

Aplicação de Testes Automáticos com Selenium WebDriver: Um Relato de Experiência no SIGAA

Alan Pontes, Anne Caroline Rocha, Raphael Freire

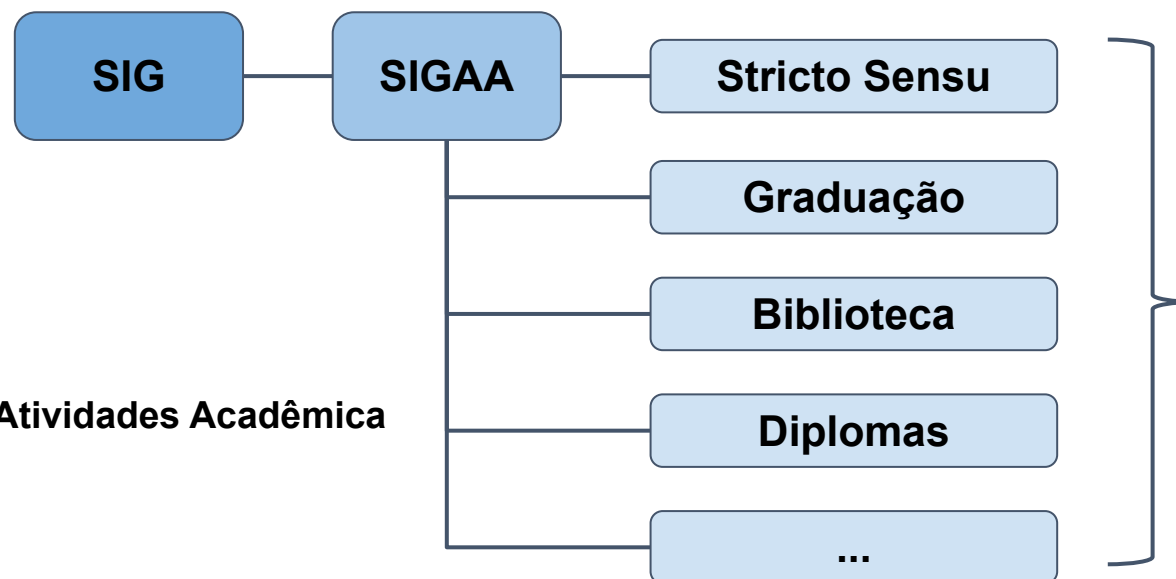
Analistas de Sistemas

Superintendência de Tecnologia da Informação (STI)

Universidade Federal da Paraíba (UFPB)

Introdução

- O SIG possui **vários sistemas**, entre eles o SIGAA, SIGRH e SIPAC.
 - Cada sistema possui diferentes subsistemas (**módulos**) que atendem a áreas específicas da universidade.



Dados e código-fonte
são compartilhados
entre os módulos dos
sistemas.

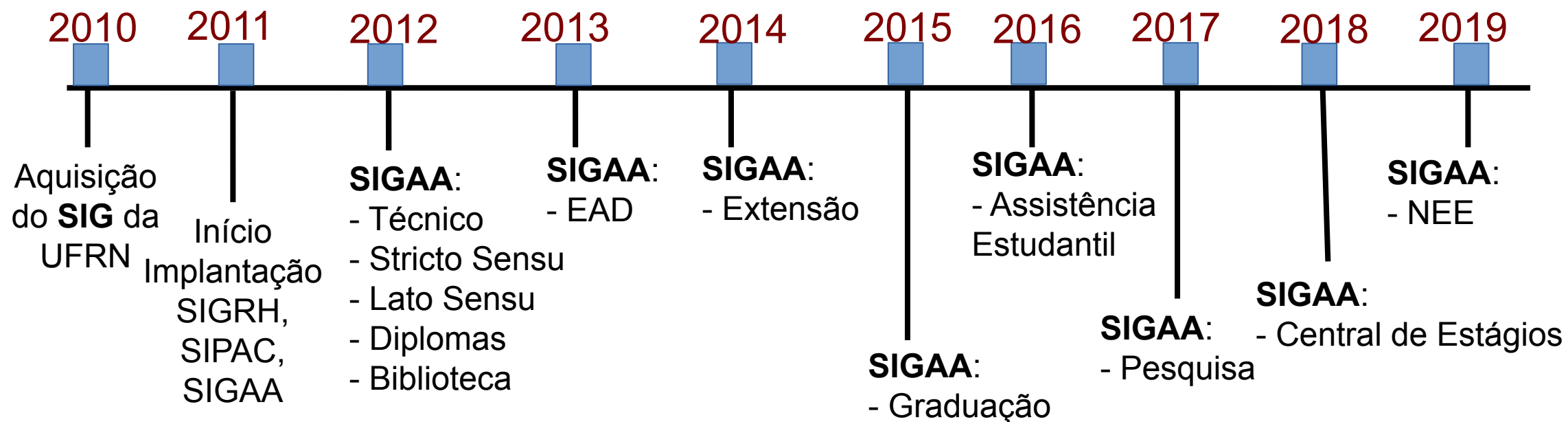
- Durante a manutenção contínua dos Sistemas Integrados de Gestão (SIG), parte dela são **erros** e outra parte são **melhorias**.

Introdução

- Como o SIG é um **sistema de grande porte**, erros podem surgir até mesmo em partes do sistema que não foram modificadas.
- Para manter a qualidade do SIG e garantir que continuará funcionando durante sua evolução, **testes de software** são necessários.
- Testes de software é uma atividade desempenhada durante o desenvolvimento de um sistema, com a finalidade de **identificar erros antes** que seja utilizado pelo usuário final.
- A cada lançamento de versão, não é viável realizar **testes funcionais manuais** em todos os módulos do SIG novamente, que são chamados **testes de regressão**.

Introdução

Cronologia da Implantação de Módulos do SIG pela UFPB



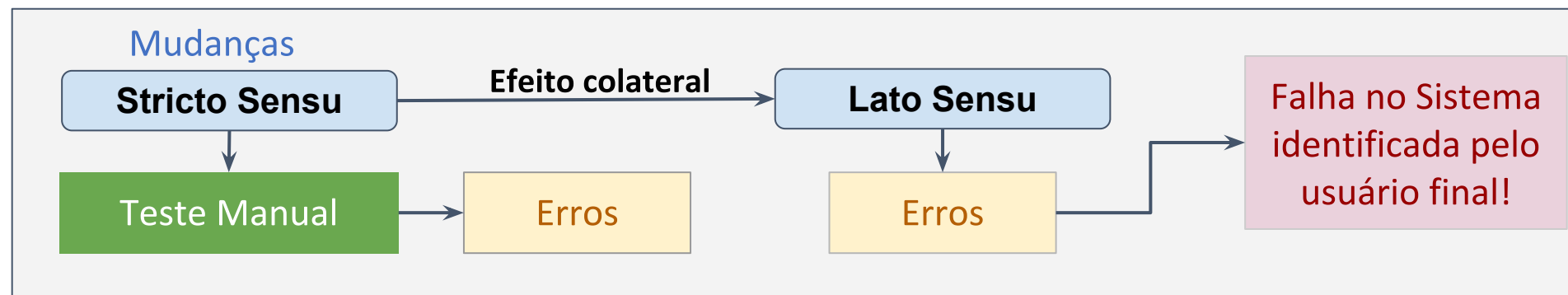
Módulos integrados podem causar falhas entre eles

SIG: Sistemas Integrado de Gestão | SIGAA: Sistema Integrado de Gestão de Atividades Acadêmicas

Introdução

- Na UFPB, ao final de 2012 foram implantados os seguintes módulos do SIGAA:
 - Stricto Sensu, Biblioteca, Lato Sensu, Técnico e Diplomas.

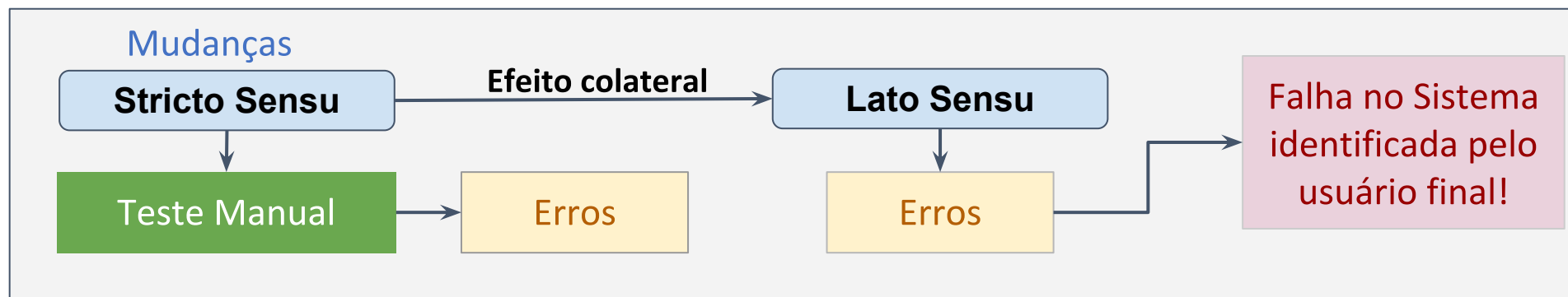
- **Problema:**



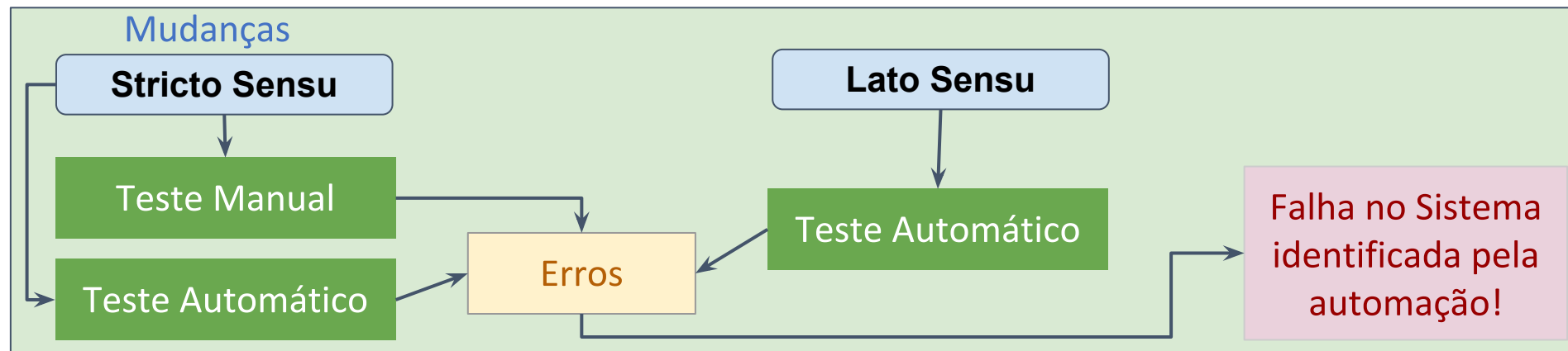
Introdução

- Na UFPB, ao final de 2012 foram implantados os seguintes módulos do SIGAA:
 - Stricto Sensu, Biblioteca, Lato Sensu, Técnico e Diplomas.

- Problema:**



- Solução:**



Métodos

- Em 2013, decidimos realizar **testes funcionais automáticos**, pois simulam a utilização do sistema pelo usuário final, através de um navegador.
- Utilizamos o **Selenium WebDriver para Java**, com isso os testes são rodados no Eclipse IDE através da ferramenta **JUnit**, o qual é integrado com uma biblioteca específica para o Selenium.

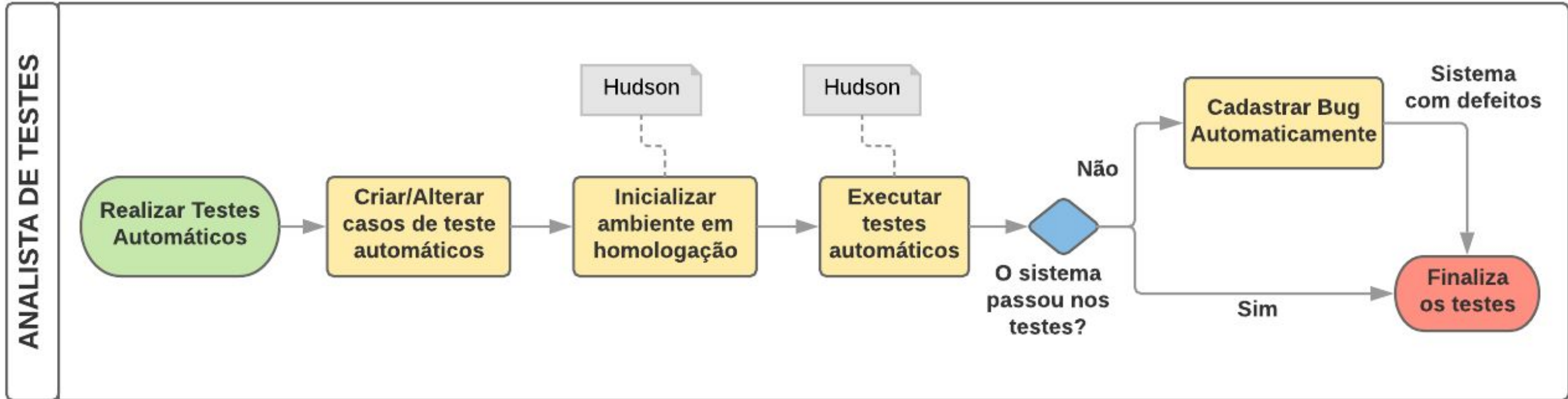


Selenium WebDriver



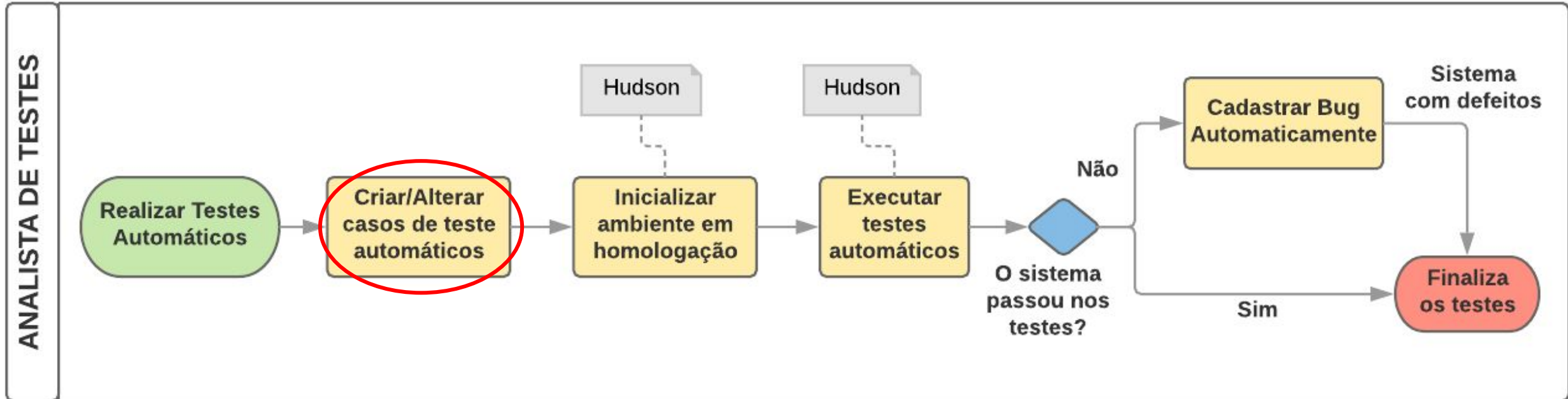
Métodos

Processo durante a execução dos testes automáticos



Métodos

Processo durante a execução dos testes automáticos



Métodos

Descrição dos Casos de Teste no TestLink

TestLink Prague 1.9.5 : caroline [admin] [[Pessoal](#) | [Sair](#)]

[Início](#) | [Requisitos](#) | [Especificação de Testes](#) | [Executar Testes](#) | [Relatórios](#) | [Administração](#) | [Eventos](#) | 003-3-

Projeto de Teste SIGAA-StrictoSensu-Fluxo

Teste Manual

FX008 - Cadastrar Discente Aprovado no Processo Seletivo (26)

- 003-3-146: Verificar cadastro de novo discente regular sem ter sido aprovado em um PS
- 003-3-154: Verificar cadastro de discente REGULAR aprovado no PS com sucesso
- 003-3-162: Verificar cadastro de discente ESPECIAL aprovado no PS com sucesso
- 003-3-147: Verificar que não é possível o cadastro de discente sem ter sido aprovado em um PS
- 003-3-148: Verificar que não é possível o cadastro de discente com o mesmo CPF
- 003-3-149: Verificar que não é possível cadastro de discente com mesmo CPF em cursos diferentes
- 003-3-150: Verificar que não é possível o cadastro de discente com o mesmo CPF já cadastrado
- 003-3-156: Verificar que os dados do candidato APROVADO foram carregados corretamente ao
- 003-3-157: Verificar obrigatoriedade do campo PS no cadastro de discente REGULAR
- 003-3-158: Verificar NÃO obrigatoriedade do campo PS no cadastro de discente ESPECIAL
- 003-3-159: Verificar obrigatoriedade do campo PS no cadastro de novo discente REGULAR
- 003-3-160: Verificar NÃO obrigatoriedade do campo PS no cadastro de novo discente ESPECIAL
- 003-3-161: Verificar cadastro de novo discente regular sem ter participado em um PS
- 003-3-163: Verificar cancelamento com sucesso da operação de cadastrar discente
- 003-3-164: Verificar cancelamento com sucesso da operação de cadastrar NOVO discente
- 003-3-165: Verificar se não há referência para UFRN ao cadastrar discente
- 003-3-167: Verificar se não há problemas de layout ao cadastrar discente
- 003-3-169: Verificar se não há problema nos campos obrigatórios ao cadastrar discente
- 003-3-171: Verificar se não há erros de ortografia ao cadastrar discente
- 003-3-166: Verificar se não há referência para UFRN ao cadastrar NOVO discente
- 003-3-168: Verificar se não há problemas de layout ao cadastrar NOVO discente
- 003-3-170: Verificar se não há problemas nos campos obrigatórios ao cadastrar NOVO discente
- 003-3-172: Verificar se não há erros de ortografia ao cadastrar NOVO discente

Automáticos (3)

- 003-3-1598: Verificar cadastro de discente REGULAR aprovado no PS com sucesso
- 003-3-1600: Verificar que não é possível o cadastro de discente com o mesmo CPF
- 003-3-1602: Verificar cancelamento com sucesso da operação de cadastrar discente

Teste Automático

FX009 - Confirmar Pagamento de Inscrições (22)

FX010 - Oferta de Vagas no Cursos (15)

Caso de Teste

003-3-1598: Verificar cadastro de discente REGULAR aprovado no PS com sucesso

Editar Deletar Mover / Copiar Deletar esta versão Criar uma nova versão Desativar esta versão Adicionar aos Planos de Teste

Exportar Comparar versões Visualizar todo o histórico da execução Visualizar Impressão

Versão 2
Criado em 02/06/2019 21:24:54 por caroline
Última modificação em 02/06/2019 21:25:37 por caroline

Objetivo do Teste:
Perfil: Coordenador Stricto

Pré-condição:

- Acessar o UC - Cadastrar Discente em *Cadastro > Gerenciar Processo Seletivo > Gerenciar Inscrições > Cadastrar Discente*.
- Tentar cadastrar um discente cujo CPF foi aprovado em um Processo Seletivo X.

Pré-condições

#	Ações do Passo	Resultados Esperados:	Execução
1	Passos para pesquisar discente 1. Informar o CPF de um candidato que foi APROVADO no Processo Seletivo X. 2. Preencher todos os campos obrigatórios. 3. Selecionar a opção Tipo = REGULAR. 4. Verificar que o Processo Seletivo X já vem preenchido e é obrigatório. 5. Confirmar a operação.	1. Sistema deverá permitir a inscrição de um candidato aprovado no Processo Seletivo. 2. O sistema gera uma matrícula para o novo discente cadastrado. 3. Sistema exibe uma mensagem de sucesso.	Manual

RN01 - Alunos regulares só podem ser cadastrados por meio de um processo seletivo.

Mensagem de sucesso



Métodos



Código para o Selenium WebDriver para JAVA

```
/**
 * Cadastrar discentes aprovados no Processo Seletivo do Stricto Sensu
 */
public class CadastrarDiscente extends RoteiroSigaa {
    @Test
    public void testCadastrarDiscente() throws Exception {
        try {

            resultado = TestLinkAPIResults.TEST_FAILED;
            //Configurar de acordo com a localização
            LoginSigaa.inicia( baseUrl, driver, PORTAL_COORDENADOR_STRICTO );
            LocalizacaoSigaa.escolherVinculo( driver, VINCULO_SERVIDOR, NOME_VINCULO_SERVIDOR );
            LocalizacaoSigaa.acessarPortal( driver, PORTAL_COORD_STRICTO_SENSU_ID_MODULO,
            PORTAL_COORD_STRICTO_SENSU_NOME_MODULO );
            //Acessa caso de uso pelo menu do SIGAA
            driver.findElement(By.xpath("//a[contains(.,'+Cadastros')]")).click(); //Menu Cadastros
            driver.findElement(By.xpath("//a[contains(.,'+Processos Seletivos')]")).click(); //Menu Procesos Seletivo
            driver.findElement(By.xpath("//a[contains(.,'Gerenciar Processos Seletivos')]")).click(); //Menu Gerenciar PS
            encontrarElemento(driver, By.id("formListaProcessosSeletivos:filtroStatus"), "PUBLICADO");//Seleciona Edital
            driver.findElement(By.xpath("//input[@value='Buscar']")).click(); //Pesquisa edital
            driver.findElement(By.xpath("//tr[td/strong[contains(text(),'João Carlos')]]/a[@title='Gerenciar Inscrições']")).
            click(); //Seleciona discente
            selectByVisibleText("CANDIDATO APROVADO");
            (...)
            assertTrue(isElementPresent(By.xpath("//li[contains(text(),'cadastrado com sucesso')]")));//Esperado msg Sucesso
            driver.findElement(By.xpath("//input[@value='<< Voltar']")).click();
            driver.findElement(By.xpath("//a[@title='Sair do SIGAA']")).click();

            resultado = TestLinkAPIResults.TEST_PASSED;
```

Xpath para encontrar elementos HTML

Assertiva semelhante ao JUnit



Métodos

Código para o Selenium WebDriver para JAVA

```
/**
 * Cadastrar discentes aprovados no Processo Seletivo do Stricto Sensu
 */
```

```
public class CadastrarDiscente extends RoteiroSigaa {
```

```
@Test
```

```
public void testCadastrarDiscente() throws Exception {
    try {
```

```
        resultado = TestLinkAPIResults.TEST_FAILED;
```

```
        //Configurar de acordo com a localização
```

```
        LoginSigaa.inicia( baseUrl, driver, PORTAL_COORDENAD
```

```
        LocalizacaoSigaa.escolherVinculo( driver, VINCULO_S
```

```
        LocalizacaoSigaa.acessarPortal( driver, PORTAL_COOR
```

```
        PORTAL_COORD_STRICTO_SENSU_NOME_MODULO);
```

```
        //Acessa caso de uso pelo menu do SIGAA
```

```
        driver.findElement(By.xpath("//a[contains(.,'+Cada
```

```
        driver.findElement(By.xpath("//a[contains(.,'+Proce
```

```
        driver.findElement(By.xpath("//a[contains(.,'Gerenc
```

```
        encontrarElemento(driver, By.id("formListaProcessos
```

```
        driver.findElement(By.xpath("//input[@value='Busca
```

```
        driver.findElement(By.xpath("//tr[td/strong[contain
```

```
        click(); //Seleciona discente
```

```
        selectByVisibleText("CANDIDATO APROVADO");
```

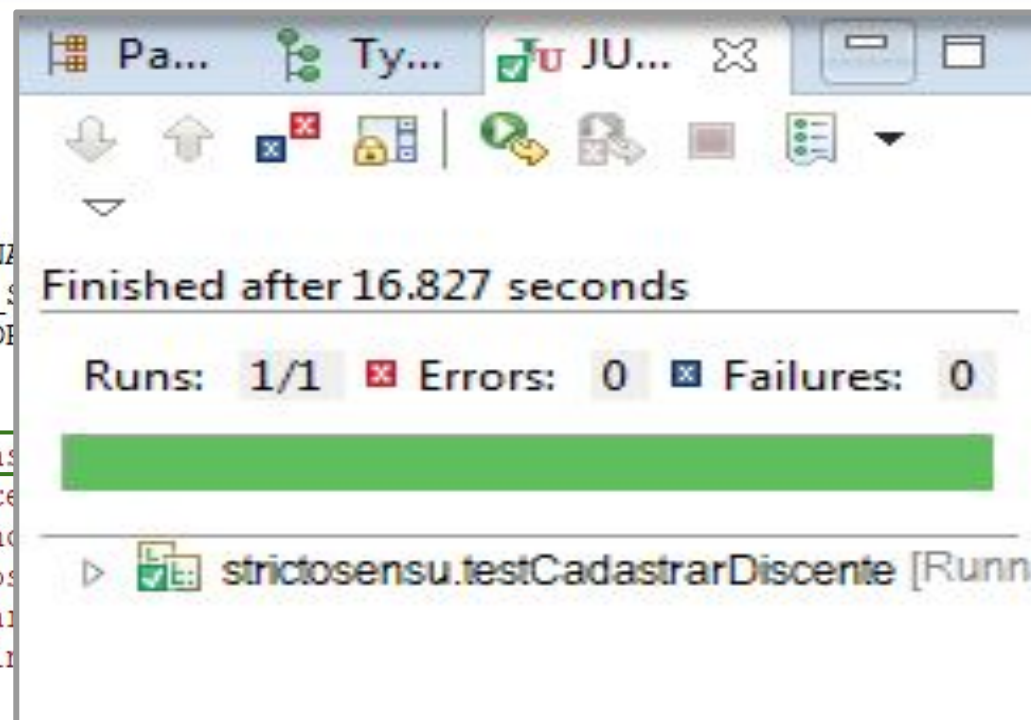
```
        (...)
```

```
        assertTrue(isElementPresent(By.xpath("//li[contains(text(),'cadastrado com sucesso')]"))); //Esperado msg Sucesso
```

```
        driver.findElement(By.xpath("//input[@value='<< Voltar']")).click();
```

```
        driver.findElement(By.xpath("//a[@title='Sair do SIGAA']")).click();
```

```
        resultado = TestLinkAPIResults.TEST_PASSED;
```

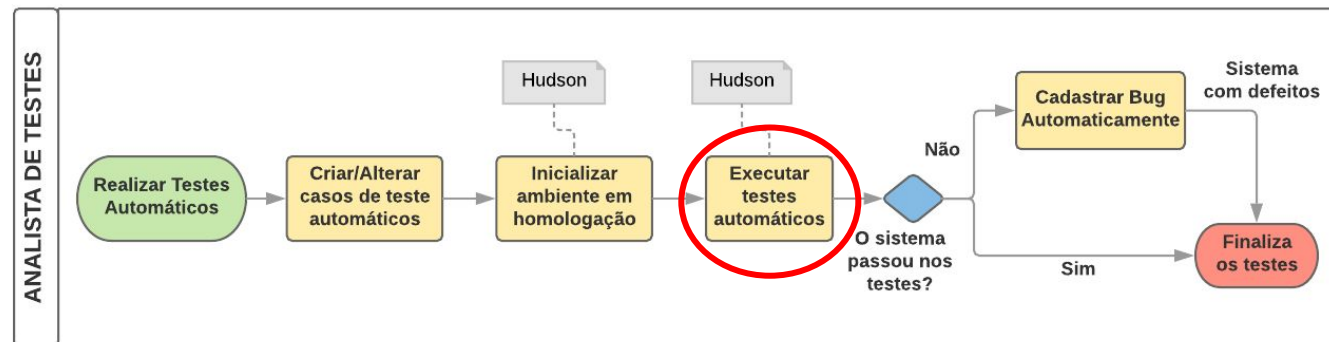


Assertiva semelhante ao JUnit

Métodos

Papel: Analista de Testes

Processo: Executar testes automáticos



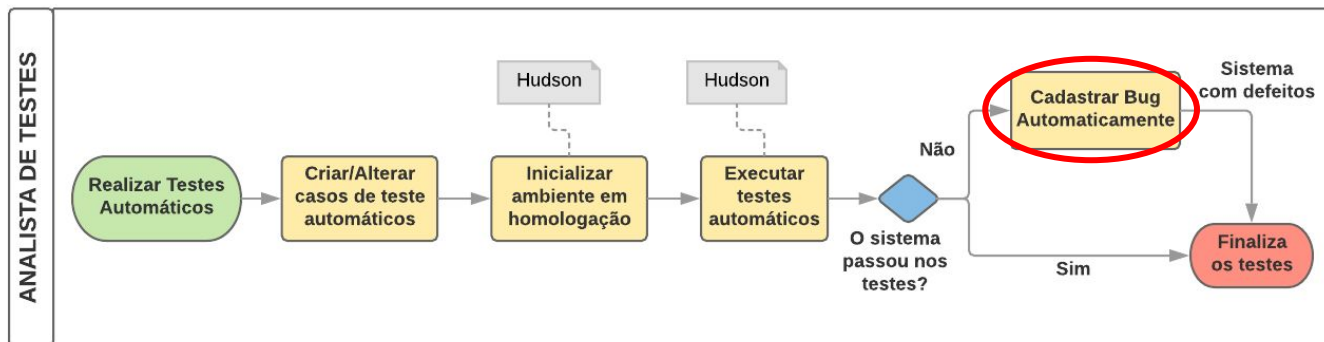
- A ferramenta de integração contínua utilizada para esta execução era o **Hudson**.
- Quando o sistema entra para o ambiente de homologação, os testes automáticos são executados.
- Em Maio/2019 alteramos para o **GitLab CI/CD**.



Métodos

Papel: Analista de Testes

Processo: Cadastrar Bug Automaticamente



Bug #25997

✎ Editar ⌚ Tempo de trabalho ★ Observar 📄 Copiar 🗑 Excluir



SIGAA - Stricto Sensu - Fluxo #22943: FX - Cadastrar Discente Aprovado no Processo Seletivo
 SIGAA - Stricto Sensu - Correção #23449: Correção - FX - Cadastrar Discente Aprovado no Processo Seletivo

`class sigaa.strictosensu.testCadastrarDiscente`

Adicionado por Redmine Admin 7 meses atrás. Atualizado menos de um minuto atrás.

Situação:	Rejeitada	% Terminado:	100%
Prioridade:	Imediata	Tempo estimado:	1.00 h
Equipe do bug:	Equipe de Testes		
Tipo do bug:	Bug de Automação		

Descrição

Não encontrou o campo CPF

💬 Responder

Unable to locate element: #form\:txtCPF

For documentation on this error, please visit: http://seleniumhq.org/exceptions/no_such_element.html

Build info: version: '3.11.0', revision: 'e59cfb3', time: '2018-03-11T20:33:15.31Z'

System info: host: 'eris.nti.ufpb.br', ip: '150.165.250.7', os.name: 'Linux', os.arch: 'i386', os.version: '3.2.0-4-686-pae', java.version: '1.8.0_162'

Driver info: org.openqa.selenium.firefox.FirefoxDriver

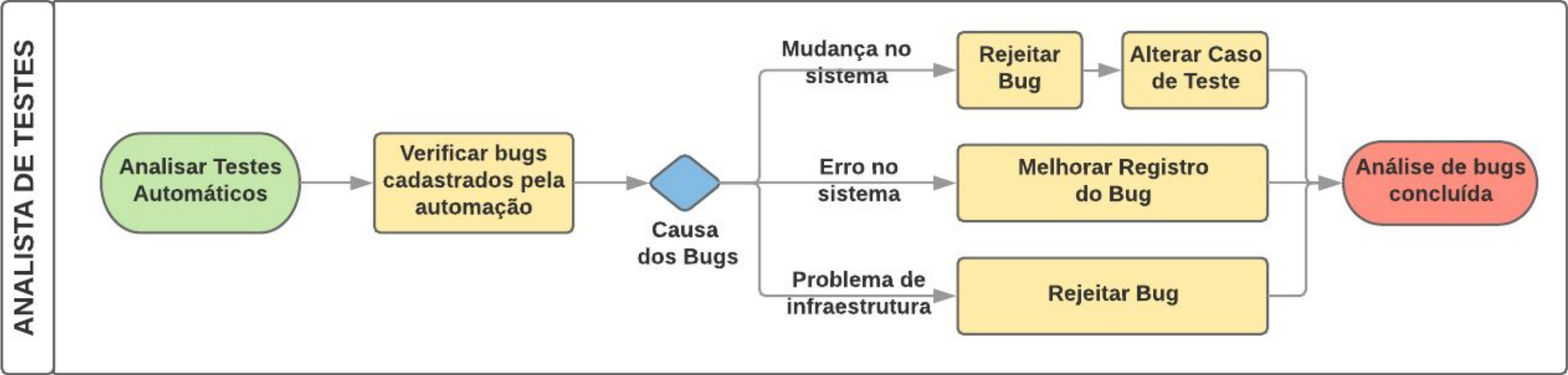
Capabilities {acceptInsecureCerts: true, browserName: firefox, browserVersion: 54.0.1, javascriptEnabled: true, moz:accessibilityChecks: false, moz:processID: 27603, moz:profile: /tmp/rust_mozprofile.VnR7kR..., pageLoadStrategy: normal, platform: LINUX, platformName: LINUX, platformVersion: 3.2.0-4-686-pae, rotatable: false, specificationLevel: 0, timeouts: {implicit: 0, pageLoad: 300000, script: 30000}}

Session ID: 76247395-689b-4891-9a71-0d8c430a2a30



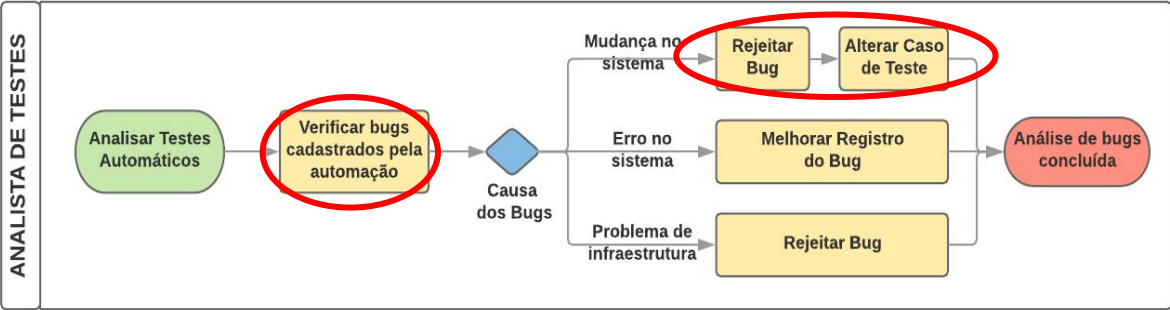
Métodos

Processo após execução dos testes automáticos



Métodos

Papel: Analista de Testes
Processo: Cadastrar Bug Automaticamente



Bug #25997

Editar Tempo de trabalho



SIGAA - Stricto Sensu - Fluxo #22943: FX - Cadastrar Discente Aprovado no Processo Seletivo
SIGAA - Stricto Sensu - Correção #23449: Correção - FX - Cadastrar Discente Aprovado no Processo Seletivo
class sigaa.strictosensu.testCadastrarDiscente

Adicionado por Redmine Admin 7 meses atrás. Atualizado menos de um minuto atrás.

Situação: Rejeitada

Prioridade: Imediata

% Terminado: 100%

Tempo estimado: 1.00 h

 Atualizado por Alan há 7 meses

#4

- Situação alterado de Revisando para Rejeitada
- % Terminado alterado de 80 para 100

Alteracao no sistema que teve como objetivo nao permitir alteracao do cpf do discente para evitar vinculacao a outro idpessoa. Porem, pode-se editar normalmente os outros dados do discente, como por exemplo nome deste!

Compartilhar Editar Excluir



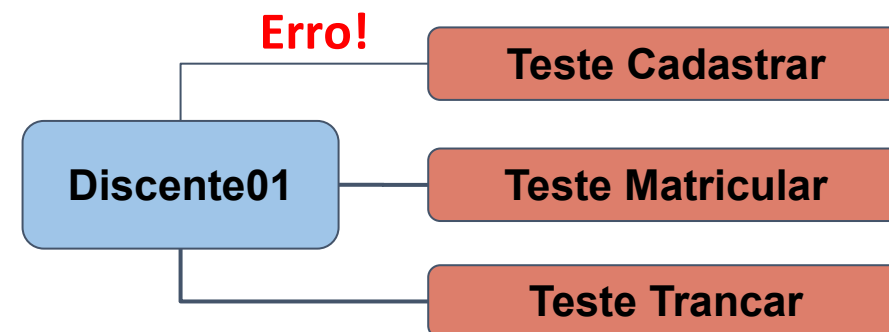
Resultados

Dados criados para realizar os testes:

- Sobre a estrutura dos Scripts de Teste, por ser um sistema de grande porte, foi necessário criar dados de usuários fictícios, que eram gravados no banco de dados e limpos ao final do teste.

Abordagem anterior:

- Dados utilizados para um teste eram aproveitados nos testes seguintes.
 - **Todos os testes seguintes ao erro não eram executados!**



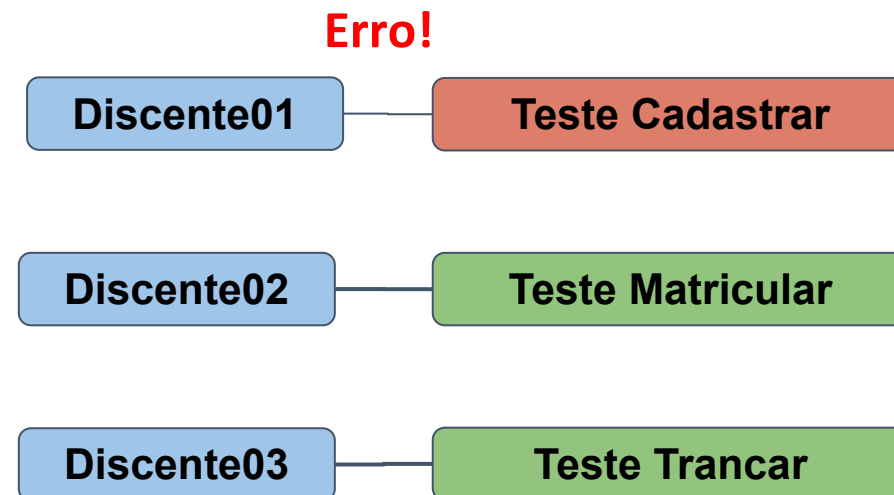
Resultados

Dados criados para realizar os testes:

- Sobre a estrutura dos Scripts de Teste, por ser um sistema de grande porte, foi necessário criar dados de usuários fictícios, que eram gravados no banco de dados e limpos ao final do teste.

A abordagem atual:

- Dados são utilizados para um conjunto pequeno de testes.
 - Os testes seguintes ao erro, podem ser executados.



Resultados

Esforço na manutenção dos scripts de teste:

- Decidimos criar testes mais simples:
 - Que realizam apenas uma função.
 - Que testam apenas o fluxo principal do caso de uso.
- Os **testes mais completos**, são realizados pela equipe de **testes manuais**.
- Os **testes automáticos** garantem apenas que as **operações básicas** do sistema continuam funcionando.

Resultados

Limitações da ferramenta

- Durante esses 6 anos realizando testes automáticos funcionais com a ferramenta Selenium WebDriver, em 2018 foi necessário atualizar para o **Selenium 3.0**.
 - Os scripts de teste **pararam de funcionar**, foi necessária atualização em todas as classes de teste.
- O **Selenium** requer uma dependência com a versão do Firefox, desta forma, incluímos um Firefox dentro do projeto de teste, o qual é executado, independente do Firefox instalado na máquina do servidor.

Resultados

Quantidade de testes automáticos criados: 240 casos de teste

- Há testes automáticos para os módulos do SIGAA:
 - Stricto Sensu
 - Diplomas
 - Graduação
 - Técnico
- Estes módulos foram escolhidos porque:
 - Possuem um maior número de usuários.
 - São módulos com forte compartilhamento de código-fonte.
- O tempo para executar todos os testes para o SIGAA dura **em média 2h30min.**

Resultados

Quantidade de bugs reportados de Fevereiro/2018 a Fevereiro/2019

Cadastro da Automação	Quantidade
Bugs	7
Falso Positivo (Infraestrutura)	13
Falso Positivo (Alteração de Sistema)	8
TOTAL	28

- **Infraestrutura:** sistema não estava disponível no momento em que os testes foram executados.
- **Alteração de sistema:** houve mudanças na interface do sistema, com isso o teste esperava um elemento da página, que não existia mais.

Conclusão

- Os testes automáticos têm sido relevantes na detecção de erros causados por **efeito colateral** entre módulos do SIGAA, mas principalmente na garantia de que as funcionalidades básicas continuam funcionando.
- Mesmo os erros rejeitados, eles são úteis para **rastrear as mudanças** no sistema.
- Com os testes automáticos, a equipe de teste pôde se dedicar mais aos **testes manuais para novas funcionalidades**.
- Com a automação, é necessário que **1 membro da equipe** esteja dedicado à manutenção dos scripts de testes e análise dos erros reportados pela ferramenta.

Conclusão

- A cada atualização de **ferramentas de infraestrutura**, são necessárias novas configurações nos ambientes de teste automático.
 - Por exemplo.: mudança do Hudson para o **GitLab CI/CD**
- Quanto mais o sistema muda, mais é necessário a **atualização dos scripts** de teste.
 - Por exemplo: Atualização da interface para atender aos requisitos de **Acessibilidade** em 2018.
- Desta forma, por não ter mudanças frequentes nos módulos implantados, os testes automáticos têm sido viáveis.

Referências

BSTQB. (2018) “Certified Tester, Foundation Level Syllabus”, Versão 2018 BR. Disponível em: https://www.bstqb.org.br/uploads/syllabus/syllabus_ctfl_2018br.pdf

Acesso em: Abril/2019

Hudson-ci. (2019) Disponível em: <http://wiki.eclipse.org/Hudson-ci>

Acesso em: 26 fevereiro 2019.

Junit. (2019) Disponível em: <https://junit.org/junit4/> Acesso em: Fevereiro/2019.

SeleniumHQ. (2019) Disponível em: <https://www.seleniumhq.org/docs>

Acesso em: Fevereiro/2019.

UFRN. (2006) “Sistemas Institucionais Integrados de Gestão-SIG”, Disponível em: <https://docs.info.ufrn.br> /, Acesso em: Fevereiro/2019.

Obrigada!

Contato: caroline@sti.ufpb.br