

# RELATÓRIO TÉCNICO

**Analista Responsável:** Everton da Silva Candido

**Projeto:** Sistema de Teste de QA da Tinnova

**Data:** 20/01/2023

## 1. Funcionamento do Sistema:

O sistema foi projetado para testar analistas de qualidades, que pretendem atuar dentro da Tinnova. O mesmo é bem simples, sendo um sistema de gestão de usuários (com opções de cadastro, visualização, edição e deleção).

## 2. Análise do Sistema a partir de Teste Exploratório:

- O sistema demonstrou falhas de segurança já em seu acesso, pois o mesmo não continha uma ferramenta de login/senha, apenas um botão de acesso que já introduz para o usuário informações gerais de todos os cadastrados.
- Devido o sistema ser simples e não possuir um sistema de permissões, qualquer usuário pode apagar, editar ou criar novos cadastros.
- O sistema não faz nenhum tipo de checagem dos dados que estão sendo inseridos, apenas se existe campos em branco. Dessa forma, caracteres especiais e não associados ao atributo em questão podem acabar gerando falhas no sistema. Foi observado que o uso de caracteres não associados a um atributo gera erro no sistema, como por exemplo, utilização de letras no campo CPF. O erro observado é que o sistema permite a gravação do registro, porém sem de fato salvar aquele campo, gerando assim registros incompletos e impossíveis de editar.
- O sistema não possui máscara em seus campos de registro, fazendo com que o usuário tenha dúvidas em relação à formatação e tipo de caracteres a ser utilizada em cada um.
- Foi observado também que as alterações nos registros não estão sendo repassadas para o banco de dados. Apesar de ser possível realizar deleções, edições e inserções as mesmas não estão sendo refletidas para o banco de dados e sim sendo salvas no cachê de navegação, causando a impressão de que todas as funcionalidades estão operacionais.
- Os campos na tela de cadastro e edição não possuem um id associado, essa é uma prática que facilita a manutenção do código e seria valido a sua inserção.

### 3. Metodologia:

**Observação:** Antes de fornecer informações referentes à metodologia utilizada, gostaria de informar que foi utilizado os comandos `visit` e `sleep` nas classes de `step` devido a questões do ambiente de desenvolvimento do qual estava utilizando, pois, as variáveis setadas na classe `env.rb` não surtiram o efeito conforme o esperado. Dessa forma, mesmo sabendo que os comandos não são considerados boas práticas de programação, os mesmos apresentaram o resultado esperado nos parâmetros de teste e foram utilizados com o intuito de justamente cumprirem com o objetivo final que no caso é a validação dos cenários de teste.

O projeto de automação dos testes foi desenvolvido utilizando a linguagem **Ruby** com o uso dos seguintes frameworks (gems) instalados:

- Cucumber
- Capybara
- Selenium Web Driver
- RSpec
- Site Prism

Além disso, o sistema operacional e a IDE utilizado foram o Windows 10 e Visual Studio Code, respectivamente.

Para a automação dos testes foi utilizada a técnica de aplicar tags em cada cenário para que cada um pudesse ser testado separadamente e isoladamente, além de facilitar a visualização do que será testado. Dessa forma, segue abaixo as tags de cada cenário (descritas também dentro do projeto):

- @introducao -> Cenário: Entrar com Sucesso
- @home\_adicionar -> Cenário: Acessar a Página de Cadastro
- @home\_sair -> Cenário: Sair do Sistema
- @home\_visualizar -> Cenário: Visualizar Usuário
- @home\_editar -> Cenário: Editar Usuário
- @home\_deletar\_sim -> Cenário: Deletar Usuário
- @home\_deletar\_nao -> Cenário: Cancelar a Deleção do Usuário
- @visualizar -> Cenário: Voltar para a Home do Sistema após a Visualização de Registro
- @editar\_valido -> Cenário: Editar Registro com Credenciais Validas
- @editar\_invalido -> Cenário: Editar Registro com Credenciais Invalidas
- @editar\_incompleto -> Cenário: Editar Registro com Credenciais Incompletas
- @registrar\_usuario\_valido -> Cenário: Cadastrar Registro com Credenciais Validas

- @registrar\_usuario\_invalido -> Cenário: Cadastrar Registro com Credenciais Invalidas
- @registrar\_usuario\_incompleto -> Cenário: Cadastrar Registro com Credenciais Incompletas

Para a execução de cada cenário individualmente, é necessário utilizar o comando:

Cucumber -t@tag\_desejada

#### 4. Resultados dos Cenários Testados:

Após análise do sistema, foi feito um projeto de automação dos cenários observados (com base em suas funcionalidades) e também pontuado cenários que em tese deveriam funcionar corretamente, mas que por conta de falhas e brechas no sistema acabam executando com erros. Os cenários mapeados que apresentaram erro, ou seja, que demonstram de fato falhas no sistema de acordo com o resultado esperado foram:

- Cenário: Cadastrar Registro com Credenciais Invalidas
- Cenário: Cadastrar Registro com Credenciais Incompletas
- Cenário: Editar Registro com Credenciais Invalidas
- Cenário: Editar Registro com Credenciais Incompletas

Para os demais cenários, os testes apresentaram o resultado esperado (as descrições dos cenários, assim como todas as demais informações sobre os mesmos foram realizadas no modelo BDD e estão disponibilizadas na pasta com o mesmo nome dentro do projeto).

Dessa forma, reforço a necessidade de correção do sistema, pois, o mesmo está apresentando brechas e falhas em seu funcionamento, o que ficou comprovado através da execução dos testes.

Para maior detalhamento dos testes executados foi disponibilizado um relatório com a execução dos mesmos (reforço novamente que os cenários que apresentaram erro são devido a falhas presentes no sistema e rastreadas pelos testes automatizados e também pelos testes manuais).

**IMPORTANTE:** para os cenários de edição e deleção não foram considerados os registros de testes como massa importante dentro do sistema. Isso porque se entende que a massa de cadastro disponibilizada e gerada era para testes em um ambiente não produtivo e, portanto, não possuía importância em sua manutenção, sendo ela apenas para o uso e teste das ferramentas do sistema em si.