

Teste funcional da peça rainha do jogo de Xadrez

**Alícia Duarte Evangelista¹, Elias Emanuel C. Lopes², Gabriel da Silva Pereira³,
Leonardo Figueiredo Santos⁴ e Luiz David B. Benevides⁵**

¹Departamento Acadêmico de Informação e Comunicação – Instituto Federal de Educação, Ciência e Tecnologia do Amazonas (IFAM) – Manaus – AM – Brazil
2023000391@ifam.edu.br, 2023002126@ifam.edu.br, 2023004130@ifam.edu.br,
2019003726@ifam.edu.br e 2024005249@ifam.edu.br

Abstract. *This article describes the testing of a chess game made in Java, composed of modules, one of which contains the Queen piece. Functional tests were completed on this piece using movement, capture and blocking requirements developed by the team as parameters. The web tool TestLink was used in conjunction with the MantisBT tool to manage the tests, where the latter is intended to track and monitor test failures. In this context, of the 12 tests executed, 6 resulted in success, 4 in failures and 2 in blocking, with the average execution time for each being 1 to 2 minutes. Finally, the analysis of the tests allowed us to verify that the queen moved only diagonally, directly influencing the result of the capture of enemy pieces.*

Resumo. *Este artigo descreve o teste de um jogo de Xadrez feito em Java composto por módulos, sendo um deles o que contém a peça Rainha. Foram executados testes do tipo funcional nessa peça utilizando como parâmetro requisitos de movimento, captura e bloqueio desenvolvidos pela equipe. Para gerenciamento dos testes foi utilizada a ferramenta web TestLink em conjunto a ferramenta MantisBT, onde a última possui a finalidade de rastreamento e acompanhamento das falhas dos testes. Nesse contexto, dos 12 testes executados, 6 resultaram em êxito, 4 em falhas e 2 em bloqueio, sendo que o tempo médio de execução para cada foi de 1 a 2 minutos. Por fim, a análise dos testes permitiu constatar que a rainha se move somente na diagonal, influenciando de forma direta o resultado de captura de peças inimigas.*

1. Introdução

Foi solicitada a realização dos testes da peça rainha do software Jogo de Xadrez desenvolvido em Java para avaliar se o comportamento obtido após a execução condiz com o comportamento esperado. Considerando isso, o trabalho apresenta a aplicação desses testes funcionais à peça Rainha de um jogo de Xadrez, desenvolvido com base em uma arquitetura modular. Um dos módulos é responsável pela lógica de movimentação dessa peça, cujas funcionalidades foram avaliadas por meio de critérios previamente definidos pela equipe, abrangendo requisitos de movimento, captura e bloqueio.

Os testes foram conduzidos segundo a abordagem de caixa preta, considerando exclusivamente os requisitos funcionais, sem análise do código-fonte. Para o gerenciamento do processo de testes, foram utilizadas as ferramentas TestLink e MantisBT, sendo esta última responsável pelo rastreamento e acompanhamento das falhas identificadas. Ao todo, foram executados 12 testes, onde 6 obtiveram êxito, 4 tiveram falhas e 2 foram bloqueados, com tempo médio de execução entre 1 e 2 minutos em cada caso de teste.

Este relatório apresenta de forma sistemática todas as etapas do processo de teste aplicado, incluindo a especificação de requisitos, elaboração do plano de testes, passos nas ferramentas utilizadas, execução prática, análise dos resultados e as considerações finais sobre a eficácia

do procedimento adotado.

1.1. Justificativa

Conforme Delamaro et al. (2007), o desenvolvimento de um software não é algo simples, podendo se tornar complexo conforme as características, necessidades e dimensões do sistema a ser criado. Qualquer sistema está sujeito a problemas que podem acarretar na criação de um produto diferente do que se pensava, problemas esses que possuem vários fatores, porém uma única origem: erro humano. Problemas de software ocorrem mesmo com a utilização de métodos eficientes e com as mais avançadas e sofisticadas ferramentas durante o desenvolvimento, uma vez que dependem de pessoas e suas respectivas habilidades, interpretações e execuções.

Para identificar os erros antes de o software ser repassado para a utilização é que existem atividades conhecidas como “Verificação, Validação e Teste”, e as mesmas baseiam-se na premissa de que tanto os processos de construção do software quanto o produto em si estejam de acordo com o que foi especificado.

A importância do teste, não somente no jogo de Xadrez fornecido, mas em qualquer software, se dá devido ao fato de que é possível eliminar erros logo nas fases iniciais da construção do software, economizando recursos e tempo.

Sendo assim, o presente trabalho é um teste do tipo funcional, pois avalia se o programa está conseguindo desempenhar suas funções anteriormente especificadas, e é dinâmico, uma vez que depende de um programa e utiliza entradas em particular para posterior verificação de conformidade com a especificação de requisitos, ou seja, realiza a análise do comportamento esperado e o obtido.

2. Objetivo dos Testes

Os testes realizados se enquadram como testes de caixa preta, uma vez que foram baseados exclusivamente nos requisitos funcionais, sem conhecimento prévio do código-fonte. O foco dos testes é verificar o comportamento externo da funcionalidade da Rainha conforme as regras do xadrez, independentemente da implementação interna e seguindo o ponto de vista do usuário.

Delamaro et al. (2007) aponta que independentemente do tipo de teste, existem fases bem definidas como planejamento, projeto de casos de teste, execução e por fim análise. O autor prossegue destacando que a atividade de teste não possui a finalidade de provar que determinado programa está correto, uma vez que existem limitações físicas de processamento e teóricas também, pois ao delimitar o teste em alguns casos, corre-se o risco de deixar de lado algum caso que poderia identificar um erro. Sendo assim, a atividade de teste bem realizada indica certo grau de confiabilidade para grande parte de seu funcionamento, mas não um mapeamento de todos os erros em um programa.

Seguindo a estrutura apontada, o plano de testes foi um artefato produzido durante o planejamento seguindo a especificação de requisitos, documento esse que também pode ser intitulado como projeto de casos de teste. A execução diz respeito a seguir os passos descritos no plano e a análise a interpretação dos resultados obtidos em cada teste, com finalidade de identificar falhas para posterior correção, visando assim satisfazer o comportamento requerido para a peça rainha do jogo de xadrez.

A Rainha no jogo de xadrez segue um comportamento definido de movimento, captura e bloqueio, assim como qualquer outra peça do tabuleiro, possuindo assim regras gerais, então pode-se entender o processo de teste do módulo da rainha como uma forma de verificação das funcionalidades essenciais dessa parte do jogo, que seriam metas a serem atingidas, sendo elas a movimentação em todas as direções, captura de peças em certos casos e o bloqueio do movimento. Tanto a peça da cor branca quanto a da cor preta devem atingir suas respectivas metas, e caso todas forem satisfeitas, o presente programa poderia ser caracterizado como confiável. Caso não, as falhas devem ser registradas e o programa deve passar por uma correção para que um novo ciclo de testes seja formalizado e efetuado, para que assim um nível máximo de requisitos seja atendido, aumentando a confiabilidade do programa.

3. Ferramentas Usadas

Para realizar as atividades de testar o jogo de xadrez, utilizamos as ferramentas TestLink e MantisBT. O TestLink foi utilizado para fazer o planejamento, elaboração e execução dos testes, enquanto o Mantis para o registro e acompanhamento dos bugs encontrados. Durante a realização dessas atividades, foi possível integrar as duas ferramentas, o que permitiu que falhas durante a execução dos testes fossem monitoradas no Mantis.

3.1 TestLink

O TestLink é uma ferramenta de apoio ao processo de teste de software, que permite gerenciar todas as etapas, desde o planejamento até a execução dos testes (Figura 1). Com ele, é possível criar projetos de teste, no qual nos permite especificar os requisitos do sistema, servindo como base para a criação dos testes. A partir desses requisitos, podemos cadastrar os casos de teste, sendo possível definir as pré-condições, os passos a serem executados e o resultado esperado. Isso ajuda a ter uma clareza quando os testes são executados.

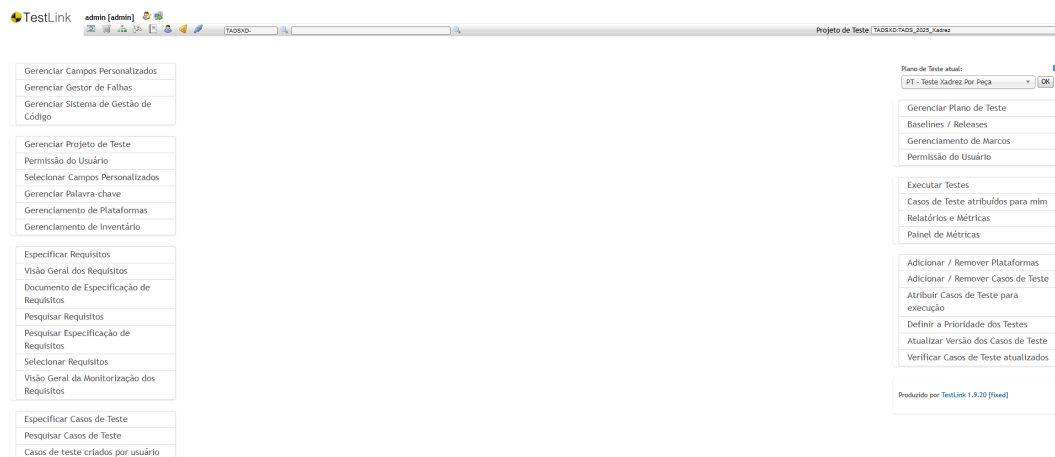


Figura 1. Tela Inicial do TestLink

Os casos de teste podem ser organizados em suítes de teste, agrupando-os por funcionalidades ou módulos específicos do sistema. Em seguida, é possível criar um plano de testes, que reúne os testes a serem executados em determinada fase do projeto. Durante esse processo, também é possível definir uma baseline, que representa uma versão de referência dos testes em execução.

O TestLink possibilita atribuir os testes a usuários específicos, que serão os responsáveis por executá-los. Na execução, o sistema registra se cada passo foi realizado com sucesso, e reportar o resultado de cada teste, podendo ter os status de passou, com falha ou bloqueado, e o usuário ainda pode adicionar observações quando os testes falharam ou foram bloqueados.

E uma funcionalidade interessante dessa ferramenta é a possibilidade de gerar relatórios das execuções dos testes, podendo ser um relatório de uma baseline, de uma suíte de teste ou até mesmo de um teste específico.

3.2 MantisBT

O MantisBT (Figura 2) é uma ferramenta de rastreamento e gerenciamento de bugs amplamente utilizada em projetos de software. Seu principal objetivo é permitir o registro, acompanhamento e controle de falhas durante o desenvolvimento ou fase de testes.

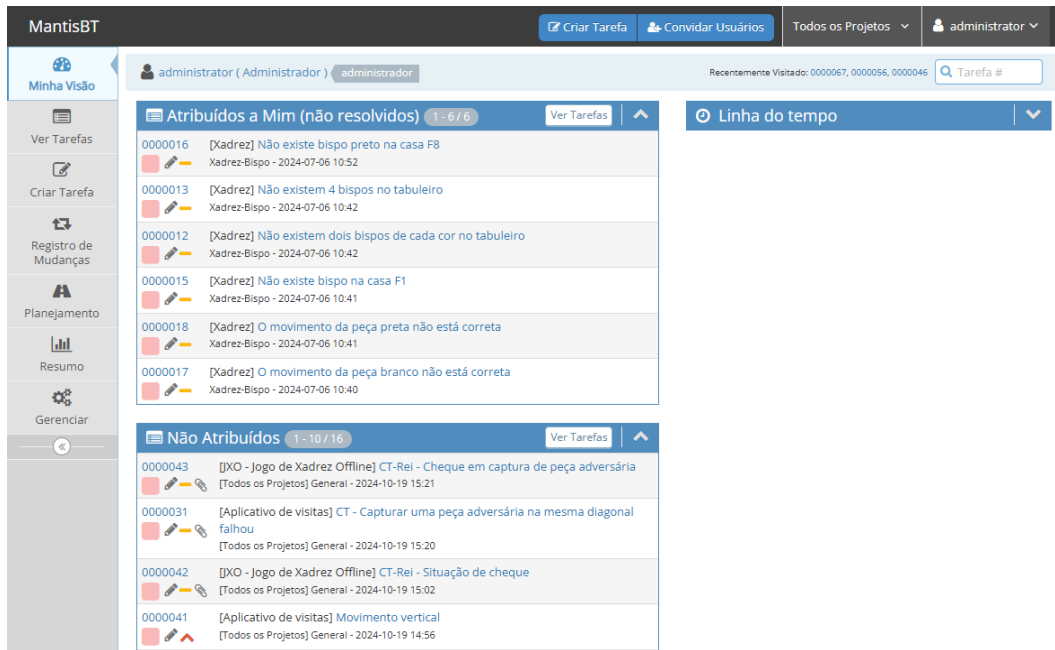


Figura 2. Tela inicial do MantisBT sendo o administrador

Por meio da criação de tarefas, é possível documentar detalhes sobre problemas identificados, como a descrição da falha, seu nível de gravidade, prioridade, e as ações esperadas para correção (Figura 3). Essas tarefas podem ser atribuídas a membros da equipe, permitindo um controle mais eficiente sobre quem será responsável pela resolução.

The screenshot displays the 'Criar Tarefa' form in MantisBT. The form is structured with several sections. The top section, 'Digite os Detalhes do Relatório', contains dropdown menus for 'Categoria', 'Reprodutibilidade', 'Gravidade', and 'Prioridade'. Below these are fields for 'Selecionar Perfil' and 'Atribuir a'. The 'Resumo' field is a short text input, while the 'Descrição' field is a larger text area with a 'Preencha este campo.' prompt. The 'Passos para Reproduzir' field is another large text area. The bottom section, 'Informações Adicionais', is a text area for additional details. The form is designed to capture comprehensive information about a reported issue.

Figura 3. Tela de criação de uma tarefa no MantisBT

A ferramenta suporta diferentes tipos de usuários, como o administrador que fica responsável pelas configurações gerais do sistema, o relator que registra as falhas encontradas e o desenvolvedor, que recebe as tarefas para análise e correção.

Durante o ciclo de vida de uma tarefa, seu estado pode ser atualizado de acordo com o progresso da correção. Os estados possíveis incluem novo, retorno, admitido, atribuído, resolvido e fechado (Figura 4).

Figura 4. Tela demonstrando uma tarefa criada e como alterar seu estado com sua progressão

Mais informações sobre as funcionalidades e configurações do MantisBT podem ser encontradas na sua documentação oficial (MANTISBT, 2025).

3.3 MantisBT e TestLink - Integração

O Mantis e o TestLink oferecem integração entre as ferramentas, o que permite associar uma tarefa a um teste que falhou. Com essa integração, é possível rastrear diretamente no TestLink quais testes resultaram em falhas e visualizar no Mantis os detalhes e o status da correção, promovendo uma melhor organização e comunicação entre as equipes de teste e desenvolvimento.

4. Processo de Testes Utilizado

O processo de testes utilizado pela equipe envolve a especificação de requisitos funcionais, voltados para garantir as regras de movimentação da Rainha; criação do plano de testes,

4.1 Especificação de requisitos e criação do plano de testes

Para definir o que se espera das funcionalidades da peça, foi feito o levantamento de requisitos. E para validar estas funcionalidades, a criação do plano de testes, que possui: casos de teste, para cada cenário específico dos requisitos funcionais; pré-condições, que definem o cenário desejado para testar uma funcionalidade; procedimentos, ou seja, as ações executadas para validar o respectivo caso de teste; e os resultados esperados para cada caso de teste de acordo com os requisitos estabelecidos.

4.2 Criação da estrutura de testes no TestLink

Após criar o plano de teste, o TestLink foi usado para gerenciar os testes da peça Rainha. Para isso, deve ser criado um projeto de teste para o software Xadrez e cadastrar uma suíte de teste para a peça Rainha.

Tendo esta estrutura como base, inicia-se o cadastro dos requisitos funcionais, bem como o cadastro dos casos de teste, para os quais são informadas as pré-condições, objetivos de cada caso de teste, as ações que devem ser executadas e seus respectivos resultados esperados.

4.3 Execução dos testes

Para a execução dos testes, os casos de teste cadastrados devem ser inseridos a uma baseline. Feito isso, em seguida cada caso de teste cadastrado é executado manualmente no tabuleiro de Xadrez com as peças Rainha Branca e Rainha Preta e demais peças solicitadas, se houver, de acordo com o respectivo pré-requisito, seguindo os passos a serem executados e validando se o resultado esperado é alcançado ou não. Para cada caso de teste, deve-se registrar os resultados obtidos e, ainda, registrar os problemas encontrados para os casos de falha ou bloqueio.

4.4 Registro de bugs no Mantis

Para rastreamento de bugs em todo o seu ciclo de vida até a correção, realiza-se o cadastro dos bugs relatados manualmente no Mantis e em seguida faz-se a integração desse cadastro no TestLink. Os bugs são atribuídos ao desenvolvedor e, após corrigidos, o software é testado novamente.

4.5 Reexecução e reteste

Após as correções, cria-se uma nova baseline (de correções), executam-se novamente os casos de teste que falharam, assim como os casos de teste bloqueados, e o status de cada caso é atualizado, completando o ciclo de teste.

4.6 Geração de relatórios

Para acompanhamento da qualidade, o TestLink gera relatórios contendo os resultados por ciclo, casos de falha, tendência de bugs e cobertura de testes.

5. Software do jogo de Xadrez

A partir de um arquivo do tipo *jar* executável, após realizar as devidas instalações, os testes foram executados.

O tabuleiro da (Figura 5) segue os requisitos de IHM que foram devidamente especificados, com o objetivo de descrever as regras de tamanho e desenho. No documento, os requisitos RIHM10 e RIHM20 descrevem respectivamente os desenhos do tabuleiro e de cada peça. Já os requisitos RIHM30 e RIHM40 descrevem o tamanho do tabuleiro e das peças, respectivamente.

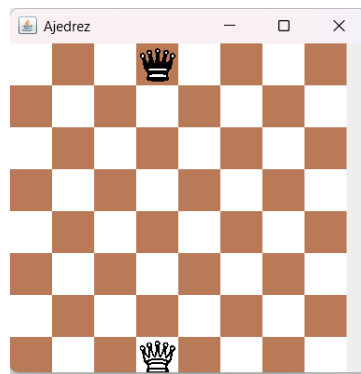


Figura 5. Imagem da Tela do teste da Rainha

5.1 Área de operação.

Os testes foram feitos utilizando arquivos do tipo java, ou seja, qualquer usuário que tenha uma máquina que suporte arquivos jar pode utilizar o jogo. Outro requisito importante é a utilização de periféricos como teclado e mouse para a interação do usuário com a tela. Importante destacar que o arquivo apenas funciona em computadores e notebooks, sendo assim, não há suporte para dispositivos mobile.

5.2 Pré-requisitos para execução do software

Para executar o software, foi necessário possuir uma máquina de execução (computador), que dispunha da versão do jdk 17 ou superior. Foi necessário também realizar o download e execução do arquivo compactado Xadrez_Rainha.jar, para assim dar prosseguimento aos testes.

6. Execução dos Testes

A execução inicial foi composta pela realização de passos pré-estabelecidos no documento de plano de testes.

6.1. Definição dos requisitos

Como se trata de um teste funcional, primeiramente é necessário entender o que o sistema está de fato proposto para realizar de acordo com as expectativas acordadas com o cliente, e isso acontece durante a fase de especificação dos requisitos. A especificação de requisitos está presente no documento “TADS-TS-Especificacao_requisitos_Xadrez”.

O documento especificava inicialmente 4 requisitos, sendo eles os requisitos RF51, RF52, RF53 e RF54, que dizem respeito à seleção da rainha (o jogador deve ser capaz de selecionar a rainha clicando na peça correspondente do tabuleiro), movimentação da rainha (a rainha deve ser capaz de se mover horizontalmente, verticalmente e diagonalmente para qualquer número de casas desocupadas em qualquer direção), captura de peças (a rainha deve ser capaz de capturar qualquer peça adversária que esteja no caminho de seu movimento, substituindo a peça adversária pela Rainha) e bloqueio de movimento (a rainha não pode pular sobre outras peças. Seu movimento é bloqueado por qualquer peça (amiga ou inimiga) que esteja no caminho), respectivamente. Porém, a equipe decidiu eliminar a RF51, uma vez que o software

fornecido não possui indicação visual de seleção e também pelo fato da indicação de cor ao selecionar a casa atual e a casa destino ser um critério opcional, conforme especificação.

6.2. Criação do plano de teste

Elaborou-se um plano de testes detalhado, abrangendo os requisitos essenciais de movimentação, captura e bloqueio. Esse plano (Tabela 1, Tabela 2 e Tabela 3) consiste na definição de casos de teste específicos para cada funcionalidade, incluindo os cenários de movimentação, captura e bloqueio, com base nas regras oficiais do jogo de xadrez.

Cada caso de teste elaborado contempla os seguintes elementos: pré-condições, que descrevem o estado inicial necessário do sistema; procedimentos, que indicam as ações a serem executadas durante o teste; e resultados esperados, que representam o comportamento correto da aplicação segundo os requisitos. O plano, portanto, estrutura-se como um instrumento essencial para assegurar que a lógica de movimentação da Rainha seja validada de forma sistemática e reproduzível.

Tabela 1. Casos de teste

N	Tela/funcionalidade	Caso de teste	Pré-condição
1	RF52 - Movimentação da Rainha	C.T.1: Movimento horizontal da Rainha Branca	1. Deve estar no turno Branco 2. Rainha Branca deve estar na casa D1 3. Caminho da Rainha até a casa H1 deve estar livre
2	RF52 - Movimentação da Rainha	C.T.2: Movimento horizontal da Rainha Preta	1. Deve estar no turno Preto 2. Rainha Preta deve estar na casa D8 3. Caminho da Rainha Preta até a casa A8 deve estar livre
3	RF52 - Movimentação da Rainha	C.T.3: Movimento vertical da Rainha Branca	1. Deve estar no turno Branco 2. Rainha Branca deve estar na casa D1 3. Caminho da Rainha Branca até a casa D5 deve estar livre
4	RF52 - Movimentação da Rainha	C.T.4: Movimento vertical da Rainha Preta	1. Deve estar no turno Preto 2. Rainha Preta deve estar na casa D8 3. Caminho da Rainha Preta até a casa D6 deve estar livre
5	RF52 - Movimentação da Rainha	C.T.5: Movimento diagonal da Rainha Branca	1. Deve estar no turno Branco 2. Rainha Branca deve estar na casa D1 3. Caminho da Rainha Branca até a casa F3 deve estar livre
6	RF52 - Movimentação da Rainha	C.T.6: Movimento diagonal da Rainha Preta	1. Deve estar no turno Preto 2. Rainha Preta deve estar na casa D8 3. Caminho da Rainha Preta até a casa A5 deve estar livre
7	RF53 - Captura de peças	C.T.7: Capturar peça preta	1. O jogo deve estar no turno branco 2. Deve haver peça adversária no tabuleiro, ou seja, preta 3. A peça preta deve ocupar uma posição que obedeça os movimentos possíveis da Rainha Branca 4. Caminho até a peça preta deve estar livre
8	RF53 - Captura de peças	C.T.8: Capturar peça branca	1. O jogo deve estar no turno preto 2. Deve haver peça adversária no tabuleiro, ou seja, branca 3. A peça branca deve ocupar uma posição que obedeça os movimentos possíveis da Rainha Preta 4. Caminho até a peça branca deve estar livre
9	RF53 - Captura de peças	C.T.9: Capturar peça amiga da Rainha Branca	1. O jogo deve estar no turno branco 2. Além da Rainha Branca, deve haver no tabuleiro pelo ao menos uma peça amiga dessa rainha, nesse turno 3. A peça amiga deve ocupar uma posição que obedeça os movimentos possíveis da Rainha Branca 4. Caminho até a peça amiga deve estar livre
10	RF53 - Captura de peças	C.T.10: Capturar peça amiga da Rainha Preta	1. O jogo deve estar no turno preto 2. Além da Rainha Preta, deve haver no tabuleiro pelo ao menos uma peça amiga dessa rainha, nesse turno 3. A peça amiga deve ocupar uma posição que obedeça os movimentos possíveis da Rainha Preta 4. Caminho até a peça amiga deve estar livre
11	RF54 - Bloqueio de movimento	C.T.11: Bloquear o movimento da Rainha Branca quando o caminho até a peça adversária selecionada estiver bloqueado	1. Deve estar no turno branco 2. Deve haver peça adversária (preta) 3. Caminho até a peça adversária escolhida deve estar obstruído por qualquer outra peça do tabuleiro
12	RF54 - Bloqueio de movimento	C.T.12: Bloquear o movimento da Rainha Preta quando o caminho até a peça adversária selecionada estiver bloqueado	1. Deve estar no turno preto 2. Deve haver peça adversária (branca) 3. Caminho até a peça adversária escolhida deve estar obstruído por qualquer outra peça do tabuleiro

Tabela 2. Procedimentos

Caso de teste	Procedimentos
C.T.1: Movimento horizontal da Rainha Branca	1. Selecionar Rainha Branca 2. Selecionar casa da posição H1
C.T.2: Movimento horizontal da Rainha Preta	1. Selecionar Rainha Preta 2. Selecionar casa da posição A8
C.T.3: Movimento vertical da Rainha Branca	1. Selecionar Rainha Branca 2. Selecionar casa da posição D5
C.T.4: Movimento vertical da Rainha Preta	1. Selecionar Rainha Preta 2. Selecionar casa da posição D6
C.T.5: Movimento diagonal da Rainha Branca	1. Selecionar Rainha Branca 2. Selecionar casa da posição F3
C.T.6: Movimento diagonal da Rainha Preta	1. Selecionar Rainha Preta 2. Selecionar casa da posição A5
C.T.7: Capturar peça preta	1. Selecionar a Rainha Branca 2. Selecionar a peça preta a ser capturada
C.T.8: Capturar peça branca	1. Selecionar a Rainha Preta 2. Selecionar a peça branca a ser capturada
C.T.9: Capturar peça amiga da Rainha Branca	1. Selecionar Rainha Branca 2. Selecionar peça amiga (branca)
C.T.10: Capturar peça amiga da Rainha Preta	1. Selecionar Rainha Preta 2. Selecionar peça amiga (preta)
C.T.11: Bloquear o movimento da Rainha Branca quando o caminho até a peça adversária selecionada estiver bloqueado	1. Selecionar Rainha Branca 2. Selecionar peça adversária
C.T.12: Bloquear o movimento da Rainha Preta quando o caminho até a peça adversária selecionada estiver bloqueado	1. Selecionar Rainha Preta 2. Selecionar peça adversária

Tabela 3. Execução dos testes

Caso de teste	Resultado esperado	Resultado obtido	Problema encontrado
C.T.1: Movimento horizontal da Rainha Branca	Movimentação deve ser concluída com sucesso	Não passou	Peça não se move na horizontal
C.T.2: Movimento horizontal da Rainha Preta	Movimentação deve ser concluída com sucesso	Não passou	Peça não se move na horizontal
C.T.3: Movimento vertical da Rainha Branca	Movimentação deve ser concluída com sucesso	Não passou	Peça não se move na vertical
C.T.4: Movimento vertical da Rainha Preta	Movimentação deve ser concluída com sucesso	Não passou	Peça não se move na vertical
C.T.5: Movimento diagonal da Rainha Branca	Movimentação deve ser concluída com sucesso	Passou	
C.T.6: Movimento diagonal da Rainha Preta	Movimentação deve ser concluída com sucesso	Passou	
C.T.7: Capturar peça preta	A captura da peça preta deve ser concluída com sucesso, ou seja, a peça preta deve sumir e a Rainha Branca deve ocupar a posição da captura	Bloqueado	Peça só se move nas diagonais, portanto a captura só acontecerá nesse tipo de movimento.
C.T.8: Capturar peça branca	A captura da peça branca deve ser concluída com sucesso, ou seja, a peça branca deve sumir e a Rainha Preta deve ocupar a posição da captura	Bloqueado	Peça só se move nas diagonais, portanto a captura só acontecerá nesse tipo de movimento.
C.T.9: Capturar peça amiga da Rainha Branca	A captura da peça amiga deve ser bloqueada e a Rainha Branca permanece na mesma posição em que foi selecionada	Passou	
C.T.10: Capturar peça amiga da Rainha Preta	A captura da peça amiga deve ser bloqueada e a Rainha Preta permanece na mesma posição em que foi selecionada	Passou	
C.T.11: Bloquear o movimento da Rainha Branca quando o caminho até a peça adversária selecionada estiver bloqueado	A captura da peça adversária deve ser bloqueada e a Rainha Branca deve permanecer na mesma posição de quando foi selecionada.	Passou	
C.T.12: Bloquear o movimento da Rainha Preta quando o caminho até a peça adversária selecionada estiver bloqueado	A captura da peça adversária deve ser bloqueada e a Rainha Preta deve permanecer na mesma posição de quando foi selecionada.	Passou	

Com base nos requisitos funcionais RF52 (Movimentação da rainha), RF53 (Captura de peças) e RF54 (Bloqueio de movimento), como anteriormente mencionado, foi elaborado um conjunto de casos de teste para validar o comportamento da peça Rainha.

6.3. Criação das suítes e casos de teste

Conforme Guerra (2017), uma suíte de testes é uma coleção de casos de teste ou specs destinados a testar um programa para verificar um determinado comportamento e existem vários benefícios ao se criar e utilizar uma suíte, por exemplo, aperfeiçoamento de

manutenibilidade ao se agrupar de forma lógica e semântica os testes, facilidade de serem mantidos, possibilidade de integração contínua, uma vez que a suíte ao ser definida no arquivo de configuração permite que os testes sejam executados pelo terminal, possibilitando a execução periódica. De início criou-se na ferramenta TestLink um projeto de teste denominado “TADS_2025_Xadrez” (Figura 6), que é basicamente o projeto de teste de todas as peças. Com o projeto de teste selecionado, criou-se uma suíte de teste para agrupar os casos de teste que serão criados conforme o plano desenvolvido.

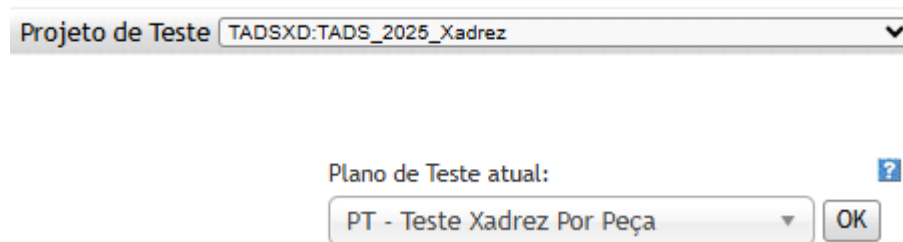


Figura 6. Projeto de teste e plano de teste da execução inicial

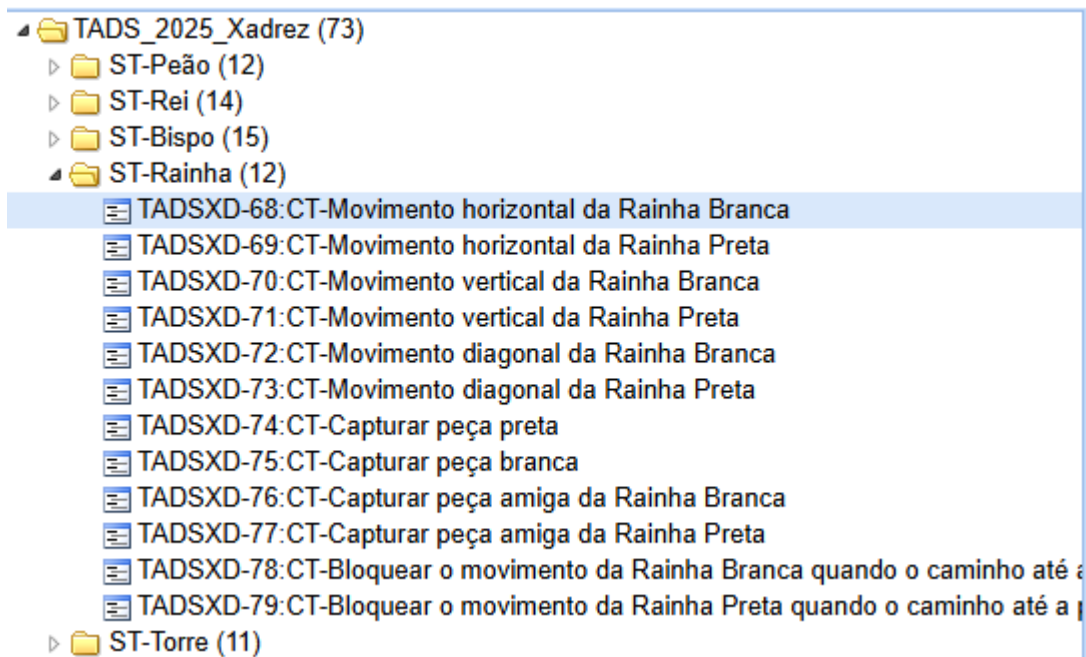


Figura 6. Estrutura dos arquivos para o teste

Cada caso de teste foi criado dentro da pasta ST-Rainha, e cada caso contém suas informações conforme plano de teste (Figura 6).

No momento de criar cada caso foi necessário fornecer entradas de dados de relevância para o teste, como por exemplo, as pré-condições, objetivos, os passos, a estimativa da duração do teste e o tipo de execução (Figura 7).

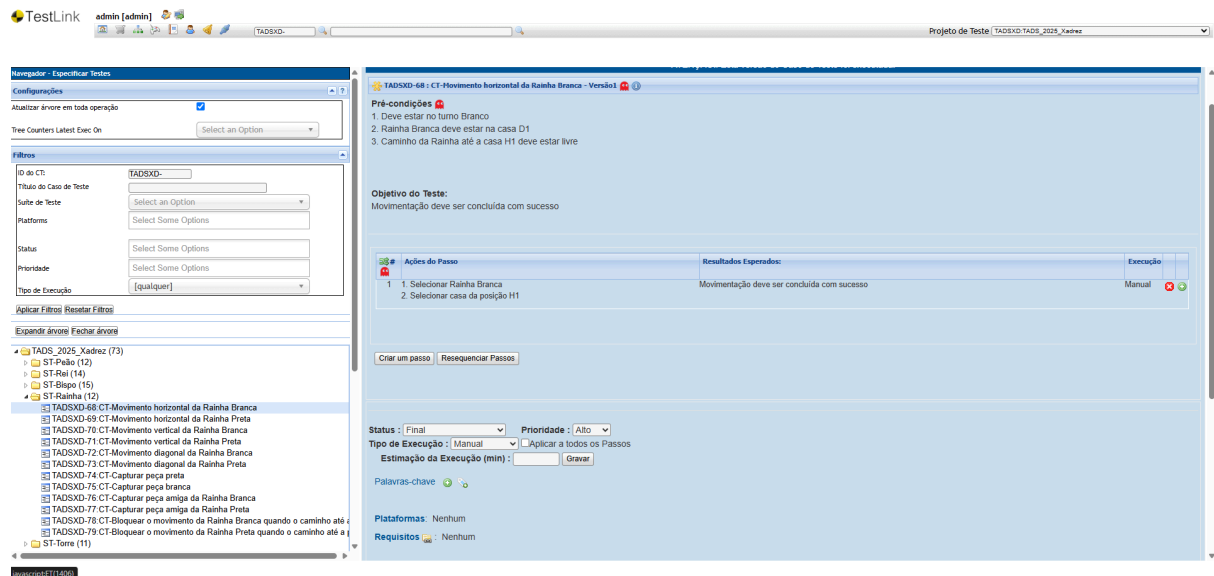


Figura 7. Detalhes de um caso de teste

Também foi necessário criar os requisitos funcionais na ferramenta (Figuras 8 e 9).

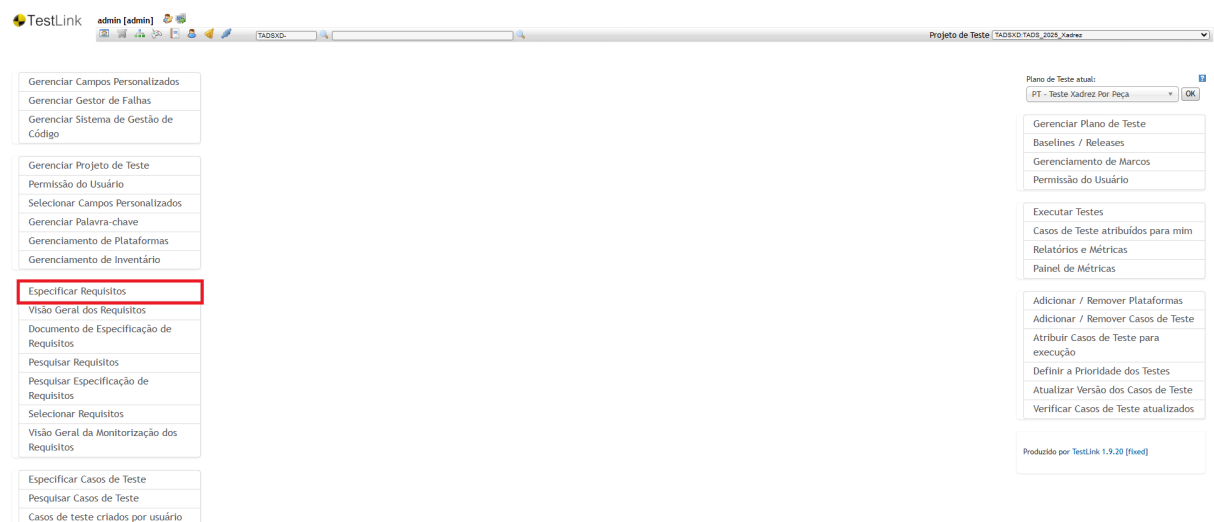


Figura 8. Campo de especificar requisitos em destaque

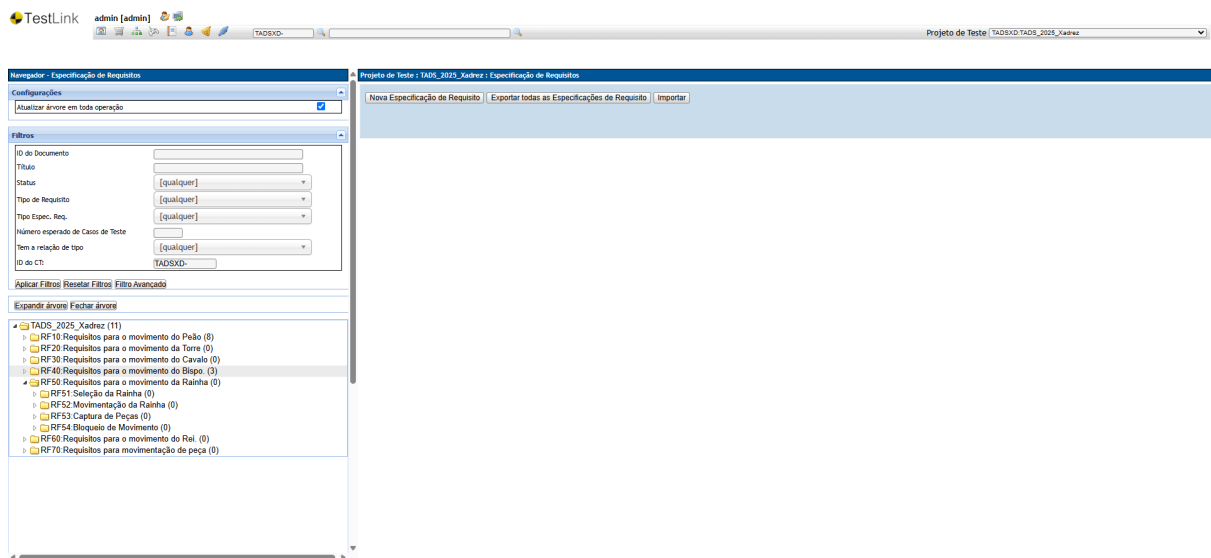


Figura 9. Criar requisito através do botão “Nova Especificação de Requisito”

Com cada caso criado, o processo seguiu para a atribuição do teste e execução (Figura 10).

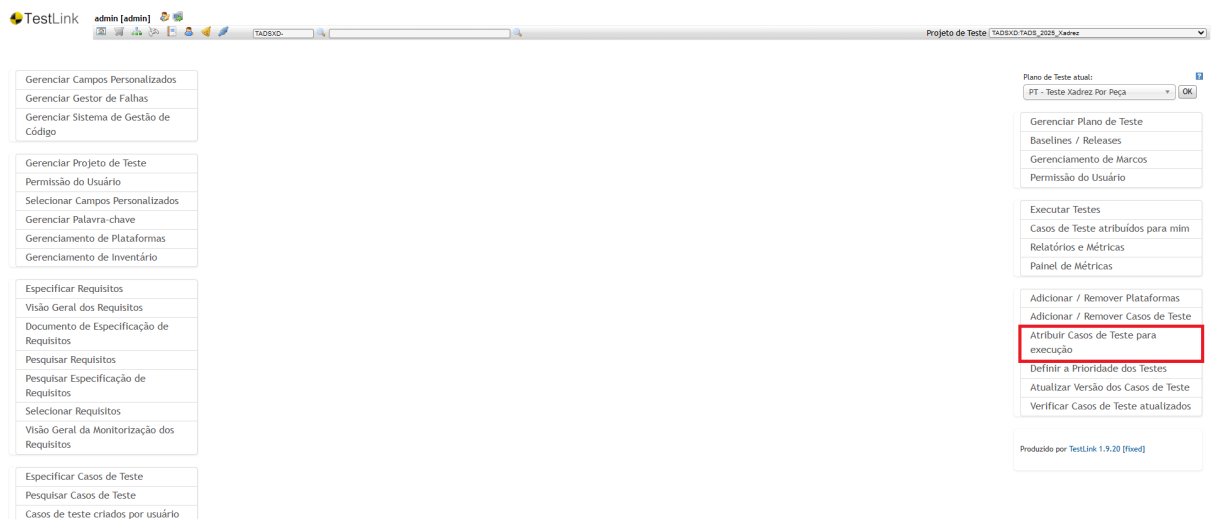


Figura 10. Campo para atribuir casos de teste destacado

Ao ser direcionado para a atribuição, selecionou-se a suíte de teste da rainha na aba da estrutura de pastas do lado esquerdo e designou-se o caso para um usuário em “Atribuir para”, ao selecionar o campo em branco na coluna de atribuir apareceu uma lista suspensa com os usuários cadastrados e possíveis de efetuar o teste. Após atribuir e gravar a atribuição, selecionou-se o ícone de engrenagem em verde para proceder com a execução do teste (Figura 11).

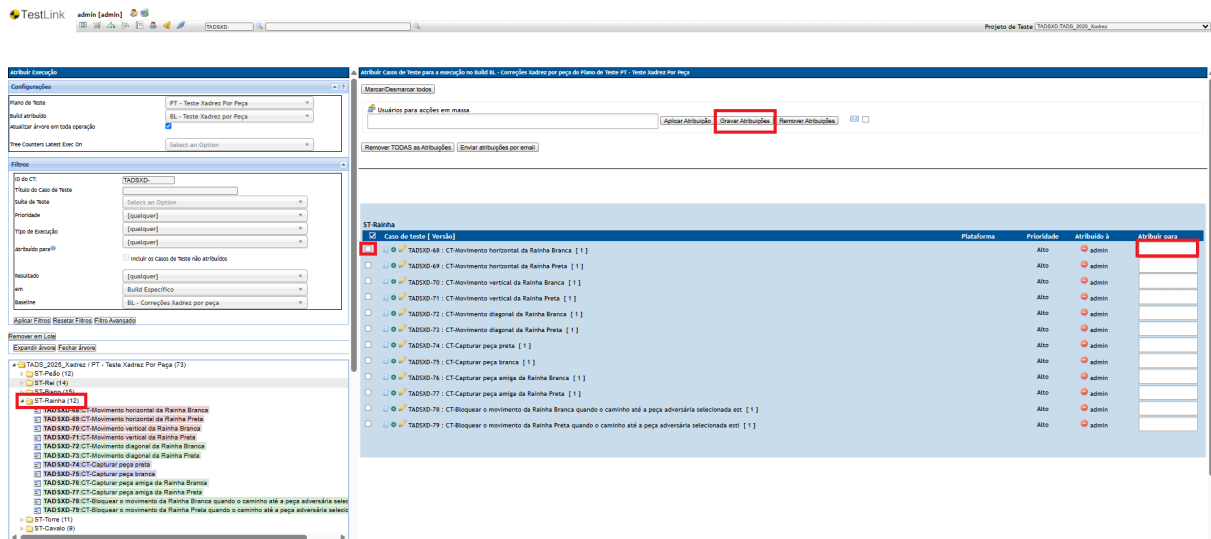


Figura 11. Tela de atribuição com a baseline inicial

Durante a execução do teste (Figura 12), o usuário alocado deve realizar o mesmo conforme as informações que foram registradas anteriormente. Foram inseridos dados essenciais como notas de execução, comentários, status (passou, falha e bloqueado), prioridade e duração.

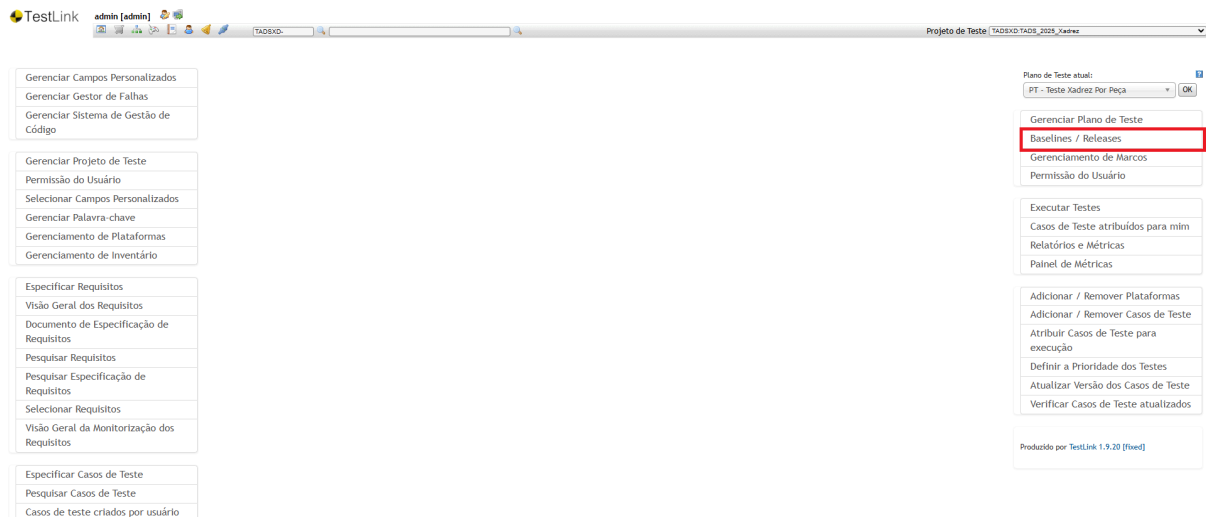


Figura 13. Campo para gerenciar baseline em destaque

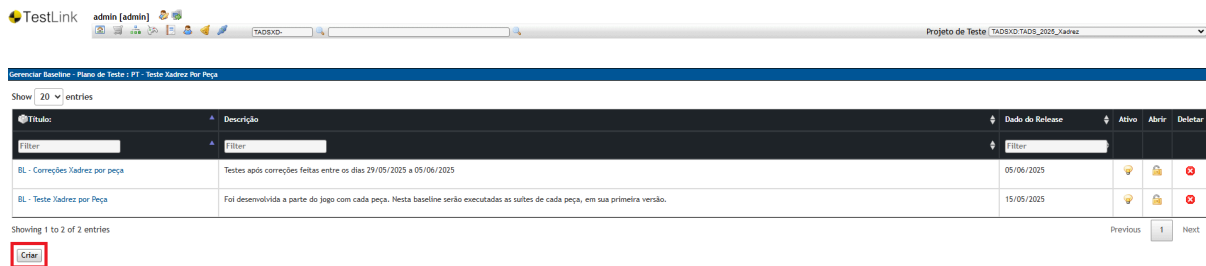


Figura 14. Tela de gerenciamento de baselines

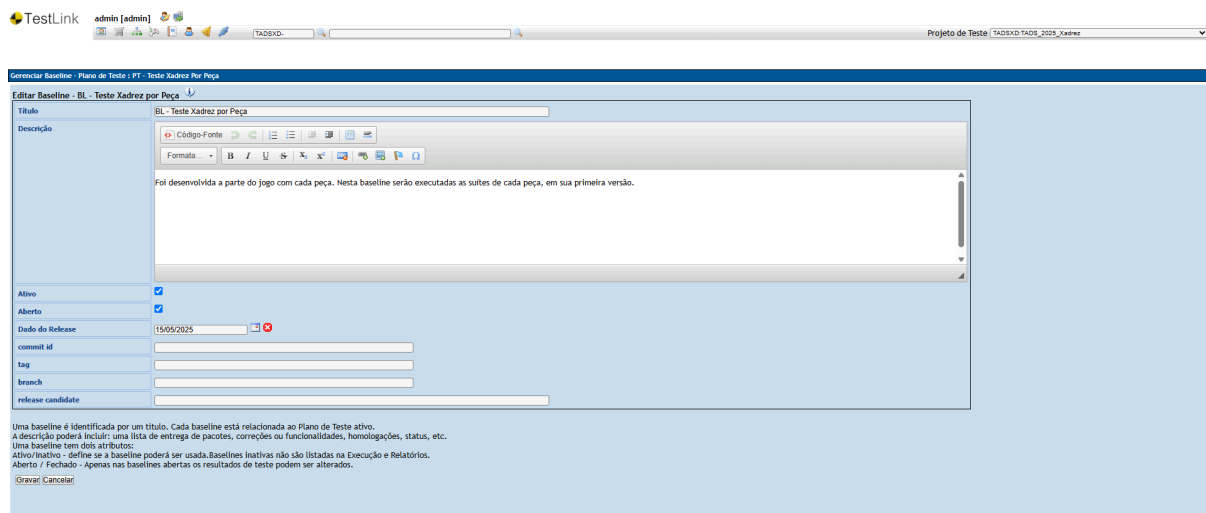


Figura 15. Tela de criação de baseline

6.5 Execução

A execução dos testes na primeira baseline da funcionalidade de movimento da Rainha revelou um cenário misto, indicando a necessidade de ajustes significativos em alguns comportamentos-chave.

Os testes de movimentação da Rainha apresentaram falhas notáveis. Quatro dos seis cenários

de movimentação esperada falharam, resultando em "Não Passou". Isso sugere que a Rainha não está conseguindo se mover corretamente em diversas direções ou para todas as casas desocupadas conforme o esperado. Apenas dois cenários de movimentação foram concluídos com sucesso.

No que tange à captura de peças, os resultados foram mais preocupantes. Ambos os testes de captura de peças adversárias (preta e branca) foram reportados como "Bloqueado". Isso indica que a lógica de captura está impedindo a Rainha de substituir a peça adversária, o que é um comportamento crítico e precisa de correção urgente.

Por outro lado, os testes relacionados ao bloqueio de movimento e captura de peças amigas/adversárias de forma indevida foram bem-sucedidos. Os quatro casos de teste que verificavam se a Rainha não capturava peças amigas ou se o movimento era bloqueado corretamente por peças adversárias (sem captura) foram aprovados. Isso demonstra que as regras de restrição de movimento e de não-captura de peças aliadas estão funcionando conforme o planejado.

Em síntese, enquanto a Rainha está respeitando as barreiras e não capturando peças amigas, a sua capacidade fundamental de movimentação e, principalmente, de captura de peças adversárias está comprometida. Esses resultados indicam que a implementação da lógica de movimento e captura da Rainha necessita de revisão e depuração para que o comportamento do jogo esteja alinhado com a baseline do teste.

7. Resultados Alcançados

Com a baseline chamada de BL - Teste Xadrez por peça, foram realizadas as execuções dos testes relacionados a peça da rainha, na suíte de teste ST-Rainha. 12 casos de teste focados exclusivamente no comportamento da peça Rainha no jogo de xadrez. Cada teste buscava verificar se os movimentos da peça, bem como suas ações de captura e bloqueio, estavam funcionando conforme os requisitos definidos.

Dos 12 testes executados, 6 foram aprovados, 4 apresentaram falhas e 2 permaneceram bloqueados (Figuras 16 e 17)

The screenshot displays the TestLink web application interface. On the left, the 'Ambiente de Teste' (Test Environment) configuration is visible, showing details for the 'ST-Rainha' test suite. The main panel on the right shows the 'Ambiente de Teste' configuration for the 'ST-Rainha' test suite, including a table of test cases and their results.

ST-Rainha	Caso de teste (Versão)	Plataforma	Prioridade	Atribuído a	Resultado
1	TAD300-01 - CT Movimento horizontal da Rainha Branca [1]	Alto	Admín		
2	TAD300-02 - CT Movimento horizontal da Rainha Preta [1]	Alto	Admín		
3	TAD300-03 - CT Movimento vertical da Rainha Branca [1]	Alto	Admín		
4	TAD300-04 - CT Movimento vertical da Rainha Preta [1]	Alto	Admín		
5	TAD300-05 - CT Movimento diagonal da Rainha Branca [1]	Alto	Admín		
6	TAD300-06 - CT Movimento diagonal da Rainha Preta [1]	Alto	Admín		
7	TAD300-07 - CT Capturar peça preta [1]	Alto	Admín		
8	TAD300-08 - CT Capturar peça branca [1]	Alto	Admín		
9	TAD300-09 - CT Capturar peça amiga da Rainha Branca [1]	Alto	Admín		
10	TAD300-10 - CT Capturar peça amiga da Rainha Preta [1]	Alto	Admín		
11	TAD300-11 - CT Bloquear o movimento da Rainha Branca quando o caminho até a peça adversária selecionada está [1]	Alto	Admín		
12	TAD300-12 - CT Bloquear o movimento da Rainha Preta quando o caminho até a peça adversária selecionada está [1]	Alto	Admín		

Figura 16. Tela mostrando a suíte de teste ST-Rainha e seus resultados

Suite de Teste: ST-Rainha (12 Items)					
TADSXD-68: CT-Movimento horizontal da ...	Alto	Passou [v1]	Com Falha [v1]	Com Falha [v1]	A rainha branca não se move na horizontal
TADSXD-69: CT-Movimento horizontal da ...	Alto	Passou [v1]	Com Falha [v1]	Com Falha [v1]	A rainha preta não se move na horizontal
TADSXD-70: CT-Movimento vertical da Ra...	Alto	Passou [v1]	Com Falha [v1]	Com Falha [v1]	A rainha branca não se move na vertical
TADSXD-71: CT-Movimento vertical da Ra...	Alto	Passou [v1]	Com Falha [v1]	Com Falha [v1]	A rainha preta não se move na vertical
TADSXD-72: CT-Movimento diagonal da R...	Alto	Passou [v1]	Passou [v1]	Passou [v1]	A movimentação da rainha branca na diagonal foi executada com sucesso!
TADSXD-73: CT-Movimento diagonal da R...	Alto	Passou [v1]	Passou [v1]	Passou [v1]	A movimentação da rainha preta na diagonal foi executada com sucesso!
TADSXD-74: CT-Capturar peça preta	Alto	Passou [v1]	Bloqueado [v1]	Bloqueado [v1]	Peça só se move nas diagonais, portanto a captura só acontecerá nesse tipo de movimento.
TADSXD-75: CT-Capturar peça branca	Alto	Passou [v1]	Bloqueado [v1]	Bloqueado [v1]	Peça só se move nas diagonais, portanto a captura só acontecerá nesse tipo de movimento.
TADSXD-76: CT-Capturar peça amiga da ...	Alto	Passou [v1]	Passou [v1]	Passou [v1]	
TADSXD-77: CT-Capturar peça amiga da ...	Alto	Passou [v1]	Passou [v1]	Passou [v1]	
TADSXD-78: CT-Bloquear o movimento da...	Alto	Passou [v1]	Passou [v1]	Passou [v1]	
TADSXD-79: CT-Bloquear o movimento da...	Alto	Passou [v1]	Passou [v1]	Passou [v1]	

Figura 17. Resultados do teste

A execução dos testes foi realizada manualmente diretamente no software do jogo de xadrez. O tempo médio de execução por teste variou entre 1 a 2 minutos, considerando a execução dos passos e o registro dos resultados. Mesmo sendo uma quantidade reduzida de testes, os resultados obtidos permitiram identificar comportamentos inconsistentes e serviram como base para a abertura de tarefas.

8. Correção

Após executar a suíte de teste da Rainha na primeira baseline, os bugs identificados foram registrados no MantisBT. Para isso, foi necessário acessar o perfil de relator cadastrado no MantisBT, criar nova tarefa, fornecer informações como resumo do bug, descrição do caso de teste, seu resultado e passos para reproduzir o teste, além de atribuir a tarefa ao perfil de desenvolvedor (Figura 18).

The screenshot shows the MantisBT web interface. The main content area displays the details of a new task being created. The task is titled '0000046: Aplicativo de visitas' and is assigned to 'rainha-dev'. The task is in the 'confirmed' state and has a priority of 'alta'. The description includes the test case details and the result. The interface also shows a sidebar with navigation options and a bottom status bar.

Figura 18. Criando nova tarefa no MantisBT

Ao criar uma nova tarefa, deve-se coletar o id gerado e cadastrá-lo no seu respectivo caso de

teste no TestLink, em “Gerenciamento de bug” (Figura 19).

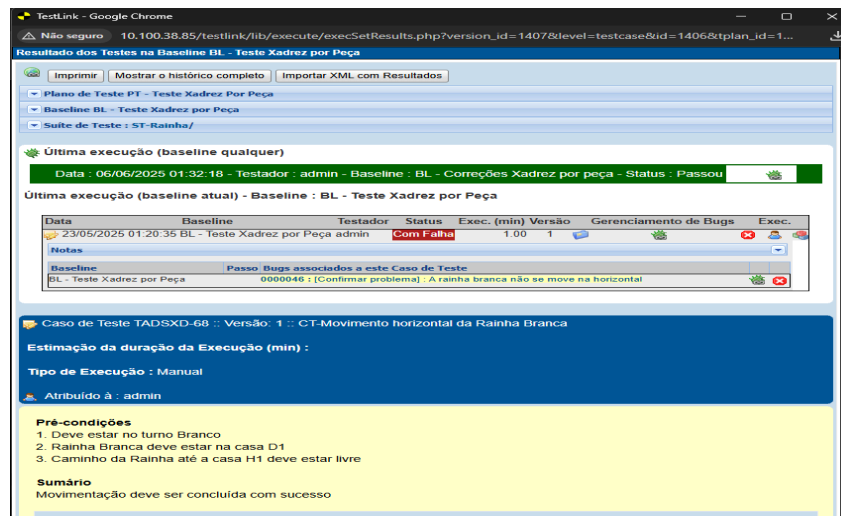


Figura 19. Associando tarefa do MantisBT aos casos de teste no TestLink

Feito isso, o grupo acompanhou a correção dos testes através do Mantis, no perfil de desenvolvedor, marcando o status como “admitido” após verificar que o bug existe e em seguida alterando para “resolvido” após finalizar a correção do bug relacionado à movimentação horizontal e vertical da Rainha.

9. Conclusão

A prática de testes de software, conforme demonstrado neste artigo, é um pilar indispensável para a garantia da qualidade e confiabilidade de qualquer sistema. A experiência do grupo com a aplicação de ferramentas como TestLink e Mantis não apenas reforçou essa premissa, mas também proporcionou aprendizados valiosos sobre a gestão e execução de testes.

O TestLink se revelou uma ferramenta eficaz para a organização e rastreabilidade dos casos de teste, facilitando o acompanhamento do progresso e a comunicação entre os membros da equipe. Da mesma forma, o Mantis foi fundamental para a identificação, registro e acompanhamento de defeitos, agilizando o processo de correção e validação. A integração e o uso conjunto dessas ferramentas permitiram uma visão mais holística do ciclo de vida do teste, desde a concepção dos cenários até a resolução dos *bugs*.

Contudo, a jornada não foi isenta de desafios. Dentre os pontos que exigiram maior atenção, destacam-se a curva de aprendizado inicial com as plataformas, a necessidade de padronização rigorosa na escrita dos casos de teste para evitar ambiguidades e a gestão de prioridades em ambientes de desenvolvimento ágil. Para futuras experiências, sugere-se a realização de treinamentos mais aprofundados no início do projeto, a criação de *templates* mais detalhados para os casos de teste e a exploração de funcionalidades de automação de testes, visando otimizar ainda mais o processo e reduzir o esforço manual.

Em suma, a aplicação prática de testes com o apoio de ferramentas dedicadas como TestLink e Mantis é crucial para a entrega de software de alta qualidade. As lições aprendidas e os desafios superados neste trabalho servirão de base para aprimorar continuamente as metodologias e práticas de teste em projetos futuros.

10. Referências bibliográficas

Delamaro, M., Jino, M. e Maldonado, J. Introdução ao teste de software, 2007. Elsevier Editora Ltda.

Guerra, L. Organizando suítes de teste com Protractor. Medium, 2017. Disponível em: <<https://medium.com/@luizaguerra/organizando-su%C3%ADtes-de-teste-com-protractor-912c06ee9178>>. Acesso em: 21 jun. 2025.

MANTISBT. *Mantis Bug Tracker Documentation*. Disponível em: <<https://mantisbt.org/documentation.php>> Acesso em: 26 jun. 2025.

TUTORIALSPPOINT. *TestLink Tutorial*. s.d. Disponível em: <https://www.tutorialspoint.com/testlink/testlink_tutorial.pdf>. Acesso em: 26 jun. 2025.