

Validação Funcional do Sistema Relax Pig por meio de Casos de Teste Elaborados com Base na Norma IEEE 829 e Implementados no Framework Selenium IDE

Ester Oribes¹, Jiani Roza¹, Cristhiano Vasconcellos¹, Carlos Santos¹

¹Instituto Federal de Educação, Ciência e Tecnologia Farroupilha (IFFAR) –
Alegrete, RS – Brasil

Email: ester.2017009874@aluno.iffar.edu.br,

{jiani.roza, cristhiano.vasconcellos, carlos.santos}@iffarroupilha.edu.br

Resumo. *Este trabalho tem como objetivo apresentar um estudo de caso sobre a validação funcional do sistema Relax Pig utilizando casos de teste baseados na Norma IEEE 829 e implementados com o framework Selenium IDE. Foram desenvolvidos e executados dez casos de teste, permitindo a identificação, documentação e correção de erros encontrados durante a fase de testes do sistema. Destaca-se o propósito de demonstrar a aplicação automatizada de testes funcionais por meio do Selenium IDE. Os testes foram realizados no sistema Relax Pig, desenvolvido para o setor de suinocultura do Instituto Federal Farroupilha Câmpus Alegrete. O foco principal deste trabalho é apresentar o processo de desenvolvimento e aplicação de testes funcionais, com o intuito de aprimorar a eficiência do processo de teste de software, assim como a qualidade e a confiabilidade do sistema investigado. Para tanto, foram desenvolvidos e executados dez casos de teste, permitindo a identificação, documentação e correção de erros encontrados durante a etapa de testes do sistema.*

1. Introdução

O processo de desenvolvimento de software envolve uma série de atividades distintas, cada qual com suas próprias metodologias e objetivos. Essas atividades são organizadas em etapas, como a definição de requisitos, análise, projeto e desenvolvimento. No entanto, existe uma etapa crucial nesse processo: o Teste de *Software*, que desempenha um papel fundamental na busca pela qualidade do produto final. O objetivo desta etapa é identificar possíveis erros no *software* em construção ou já concluído, permitindo a correção dessas falhas e aumentando a qualidade do produto final [Pereira et al. 2013].

Os testes podem ser derivados de requisitos, especificações, artefatos do projeto, ou códigos-fonte. Cada atividade de desenvolvimento é acompanhada por um nível diferente de teste. Dentre as várias técnicas, o Teste Funcional se destaca ao verificar as funções do sistema e testar os requisitos funcionais implementados [De Souza 2013].

Entretanto, realizar teste de software pode ser uma tarefa complexa e exigir muitos recursos e tempo. A fim de facilitar esta atividade foram desenvolvidas ferramentas que possibilitam a automação dos testes, como exemplo, a ferramenta de código aberto Selenium, amplamente citada na literatura para realização de testes automatizados de Sistemas *Web* [Santori 2019].

Este trabalho tem como principal objetivo realizar a validação funcional do sistema Relax Pig [Motta et al. 2024] por meio da aplicação de casos de teste construídos

com base na Norma IEEE 829 [IEEE 1998, IEEE 2008], e implementados no *framework* Selenium IDE. O sistema foi desenvolvido para auxiliar nas pesquisas relacionadas com o monitoramento do estresse suíno do Instituto Federal Farroupilha Câmpus Alegrete.

Por meio da realização dos casos de testes foi possível encontrar e corrigir erros antes da colocação do *software* em produção. Além disso, foi possível validar as funcionalidades principais do sistema a fim de verificar se estes foram devidamente implementados, garantindo, por conseguinte, a maturidade e qualidade do sistema.

2. Revisão Teórica

O teste de software é um componente essencial para avaliar a qualidade e o desempenho de sistemas e aplicativos. Sua importância reside na detecção de defeitos antes da implantação, evitando problemas graves e dispendiosos [Silva 2019]. Além disso, o teste assegura que o *software* atenda aos requisitos e padrões de qualidade, como usabilidade, desempenho e segurança. Investir em testes desde o início economiza tempo e dinheiro, evitando retrabalhos e melhorando a experiência do usuário por meio da correção de problemas de usabilidade [Júnior et al. 2017].

Um dos aspectos fundamentais dos testes de *software* são as técnicas utilizadas, as quais desempenham um papel essencial na definição e elaboração dos testes. Essas técnicas fornecem orientações para a criação dos casos de teste e ajudam a determinar o planejamento adequado. Uma abordagem comum é a realização de testes funcionais onde é levando em consideração apenas os conjuntos de entrada e saída esperados do sistema. Nessa técnica, o engenheiro de *software* não tem conhecimento da estrutura interna do sistema e também conhecidos como testes de caixa-preta [Silva 2019].

O objetivo deste tipo de teste é validar o aspecto funcional do *software*, levando em conta os requisitos funcionais definidos para o sistema. Durante a execução dos testes funcionais, o engenheiro de *software* foca nos casos de teste que validam as diferentes funcionalidades do sistema. Este processo envolve a identificação dos requisitos funcionais definidos para o *software* e a criação de casos de teste que verifiquem se esses requisitos estão sendo atendidos. Os testes funcionais podem ser realizados manualmente ou com o auxílio de ferramentas de automação, dependendo da complexidade do software e dos recursos disponíveis [Valente 2020].

Em seguida, após a aplicação dos testes funcionais, ocorre a verificação dos resultados obtidos, comparando-os com os critérios de aceitação estabelecidos no plano de teste. Se todos os critérios forem atendidos e os resultados estiverem conforme o esperado, é possível prosseguir para o próximo requisito funcional. No entanto, se algum teste não for aprovado, é necessário realizar a reimplementação ou a refatoração do código correspondente ao requisito em questão.

Esta etapa envolve a revisão do código implementado, a identificação e correção dos problemas encontrados. Após essas etapas, o processo de teste é repetido para o requisito funcional corrigido [Sommerville 2019]. Após a reimplementação ou refatoração é fundamental verificar se as modificações realizadas atenderam aos critérios de aceitação estabelecidos. Esta etapa é crucial para garantir a qualidade e o desempenho do sistema.

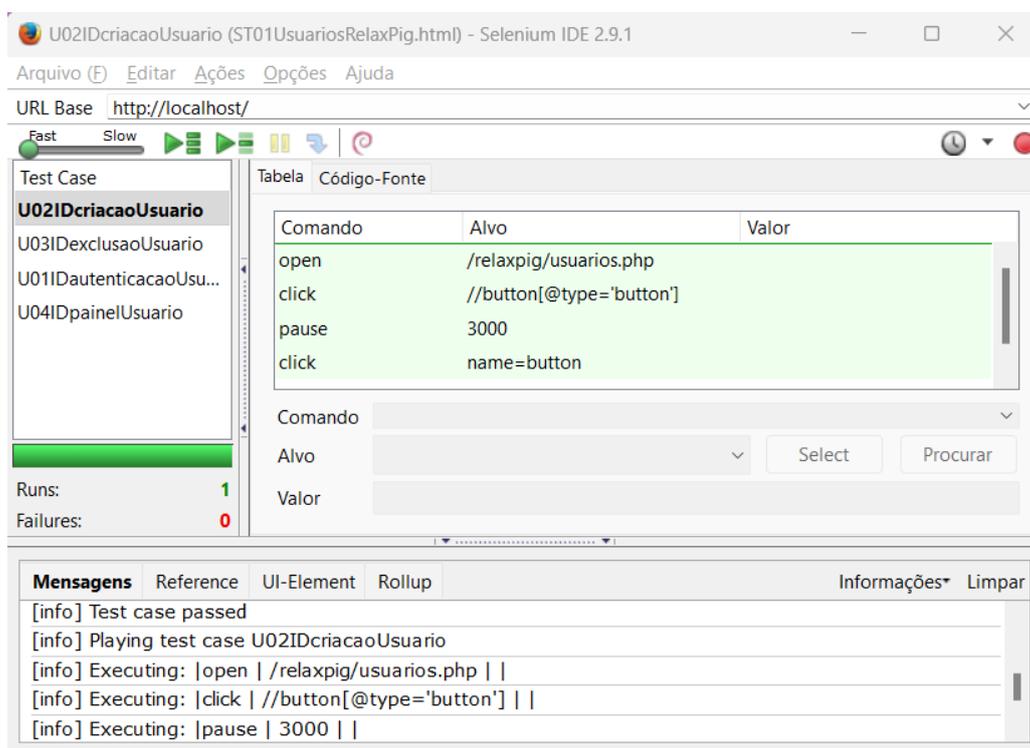
Quanto ao *framework* Selenium, trata-se de um *software* amplamente mencionado na literatura para automação de testes de sistema *Web*, sendo uma ferramenta al-

tamente flexível, que pode ser utilizada em diferentes navegadores, sistemas operacionais e linguagens de programação, além de ser compatível com vários *frameworks* de teste [Vieira 2021].

Assim, uma das principais vantagens do Selenium é sua capacidade de testar aplicações *Web* independentemente do navegador ou sistema operacional utilizado. O Selenium também permite criar e executar *scripts* de teste, além de fornecer suporte a várias linguagens de programação, como Java, C++, Python, PHP e Ruby [Peres 2016]. Além disso, fornece uma variedade de *plugins* e extensões que possibilitam a criação de *scripts* de teste robustos, adaptados às necessidades específicas de cada sistema. Esses recursos adicionais expandem as funcionalidades básicas da ferramenta, permitindo uma maior personalização dos testes [Santori 2019].

Por fim, a integração do Selenium com projetos de teste existentes é simples. Os *scripts* de teste podem ser facilmente incorporados e gerenciados dentro destes projetos, proporcionando fácil integração. A interface do Selenium IDE é simples e intuitiva [Pereira et al. 2013], conforme ilustrado na Figura 1, que apresenta casos de testes implementados para avaliação do sistema Relax Pig.

Figura 1. Interface do Framework Selenium IDE e Casos de Teste aplicados no Sistema Relax Pig. Fonte: Autoria Própria.



A Figura 1 retrata a utilização do *framework* Selenium IDE executando a *suíte* de casos de testes desenvolvidos para a validação do painel de usuário, autenticação do usuário, criação do usuário e exclusão do usuário do sistema Relax Pig.

3. Trabalhos Relacionados

A seguir são apresentados alguns trabalhos desenvolvidos cuja automatização de testes foi realizada por meio do *framework* Selenium.

3.1. Avaliação da ferramenta de testes Selenium no desenvolvimento guiado por teste de uma aplicação Web

[Santori 2019] apresenta a avaliação das vantagens e desvantagens de uso da ferramenta Selenium na automação de testes funcionais durante o desenvolvimento de uma aplicação Web chamada de Fantasy Game. A partir deste estudo foi criado um arcabouço para avaliação da ferramenta. Os critérios selecionados para avaliação foram respondidos de forma subjetiva para descrever as características e funcionalidades do Selenium, demonstrando que a ferramenta dispõe de uma variedade de recursos para a automação de testes, permitindo que seja possível a implementação de testes complexos de forma simplificada.

3.2. Qualidade de software proposta de automação de testes de um processo ágil para uma empresa de software

Já o trabalho proposto por [Braga 2019] aborda uma proposta para a automação de testes e a adoção de uma metodologia ágil para melhorar a qualidade de uma empresa de fábrica de *software*. O trabalho apresentou uma pesquisa bibliográfica sobre engenharia de software, metodologias ágeis, Scrum, Kanban, desenvolvimento orientado por comportamento e testes de *software*. Além disso, foram apresentadas soluções para automação de testes, dentre as quais a ferramenta Selenium. A partir do estudo foi apresentado uma nova proposta de processo de desenvolvimento de *software* e automação de testes.

3.3. Desenvolvimento de testes automatizados e testes manuais para um sistema Web: uma análise comparativa

O trabalho de [Silva 2021] apresenta o estudo de testes automatizados e manuais em sistema Web. Após a realização dos testes automatizados e manuais os autores verificaram a necessidade de realizar uma análise comparativa entre as abordagens e verificar qual delas agregaria mais benefícios ao produto testado. Para a realização da análise foi realizada uma pesquisa em repositórios *online* juntamente com a realização e comparação dos testes realizados no sistema Web investigado. Após o recolhimento e análise dos dados foi possível constatar a necessidade do uso de ferramentas para a automatização dos testes.

Com base na investigação e a análise dos trabalhos relacionados evidenciou-se a importância da aplicação de testes automatizados e as principais ferramentas para testes funcionais disponíveis na literatura. Desse modo, este estudo pretende apresentar os testes automatizados aplicados com o *framework* Selenium a fim de avaliar as principais funcionalidades do sistema Relax Pig. Na próxima Seção é apresentada a metodologia utilizada.

4. Metodologia

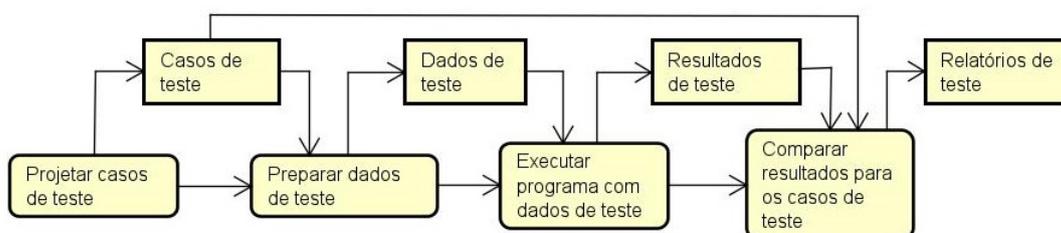
Neste trabalho, foi realizado um estudo de caso e validação funcional do sistema Relax Pig seguindo as diretrizes da Norma IEEE 829 [IEEE 1998, IEEE 2008]. Essa norma estabelece um padrão para a documentação dos processos de teste de software, fornecendo orientações precisas para a criação de documentos essenciais, como plano de teste, casos de teste, relatório de incidentes e resumo de teste. A revisão da Norma 829-2008 [IEEE 2008] aprimora e atualiza as práticas recomendadas para essa documentação, as quais foram implementadas no *Framework* Selenium IDE. Com base na análise dos resultados obtidos com os casos de teste busca-se melhorar a qualidade do sistema.

Além disso, a elaboração dos casos de teste propostos neste trabalho seguiram os padrões de teste de *software* definidos na ISO/IEC/IEEE 29119-1 [IEEE 2013], que visa apresentar um conjunto de padrões para teste de *software* acordados internacionalmente, apresentando conceitos e vocabulário sobre o tema.

A Figura 2 apresenta a metodologia na qual este estudo baseou-se para realizar os testes de *software*. Esta metodologia foi adaptada de [Sommerville 2019], sendo composta basicamente por fases, como planejamento, projeto dos casos de teste, execução e análise dos resultados.

O processo de realização dos testes é iterativo, repetindo-se para cada novo requisito funcional identificado no sistema analisado. Essa abordagem sistemática garante que cada parte do sistema seja testada e validada antes de avançar para a próxima fase de desenvolvimento, garantindo assim a qualidade e confiabilidade do *software* em evolução.

Figura 2. Modelo de Processo de Teste de Software. Fonte: Adaptado de [Sommerville 2019].



A validação e o fluxo de desenvolvimento seguem um ciclo iterativo de execução de testes, identificação de problemas, correção e validação. Este processo visa garantir que as funcionalidades implementadas atendam as necessidades dos usuários e que o sistema final seja confiável e de qualidade.

O Levantamento de Casos de Testes se refere à definição dos testes a serem automatizados e aplicados no sistema Relax Pig. Este processo ocorreu por meio da análise do documento de requisitos e diagrama de casos de uso do sistema investigado [Motta et al. 2024]. A etapa de Aplicação dos Testes consiste em apresentar a automação dos testes realizada com o auxílio do *framework* Selenium.

A última etapa da metodologia consistiu na Análise dos Resultados obtidos com a aplicação dos testes automatizados, visando verificar possíveis erros e melhorar a confiabilidade e qualidade do sistema Relax Pig.

5. Experimentos e Resultados

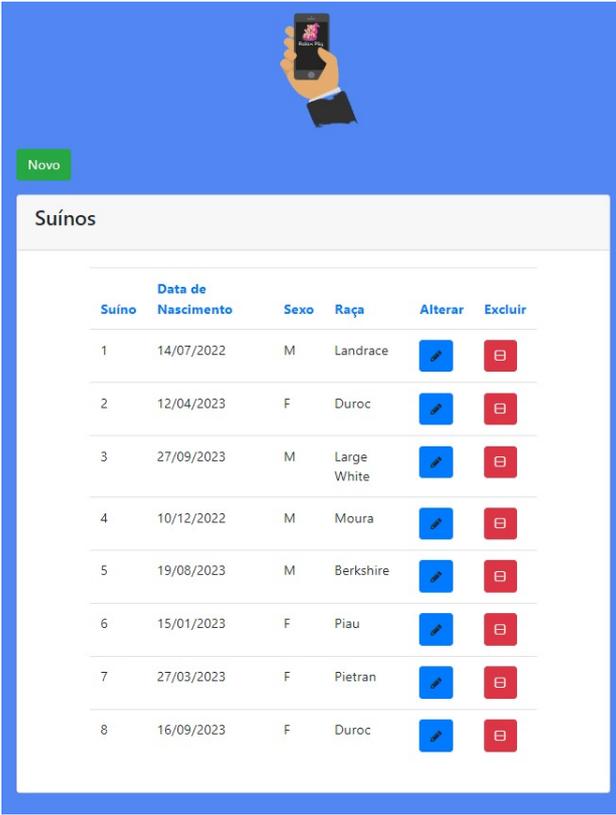
Nesta Seção são apresentadas as principais funcionalidades do sistema Relax Pig e os detalhes da automação dos testes de *software*. Também são apresentadas as especificações dos casos de testes e a aplicação dos testes realizados juntamente com os resultados obtidos.

5.1. Descrição do Sistema

No Instituto Federal Farroupilha Câmpus Alegrete (IFFAR-CA) a suinocultura é realizada no sistema intensivo, utilizando baias e gaiolas para o manejo dos suínos. No entanto, essa prática pode levar ao desenvolvimento da Síndrome do Estresse Suíno (em inglês *Porcine Stress Syndrome* - PSS) [Motta et al. 2024].

Com o objetivo de reduzir ou evitar essa síndrome nos suínos foi implementado um sistema designado por Relax Pig, a fim de auxiliar os pesquisadores da Suinocultura do IFFAR-CA no monitoramento destes animais. Com o sistema Relax Pig é possível registrar os dados dos suínos nas baias/gaiolas, armazenando informações como raça, sexo, idade, temperatura e umidade do ambiente. A Figura 3 apresenta a tela de Cadastro de Suínos do Sistema Relax Pig.

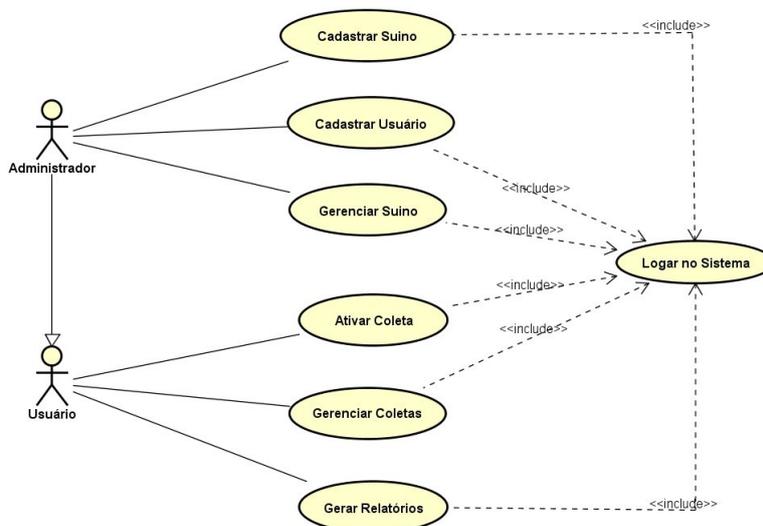
Figura 3. Tela de Cadastro de Suínos do Sistema Relax Pig.
Fonte: [Motta et al. 2024].



Suíno	Data de Nascimento	Sexo	Raça	Alterar	Excluir
1	14/07/2022	M	Landrace		
2	12/04/2023	F	Duroc		
3	27/09/2023	M	Large White		
4	10/12/2022	M	Moura		
5	19/08/2023	M	Berkshire		
6	15/01/2023	F	Piau		
7	27/03/2023	F	Pietran		
8	16/09/2023	F	Duroc		

Dentre as principais funcionalidades do sistema, citam-se o cadastro dos suínos, com todas as informações relevantes do animal, e o cadastro de usuários, responsáveis pela coleta de dados, e o registro das baias/gaiolas utilizadas. Com relação aos usuários, é importante destacar que o sistema provê controle de acesso às principais funcionalidades do sistema, como o cadastro de suínos, usuários e baias, que é restrito aos usuários administradores do sistema. Já a ativação da coleta de dados, o encerramento manual da coleta, e o acesso aos relatórios estão disponíveis para todos os usuários. A Figura 4 apresenta o Diagrama de Caso de Uso do Sistema Relax Pig [Motta et al. 2024].

Figura 4. Diagrama Caso de Uso do Sistema Relax Pig. Fonte: [Motta et al. 2024].



5.2. Casos de Teste

Para definir o escopo dos testes existem várias abordagens, sendo uma delas a especificação de casos de teste com um *design* predefinido, que deve conter detalhes de cada caso de teste, bem como os procedimentos de teste para determinar se um teste foi aprovado ou reprovado [De Souza 2013].

A metodologia de teste empregada neste trabalho segue os princípios estabelecidos na norma IEEE 829-1998 [IEEE 1998, Karouw and Wowor 2016, Javed and Mohammadian 2019]. Esta norma define uma série de documentos essenciais para conduzir as atividades de teste em um produto de *software*. Estes documentos serão utilizados no planejamento, especificação e registro dos testes realizados.

O documento referente ao Plano de Teste foi utilizado no planejamento da execução dos testes, no qual são identificadas as funcionalidades. As informações presentes são: nome do projeto, nome das pessoas que participaram dos testes e suas respectivas responsabilidades, cronograma com datas e horários para os testes, critérios para considerar o teste como finalizado, funcionalidades e módulos que serão testados, equipamentos e/ou *softwares* necessários, etc. A Tabela 1 apresenta o Plano de Teste do Sistema Relax Pig.

Durante o planejamento da execução dos testes é realizada a identificação das funcionalidades a serem testadas. Isso inclui a descrição precisa do funcionamento correto de cada teste, delineando um passo-a-passo das ações a serem realizadas pelo testador. A especificação do caso de teste descreve os passos que devem ser executados durante a verificação do sistema. Geralmente, é descrita a ação do ator e a resposta do sistema para um determinado cenário.

O caso de teste visa facilitar a criação do documento de testes na etapa de especificação. Foi realizada a compilação dos documentos da especificação do projeto de teste, especificação dos casos de teste e especificação dos procedimentos de teste. As informações neste documento abrangem o nome do projeto, um número identificador

Tabela 1. Plano de Teste do Sistema Relax Pig. Fonte: Autoria Própria.

Plano de Teste
Nome do Projeto: Validação Funcional do Sistema Relax Pig por meio de Casos de Teste Elaborados com Base na Norma IEEE 829 e Implementados no Framework Selenium IDE
Pessoas Envolvidas / Responsabilidade
Usuário 1 - Criação de casos de testes e execução dos testes.
Usuário 2 - Codificação e refatoração de código.
Funcionalidade ou Módulos
1) Validação do Usuário 2) Autenticação de usuário; 3) Criação de usuário; 4) Exclusão de usuário; 5) Criação de suíno; 6) Exclusão de suíno; 7) Criação de baia; 8) Exclusão de baia; 9) Criação de nova coleta de dados; e, 10) Gerar relatório da coleta de dados.
Equipamentos / Softwares
O sistema deve funcionar em um servidor Web com acesso via browser em desktop e dispositivos móveis, Ferramenta selenium.
Cronograma
Data de Início e Fim do Projeto: 05/04/2023 - 30/04/2024.
Data de Início e Fim do Teste: 14/12/2023 - 14/01/2024.
Local dos Testes:
O sistema não possui um local específico para testes. Os mesmos serão executados em equipamentos utilizados pelos pesquisadores envolvidos em locais aleatórios.
Critérios para considerar o teste finalizado
O teste será considerado como finalizado ao preencher a coluna resultado do teste do documento de casos de teste como “Executado com sucesso”.
Observações
O relatório de incidente é compartilhado com o pesquisador responsável pela codificação e refatoração e também anexado à documentação do software. Serão criados os casos de testes, os relatórios de incidentes e o relatório resumo de teste.

único para cada teste, o módulo ou rotina sujeito ao teste, uma breve descrição do teste, um roteiro contendo a descrição e os passos a serem seguidos, o resultado esperado que descreve como o sistema deve se comportar, o resultado do teste obtido pelo testador com base no caso de teste, incluindo a data e o resultado, e, por último, a data e a identificação do usuário, assim como o desenvolvedor responsável pela codificação e refatoração específica do código associado àquele caso de teste.

As Tabelas 2 e 3 apresentam, respectivamente, os casos de teste relativos ao cadastro e exclusão de de suíno, e à criação e exclusão de baia do sistema Relax Pig. Estes testes avaliam tanto a validação dos campos obrigatórios quanto a funcionalidade de exclusão e implicação na integridade referencial das informações persistidas no sistema. Estes casos de teste visam avaliar importantes funcionalidades do sistema, como a inclusão de suínos e baias.

O relatório de incidentes (Tabela 4), é uma compilação dos documentos diários de Teste e Relatórios de Incidente, com base na aplicação dos testes realizados no sistema Relax Pig. Este documento contém as seguintes informações: número de casos de teste, status, nome da pessoa responsável pela correção, prioridade para a correção do erro pelo

Tabela 2. Casos de teste de cadastro e exclusão de suínos do sistema Relax Pig.
Fonte: Autoria Própria.

Casos de testes						
Nome Projeto: Validação Funcional do Sistema Relax Pig por meio de Casos de Teste						
Elaborados com Base na Norma IEEE 829 e Implementados no Framework Selenium IDE						
ID	Módulo	Descrição	Roteiro	Resultado Esperado	Testador	Desenvolvedor
2.1	Criação do cadastro do Suíno	Verificar a Validação dos campos	1) Escolher a opção cadastrar Suínos 2) Deixar os campos raça, sexo e idade em branco	Mostrar a mensagem "Campo de preenchimento é obrigatório".	Usuário 1 - 21/12/2023 Executado com sucesso	Usuário 2 - 28/12/2023
2.2	Exclusão do cadastro do Suíno	Excluir dados cadastrados no sistema	1) Pesquisar e selecionar o suíno no sistema 2) Escolha a opção Excluir	Mostrar a mensagem "Deseja realmente excluir esse registro?" Ou se já existe movimentação no banco para o suíno "O suíno não pode ser excluído, pois o mesmo está associado a coletas já registradas no sistema!".	Usuário 1 - 21/12/2023 Executado com sucesso	Usuário 2 -

Tabela 3. Casos de teste de criação e exclusão de baia do sistema Relax Pig.
Fonte: Autoria Própria.

Casos de testes						
Nome Projeto: Validação Funcional do Sistema Relax Pig por meio de Casos de Teste						
Elaborados com Base na Norma IEEE 829 e Implementados no Framework Selenium IDE						
ID	Módulo	Descrição	Roteiro	Resultado Esperado	Testador	Desenvolvedor
3.1	Criação da Baia	Verificar a Validação dos campos	1) Escolher a opção inserir Novo 2) Deixar os campos nome em branco	Mostrar mensagem na tela "Campo de preenchimento é obrigatório".	Usuário 1 - 21/12/2023 Executado com sucesso	Usuário 2 - 28/12/2023
3.2	Exclusão da Baia	Excluir dados cadastrados no sistema	1) Pesquisar e selecionar o baia no sistema 2) Escolha a opção Excluir	Mostrar mensagem na tela "Deseja realmente excluir esse registro?" Ou se já existe movimentação no banco para o baia "A baia não pode ser excluído, pois está associada a coletas!".	Usuário 1 - 21/12/2023 Executado com sucesso	Usuário 2 - 28/12/2023

desenvolvedor, descrição detalhada do erro (que pode incluir a mensagem de erro exibida na tela), e nome e data de quem fez a correção. Após a correção do erro e sua inclusão no relatório de incidentes, o módulo testado é analisado novamente em busca de erros. Consequentemente, a coluna de resultados de teste na tabela de casos de teste é atualizada com informações indicando que o teste foi executado com sucesso ou com a data do teste e a descrição do erro.

Já o relatório Resumo de Teste (Tabela 5), é o documento final, e foi preenchido com as seguintes informações: nome do projeto, data de início e término do teste, descrição detalhada do teste realizado, pessoas envolvidas, o número de casos de teste criados antes e durante a execução e o número de execuções bem-sucedidas. Além disso, devem ser fornecidos o *status*: concluído com sucesso, concluído com restrições e não concluído. Também é importante incluir o percentual de casos de teste executados, o percentual de casos executados com sucesso e com erros, e o percentual de casos de teste corrigidos pelo desenvolvedor.

Tabela 4. Relatório de Incidente do Sistema Relax Pig. Fonte: Autoria Própria.

Relatório de Incidente					
Nome Projeto: Validade Funcional do Sistema Relax Pig por meio de Casos de Teste Elaborados com Base na Norma IEEE 829 e Implementação no Framework Selenium IDE					
ID	Status	Responsável pela correção	Prioridade de Correção	Descrição do erro	Data e Nome de quem corrigiu
1.1	Pronto para testar novamente	Usuário 2 -	Alta	Não foi possível acessar ao painel do usuário	1/01/2024 - Usuário 2
1.4	Pronto para testar novamente	Usuário 2 -	Alta	O sistema permite a exclusão do usuário mesmo já estando em movimentação no banco de dados.	1/01/2024 - Usuário 2

Tabela 5. Relatório de Resumo de Teste do Sistema Relax Pig. Fonte: Autoria Própria.

Relatório Resumo de Teste	
Nome Projeto:	Validação Funcional do Sistema Relax Pig por meio de Casos de Teste Elaborados com Base na Norma IEEE 829 e Implementados no Framework Selenium ID
Data início teste:	14/12/2023
Data fim teste:	14/01/2024
Descrição teste	
A criação dos casos de teste previamente facilitou a detecção e correção de erros. O relatório de incidência de teste demonstrou de forma clara a correção do pesquisador responsável pela codificação e refatoração é o momento certo para o retorno para a pesquisadora responsável pela automação dos testes do sistema, de forma que novos testes fossem realizados.	
Pessoas envolvidas	
Usuario1, Usuario2	
Números do teste	
Casos de teste criados antes do teste	10
Casos de teste criado durante o teste	0
Casos de teste executados	10
Casos de teste com erro	2
Casos de testes enviados para correção	2
Percentual	
Casos de teste executados	10 (100%)
Casos de teste executados com sucesso	8 (80%)
Casos de teste com incidência de erro	2 (20%)
Casos de teste corrigidos pelo desenvolvedor	2 (100%)

5.3. Aplicação dos Testes

Para avaliar o desempenho do sistema Relax Pig uma série de testes foi conduzida abrangendo diversas funcionalidades. Os testes realizados incluíram autenticação de usuário,

criação de usuário, dentre outros. A seguir, são apresentados, para fins de exemplificação, alguns resultados dos testes realizados juntamente com os respectivos códigos HTML (*HyperText Markup Language*) produzidos.

O primeiro teste automatizado consistiu na validação do painel de usuário. Durante este teste foram simuladas situações de inserção de credenciais visando à confirmação da integridade funcional do referido painel. Durante esse teste foi identificada uma inconsistência na entrada para acessar o painel de usuário, resultando na apresentação de um painel em branco. É crucial destacar que este erro tem um impacto considerável no funcionamento do sistema. No teste de validação dos usuário no sistema foram consideradas situações como a inserção de usuário e senha incorretos. O processo de validação de usuários obteve êxito, evidenciado pelo seguinte resultado: ao tentar efetuar *login* com senha incorreta, uma mensagem indicando 'usuário não cadastrado' foi devidamente exibida. A Figura 5(a) apresenta o código HTML relacionado ao teste na Validação do painel do usuário.

Figura 5. Teste realizado na Validação do Painel do Usuário (a) e Teste realizado na Exclusão de Usuário (b). Fonte: Autoria Própria.

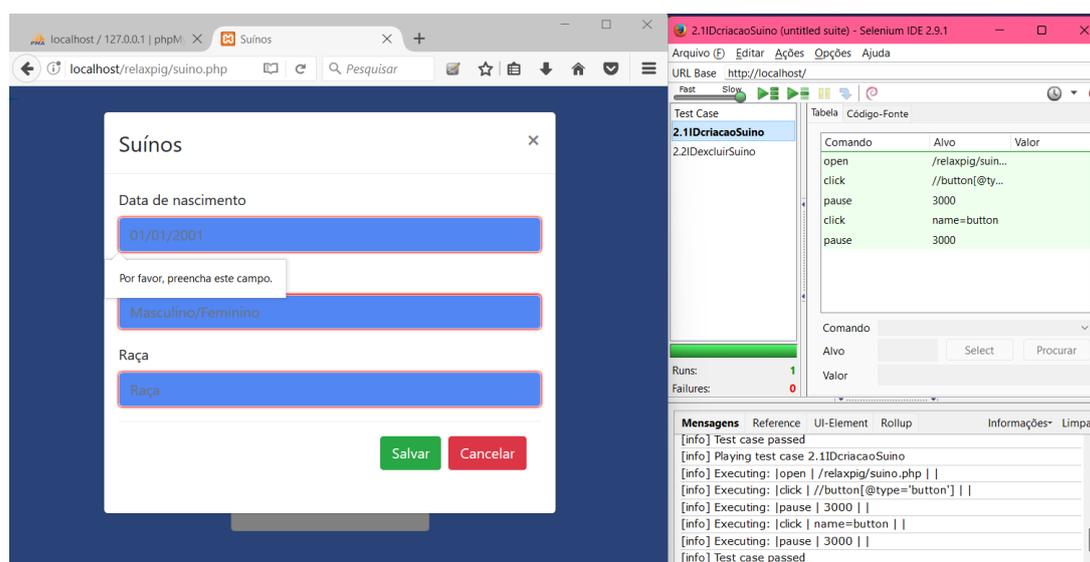
Código do teste em HTML	Código do teste em HTML
<pre> <?xml version="1.0" encoding="UTF-8"?> <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd"> <html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en"> <head profile="http://selenium-ide.openqa.org/profiles/test-case"> <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" /> <link rel="selenium.base" href="http://localhost/" /> <title>U04IDpainelUsuario</title> </head> <body> <table cellpadding="1" cellspacing="1" border="1"> <thead> <tr><td rowspan="1" colspan="3">U04IDpainelUsuario</td></tr> </thead><tbody> <tr> <td>open</td> <td>/relaxpig/index.php</td> <td></td> </tr> <tr> <td>type</td> <td>name=usuario</td> <td>usuario</td> </tr> <tr> <td>type</td> <td>name=senha</td> <td>usuario</td> </tr> <tr> <td>clickAndWait</td> <td>//input[@value='Login']</td> <td></td> </tr> </tbody></table> </body> </html> </pre>	<pre> <?xml version="1.0" encoding="UTF-8"?> <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd"> <html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en"> <head profile="http://selenium-ide.openqa.org/profiles/test-case"> <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" /> <link rel="selenium.base" href="http://localhost/" /> <title>U03IDexclusaousuario</title> </head> <body> <table cellpadding="1" cellspacing="1" border="1"> <thead> <tr><td rowspan="1" colspan="3">U04IDexclusaousuario</td></tr> </thead><tbody> <tr> <td>open</td> <td>/relaxpig/painel_admin.php</td> <td></td> </tr> <tr> <td>clickAndWait</td> <td>link=Usuários</td> <td></td> </tr> <tr> <td>click</td> <td>//tr[2]/td[7]/a/i</td> <td></td> </tr> <tr> <td>assertConfirmation</td> <td>Tem certeza de que deseja excluir este usuario?</td> <td></td> </tr> </tbody></table> </body> </html> </pre>
(a)	(b)

O teste seguinte concentrou-se na criação de usuário, com o objetivo de verificar a funcionalidade dos campos de preenchimento obrigatório e a validação do botão de Salvar. Este teste transcorreu sem intercorrências, resultando na exibição da mensagem 'Preencher campos obrigatórios', impedindo qualquer avanço sem o preenchimento adequado.

Na sequência, foi realizado o teste de exclusão. cujos objetivos principais são: a) alertar o administrador sobre a exclusão de outros usuários; e, b) impedir a exclusão de usuários envolvidos em operações já persistidas em tabelas do banco de dados. Este teste resultou um erro: mesmo usuários envolvidos em operações em andamento puderam ser excluídos. Este erro comprometeu a integridade referencial das informações do sistema, impactando negativamente na consistência e confiabilidade dos dados persistidos previamente. A Figura 5(b) apresenta o código HTML relacionado ao teste realizado na Exclusão de Usuário.

Por fim, foram realizados testes de criação de cadastro de suíno e baía, assim como a exclusão de ambos. Os resultados obtidos durante estes testes foram satisfatórios, não havendo a geração de erros. A criação de novas coletas de dados e geração de relatórios, incluindo a validação do preenchimento dos campos obrigatórios dos formulários também foram bem-sucedidos nos testes funcionais aplicados. A Figura 6 apresenta o teste relacionado ao cadastro de suíno realizado no *Framework Selenium IDE*.

Figura 6. Teste realizado quanto à criação de suíno no *Framework Selenium IDE*.



A Figura 6 demonstra a aplicação da *suite* de casos de testes de criação e exclusão de suíno de forma automatizada por meio da ferramenta Selenium IDE no sistema Relax Pig. É possível verificar que o teste foi executado corretamente sem a ocorrência de nenhum erro.

6. Análise dos Resultados

A detecção de erros que interferem no funcionamento do sistema é fundamental para garantir sua eficácia e eficiência. No contexto apresentado foram identificadas questões que comprometiam funcionalidades básicas do sistema. Estes erros foram reportados ao desenvolvedor responsável pelo sistema Relax Pig com o objetivo de que fossem implementadas as correções necessárias a fim de sanar os erros encontrados. Foram criados e executados um total de dez casos de teste.

Nos primeiros testes funcionais aplicados, uma das principais inconsistências residia no painel de acesso do usuário. Mesmo com as credenciais de *login* e senha atua-

lizadas no banco de dados, não estava sendo possível realizar o acesso ao sistema. Este erro impactava em uma funcionalidade básica do sistema, tornando imprescindível sua correção.

Durante os testes, outro erro identificado foi a tentativa de exclusão de um usuário. Mesmo quando o usuário já possuía movimentações registradas no banco de dados ainda sim era permitido a exclusão deste. Para mitigar este problema, o sistema foi ajustado para solicitar a confirmação antes de proceder com a exclusão, exibindo a seguinte mensagem "Deseja realmente excluir esse registro?".

Além disso, uma nova funcionalidade foi implementada no sistema, a fim de verificar a presença de movimentações associadas aos dados antes de se proceder a exclusão. Caso hajam registros de movimentações o sistema verifica a integridade referencial das tabelas de banco de dados do sistema. Se for identificada alguma movimentação existente o sistema não permite a exclusão. Nesse cenário, uma mensagem será apresentada informando: "Já existe movimento para este usuário no sistema, portanto, não poder ser excluído!".

Ao comunicar as situações observadas nos testes ao desenvolvedor, as correções foram aplicadas com a devida urgência, visando restabelecer a plena funcionalidade do painel do usuário. Esta medida foi implementada para garantir a integridade dos dados e evitar a exclusão involuntária de registros com movimentação. Todos os erros encontrados na etapa de testes do sistema foram devidamente documentados e corrigidos.

Por meio dos testes executados pode-se perceber a importância da automação dos testes funcionais, a fim de facilitar o trabalho do testador. Também foi possível verificar a importância da utilização de ferramentas como o *framework* Selenium, que permite o armazenamento de testes em conjunto, formando suítes de testes que podem ser executados sequencialmente. Entretanto, sugere-se que a automação dos testes seja iniciada individualmente, módulo por módulo do sistema, de forma a garantir que as funcionalidades sejam tratadas de acordo com suas particularidades.

Uma das vantagens do processo de automatização de testes observado no Selenium foi a possibilidade de gravação do teste, em que o testador não precisa necessariamente interagir diretamente com o sistema. Os passos são executados automaticamente, e os resultados podem ser visualizados através do *feedback* fornecido pela ferramenta, que detalha o que acontece durante a execução do teste. Caso um erro seja encontrado a própria ferramenta indica sua natureza, contribuindo para a redução do tempo necessário para a correções e ajustes do sistema analisado.

7. Conclusões

A validação e verificação de software são atividades dispendiosas no processo de desenvolvimento de software, pois podem exigir uma quantidade significativa de recursos. A exatidão e o custo associado a esta atividade dependem principalmente da importância da aplicação a ser desenvolvida. Diferentes categorias de aplicações requerem uma atenção distinta a estas atividades. Realizar verificações não se limita apenas à criação e execução de casos de teste, mas também envolve questões de planejamento, gerenciamento e análise de resultados.

Neste artigo foi apresentada a validação funcional do sistema Relax Pig por meio

do *framework* Selenium. Os testes aplicados e os resultados obtidos permitiram avaliar as funcionalidades disponíveis no sistema investigado. Também foi possível realizar a aplicação automática de testes funcionais. Esta abordagem promove maior eficiência e confiabilidade aos processos de teste, contribuindo para a qualidade do produto de software desenvolvido. Foram criados e executados um total de dez casos de teste. Todos os erros encontrados na etapa de testes do sistema investigado foram devidamente documentados e corrigidos.

Com a validação funcional do sistema Relax Pig utilizando o *framework* Selenium no ambiente de testes, em conformidade com a Norma IEEE 829, foi possível evidenciar a importância da etapa de testes no contexto de desenvolvimento do *software*, podendo inclusive servir como arcabouço para a automatização de testes de futuros *softwares* desenvolvidos no Instituto Federal Farroupilha.

Como trabalhos futuros, pretende-se realizar a criação de novos casos de teste voltados para a validação de novas funcionalidades do sistema Relax Pig, tais como a coleta automática de dados por meio da utilização de sensores integrados com uma plataforma Arduíno, a fim de verificar a integridade e confiabilidade do sistema perante estes novos cenários de uso. Além disso, pretende-se também realizar testes de aceitação do sistema Relax Pig juntamente com os *stakeholders*.

Referências

- Braga, F. A. (2019). Qualidade de software: proposta de automação de testes e de um processo ágil para uma empresa de software. <https://repositorio.animaeducacao.com.br/handle/ANIMA/10942>. [Online; accessed 17-June-2023].
- De Souza, K. P. (2013). A importância da atividade de teste no desenvolvimento de software. https://abepro.org.br/biblioteca/enegep2013_TN_STO_177_007_23030.pdf. [Online; accessed 11-June-2023].
- IEEE (1998). Ieee standard for software test documentation. *IEEE Std 829-1998*, pages 1–64.
- IEEE (2008). Ieee standard for software and system test documentation. *IEEE Std 829-2008*, pages 1–150.
- IEEE (2013). Iso/iec/ieee international standard - software and systems engineering —software testing —part 1:concepts and definitions. *ISO/IEC/IEEE P29119-1/FDIS, setembro de 2021*.
- Javed, Z. and Mohammadian, M. (2019). A novel model-driven framework for software and system testing. In *2019 4th International Conference on Information Systems and Computer Networks (ISCON)*, pages 750–756.
- Júnior, M., Andrade, S., and Delamaro, M. (2017). Capítulo 6: Automatização de teste de software com ênfase em teste de unidade. https://www.researchgate.net/publication/313360663_Capitulo_6_Automatizacao_de_teste_de_software_com_enfase_em_teste_de_unidade. [Online; accessed 25-June-2023].

- Karouw, S. and Wowor, H. (2016). Using cloud computing for building disaster information system prototype. In *2016 International Conference on Informatics and Computing (ICIC)*, pages 257–261.
- Motta, M., Santos, C., Vasconcellos, C., and Fontoura, J. (2024). Relax pig: Sistema para auxílio no monitoramento da síndrome do estresse suíno. Trabalho de Conclusão de Curso (TCC) apresentado ao Instituto Federal Farroupilha - Campus Alegrete, como requisito parcial para obtenção do título de Especialista em Computação Aplicada.
- Pereira, A., Oliveira, D., and Lima, G. (2013). Estudo de Caso da Ferramenta de Teste Selenium aplicado ao Módulo de Pedidos Online do Sistema FALCONFV. <https://pergamum.ifmg.edu.br/acervo/86514/>. [Online; accessed 10-June-2023].
- Peres, H. (2016). *Automatizado testes de software com selenium*. Simplíssimo Livros Ltda, 1ª edição edition.
- Santori, R. P. (2019). Avaliação da ferramenta de testes selenium no desenvolvimento guiado por teste de uma aplicação web. <https://bdm.unb.br/handle/10483/24474>. [Online; accessed 17-June-2023].
- Silva, C. G. (2019). Estudo Comparativo de Ferramentas de Testes de Ponta a Ponta Automatizados em Sistemas Web. <https://repositorio.ufrn.br/handle/123456789/43656>. [Online; accessed 10-June-2023].
- Silva, M. (2021). Desenvolvimento de Testes Automatizados e Testes Manuais para um Sistema Web: Uma Análise Comparativa. <https://memoria.ifrn.edu.br/handle/1044/2125>. [Online; accessed 17-June-2023].
- Sommerville, I. (2019). *Engenharia De Software*. PEARSON BRASIL.
- Valente, M. T. (2020). *Engenharia de software moderna: princípios e práticas para desenvolvimento de software com produtividade*. Independente, 1ª edition.
- Vieira, J. (2021). Avaliação e comparação entre dois frameworks de desenvolvimento de testes. <https://repositorio.animaeducacao.com.br/handle/ANIMA/19950>. [Online; accessed 17-June-2023].