**Anotações do Curso Do Requisito à Execução de Testes Focado em Sistemas Web**

Obs.: As informações aqui contidas são baseadas no meu estudo. Podem conter erros e serão corrigidas ao decorrer do tempo.

## Introdução

As etapas para a elaboração de testes de um sistema passa pelas seguintes etapas:

* Levantamento dos Requisitos do Sistema
  + Criação do Documento e Protótipos do Projeto junto com a equipe de planejamento.
* Leitura do documento do sistema
  + Estudar o Documento do Projeto
* Leitura do protótipo do sistema
  + Estudar o Protótipo das Telas do Projeto ou documento similar.
* Planejamento:
  + O objetivo é levantar todas as funcionalidades seus campos
* Mapeamento
  + O objetivo é rastrear cada campo das funcionalidades e listar todos os casos de testes possível para cada um deles**.**
* Modelagem
  + O objetivo é elaborar de forma detalhada todo o passo a passo para a execução de cada caso de teste.
* Elaborar plano de teste
  + É o momento onde é decidido o prezo e o tempo máximo para realizar os testes.
  + Também é inserido no plano de teste os tipos de testes, plataformas a serem testadas entre outras informações adicionais
* Execução de Testes
  + O objetivo é que após executar os testes que foram modelados na etapa anterior, fazer a coleta de evidencias e reportar para a equipe como está o andamento dos testes.
* Abertura e Organização de Bugs
  + O objetivo é encontrar bugs no sistema durante a realização de testes e reportar para a equipe de desenvolvimento.
* Retestes e regressões
  + As regressões são testes que são refeitos quando há mudança de versão do sistema.
* Entrega do Projeto
* Encerramento do Case

## 1. Planejamento

* Após o QA estudar o documento do projeto, o protótipo das telas do sistema, etc., a próxima fase será a fase de planejamento.
* O objetivo da fase de planejamento do QA é levantar todos os campos, seja campos de texto, botões, etc., indicando a funcionalidade (ou tela) do sistema que aquele campo pertence. É importante também indicar o nível de acesso.
  + **Funcionalidade do sistema**:
    - São tarefas que o usuário poderá executar através do sistema. Exemplo de funcionalidades de um sistema:
      * Cadastro de usuário
      * Atualizar dados
      * Pagamento
      * Login, etc.
    - Geralmente as funcionalidades do sistema são divididas em telas. Exemplo:
      * Tela de cadastro de usuário
      * Tela de atualizar dados
      * Tela de pagamento
      * Tela de login, etc.
    - É importante ressaltar que todas essas tarefas são facilmente executáveis e também é importante que haja botões de ajuda presentes em todo o sistema.
  + **Campos e botões**:
    - São item necessários para a execução das funcionalidades. Exemplo: a tela de envio de e-mail contém formulário com os campos: campo de e-mail, campo de assunto, campo de mensagem e botão enviar.
* Oque fazer?
  + Criar a Planilha de Funcionalidades.
* Como fazer?
  + **1º Etapa** - Listar em uma coluna todos os nomes das funcionalidades (telas).
  + Verificar qual nível de acesso do sistema as funcionalidades pertencem (Se é nível de usuário, nível de administrador, nível de empresa, etc.)
    - Ex.: O sistema é dividido em funcionalidades da empresa e funcionalidades do usuário. A funcionalidade do usuário tem as funcionalidades: tela inicial, tela de acesso, tela de dados do candidato, Login, etc.
  + **2º Etapa** - Listar em uma coluna todos os nomes de todos elementos encontrados em uma tela de sistema pertencente uma funcionalidade/tela, sem exceção. Ou seja, listar campos de texto, botões, links, campos de seleção, etc. encontrados em cada tela do sistema e indicar qual funcionalidade ela pertence. Todos elementos de uma página devem ser avalidados.
    - Ex.: Na parte de usuário tem a funcionalidade ligada a tela de cadastro de usuário com os campos de “nome do usuário”, “login”, “senha”, botão “cadastrar”, botão “fechar”, etc.)
    - Obs.: Na 2º etapa separar funcionalidades genéricas, ou seja, todas as funcionalidades e os campos na mesma tela comuns para dois ou mais usuários diferentes. Exemplo:
      * As funcionalidades genéricas são as funcionalidades comuns a empresa e usuário. Será listada em uma coluna com seus campos
      * As funcionalidades exclusivas da empresa, aquelas só pertencem a empresa, serão listadas em outra coluna com seus campos
      * A as funcionalidades exclusivas do usuário, aquelas que só pertencem ao usuário, serão listados em outra coluna com seus campos.
* **Importante**: Após terminar a etapa de planejamento, a equipe de QA deve revisar ela junto com o time (desenvolvedores, gerente de projeto, etc.). Só assim deverá seguir para a próxima etapa que é a mapeamento.

## 2. Mapeamento

* O objetivo é listar todos os casos de testes possível para ser aplicado em cada campo do sistema. Os **casos de testes** são todos os tipos de testes possíveis que um testador terá que executar sobre um campo da tela. Cada campo pode gerar de 1 até n casos de testes.
* Oque fazer?
  + Criar a planilha Matriz de Rastreabilidade
* Como fazer?
  + Criar e preencher uma tabela com as seguintes colunas: Nome do Caso de Teste, Complexidade, Tipo de Teste, Necessidade de massa de Teste, Funcionalidade associada ao CT, Passível de automação, Passível de automação, Regressivo Obrigatório, Prioridade, Versão.
    - **Nome do Caso de Teste:**
      * Levantamento de casos de testes. Listar todos os casos de testes que um testador deve executar sobre um campo ao cadastrar produtos em sistema. Um campo pode gerar de 1 a N casos de testes pois também deve-se criar casos de testes para entradas negativas, entradas que o sistema deve recusar. Alguns exemplos de modelos de geração de casos de teste, incluindo também casos de testes para entradas negativas:
        + **Casos de testes positivos** são escritos para que o sistema tenha resposta positiva para uma entrada de dados corretos e válidos.
        + **Casos de testes negativos** são escritos para que o sistema recuse entrada de informações erradas ou inválidas.

Exemplo: Uma senha errada foi inserida em um campo de senha pelo QA de proposito, logo o sistema deve enviar a mensagem “Sua senha está incorreta. ”

* + - * + **Obs.:** Os valores que alguns campos podem receber geralmente são quatro tipos: (1) caracteres normais, (2) numéricos, (3) especiais e (3) variados.

**Caracteres normais** (apenas letras, acentos, letras com acentos ou pontuações, conforme o idioma do sistema):

*Ex.: nome do usuário, nome da empresa, apelido, etc.*

**Caracteres numéricos** (apenas números):

*Ex.: CPF, CNPJ, Telefone, Valor, etc.*

**Caracteres especiais** (apenas caracteres diferentes dos normais e também diferentes dos numéricos):

*Ex.: @$ %\*+*

**Caracteres variados** (combinações dos caracteres normais, especiais ou numéricos):

*Ex.: valor em formato monetário, etc.*

* + - * Modelo de casos de testes para validar campos de um formulário:
        + Criar casos de teste para validar um campo de texto que só deve aceitar **caracteres normais** (são apenas letras, acentos, letras com acentos ou pontuações. Ex.: campo de nome do usuário, etc.):

Validar o campo “nome” com caracteres normais.

Validar o campo “nome” com caracteres numéricos

Validar o campo “nome” com caracteres especiais.

Validar o campo “nome” com caracteres variados.

Tabela:

|  |  |
| --- | --- |
| Entrada | Saída |
| João da Silva | Ok |
| 1234 | Erro |
| %\*# | Erro |
| João 1234 %\*# | Erro |

* + - * + Criar casos de testes para validar um campo de texto que só deve aceitar **caracteres números** (apenas números. Ex.: campo de CPF, CNPJ, telefone, valores numéricos, etc.):

Validar o campo “CPF” com caracteres numéricos.

Validar o campo “CPF” com caracteres normais.

Validar o campo “CPF” com caracteres especiais.

Validar o campo “CPF” com caracteres variados.

Tabela:

|  |  |
| --- | --- |
| Entrada | Saída |
| 11122233344 | Ok |
| Um dois três | Erro |
| %\*# | Erro |
| Um dois 1234 %\*# | Erro |

* + - * + Criar casos de testes para validar um campo de texto que só deve aceitar apenas **caracteres normais, números ou ambos** (Ex.: campo de um código específico de um produto):

Validar o campo “código” com caracteres normais.

Validar o campo “código” com caracteres numéricos.

Validar o campo “código” com caracteres normais e números.

Validar o campo “código” com caracteres especiais.

Validar o campo “código” com caracteres variados.

Tabela:

|  |  |
| --- | --- |
| Entrada | Saída |
| abc | Ok |
| 123 | Ok |
| abc123 | Ok |
| %\*# | Erro |
| abc123%\*# | Erro |

* + - * + Criar casos de testes para validar um campo de texto que deve aceitar **caracteres variados** (qualquer texto em geral ou caractere. Ex.: campo de senha, comentário, descrição, endereço, localidade, valor em formato monetário – por exemplo R$ 20,00, US$ 10,00, etc.):

Validar o campo “comentário” com caracteres variados.

Tabela:

|  |  |
| --- | --- |
| Entrada | Saída |
| Meu nome é João. Tenho 30 anos. Meu salário é R$ 3.800 | Ok |

* + - * + Criar casos de testes para validar um campo de texto que só deve aceitar texto em **formato de e-mail**:

Validar o campo “e-mail” com caracteres no formato de e-mail

Validar o campo “e-mail” com caracteres em formato de e-mail inválido

Tabela:

|  |  |
| --- | --- |
| Entrada | Saída |
| usuario@abc.com | Ok |
| usuário.abc.com | Erro |

* + - * + Criar casos de testes para validar um campo que só deve aceitar texto no **formato de data**:

Validar o campo “data” no formato "DD/MM/AAAA"

Validar o campo “data” no formato diferente de "DD/MM/AAAA“

Tabela:

|  |  |
| --- | --- |
| Entrada | Saída |
| 21/08/2019 | Ok |
| 21-08-2019 | Erro |
| 21/08/19 | Erro |
| 21-08-19 | Erro |

* + - * + Criar casos de testes para validar um campo de seleção tipo **radio button**  (Ex.: selecionar sexo, selecionar idade, etc.):

Validar a seleção da opção “Masculino”.

Validar a seleção da opção “Feminino”

* + - * Modelo de casos de teste para validar botão de envio de formulário:
        + Criar casos de testes para validar um **botão de envio de formulário de informações** com campos obrigatórios (ex.: validar o envio de um formulário de cadastro)

Validar o cadastro preenchendo todos os campos

Validar o cadastro sem o preenchimento do campo 1

Validar o cadastro sem o preenchimento do campo 2

Validar o cadastro sem o preenchimento do campo 3

Validar o cadastro sem o preenchimento do campo 4

........

Tabela:

|  |  |
| --- | --- |
| Entrada | Saída |
| Preencher todos os campos | Formulário enviado |
| Preencher todos os campos, menos o campo 1 | Formulário não enviado |
| Preencher todos os campos, menos o campo 2 | Formulário não enviado |
| Preencher todos os campos, menos o campo 3 | Formulário não enviado |
| Preencher todos os campos, menos o campo 4 | Formulário não enviado |

* + - * Modelo de casos de teste para validar botão ou links de ações simples:
        + Criar casos de testes para validar um **botão para ações simples** dentro da aplicação (ex.: validar botão imprimir boleto, botão copiar código, botão fechar caixa de mensagem, etc.):

Validar o botão “imprimir”

Tabela:

|  |  |
| --- | --- |
| Entrada | Saída |
| Clicar no botão imprimir | Abrir janela de impressão |

* + - * + Criar casos de testes para validar um **link** de redirecionamento:

Validar o link “acessar página principal”

Tabela:

|  |  |
| --- | --- |
| Entrada | Saída |
| Clicar no link de redirecionamento para o site “www.abc.com” | Redirecionar página para o site “www.abc.com” |

* + - * Modelo de casos de teste para validar combinações de campos de seleção e botão:
        + Criar casos de testes para validar um campo de seleção tipo radio(Ex.: 2 opções selecionáveis X e Y):

Validar a seleção da opção X

Validar a seleção da opção Y

Validar a seleção da opção X e clicar no botão confirmar

Validar a seleção da opção Y e clicar no botão confirmar

* + - * Obs.: Alguns casos deverão ser focados para testar se o sistema gera respostas negativas e outros focados em testar se o sistema a gera respostas positivas.
      * Para escrever os casos de testes primeiro cada caso deve começar com um verbo no infinitivo indicando o campo, a funcionalidade (local ou tela onde o campo se encontra) e o tipo de entrada
        + Exemplo: Validar o campo “nome” na tela de “acesso do candidato” com caracteres normais.
        + Validar + campo + local/funcionalidade + tipo de entrada
      * Os casos de testes deverão ser listados na Matriz de Rastreabilidade seguindo o modelo de numeração sequencial: CT-0001, CT-0002, CT-0003, ..., CT-9999. (Cada caso de teste deve receber uma numeração exclusiva e única).
        + Exemplo:

CT-0001 Validar o campo “nome” com caracteres normais.

CT-0002 Validar o campo “nome” com caracteres numéricos

CT-0003 Validar o campo “nome” com caracteres especiais.

CT-0004 Validar o campo “nome” com caracteres variados.

CT-0005 ......

* + - * Caso a aplicação possuía campos com nomes parecidos em locais diferentes, indicar o local que cada campo está localizado.
        + Exemplo:

CT-0001 Validar o campo “nome” **na tela "Anunciar vaga"** com caracteres normais.

CT-0032 Validar o campo “nome” **na tela "Cadastro de Usuário"** com caracteres normais

* + - **Complexidade** (Muito Simples, Simples, Médio, Complexo, Muito Complexo.)
      * Mede a complexidade do caso de teste na hora de executa-lo:
        + Exemplos:

Teste Muito Simples:

Seleção de uma opção de um campo

Clicar em links ou botão para uma página qualquer do site.

Teste Simples:

Preenchimento de um campo

Clicar em botões simples como: botão imprimir, botão copiar código.

Teste Médio:

Clicar no botão de envio de formulário ou concluir e finalizar cadastro de usuário. Formulário de Login. Porque esse tipo de botão é considerado de complexidade média? Para realizar a tarefa é necessário que os campos do formulário estejam preenchidos como nome, telefone, CPF, idade, como enviar um e-mail. Logo não é uma tarefa simples pois é necessário preencher os campos para que a tarefa de clicar no botão seja realiza-la.

Teste Complexo:

* + - **Tipo de Teste** (Funcional Positivo, Funcional Negativo, Não Funcional, Integração, Usabilidade)
      * O Identifica qual é tipo de teste descrito:
        + Tipos:

Funcional Positivo: Entradas válidas que o sistema deve aceitar. É um caso de teste na qual esperamos um resultado positivo do sistema. Deve ser o primeiro caso a ser descrito.

Exemplo: Um caso de teste será feito para que o testador coloque caracteres normais no campo nome, o resultado esperado é que o sistema permita o uso de caráteres no campo.

Funcional Negativo: São entradas que o sistema não deve aceitar. É um caso de teste na qual esperamos um resultado negativo do sistema. Deve escrito após os casos de testes funcionais positivos.

Exemplo: Um caso de teste é feito para que o testador coloque no campo de nome caracteres especiais e números com o objetivo de verificar se o sistema recusa essas entradas. A mensagem de erro esperada é “os caracteres inseridos no campo nome são inválidos. Insira apenas caracteres normais de texto”.

Não funcional: Casos de testes voltados para o código e utilizado pelos desenvolvedores.

Integração: Casos de testes voltados para o código e utilizado pelos desenvolvedores.

Usabilidade: Caso de teste voltado para o design do sistema.

* + - **Necessidade de massa de Teste** (Sim ou Não)
      * Identifica se o caso de teste envolve:
        + Anexar arquivo ao campo

Exemplo: anexar imagem, arquivo de texto, pdf, etc.

* + - * + Quando se usa uma informação real para preencher os campos.

Exemplo: CPNJ válido, Número do cartão de crédito válido, etc.

* + - * + Quando o teste necessita de informações registradas no banco de dados para ser executado

Exemplo: Para fazer login é necessário email e senha cadastrados no banco de dados.

* + - * + Envio de formulário na qual ao menos um campo do formulário está marcado como “sim” para “Necessidade de massa de teste”
      * Obs.: O teste de massa não se aplica a casos de testes funcionais negativos quando não se usa dados válidos para o teste.
    - **Funcionalidade associada ao CT** (Seleção de funcionalidade)
      * Identifica qual a funcionalidade está ligada ao caso de teste.
      * Identifica o nome da tela ou funcionalidade onde contém o campo na qual está o caso de teste participa.
      * Exemplo: Inicial; Acesso Empresas; Anunciar Vagas; Compra Efetuada; Assinatura; Pagamento;
    - **Passível de automação** (Sim ou Não)
      * Geralmente é aplicado em tarrafas de complexidade de teste média ou complexa. Verifica se o teste é passível de automação quando contém fluxo de dados.
        + Por exemplo, caso o teste envolva clicar no botão enviar formulário na qual precise de vários campos preenchidos (momento de finalizar uma inscrição).
      * Não é passível de automação quando a tarefa é simples ou muito simples, como preenchimento de um campo, inserção de uma foto, inicialmente a coluna “passível de automação” é marcado como “não”.
      * Quando não há informação sobre a necessidade de automação no momento.
      * Obs.: Para testes automatizados é necessário uso de ferramentas especificas.
    - **Regressivo obrigatório** (Sim ou Não)
      * Caso o sistema seja atualizado ou melhorado nas próximas versões, será necessário fazer alguns testes novamente.
      * Nem todos os testes serão necessários fazer, mas alguns testes mais importantes para o sistema deverão ser refeitos.
      * Geralmente os testes serão aplicados a casos de testes a partir da complexidade média ou complexos, como envio de formulário.
      * Incluir de preferência apenas testes funcionais positivos.
      * Não incluir funcionais negativos, exceto para Login.
    - **Prioridade** (Sim ou Não)
      * Identifica quais testes tem prioridade ou não para ser executados. Caso a quantidade de casos de testes seja grande e sua execução seja extensa na qual não dê tempo para a entregar no prazo. Alguns testes deverão ter prioridade sobre os outros. Os casos de testes mais importantes devem ser feitos primeiro. O nível varia de P1, P2, P3 e P4 (da maior prioridade até a menor prioridade).
      * Prioridade P1:
        + Teste mais de maior prioridade (P1) são aqueles mais importantes e que necessitam do preenchimento de todos os campos para ser realizado, como clicar no botão “Finalizar Cadastro”, botão de envio de formulário, formulário de Login.

Exceção: Os testes funcionais negativos do mesmo caso de teste acima terão uma prioridade menor (P2), exceto para Login.

* + - * + A prioridade (P1) também inclui botão de imprimir, copiar código de barras, etc.
      * Prioridade P2:
        + Geralmente tipos de testes funcionais positivos de campos de texto ou datas tem prioridade (P2).

Exceção: Os funcionais negativos do mesmo caso de teste acima têm prioridade (P3).

* + - * + Quando a validação de todos os caracteres de um campo de texto tem grande importância, neste caso testes funcionais positivos e negativos terão a mesma prioridade (P2). Exemplo: campo CNPJ é obrigatório para certos caracteres.
      * Prioridade P3:
        + Geralmente campos de prioridade (P3) são aplicados a um campo de seleção, botões que devem ser clicados após uma seleção de campo, links para uma página qualquer do site.

**Versão**

* + - * Identifica qual versão do sistema o caso de teste está sendo feito. Exemplo: V 1.0
* Importante: Após terminar a etapa de mapeamento, a equipe de QA deve revisar ela junto com o time (desenvolvedores, gerente de projeto, etc.). É importante verificar quais os testes são necessários e quais devem ser adicionados ou removidos da lista de casos de teses. Só assim deverá seguir para a próxima etapa que é a modelagem.

## 3. Modelagem

* O objetivo é elaborar de todo o passo a passo, contando de forma mais detalhadas detalhada em relação a etapa de mapeamento, tudo que um testador precisa saber para executar cada caso de teste.
* Na etapa de modelagem é onde os cenários de testes são criados.
* Oque fazer?
  + Será necessário criar a planilha de Modelagem
* Como fazer?
  + A planilha de Modelagem terá uma tabela com as seguintes colunas:
    - **Nome do Caso de Teste**
      * Contém o mesmo nome do caso de teste criado na etapa de Mapeamento.
      * Exemplo:
        + CT-0001 Validar inserção de foto na tela de acesso de empresas.
    - **Descrição**
      * Descreve o caso de teste de forma mais detalhada para que o testador possa entender.
        + Modelo:

Verbo infinitivo + campo + local/funcionalidade

* + - * + Exemplo:

Verificar como inserir a foto na tela de cadastro de empresas na funcionalidade "Acesso Empresas".

* + - **Execução (Manual ou Automatizado)**
      * Informa se e teste será executado manualmente ou automatizado.
    - **Pré-Requisitos**
      * São informações que a pessoa precisa ter para realizar o teste.
        + Exemplo:

1. Ter uma foto disponível no formato esperado.

* + - **Massa de Testes**
      * Descrição da massa de teste que explica que o testador precisará para executar o teste. Identificar quais formatos de arquivo devem ser usados no teste, etc.
        + Exemplo:

1. Foto formato jpeg

* + - * Obs.: Deixar o item me branco caso o case não tenha necessidade massa de teste
    - **Passo**
      * Descreve o passo a passo que o testador deverá fazer para executar um teste.
        + Exemplo de 5 passos para realizar a tarefa inserir foto:

1. Acessar o site facilitatech.com.br

2. Clicar na opção empresas

3. Clicar em "Inserir foto"

4. Selecionar uma foto no formato esperado

5. Clicar em "Ok"

* + - * Obs.: Começar cada passo da lista com verbo no infinitivo.
      * Obs.: Para a elaboração desta etapa é necessário de preferência que as telas do sistema já estejam prontas e desenvolvidas pelos desenvolvedores pois elas serão necessárias para que o teste seja realizado.
    - **Resultado Esperado**
      * Descreve qual é o resultado esperado pelo sistema em cada passo descritos para realizar aquela tarefa.
        + Exemplos dos resultados esperados.

Campo selecionado corretamente.

Campo preenchido corretamente sem mensagens de erro.

Campo preenchido corretamente, com mensagem de erro adicional pelo tipo de caractere.

Site aberto corretamente.

Sistema vai redirecionar para a tela de cadastro empresas

* + - **Status**
      * Indica se o teste falhou ou não

Exemplo 1 - Teste para preenchimento de campo

* **Nome do Caso de Teste**
  + CT-0006 Validar o campo "Nome Fantasia" com caracteres normais.
* **Descrição**
  + Verificar o preenchimento do campo "Nome Fantasia" com caracteres normais, que devem aceitos.
* **Execução**
  + Manual
* **Pré-Requisitos**
  + 1. Testador adicionar apenas caracteres normais.
* **Massa de Testes**
  + Não tem
* **Passo**

|  |  |
| --- | --- |
| **Passo** | **Resultado Esperado** |
| 1. Acessar o site facilitatech.com.br | 1. Site aberto corretamente. |
| 2. Clicar na opção empresas | 2. Sistema vai redirecionar para a tela de cadastro empresas |
| 3. Preencher campo "Nome Fantasia" com caracteres normais | 3. Campo preenchido corretamente sem mensagens de erro. |

* **Status**
  + Indica se o teste falhou ou não

Exemplo 2 - Teste funcional negativo para preenchimento de campo

* **Nome do Caso de Teste**
  + CT-0026 Validar o campo "Contato" com caracteres especiais.
* **Descrição**
  + Verificar o campo "Contato" com caracteres especiais, que não devem ser permitidos pelo sistema.
* **Execução** 
  + Manual
* **Pré-Requisitos**
  + 1. Testador para adicionar apenas caracteres especiais para forçar o erro no sistema
* **Massa de Testes**
  + Não tem
* **Passo**

|  |  |
| --- | --- |
| **Passo** | **Resultado Esperado** |
| 1. Acessar o site facilitatech.com.br | 1. Site aberto corretamente. |
| 2. Clicar na opção empresas | 2. Sistema vai redirecionar para a tela de cadastro empresas |
| 3. Preencher o campo 'Contato' com caracteres especiais"!@#$%¨&¨\*". | 3. Campo preenchido corretamente, com mensagem de erro adicional pelo tipo de caractere. |

* **Status**
  + Indica se o teste falhou ou não

Exemplo 3 - Teste para envio de formulário

* **Nome do Caso de Teste**
  + CT-0028 Validar o cadastro na tela "Acesso Empresas" preenchendo todos os campos.
* **Descrição**
  + Verificar o cadastro completo na tela de "Acesso empresas"
* **Execução** 
  + Manual
* **Pré-Requisitos**
  + 1. Ter um número de CNPJ válido.
  + 2. Testador adicionar apenas informações válidas nos campos.
* **Massa de Testes**
  + 1. CNPJ Válido
* **Passo**

|  |  |
| --- | --- |
| **Passo** | **Resultado Esperado** |
| 1. Acessar o site facilitatech.com.br | 1. Site aberto corretamente. |
| 2. Clicar na opção empresas | 2. Sistema vai redirecionar para a tela de cadastro empresas |
| 3. Clicar em "Inserir foto" | 3. Windows abre uma nova janela de acesso a área de trabalho. |
| 4. Selecionar uma foto no formato esperado | 4. Selecionada conforme esperado |
| 5. Preencher o campo razão social com caracteres válidos | 5. Campo preenchido corretamente. |
| 6. Preencher o campo nome fantasia com caracteres válidos | 6. Campo preenchido corretamente. |
| 7. Preencher o campo data da fundação com caracteres numéricos no formato "DD/MM/AAAA" | 7. Campo preenchido corretamente. |
| 8. Preencher o campo CNPJ com número válido. | 8. Campo preenchido corretamente. |
| 9. Preencher o campo Estado com caracteres válidos. | 9. Campo preenchido corretamente. |
| 10. Preencher o campo Cidade com caracteres válidos | 10. Campo preenchido corretamente. |
| 11. Preencher o campo Contato com caracteres numéricos válidos. | 11. Campo preenchido corretamente. |
| 12. Clicar em "Próximo" | 12. Cadastro realizado com sucesso. |

* **Status**
  + Indica se o teste falhou ou não

Exemplo 4 - Teste para validar um campo de seleção

* **Nome do Caso de Teste**
  + CT-0055 Validar a seleção da opção "Anual" na tela de "Assinatura" e clicar no botão "Próximo"
* **Descrição**
  + Verificar o click na opção 'anual' da tela de assinatura'.
* **Execução** 
  + Manual
* **Pré-Requisitos**
  + Não tem
* **Massa de Testes**
  + Não tem
* **Passo**

|  |  |
| --- | --- |
| **Passo** | **Resultado Esperado** |
| 1. Selecionar a opção anual na tela de assinaturas. | 1. Campo selecionado corretamente. |
| 2. Clicar em "Próximo" | 2. Sistema redirecione para a tela de pagamentos. |

* **Status**
  + Indica se o teste falhou ou não

Exemplo 5 - Teste para validar botão imprimir

* **Nome do Caso de Teste**
  + CT-0077 Validar a opção de pagamento por boleto pelo botão "Imprimir"
* **Descrição**
  + Verificar pagamento com boleto pela opção imprimir
* **Execução** 
  + Manual
* **Pré-Requisitos**
  + Não tem
* **Massa de Testes**
  + Não tem
* **Passo**

|  |  |
| --- | --- |
| **Passo** | **Resultado Esperado** |
| 1. Clicar em "Imprimir" na opção boleto. | 1. Sistema abre o uma nova janela com o boleto para impressão. |
| 2. Clicar no botão "Concluir" | 2. Sistema redireciona para a tela de "Compra Efetuada". |

* **Status**
  + Indica se o teste falhou ou não

Exemplo 6 – Teste para inserção de foto em um campo

* **Nome do Caso de Teste**
  + CT-0001 Validar inserção de foto na tela de acesso de empresas.
* **Descrição**
  + Verificar como inserir a foto na tela de cadastro de empresas na funcionalidade "Acesso Empresas".
* **Execução (manualmente ou automatizado)**
  + Manual
* **Pré-Requisitos**
  + 1. Ter uma foto disponível no formato esperado.
* **Massa de Testes**
  + 1. Foto formato jpeg
* **Passo**

|  |  |
| --- | --- |
| **Passo** | **Resultado Esperado** |
| 1. Acessar o site facilitatech.com.br | 1. Site aberto corretamente. |
| 2. Clicar na opção empresas | 2. Sistema vai redirecionar para a tela de cadastro empresas |
| 3. Clicar em "Inserir foto" | 3. Windows abre uma nova janela de acesso a área de trabalho. |
| 4. Selecionar uma foto no formato esperado | 4. Selecionada conforme esperado |
| 5. Clicar em "Ok" | 5. Sistema adiciona a foto ao site da Facilita Tech. |

* **Status (Passou, Falhou, Bloqueado)**
  + Passou:
    - O teste passou. Indica que o teste conseguiu ser finalizado, e o resultado final do teste condiz com o resultado esperado.
  + Falhou:
    - O teste falhou. Indica que o teste conseguiu ser finalizado, porém o resultado final do teste é diferente do resultado esperado, talvez por causa de bugs ou falhas durante o desenvolvimento.
  + Bloqueado:
    - O teste deve ser bloqueado quando não há como finalizar ele. Logo não há como saber se o teste passa ou falha, talvez por causa de bus ou outros problemas, que impediram finalização dos testes.
    - Indica o número de testes que não foram executados, e testes que foram executados com sucesso, mas com falha.

## 4. Execução de Testes

### Coleta de evidencias

* E objetivo é que após executar os testes que foram modelados na etapa anterior, fazer a coleta das evidencias de cada um deles, para que o resultado seja entregue aos desenvolvedores o mais detalhado possível; também informar aos gerentes do projeto, com dados e números, como está o andamento dos testes.
* Obs.: Nem sempre um caso de teste executado com sucesso significa que a funcionalidade está funcionando perfeitamente. Mesmo que um teste seja bem-sucedido, é possível que ainda haja outros problemas no sistema para corrigir que não estejam relacionados ao caso de teste, como problemas no layout, etc. Todos esses problemas precisam, além do resultado do teste, devem ser descritos na coleta de evidencia.
* Coleta de evidencias
  + O objetivo é coletar evidencias. É a etapa mais importante do QA.
  + É um documento onde o resultado de cada “Passo” e “Resultado Esperado” de um caso de teste é evidenciado, tela por tela, através de prints ou vídeos
  + O documento pode ser feito pelo Word ou PowerPoint.
  + Este documento deve conter as seguintes informações: nome do projeto, nome do cliente, nome do caso de teste, funcionalidade referente, número da evidencia (de 001 a 999), nome do testador, versão do sistema, data da coleta, autor, observações.
    - Exemplo e modelo de documento:

|  |
| --- |
|  |

* + O documento também deve ser exibido prints para “Passo” e “Resultado esperado” de cada passo da etapa do caso de teste. Ou seja, deve-se evidenciar um print de antes e depois de cada etapa. Outra alternativa é a capturar tela em vídeo.
    - Exemplo de um documento contendo prints de telas de um sistema web:

|  |
| --- |
| **1 - Selecionar a opção anual na tela de assinaturas.**    **Campo selecionado corretamente.**    **2 - Clicar em "Próximo"**    **Sistema redirecione para a tela de pagamentos.** |

### Status Report e Controle de Execução

* + O objetivo é reportar o status de como está andamento dos testes. Os stakeholders e o time de desenvolvimento precisam saber como está o andamento dos testes através deste relatório pois ele usado como peça para entender como será o cronograma de entrega do projeto.
  + Durante a faze de execução o QA precisa mandar diariamente o status report dos testes para as áreas interessadas.
  + Cada empresa teu o seu modelo de status report.
  + Será necessário criar uma planilha de “Controle de Execução”, que exibe as estatísticas dos testes especificando qual funcionalidade pertence; também a planilha de “status report”, que é semelhando a planilha “Controle de Execução”, mas exibe as estatísticas dos testes sem especificar qual funcionalidade pertence, o foco do “status report” é apenas a visão geral de como está o andamento dos processos totais.
  + ***Controle de Execução***
    - A planilha de controle de execução terá as seguintes colunas:
      * **Funcionalidade**
        + Indica cada funcionalidade
      * **Testes Planejados**
        + Indica o total de casos de testes de uma funcionalidade
      * **Testes Executados**
        + Indica o total de casos de testes executados até o momento de uma funcionalidade.
        + Deve ser atualizado diariamente
      * **Bloqueados**
        + Indica o número de testes que não foram executados, e testes que foram executados com sucesso, mas com falha.
      * **Não aplicáveis**
        + Indica o número de testes de uma funcionalidade que não podem mais ser executados após uma mudança de versão do sistema.
      * **Status (Completa, Em processo, À fazer)**
        + Indica como vai o andamento dos testes. Indica se todos os testes de uma funcionalidade foram executados (Completa), se está em andamento (Em processo) ou ainda não começou a ser executado (Á fazer).
      * **Observações**
        + O objetivo é acrescentar informações adicionais em relação a execução de teste de cada funcionalidade.

Exemplo de observação da funcionalidade “Acesso”:

2 testes bloqueados por bugs / 1 teste bloqueado por acesso a ambiente / 2 testes bloqueados por falta de massa de testes

* + ***Status report***
    - A planilha de Status report terá as seguintes linhas:
    - **Executados com sucesso**
      * Indica o número de testes já executados pelo QA com sucesso.
    - **Executados com falha**
      * Indica o número de testes já executados pelo QA com falha.
    - **Pendentes**
      * Indica o número de testes que ainda não foram executados
    - **Em processo**
      * Indica o número de testes em processo.
    - **Bloqueados**
      * Indica o número de testes bloqueados.
    - **Não aplicáveis**
      * Indica o número de testes não aplicáveis para a versão atual do sistema.
    - As somas de todos os valores de todos os itens acima devem ser iguais o total de testes do planejamento. É através deste relatório que os stakeholders saberá como será o cronograma de entrega do projeto e qual será o prazo para realizar todos os testes.

## 5. Abertura e Organização de Bugs

* O objetivo é encontrar bugs no sistema durante a realização de testes e reportar para a equipe de desenvolvimento.
* Bug é uma inconsistência encontrada no sistema durante a realização de testes.
  + Ao realizar um teste, um comportamento não esperado e estranho do sistema pode acontecer, podendo infelizmenteo atrapalhar a finalização de uma execução de um teste.
  + Eles devem ser reportados com detalhe, junto com os passos executados no sistema que fizeram o bug surgir.
* Tipos de bug:
  + Bug funcional
    - Está ligado a funcionalidade de um software, da parte lógica de uma máquina ou computador. Isto é, quando você faz uma ação e o site ou aplicativo não funciona como esperado.
      * Exemplo:
        + Se considera um bug funcional quando você clica num botão e o aplicativo trava, por exemplo, obrigando a reiniciar o processo.
  + Bug de conteúdo
    - O bug de conteúdo tem relação com o conteúdo de aplicativos ou sites, como, por exemplo, problemas com texto, rótulos, links, dados, entre outros.
      * Exemplo:
        + Assim, trata-se de um bug de conteúdo quando você clica em um link, mas o redirecionamento não acontece ou então o link está “quebrado”.
        + Você está em um site sobre empreendedorismo e quer comprar um curso sobre planejamento financeiro. Ao buscar o produto, ele não aparece nos resultados. Isso é resultado de um bug de conteúdo.
  + Bug visual
    - O bug visual, por sua vez, consiste nos erros com interfaces gráficas (imagens) de sites ou aplicativos.
      * Exemplo:
        + Um texto está desalinhado ou se sobrepõe a uma imagem, além de problemas na própria estrutura do layout da página.
* Alguns fatores podem originar um bug como por exemplo:
  + Falha no momento do desenvolvimento
  + Erro de funcionalidade
  + Requisitos solicitado pelo cliente que no momento não faz sentido para o produto dele
  + É a responsabilidade do QA encontra-las.
* Nem sempre uma inconsistência encontrada em um sistema deve ser classificada como bug. As vezes pode ser inconsistência provocado por algum erro de configuração ou algo similar. Então antes reportar um bug é necessário executar o teste mais de uma vez. Se o comportamento de bug persistir, então só assim o problema será classificado como bug.
* O documento de coleta de evidencias terá o papel importante no momento de descrever o bug pois é nele que terá as informações como prints que ajudarão os desenvolvedores a entender o problema.
* Algumas empresas possuem suas próprias ferramenta de controle de bugs. Também pode ser usado planilhas para controle de bugs.
* ***Controle de Bugs***
  + Exemplo abaixo de colunas da planilha “Controle de Bugs”:
  + **Nº do Defeito** 
    - Número criado para identificar um bug (Ex.: 1, 2, 3, ...)
    - Obs.: Um único defeito pode estar atrelado em mais de um caso de teste.
  + **Descrição**
    - Descreve qual é o comportamento inesperado que está acontecendo no sistema.
      * Exemplos:
        + Na tela de assinatura não é possível selecionar o campo "Anual"
        + Na tela de pagamentos com a opção cartão de crédito não é possível adicionar o código de segurança
  + **ID Caso de Teste** 
    - Número do caso de teste atrelado ao bug (Ex.: CT-001, CT-002, … )
    - Obs.: Se houver mais de um caso de teste atrelado a esse bug, selecionar apenas um caso para ser usado como referência.
  + **Origem do Defeito**
    - Indica qual funcionalidade, tela ou local do sistema está a origem do defeito. Por exemplo, o defeito pode estar na tela de assinaturas, na tela de pagamentos, etc.
  + **Prioridade (P1, P2, P3, P4)**
    - Identifica a prioridade de correção do bug. Geralmente bugs de prioridade maior indica que aquele bug deve ser corrigido imediatamente, por exemplo, bug bloqueia alguma funcionalidade. Alguns bugs de prioridade menor, como problemas no layout, campos de texto fora do luar. Alguns desses bugs podem ser corrigidos em uma outra versão do sistema, outras não.
    - Exemplos:
      * Na tela de assinatura não é possível selecionar o campo "Anual" (P1).
        + Obs.: Se não corrigir o bug na tela de assinatura não tem como fazer o pagamento, se não fizer o pagamento o principal objetivo do sistema não será cumprido.
      * Na tela de pagamentos com a opção cartão de crédito não é possível adicionar o código de segurança (P2).
  + **Severidade (Altíssimo, alto, médio, baixo, baixíssimo).**
    - Identifica o impacto que o bug pode causar no sistema. Por exemplo, bugs que bloqueiam uma funcionalidade tem uma severidade alta, também bugs atrelados a mais de um caso de teste.
    - Exemplos:
      * Na tela de assinatura não é possível selecionar o campo "Anual" (Médio).
        + Obs.: Esse problema não afeta os casos de testes anteriores, mas afeta somente os próximos.
      * Na tela de pagamentos com a opção cartão de crédito não é possível adicionar o código de segurança (Alto).
  + **Data de abertura**
    - Data que o bug foi identificado no sistema.
  + **Status** 
    - *Aberto*:
      * Momento que o QA abre o bug.
      * O QA deve notificar o desenvolvedor sobre o bug encontrado.
    - *Em Análise:*
      * Momento que o desenvolvedor analisa se o problema é um bug, um erro de execução do QA, ou uma inconsistência pontual no sistema.
    - *Em Desenvolvimento:*
      * Após identificar o bug, o desenvolvedor altera o status para “Em desenvolvimento”.
    - *Em QA:*
      * Após a correção do bug pelo desenvolvedor, o caso volta para o QA, que novamente fará um novo teste sobre o caso.
    - *Em Aceite:*
      * Após passar pela avaliação do QA, será a vez do dono do produto avaliar o bug.
    - *Devolvido*:
      * Quando o desenvolvedor entende que aquela solicitação não é um bug, podendo ser uma inconsistência no ambiente ou na configuração. Então ele devolve o bug para o QA pois não identificado nenhuma correção.
    - *Corrigido*:
      * Após a avaliação do caso passar no teste com dono do produto, o status muda para corrigido.
      * Os bugs corrigidos devem ser testados outras vez e imediatamente pelo QA, gerente de projeto ou dono do produto.
    - *Versão*
      * Identifica qual é a versão do sistema.
* Obs.: Os bugs corrigidos devem também ser associados a regressões futuras juntos com os casos de testes regressivos obrigatórios.

## 6. Retestes e regressões

* As regressões são testes que são refeitos quando há mudança de versão do sistema.
* A regressão garante que após a alteração do sistema, de uma versão para a outras, as principais funcionalidades do sistema não sejam afetadas.
* Mesmo que apenas uma funcionalidade seja atualizada na próxima versão, os casos mais importantes das outras funcionalidades do sistema devem ser retestadas também. Além de bugs corrigidos nas versões anteriores do sistema.
* Nem todos os casos de testes devem ser regressivos. Pelo menos um caso de teste por cada funcionalidade deve entrar na lista de regressão. Ou seja, somente os casos de testes mais importantes para a funcionalidade ou principais ações do sistema devem ser classificados como regressivos. Por exemplo, caso de teste funcional positivo para envio de um formulário.
* Os casos de teste são definidos como regressivo na fase de “Mapeamento” e marcados como “Regressivo obrigatório”, e também na fase de “Abertura e Organização de Bugs” com os bugs corrigidos.
* Em alguns casos geralmente as regressões são feitas de forma automatizada.
* Se uma empresa já tem um processo de automação definido, é importante deixar os testes regressivos já programados. Basta apenas executar, assim não tomará o tempo. Casos não haja automatização das regressões, será necessário o QA estimar um tempo para trabalhar com as regressões.
* As regressões são muito importantes pois devem acontecer após alterações no sistema e precisam ser executadas constantemente durante todo o período de desenvolvimento do sistema.
* Os bugs corrigidos devem também ser associados a regressões futuras juntos com os casos de testes regressivos obrigatórios.

## 7. Entrega do Projeto

* Ferramentas de Gerenciamento de Teste
  + O planejamento de teste, execução de teste, controle e lista de regressão podem ser feitos em ferramentas específicas.
  + Exemplos de ferramentas de gerenciamento de teste
    - TestLink
    - Jira
    - Zephyr
* Finalização dos Projetos
  + Nem sempre o projeto é entregue com 100% dos testes realizados ou 100% dos bugs corrigidos, porque todas as empresas trabalham com prazo.
  + Por conta do prazo de entrega, os testes mais importantes devem ter prioridade de execução. Os testes menos importantes, caso não haja prazo para executá-los, podem ser corrigidos em uma próxima versão do produto.
  + Mesmo que 100% dos testes não tenham sido feitos, o importante é analisar as seguintes questões:
    - O produto tem qualidade?
    - O produto atenderá as necessidades do meu cliente?
    - Se a resposta é sim, então o produto está pronto para ser entregue.
    - Obs.: As funcionalidades principais e bugs, que podem impactar no sistema, devem ser corrigidos.
  + Formalização Final:
    - O QA deverá fazer uma formalização final através de e-mail ou reunião, mostrando tudo que foi executado, os bugs encontrados, bugs abertos, os testes que foram descartados, os testes que não foram executados por falta de tempo e as possíveis melhorias para as versões futuras.

## 8. Encerramento do Case

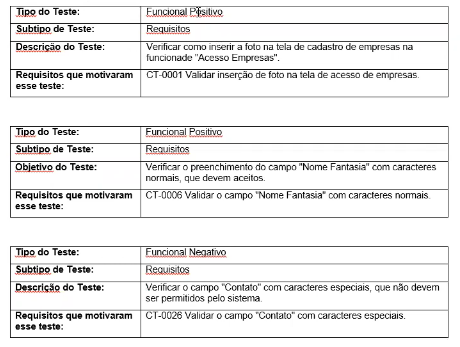
* Na reunião de encerramento o QA deve apresentar para a equipe de desenvolvedores, gerente do produto, stakeholders, todo o seu trabalho de QA no projeto.
* Você deve levantar os seguintes tópicos:
  + O que foi planejado?
  + O que foi entregue?
  + O que você encontrou com a sua visão de QA como melhorias para as futuras para o produto?
* Na metodologia ágil essa reunião é chamada de “demo”.
* Na reunião você mostrará para as partes interessadas o que foi planejado, o que estava documentado e o que foi desenvolvido e testado.
* Você deverá mostrar as principais funcionalidades de seu produto e simulando o cenário mais real possível para a entrega para o cliente.
* Documento de revisão de melhorias
  + É o documento onde é adicionado as melhorias encontradas no sistema ao longo do projeto.
  + Uma planilha pode ser usada para criar a lista de melhorias do sistema.
  + ***Lista de melhorias***
    - Exemplo abaixo de colunas da planilha “Lista de melhorias”:
    - **Funcionalidade**
      * Nome da funcionalidade
    - **Descrição**
      * Descreve a melhorias que pode ter em alguma parte do sistema
      * Exemplo:
        + Adicionar seleção para o campo estado
        + Formato de preenchimento do campo contato. Ex.: (11) 9999-9999.
    - **Versão**
      * Descreve a versão do sistema onde a melhoria foi encontrada.
  + Algumas melhorias o cliente poderá decidir se ele aceita ou não.

## 9. Plano de Testes

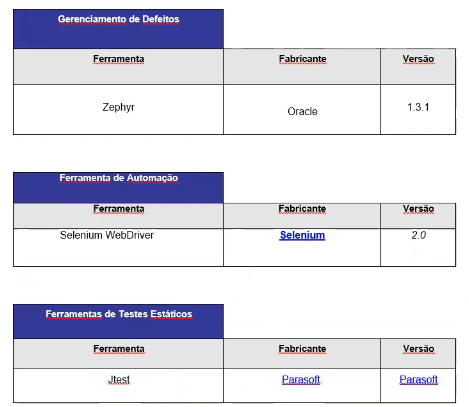
* É um guia de execução elaborado antes da fase de execução.
* É um documento opcional, não obrigatório, usado para que terceiros entendam o que o QA está fazendo.
* O objetivo do plano de teste é colocar todas as informações relevantes para o cliente.
* O documento precisa ser fácil de ser entendido.
* É como se fosse um diário do QA.
* É o momento onde é decidido o prezo e o tempo máximo para realizar os testes.
* É onde é separado os tipos de testes, quais os testes que tem maior prioridade.
* É onde é explicado quais as estratégias de teste usadas para a execução de um produto.
* O plano de teste pode ser atualizado ao longo do projeto.
* ***Oque deve conter no plano de testes?***
  + Descrição do objetivo do projeto.
  + Arquivos dos documentos do projeto anexados ao plano de testes.
    - Exemplo:
      * Protótipo das telas do sistema.
  + As técnicas de teste usadas.
    - Exemplo:
      * Será executado testes funcionais?
      * Será executado testes exploratórios?
      * Será executado testes manuais?
      * Será executado testes automatizados?
      * Será executado testes de carga?
      * Será executado testes de stress?
  + O processo que se encontra o QA e quais as etapas seguintes.
    - Exemplo:
      * O plano de teste está sendo criado na fase de planejamento porque não há informações necessárias de cenário para montar os casos de testes.
  + O estágio de teste que está sendo executados.
    - Exemplo:
      * O teste está sendo executado a cada 15 dias.
      * O teste será executa daqui a 3 meses.
  + As ferramentas usadas para gerenciar os testes usadas.
    - Exemplo:
      * Jira
      * TestLink
  + Os critérios de conclusão com êxito.
    - Exemplo:
      * O plano será cumprido quando executar 100% dos casos de testes.
      * O plano será cumprido quando executar 50% dos casos de testes.
  + A programação dos testes.
    - É colocar os testes em uma ordem lógica.
      * Exemplo:
        + O CT-008 deve ser feito após o CT-003
  + A data de entrega prevista.
    - Exemplo:
      * Quando os testes devem ser planejados
      * Quando os testes devem ser executados
      * Quando os testes devem terminar
* ***Exemplo de Estrutura do documento de Plano de Teste***
  + Capa
  + Índice
  + 1. Introdução
    - 1.1 Propósito do Sistema
      * Como fazer?
        + Copiar informações do documento do produto.
    - 1.2 Referências e Documentação do Projeto
      * Como fazer?
        + Colocar o link do documento.
  + 2. Abordagem de Testes
    - 2.1 Categorização dos Requisitos Funcionais x Funcionais
      * Requisitos Funcionais
        + Requisito Funcional Positivo
        + Requisito Funcional Negativo



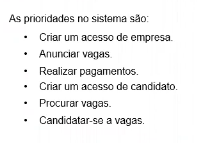
* + - 2.2. Detalhamento da abordagem dos Testes
      * Tipo de Teste
      * Subtipo de Teste
      * Descrição de Teste
      * Requisitos que motivaram esse teste



* + - 2.3. Ferramentas
      * Gerenciamento de Defeitos
        + Ferramenta
        + Fabricante
        + Versão
      * Ferramenta de Automação
        + Ferramenta
        + Fabricante
        + Versão
      * Ferramenta de Testes Estáticos
        + Ferramenta
        + Fabricante
        + Versão



* + 3. Ambiente de Teste
    - 3.1 Definição do Ambiente de Teste
      * Como fazer?
        + Descrever qual ambiente que os testes serão executados, se será na máquina dos desenvolvedores ou em ambientes específicos.
        + As ferramentas usadas para execução dos testes, IDE’s, banco de dados, frameworks, linguagens usadas no front-end e back-end, servidores, etc.
  + 4. Programação dos Testes
    - 4.1 Objetivos e Prioridades
      * Como fazer?
        + Listar as principais funções e funcionalidades do sistema



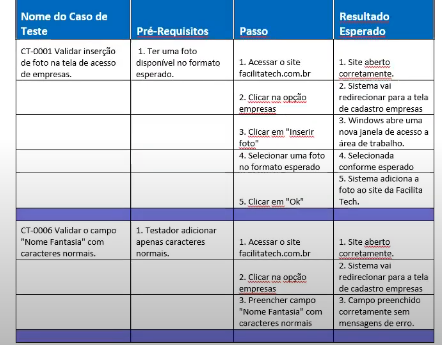
* + 5. Casos de Teste
    - * Como fazer?
        + Adicionar à lista dos casos de testes da última etapa, etapa de Modelagem.
        + Listar apenas

Nome do Caso de Teste

Pré-Requisitos

Passo

Resultado Esperado



* + 6. Cronograma de Execução

(Adicionar mais informações no documento se necessário)

## Vocabulário

**Stakeholders**

São os funcionários, fornecedores, clientes e outros parceiros de negócios. Eles precisam ser analisados para garantir que não tragam riscos inerentes para a companhia. Por isso, conhecer os stakeholders é fundamental para que as empresas consigam tomar decisões melhores e alcançar seus objetivos sem abrir mão da segurança dos negócios. Eles podem ser clientes, colaboradores, fornecedores, comunidade e gestores.

**Teste Funcional**

Testa os requisitos funcionais da aplicação e software. Resumidamente verifica se a aplicação está apta a realizar as funções na qual foi desenvolvida para fazer (Teste de Caixa-branca, Teste de Caixa-preta).

Exemplos:

* Testar recursos em uma página da web, como pesquisas, menus, links;
* Executar testes de sistema para avaliar funções executadas para concluir uma tarefa importante, como comprar um produto em um site ou e-commerce;
* Realizar o teste de aceitação do usuário, para avaliar se o software faz o que promete.

**Testes não funcionais**

Testa aspectos não-funcionais do software, como por exemplo, os testes de desempenho, teste de carga ou teste de usabilidade. As técnicas não funcionais verificam atributos de um componente ou sistema que não se relacionam com a funcionalidade (por exemplo, confiabilidade, eficiência, usabilidade, performance, manutenção e portabilidade).

**Fontes:**

<http://cachoeiradosul-portais.govcloud.com.br:8080/cidadao/help/Funcionalidade_do_sistema.htm>

<https://www.techtudo.com.br/listas/2021/11/caracteres-especiais-entenda-o-que-sao-e-conheca-atalhos.ghtml>

<https://www.pontotel.com.br/stakeholders/#:~:text=Os%20stakeholders%20s%C3%A3o%2C%20ent%C3%A3o%2C%20todos,e%20comunidade%2C%20chamados%20de%20externos>.

<https://king.host/blog/2022/08/o-que-e-bug/>

<https://blog.vericode.com.br/testes-nao-funcionais-o-que-sao-e-como-podem-ajudar-a-melhorar-as-aplicacoes/>

<https://www.devmedia.com.br/testes-funcionais-de-software/23565>

<https://visuresolutions.com/pt/blog/non-functional-requirements>

<https://www.testar.me/teste-de-software>

<https://www.primecontrol.com.br/o-que-sao-cenarios-scripts-e-casos-de-teste/>

<https://blog.neoway.com.br/stakeholders/#:~:text=Stakeholders%20s%C3%A3o%20todas%20as%20pessoas,direta%20ou%20indiretamente%20%E2%80%93%20por%20ela>.

<https://www.portugal-a-programar.pt/forums/topic/42855-passar-de-caracteres-especiais-para-normais/>

<https://www.projetoacbr.com.br/forum/topic/45734-caracteres-especiais-na-importa%C3%A7%C3%A3o-de-xml/>

<https://learn.microsoft.com/pt-br/power-platform/power-fx/reference/function-ismatch>

<https://www.portugues.com.br/gramatica/particularidades-alguns-sinais-pontuacao.html#:~:text=Usualmente%2C%20os%20sinais%20de%20pontua%C3%A7%C3%A3o,%2C%20par%C3%AAnteses%2C%20v%C3%ADrgula%20e%20colchetes>.