade.

O escopo deste plano de teste é verificar e validar de forma abrangente as seguintes funcionalidades do produto:

- **Checkout Process for Ticket Purchase:** Verificar se os usuários conseguem concluir com sucesso o processo de compra de ingressos, tanto como usuários logados quanto como convidados.
- **Authentication:** Testar o processo de autenticação para garantir que os usuários possam fazer login e cadastro com credenciais válidas.
- **Insurance Form:** Verificar se o formulário de seguro está funcionando corretamente.

## Tipos de Teste

- Testes automatizados (E2E)
  - Nível de teste: Teste de sistema
  - Tipo de teste: Teste funcional

## Casos de testes

### Funcionalidade: Processo de checkout para inscrição em evento

#### ▼ Cenário 01: Usuário cadastrado realiza o checkout com sucesso



Como usuário cadastrado, eu quero realizar inscrição na plataforma para que eu possa participar do evento

- **Pré-condição:**
  - Usuário deve possuir cadastro prévio na plataforma
  - Usuário deve acessar URL: <https://checkout.einscricao.app/tickets>
- Dados do teste:

```
e-Mail: 'automatedtesteseinscricao@gmail.com'
```

```
password: '123456'
```

- Resultado esperado:
  - Usuário consiga se inscrever com sucesso e visualize todas as informações referentes ao pedido
- **Passos para execução**

Dado que o usuário está na página de checkout  
Quando seleciona uma quantidade de ingressos maior que zero  
E clica no botão "Continuar"  
E realiza o login com credenciais válidas  
E seleciona participante existente  
E clica no botão "Finalizar"  
Então é redirecionado para a página de checkout concluído  
E visualiza a mensagem de pedido feito com sucesso

E visualiza o QR code e o código de inscrição  
E tem a opção de salvar o comprovante  
E tem a opção de visualizar suas inscrições  
E consegue avaliar sua experiência

#### ▼ Cenário 02: Usuário guest realiza o checkout com sucesso



Como usuário visitante, eu quero realizar inscrição na plataforma para que eu possa participar do evento

- Resultado esperado:
  - Usuário consiga se inscrever com sucesso e visualize todas as informações referentes ao pedido
  - Usuário deve acessar URL: <https://checkout.einscricao.app/tickets>

- **Passos para execução**

Dado que o usuário está na página de checkout  
Quando seleciona uma quantidade de ingressos maior que zero  
E clica no botão "Continuar sem conta"  
E preenche os dados válidos do participante  
E clica no botão "Finalizar"  
Então é redirecionado para a página de checkout concluído  
E visualiza a mensagem de pedido feito com sucesso  
E visualiza o QR code e o código de inscrição  
E tem a opção de salvar o comprovante  
E tem a opção de visualizar suas inscrições  
E consegue avaliar sua experiência

#### Funcionalidade: Preenchimento do formulário de seguro

##### ▼ Cenário 01: Preenchimento completo do formulário de seguro com dados válidos



Como usuário, eu quero preencher o formulário de seguro para contratar serviços relacionados

- Resultado esperado:
  - Usuário consegue preencher **todas as informações obrigatórias** do formulário e visualize mensagem de sucesso ao final
  - Usuário deve acessar URL: <https://checkout.einscricao.app/tickets>

- **Passos para execução**

**Dado** que o usuário está na aba inicial do formulário de seguros

**Quando** todas as abas existentes forem preenchidas com as informações obrigatórias e dados válidos

**Então** a mensagem de sucesso "Sending e-mail success!" deve ser visualizada.

## Ambientes e Ferramentas

Os testes serão feitos do ambiente de homologação, e contém as mesmas configurações do ambiente de produção, utilizando o ambiente:

- Sistema operacional: Windows 10
- Navegador: Chrome
- Dispositivo: Desktop

Ferramentas utilizadas:

- Cypress
- Github

## Entregáveis

- Plano de teste
- Casos de teste
- Relatório de resultados do teste
- Relatório de bugs

## Classificação de bugs

- **Major:** Impacto na funcionalidade evidente, pode ou não bloquear o teste.
- **Minor:** Quase nenhum impacto na funcionalidade, mas ainda é um erro válido ou não atinge os critérios de aceite.
- **Minimal:** funciona perfeitamente como deveria, mas poderia ser melhor.

## Definition of done

- Todos os casos de testes devem ser executados e aprovados
- Todos os bugs major e minors devem ser corrigidos

## Sugestões de melhorias para implementação

- Refinar o código da automação incorporando boas práticas, como o uso de constantes, utilitários, variáveis de ambiente, uma URL base, entre outros, especialmente para a versão mobile
- Front-end deve adotar a convenção de nomeação `data-test` para identificar seletor de elementos destinados a testes, proporcionando uma estrutura mais robusta e fácil de manter.
- Aplicar interceptações para aprimorar as validações e gerenciar os tempos de espera de maneira eficiente. Isso não apenas otimizará o desempenho, mas também garantirá uma execução mais estável e confiável dos testes automatizados

## Problemas encontrados durante a execução

- ERRO "e is not defined" é um erro que ocorreu no código da sua aplicação, não no código do Cypress. O Cypress detecta erros não capturados originados da sua aplicação e automaticamente encerra o teste quando isso acontece.

O problema parece estar relacionado a uma variável ou objeto chamado `e` que não foi definido no escopo onde está sendo usado.

**Workaround:** adicionado arquivo em `commands.js` para habilitar a detecção de eventos não capturados.

```
Cypress.on('uncaught:exception', (err, runnable) => {  
  // Retorne false para impedir que o erro encerre o teste  
  return false;  
});
```