

1. Cenário geral do Site do Laboratório de Segurança e Criptografia Aplicada (LASCA) - UNICAMP

Este documento tem o objetivo de documentar as informações necessárias para planejar e controlar os testes de validação do projeto WEB - LASCAWeb. O documento descreve o plano geral de testes referente ao sistema Django Admin de forma a direcionar os esforços de teste e os Casos de Teste a serem executados para validar o produto.

Escopo

Este documento descreve o Plano de Testes a ser usado pelo projeto WEB - LASCAWeb para avaliar a qualidade funcional. O teste que será coberto por este documento será: Utilização do módulo de *Publications*.

Este módulo permite adicionar publicações de journals, congressos e outros que fossem publicados pelos membros do LASCA.

Referências

[1] <https://www.lasca.ic.unicamp.br/site/publications/>

[2] <https://github.com/ecram/djangolab>

2. Estratégia(s) de Teste (como será testado)

Para o nosso projeto foi utilizado o Selenium IDE que é uma versão para Browsers Firefox para a realização de testes funcionais em Web Sites. Também através desses testes foram garantidos a injeção e extração de dados do banco, que no nosso caso é o Postgres.

Objetivo dos Testes Funcionais

Assegurar o correto funcionamento dos recursos oferecidos pelo software, incluindo o registro de novas publicações, processamento dos dados de entrada e pesquisa de publicações.

Técnica Utilizada: Teste Funcional com Particionamento em Classes de Equivalência

Os testes foram feitos verificando-se o fluxo dos dados de entrada para cada um deles, verificando-se a resposta do programa, se foi adequada às entradas, reportando erros se os dados eram inválidos ou gerando saídas corretas, a partir de dados válidos, mais conhecido como par entrada, resultado esperado.

A fim de utilizar o critério de particionamento em classes de equivalência, foi dividido o domínio de entradas em classes de equivalência (subconjuntos) que, de acordo com a especificação do módulo *Publications*, utilizando as aulas da semana 3.

Critérios

Foi executado cada caso de uso, seu fluxo de caso ou função, utilizando dados válidos ou inválidos, para verificar o seguinte:

- O resultado esperado ocorre quando entradas válidas são utilizadas.
- Erros apropriados ocorrem quando entradas inválidas são utilizadas.
- Se a condição de entrada especifica um intervalo, então define-se uma classe válida e duas inválidas.
- Se a condição de entrada especifica uma quantidade, então define-se uma classe válida e duas inválidas.
- Se a condição especifica conjunto determinados de valores, define-se uma classe válida para cada conjunto e uma classe inválida com outro valor qualquer.
- Se a condição de entrada é específica ("deve ser assim"), então define-se uma classe válida e uma inválida.

Ferramenta: Selenium WebDriver

O WebDriver foi projetado para fornecer uma interface de programação mais simples e concisa. O Selenium-WebDriver suporta páginas web dinâmicas. O objetivo do WebDriver é fornecer uma API orientada a objetos bem projetada que forneça suporte aprimorado para problemas modernos de teste de aplicativos da web avançados.

3. Projeto de Casos de Teste (como será testado)

Para a definição dos Casos de Teste foram utilizados o modulo de Publications.

CASO Nº	CT01 – Realizar acesso ao Django Admin.
OBJETIVO DO TESTE	Verificar se o usuário consegue efetuar o login.
PASSOS	1. Acessar a página de login de Django Admin: Site: www.lasca.ic.unicamp.br → Login 2. Informe o login de rede: 3. Informe a senha:
CRITÉRIOS DE ÊXITO	O usuário deve conseguir acessar a próxima tela onde irá selecionar uns dos módulos do sistema.
DADOS DE ENTRADA E RESULTADOS ESPERADOS	CT0101 ("user", "password", "Site administration") CT0102 ("user", "", "Please correct the error below.")

	CT0103 (“”, “password”, “Please correct the error below.”) CT0104 (“user”, “p@ssw0rdx20”, “Please correct the error below.”)
--	---

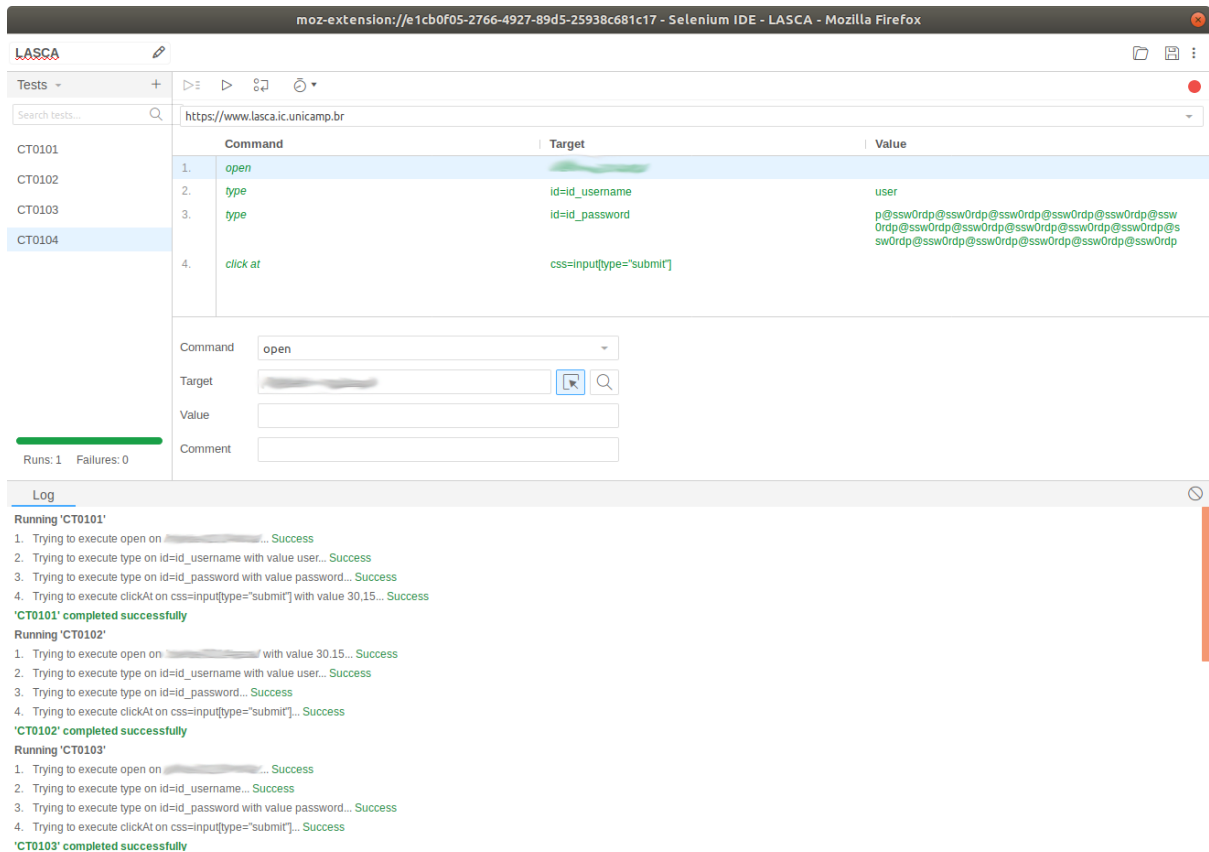
CASO Nº	CT02 – Adicionar uma nova publicação pelo formulário.
OBJETIVO DO TESTE	Verificar se o usuário consegue adicionar uma nova publicação.
PASSOS	<ol style="list-style-type: none"> 1. Acessar a página de login de Django Admin: Site: www.lasca.ic.unicamp.br → Login. 2. Informe o login de rede correta 3. Informe a senha correta 4. Selecionar modulo Publications. 5. Seleccionar em adicionar nova publicação por formulário 6. Ingressar dados solicitados: <ol style="list-style-type: none"> 1. Informe o Titulo (<i>Title</i>) 2. Ano (<i>Year</i>) 3. Month 4. PDF (arquivo maior a 25Mb) 5. URL (congress)
CRITÉRIOS DE ÊXITO	O usuário deve conseguir adicionar por formulário uma nova publicação ou receber mensagens de erros em caso de introduzir dados inválidos.
DADOS DE ENTRADA E RESULTADOS ESPERADOS	CT0201 (Title, “Software Testing”, “valid”) CT0202 (Title, “”, invalid) CT0203 (Title, “32434%%#@\$\$%\$#”, invalid) CT0204 (Ano, 2018, valid) CT0205 (Ano, -2018, invalid) CT0206 (Ano, “ano”, invalid) CT0207 (Month, 09, valid) CT0208 (Month, 13, invalid) CT0209 (Month, “mes”, invalid) CT0210 (PDF, “normal_file.pdf”, valid)

	CT0211 (PDF, "big_files.pdf", invalid) CT0212 (PDF, "file.exe", invalid) CT0213 (URL, "http://www.ieee.org", valid) CT0214 (URL, "", invalid) CT0215 (URL, "unknown", invalid)
--	--

CASO Nº	CT03 – Adicionar uma nova publicação pelo formulário.
OBJETIVO DO TESTE	Verificar se o usuário consegue adicionar uma nova publicação.
PASSOS	<ol style="list-style-type: none"> 1. Acessar a página de login de Django Admin: Site: www.lasca.ic.unicamp.br → Login. 2. Informe o login de rede correta 3. Informe a senha correta 4. Selecionar modulo Publications. 5. Seleccionar em adicionar nova publicação por BibTex 6. Ingressar o bibtex
CRITÉRIOS DE ÊXITO	O usuário deve conseguir adicionar uma nova publicação por BibTex ou receber mensagens de erros em caso de introduzir dados inválidos.
DADOS DE ENTRADA E RESULTADOS ESPERADOS	CT031 (BibTex, "BibTex file", valid) CT032 (BibTex, "Other files", invalid) CT033 (BibTex, "", invalid)

4. Execução (quando e como será testado)

A utilização de Selenium WebDriver IDE permitiu desenvolver os testes funcionais. Foi feito a gravação de cada um dos testes, fazendo modificações para testar cada um dos casos de testes. A seguir podemos observar uma imagem do Testes.



Foi adicionado ao sistema cada um dos testes descritos na seção de casos de testes.

5. Análise dos Resultados e próximos passos

Os testes realizados obtiveram sucesso em todos os casos, gerando erros quando o login era inexistente, ou a senha inválida. O módulo de *Publications* apresentou alguns erros, não consegue detectar várias modificações feitas. Resultado dos teste, passou 18/22.

Para solucionar os erros apresentados no módulo de *Publications* utilizaremos testes estruturais e de mutação. Serão desenvolvidas funções em Python.