Cenários de Teste

Cenários Gerais

https://www.softwaretestinghelp.com/sample-test-cases-testing-web-desktop-applications/

1. Todos os campos obrigatórios devem ser validados e devem estar indicados por um símbolo (*) de asterisco, preferencialmente em cor vermelha.
2. As mensagens de erro de validação devem ser exibidas corretamente na posição correta.
3. Todas as mensagens de erro devem ser exibidas no mesmo estilo CSS (por exemplo, usando a cor vermelha).
4. As mensagens de confirmações gerais devem ser exibidas usando o estilo CSS diferente do estilo das mensagens de erro (por exemplo, usando a cor verde).
5. O texto das dicas de “elementos” deve ser significativo.
6. Os campos ‘*drop-down*’ devem ter a primeira entrada em branco ou texto como ‘selecionar’.
7. Funcionalidade ‘Excluir' deve sempre solicitar uma confirmação.
8. A opção Selecionar / desmarcar todos os registros deve ser fornecida se a página suportar a funcionalidade de adicionar / excluir / atualizar registros.
9. Os valores financeiros devem ser exibidos com símbolos de moeda corretos.
10. A ordenação padrão da página deve ser fornecida.
11. A funcionalidade ‘Redefinir’ (*Reset Button*) deve definir valores padrão para todos os campos.
12. Todos os valores numéricos devem ser formatados corretamente.
13. Os campos de entrada devem ser verificados quanto ao valor máximo do campo. Valores de entrada maiores que o limite máximo especificado não devem ser aceitos ou armazenados no banco de dados.
14. Verifique todos os campos de entrada relativamente a caracteres especiais.
15. Os rótulos (*labels*) dos campos devem ser padrão, por exemplo: o campo que aceita nome do usuário deve ser rotulado corretamente como "Nome".
16. Verifique a funcionalidade de ordenação da página após a execução das operações de adicionar/editar/excluir em qualquer registro.
17. Verifique a funcionalidade de tempo limite (*Timeout*). Os valores de tempo limite devem ser configuráveis. Verifique o comportamento da aplicação após alcançar o tempo limite da operação.
18. Verifique os cookies usados em uma aplicação.
19. Verifique se os arquivos para download (downloadable files) estão apontando corretamente para os caminhos e nomes de arquivo, em que serão salvos.
20. Todas os recursos devem ser configurados em arquivos de configuração ou banco de dados, não no código-fonte.

21. As convenções padrão devem ser seguidas para a nomeação de recursos chave.
22. Valide marcadores para todas as páginas da Web (valide HTML e CSS para erros de sintaxe) para garantir que sejam compatíveis com os padrões.
23. Falha na aplicação ou páginas indisponíveis devem ser redirecionadas para a página de erro.
24. Verifique os textos em todas as páginas quanto a erros ortográficos e gramaticais.
25. Verifique os campos de entrada numéricos com os valores de entrada de caracteres. A mensagem de validação adequada deve aparecer.
26. Verifique números negativos, se permitido, para campos numéricos.
27. Verifique os campos de valor com valores numéricos decimais.
28. Verifique a funcionalidade dos botões disponíveis em todas as páginas.
29. O usuário não poderá enviar (*submit*) a página duas vezes pressionando o botão enviar em rápida sucessão.
30. Erros de divisão por zero devem ser tratados para quaisquer cálculos.
31. Os dados de entrada com a primeira e a última posição em branco devem ser manipulados corretamente.

Cenários de Teste de Usabilidade e Interface Gráfica de Usuário (GUI)

1. Todos os campos na página (por exemplo: *textbox, radio options, drop-down lists*) devem estar alinhados corretamente.
2. Os valores numéricos devem ser justificados corretamente, a menos que especificado de outra forma.
3. Deve haver espaço suficiente entre rótulos (*labels*) de campos, colunas, linhas, mensagens de erro etc.
4. Barras de rolagem devem ser ativadas apenas quando necessário.
5. O tamanho da fonte, estilo e cor do título, texto da descrição, rótulos, dados internos dos campos e informações da grade devem ser padrão, conforme especificado no documento de requisitos.
6. Os valores em uma '*textbox*' devem estar dispostos em multilinhas.
7. Os campos desabilitados devem estar acinzentados e os usuários não devem poder focar nesses campos.
8. Ao clicar em um campo de entrada de texto, o ponteiro da seta do mouse deve ser alterado para o cursor.
9. O usuário não poderá digitar valores em listas '*drop-down*'.
10. As informações preenchidas pelos usuários devem permanecer intactas quando houver uma mensagem de erro no envio da página. O usuário deve poder enviar o formulário novamente, corrigindo os erros.
11. Verifique se os *labels* dos campos são usados adequadamente nas mensagens de erro.
12. Os valores de campos '*drop-down*' devem ser exibidos em uma ordem de classificação definida.
13. Tab e Shift + Tab devem funcionar corretamente.

14. As opções padrão em ‘*radio buttons*’ devem ser pré-selecionadas no carregamento da página.
15. As mensagens específicas de ajuda tanto em nível de campo, quanto de página devem estar disponíveis.
16. Verifique se os campos são realçados corretamente em caso de erros.
17. Verifique se as opções de listas ‘*drop-down*’ são legíveis e não truncadas devido ao limite de tamanho do campo.
18. Todos os botões em uma página devem ser acessíveis por atalhos de teclado e o usuário deve poder executar todas as operações usando um teclado.
19. Verifique todas as páginas para imagens quebradas (imagens que não são carregadas corretamente).
20. Verifique todas as páginas para links quebrados (inacessíveis).
21. Todas as páginas devem ter um título.
22. As mensagens de confirmação devem ser exibidas antes de executar qualquer operação de atualização ou exclusão.
23. A ampulheta deve ser exibida quando a aplicação estiver executando algum processamento.
24. O texto da página deve ser justificado.
25. O usuário deve poder selecionar apenas uma opção de ‘*radio button*’ e qualquer combinação para caixas de seleção.

Cenários de Teste para Critérios de Filtros

1. O usuário deve poder filtrar os resultados usando todos os parâmetros na página.
2. A funcionalidade de refinar a pesquisa deve carregar a página de pesquisa com todos os parâmetros de pesquisa selecionados pelo usuário.
3. Quando houver pelo menos um critério de filtro necessário para executar a operação de pesquisa, verifique se a mensagem de erro adequada é exibida quando o usuário envia a página sem selecionar nenhum critério de filtro.
4. Quando a seleção de, pelo menos, um critério de filtro não é obrigatória, o usuário deve poder enviar a página e os critérios de pesquisa padrão devem ser usados para consultar os resultados.
5. Mensagens de validação adequadas devem ser exibidas para todos os valores inválidos para os critérios de filtro.

Cenários de Teste para *Grid* de Resultados

1. O símbolo de carregamento da página deve ser exibido quando estiver demorando mais do que o tempo padrão para carregar a página de resultados.
2. Verifique se todos os parâmetros de pesquisa são usados para buscar dados mostrados na grade de resultados.
3. O número total de resultados deve ser exibido na grade de resultados.
4. Os critérios de pesquisa usados devem ser exibidos na grade de resultados.
5. Os valores da grade de resultados devem ser classificados por coluna padrão.
6. As colunas ordenadas devem ser exibidas com um ícone de ordenação.
7. As grades de resultados devem incluir todas as colunas especificadas com valores corretos.

8. A funcionalidade de ordenação crescente e decrescente deve funcionar para colunas suportadas pela ordenação de dados.
9. As grades de resultados devem ser exibidas com espaçamento adequado entre colunas e linhas.
10. A paginação deve ser ativada quando houver mais resultados do que a contagem de resultados padrão por página.
11. Verifique a funcionalidade de paginação Próxima, Anterior, Primeira e Última página.
12. Registros duplicados não devem ser exibidos na grade de resultados.
13. Verifique se todas as colunas estão visíveis e a barra de rolagem horizontal está ativada, se necessário.
14. Verifique os dados para colunas dinâmicas (colunas cujos valores são calculados dinamicamente com base nos valores de outras colunas).
15. Para grades que mostram resultados de relatórios, verifique os totais por linhas e por colunas.
16. Para grades que mostram resultados de relatórios, verifique os totais por linha, quando a paginação é ativada e o usuário navega para a próxima página.
17. Verifique se símbolos adequados são usados para exibir valores de coluna, por exemplo O símbolo '%' deve ser exibido para o cálculo da porcentagem.
18. Verifique na *grid* os dados retornados para saber se a faixa de valores está habilitada.

Cenários de Teste para uma Tela.

1. Verifique se o tamanho padrão da tela está correto.
2. Verifique se o tamanho da tela filha está correto..
3. Verifique se há algum campo na página com foco padrão (em geral, o foco deve ser definido no primeiro campo de entrada da tela).
4. Verifique se as telas filhas estão sendo fechadas ao fechar a tela mãe.
5. Se a tela filha for aberta, o usuário não poderá usar ou atualizar nenhum campo na tela principal ou em segundo plano.
6. Verifique a funcionalidade para minimizar, maximizar e fechar a tela.
7. Verifique se a tela é redimensionável.
8. Verifique a funcionalidade da barra de rolagem para ver as telas mãe e filha.
9. Verifique a funcionalidade do botão Cancelar para a tela filha.

Test Cenários para Teste de Banco de Dados

1. Verifique se os dados estão sendo salvos corretamente no banco de dados após o envio bem-sucedido da página.
2. Verifique valores das colunas que não aceitam valores nulos.
3. Verifique a integridade dos dados. Os dados devem ser armazenados em tabelas únicas ou múltiplas, com base no design.
4. Os nomes dos índices devem ser fornecidos de acordo com os padrões, p. IND_ <Tablename> _ <ColumnName>.
5. As tabelas devem ter uma coluna de chave primária.

6. As colunas da tabela devem ter informações de descrição disponíveis (exceto as colunas de auditoria, como data de criação, criada por, etc.).
7. Para cada operação de adicionar / atualizar no banco de dados, o arquivo de log da operação da aplicação deve ser atualizado.
8. Os índices obrigatórios das tabelas devem ser criados.
9. Verifique se os dados estão confirmados no banco de dados somente quando a operação for concluída com êxito.
10. Os dados devem ser revertidos (roolback) em caso de falha nas transações.
11. O nome do banco de dados deve ser fornecido de acordo com o tipo de aplicação, ou seja, teste, UAT, sandbox, ao vivo (embora esse não seja um padrão, é útil para a manutenção do banco de dados).
12. Os nomes lógicos do banco de dados devem ser fornecidos de acordo com o nome do banco de dados (novamente, isso não é padrão, mas útil para a manutenção do banco de dados).
13. Os procedimentos armazenados não devem ser nomeados com o prefixo "sp_".
14. Verifique se os valores das colunas de auditoria da tabela (como a data de criação, criada por, atualizada, atualizada por, são excluídos, dados excluídos, excluídos por etc.) são preenchidos corretamente.
15. Verifique se os dados de entrada não estão truncados ao salvar. O comprimento do campo mostrado ao usuário na página e no esquema do banco de dados deve ser o mesmo.
16. Verifique os campos numéricos com os valores mínimo, máximo e precisão de ponto flutuante.
17. Verifique os campos numéricos com valores negativos (para aceitação e não aceitação).
18. Verifique se as opções dos '*radio buttons'* e das listas '*drop-down*' estão salvas corretamente no banco de dados.
19. Verifique se os campos do banco de dados foram projetados com o tipo e comprimento de dados corretos.
20. Verifique se todas as restrições da tabela, como chave primária, chave estrangeira etc., estão implementadas corretamente.
21. Teste '*stored procedures'* e '*triggers*' com algum dado de entrada.
22. Os espaços iniciais e finais do campo de entrada devem ser truncados antes de confirmar os dados no banco de dados.
23. Valores nulos não devem ser permitidos para a coluna Chave primária.

Cenários de Teste para a Funcionalidade de *Upload* de Imagem

(Aplicável também para a funcionalidade de upload de outros tipos de arquivos)

1. Verifique o caminho da imagem a que se refere o *upload*.
2. Verifique o *upload* da imagem e altere a funcionalidade, se necessário.
3. Verifique o *upload* de imagens com arquivos de diferentes tipos de extensão (por exemplo: JPEG, PNG, BMP, etc.).
4. Verifique a funcionalidade de *upload* de imagens com arquivos que contenham espaço ou qualquer outro caractere especial permitido no nome do arquivo.
5. Verifique a duplicidade de nomes de imagens carregadas.

6. Verifique o *upload* de imagens com tamanho de arquivo maior que o tamanho máximo permitido. A mensagem de erro adequada deve ser exibida.
7. Verifique a funcionalidade de *upload* de imagens com outros tipos de arquivo (por exemplo, txt, doc, pdf, exe etc.). Uma mensagem de erro adequada deve ser exibida.
8. Verifique se as imagens com altura e largura especificadas (se definidas) são aceitas de outra forma.
9. A barra de progresso do *upload* da imagem deve aparecer para imagens de tamanho grande.
10. Verifique se a funcionalidade Cancelar funciona durante o processo de *upload*.
11. Verifique se a caixa de diálogo de seleção de arquivos lista apenas os arquivos dos tipos suportados.
12. Verifique a funcionalidade de *upload* para várias imagens.
13. Verifique a qualidade da imagem após o *upload*. A qualidade da imagem não deve ser alterada.
14. Verifique se o usuário é capaz de usar / visualizar as imagens carregadas.

Cenários de Teste para a Funcionalidade de Envio de E-mails

(Os casos mais importantes para composição ou validação de e-mails não estão incluídos aqui)
(Certifique-se de usar endereços fictícios de e-mail antes de executar os testes)

1. O *template* de e-mail deve usar CSS padrão para todos os e-mails.
2. Os endereços de e-mail devem ser validados antes do envio.
3. Caracteres especiais no *template* do corpo do e-mail devem ser tratados adequadamente.
4. Caracteres específicos de idioma (por exemplo, caracteres de idioma russo, chinês ou alemão) devem ser tratados adequadamente no *template* do corpo do e-mail.
5. O assunto do e-mail não deve ficar em branco.
6. Os campos de espaço reservado usados no *template* de e-mail devem ser substituídos por valores reais, por exemplo {Nome} {Sobrenome} deve ser substituídos individual e corretamente por nome e sobrenome, para todos os destinatários.
7. Se relatórios com valores dinâmicos forem incluídos no corpo do e-mail, os dados do relatório deverão ser calculados corretamente.
8. O nome do remetente do e-mail não deve ficar em branco.
9. Os e-mails devem ser verificados em diferentes clientes, como Outlook, Gmail, Hotmail, Yahoo!, etc.
10. Verifique a funcionalidade de envio de e-mail usando os campos TO, CC e BCC.
11. Verifique os e-mails com texto sem formatação.
12. Verifique os e-mails em formato HTML.
13. Verifique o cabeçalho e o rodapé do e-mail quanto ao logotipo da empresa, política de privacidade e outros links.
14. Verifique e-mails com anexos.

15. Verifique a funcionalidade de envio de e-mail para destinatários únicos, múltiplos ou da lista de distribuição.
16. Verifique se uma resposta ao endereço de e-mail está correta.
17. Verifique o envio de um grande volume de e-mails.

Cenários de Teste para a Funcionalidade de Exportar para Excel.

1. O arquivo deve ser exportado com a extensão correta.
2. O nome do arquivo exportado para Excel deve estar de acordo com os padrões, por exemplo se o nome estiver usando data / hora, ele deve ser substituído corretamente por uma data / hora real do momento da exportação.
3. Verifique o formato da data se o arquivo exportado para Excel contiver este dado.
4. Verifique a formatação de valores numéricos ou monetários. A formatação deve ser a mesma mostrada na página.
5. O arquivo exportado deve ter colunas com nomes adequados.
6. A ordenação padrão na página deve ser a mesma no arquivo exportado.
7. Os dados do arquivo do Excel devem ser formatados corretamente com os valores do texto do cabeçalho e rodapé, data, número das páginas, etc. para todas as páginas.
8. Verifique se os dados exibidos em uma página e no arquivo exportado para Excel são os mesmos.
9. Verifique a funcionalidade de exportação quando a paginação estiver ativada.
10. Verifique se o botão de exportação está mostrando o ícone adequado, de acordo com o tipo de arquivo exportado, por exemplo: ícone xls para arquivos do Excel.
11. Verifique a funcionalidade de exportação para arquivos com um tamanho muito grande.
12. Verifique a funcionalidade de exportação para páginas contendo caracteres especiais. Verifique se esses caracteres especiais foram exportados corretamente no arquivo do Excel.

Cenários para Teste de Performance

1. Verifique se o tempo de carregamento da página está dentro da faixa aceitável.
2. Verifique o carregamento da página em conexões lentas.
3. Verifique o tempo de resposta para qualquer ação em condições de carga leve, normal, moderada e pesada.
4. Verifique o desempenho dos *stored_procedures* e *triggers* armazenados do banco de dados.
5. Verifique o tempo de execução da consulta ao banco de dados.
6. Verifique o teste de carga da aplicação.
7. Verifique o teste de estresse da aplicação.
8. Verifique o uso da CPU e da memória sob a condição de pico de carga.

Cenários para Teste de Segurança

1. Verifique a injeção de SQL.
2. As páginas seguras devem usar o protocolo HTTPS.
3. Falha na página não deve revelar informações da aplicação ou do servidor. A página de erro deve ser exibida para isso, inclusive informar o arquivo de log da aplicação.
4. Escape de caracteres especiais como entrada.
5. As mensagens de erro não devem revelar nenhuma informação sensível.
6. Todas as credenciais devem ser transferidas por um canal criptografado.
7. Teste a segurança da senha e a aplicação da política de senha.
8. Verifique a funcionalidade de logout da aplicação.
9. Verifique se há ataques de força bruta.
10. As informações sobre cookies devem ser armazenadas apenas no formato criptografado.
11. Verifique a duração do cookie da sessão e o encerramento da sessão após o tempo limite ou logout.
12. Os tokens de sessão devem ser transmitidos por um canal seguro.
13. A senha não deve ser armazenada em cookies.
14. Teste para ataques de negação de serviço (DoS).
15. Teste para vazamento de memória.
16. Teste o acesso de aplicações não autorizadas, manipulando valores de variáveis na barra de endereços do navegador.
17. Teste a manipulação de extensões de arquivos para que os arquivos ‘exe’ não sejam carregados e executados no servidor.
18. Os campos confidenciais, como senhas e informações do cartão de crédito, não devem ter o preenchimento automático ativado.
19. A funcionalidade de upload de arquivos deve usar restrições de tipo de arquivo e também antivírus para verificar os arquivos enviados.
20. Verifique se a listagem de diretórios é proibida.
21. A senha e outros campos confidenciais devem ser mascarados durante a digitação.
22. Verifique se a funcionalidade de senha esquecida está protegida com recursos como validade temporária da senha após o horário especificado e se pergunta de segurança antes de alterar ou solicitar uma nova senha.
23. Verifique a funcionalidade CAPTCHA.
24. Verifique se os eventos importantes estão registrados nos arquivos de log.
25. Verifique se os privilégios de acesso estão implementados corretamente.

Cenários para o Teste de Penetração (*Pen Testing*)

https://www.softwaretestinghelp.com/penetration-testing-guide/

1. Verifique se a aplicação web é capaz de identificar ataques de spam nos formulários de contato usados no site.
2. Servidor proxy – verifique se o tráfego de rede é monitorado por dispositivos proxy. O servidor proxy dificulta que os hackers obtenham detalhes internos da rede, protegendo o sistema contra ataques externos.

3. Filtros de spam – Verifique se o tráfego de entrada e saída de e-mails é filtrado e se os e-mails não solicitados estão bloqueados.
4. Muitos clientes de e-mail vêm com filtros de spam incorporados que precisam ser configurados conforme suas necessidades. Essas regras de configuração podem ser aplicadas a cabeçalhos, assunto ou corpo de e-mail.
5. Firewall – Verifique se toda a rede ou computadores estão protegidos com o Firewall. Um firewall pode ser um software ou hardware para bloquear o acesso não autorizado a um sistema. Um firewall pode impedir o envio de dados para fora da rede sem a sua permissão.
6. Tente explorar todos os servidores, sistemas de desktop, impressoras e dispositivos de rede.
7. Verifique se todos os nomes de usuário e senhas estão criptografados e transferidos por uma conexão segura, como https.
8. Verifique as informações armazenadas nos cookies do site. Não deve estar em um formato legível.
9. Verifique as vulnerabilidades encontradas anteriormente para verificar se a correção está funcionando.
10. Verifique se não há porta aberta na rede.
11. Verifique todos os dispositivos de telefone.
12. Verifique a segurança da rede WIFI.
13. Verifique todos os métodos HTTP. Os métodos PUT e Delete não devem ser ativados em um servidor web.
14. Verifique se a senha atende aos padrões exigidos. A senha deve ter pelo menos 8 caracteres, contendo pelo menos um número e um caractere especial.
15. O nome de usuário não deve ser como “admin” ou “administrador”.
16. A página de login da aplicação deve ser bloqueada após algumas tentativas malsucedidas de login.
17. As mensagens de erro devem ser genéricas e não devem mencionar detalhes específicos, como “Nome de usuário inválido” ou “Senha inválida”.
18. Verifique se caracteres especiais, tags HTML e scripts são tratados corretamente como um valor de entrada.
19. Os detalhes internos do sistema não devem ser revelados em nenhuma das mensagens de erro ou alerta.
20. Mensagens de erro personalizadas devem ser exibidas ao usuário final em caso de falha da página da web.
21. Verifique o uso de entradas do registro. Informações confidenciais não devem ser mantidas no registro.
22. Todos os arquivos devem ser escaneados (contra vírus) antes de serem enviados ao servidor.
23. Dados confidenciais não devem ser transmitidos em URLs durante a comunicação entre diferentes módulos internos da aplicação Web.
24. Não deve haver nenhum nome de usuário ou senha codificados no sistema.
25. Verifique todos os campos de entrada com uma *string* longa com e sem espaços.
26. Verifique se a funcionalidade de redefinição de senha é segura.
27. Verifique a aplicação para SQL Injection.
28. Verifique a aplicação for Cross Site Scripting.

29. Validações importantes de entrada devem ser feitas no lado do servidor, em vez de verificações de JavaScript no lado do cliente.
30. Recursos críticos no sistema devem estar disponíveis apenas para pessoas e serviços autorizados.
31. Todos os logs de acesso devem ser mantidos com permissões de acesso adequadas.
32. Verifique se a sessão do usuário termina após o logoff.
33. Verifique se a navegação no diretório está desabilitada no servidor.
34. Verifique se todos as versões das aplicações e banco de dados estão atualizados.
35. Verifique a manipulação de URL para checar se uma aplicação Web não está mostrando nenhuma informação indesejada.
36. Verifique o vazamento de memória e o estouro de buffer.
37. Verifique se o tráfego de rede recebido é escaneado para encontrar ataques de Trojan.
38. Verifique se o sistema está protegido contra ataques de força bruta – um método de tentativa e erro para encontrar informações confidenciais como senhas.
39. Verifique se o sistema ou a rede está protegido contra ataques de negação de serviço (DoS). O hacker pode direcionar a rede ou um único computador com solicitações contínuas, devido à sobrecarga de recursos no sistema de destino, resultando na negação de serviço para solicitações legítimas.
40. Verifique a aplicação contra ataques de injeção de script HTML.
41. Verifique contra ataques COM e ActiveX.
42. Verifique se há ataques de falsificação (*Spoofing attack*). A falsificação pode ser de vários tipos – falsificação de endereço IP, falsificação de ID de e-mail,
43. Falsificação ARP, falsificação de referenciador, falsificação de identificação de chamadas, envenenamento de redes de compartilhamento de arquivos, falsificação de GPS.
44. Verifique se há um ataque não controlado no formato de *String* - um ataque de segurança que pode causar o travamento do aplicativo ou a execução de um script prejudicial.
45. Verificar ataque de injeção XML – usado para alterar a lógica pretendida da aplicação.
46. Verifique contra ataques de canonização (*canonicalization attacks*).
47. Verifique se as páginas de erro estão exibindo alguma informação que possa ser útil para um hacker entrar no sistema.
48. Verifique se algum dado crítico como a senha está armazenado em arquivos secretos no sistema.
49. Verifique se a aplicação está retornando mais dados do que o necessário.