

# **DOCUMENTO DE TESTES**

24/05/2023

### **AELY Development Team**

This is a project for the Software Engineering Laboratory Discipline. In this Project we have in mind to make a virtual game store where we will provide the Keys to release them on some platforms like Steam or Epic Games.

# Sumário

1. Histórico de revisões	. 3
2. Introdução	4
2.1. Objetivos do documento	4
2.2. Escopo Do Produto	4
2.3. Referências	4
3. Testes	6
3.1. Teste de banco de dados	6
3.2. Teste funcional	7
3.3. Teste de ciclo de negócio	8
3.4. Teste da interface do usuário	9
3.5. Teste de performance	9
3.6. Teste de carga	10
3.7. Teste de estresse	10
3.8. Teste de segurança e controle de acesso	11
3.9. Teste de falha/recuperação	12
4. Estratégias de Testes	12
4.1. Teste de Banco de dados	12
4.2. Teste funcional	12
4.3. Teste de ciclo de negócio	13
4.4. Teste da interface do usuário	13
4.5. Teste de performance	13
4.6. Teste de carga	13
4.7. Teste de estresse	14
4.8. Teste de segurança e controle de acesso	14
4.9. Teste de falha/recuperação	14
5. Recursos Necessários	15
5.1. Ferramentas	15

5.2. Recursos Humanos	15
5.3. Recurso do sistema	. 15
6. Execução dos testes	. 16
6.1. Planejamento	16
6.2. Modelo de reportação de erro	. 16

# 1. HISTÓRICO DE REVISÕES

Data	Versão	Descrição	Autor
18/05/23	1.0	Definição inicial do documento de testes	Lucas Claudiano
25/05/23	1.0	Revisão do documento de testes	Lucas Claudiano

### 2. INTRODUÇÃO

#### 2.1. OBJETIVOS DO DOCUMENTO

O propósito deste documento é identificar os componentes de software e requisitos a serem testados, além de descrever as estratégias de teste a serem utilizadas. O plano de testes também fornecerá uma estimativa dos esforços e recursos necessários, e descreve o planejamento dos testes para que sua execução seja acompanhada de forma prática e organizada, visando atingir os objetivos estabelecidos.

#### 2.2. ESCOPO DO PRODUTO

O sistema da AELY GAMES será submetido a testes de unidade, integração, sistema e aceitação. Os testes de unidade e integração abordarão a qualidade funcional, o banco de dados, a interface gráfica e o controle de acesso. Enquanto o teste de unidade avaliará cada componente individualmente, o teste de integração avaliará a integração de todos esses componentes.

Os testes de sistema avaliarão o funcionamento e desempenho do sistema como um todo, verificando seu desempenho. O teste de aceitação consistirá na utilização do sistema final pelo usuário para testar cada requisito implementado, a fim de validar o produto.

### 2.3. REFERÊNCIAS

Este documento é baseado no fluxo de testes do RUP e está diretamente relacionado com o Documento de Requisitos. Além desse documento padrão do processo, este Plano de Testes também toma por base o documento do sistema como um todo, pois neste há definições de várias regras de negócio que regem o sistema como um todo.

### - Documento de Requisitos:

 $\frac{https://docs.google.com/document/d/1v05UL6\_m-co8F4fKiwE1xFLVcmyWs0-x/edit?usp=drive\_left in kande 110353981874825999683&rtpof=true&sd=true$ 

- Rational Unified Process (RUP):

http://pt.wikipedia.org/wiki/IBM Rational Unified Proces

#### 3. TESTES

#### 3.1. TESTE DE BANCO DE DADOS

#### Descrição

Os testes do banco de dados e das regras de negócio devem ser conduzidos de forma independente, sem envolver a interface gráfica. Esses testes estão relacionados às operações básicas de manipulação dos dados no sistema, como inserção, remoção, consulta e atualização.

- Verificar se usuários podem ser cadastrados e removidos do sistema;
- Verificar se jogos podem ser cadastrados e removidos do sistema;
- Verificar se jogos podem ser associados e desassociados aos usuários;
- Verificar se jogos podem ser desativados ou reativados corretamente;
- Verificar se dados dos jogos podem ser atualizados corretamente (código, título, categoria, preço, descrição, Desenvolvedora;
- Verificar se lista de desejo podem ser associados e desassociados aos usuários;
- Verificar se dados do usuário podem ser atualizados corretamente;
- Verificar se categorias dos jogos podem ser adicionadas e removidas do sistema;
- Verificar se categorias de jogos podem ser editadas;
- Verificar se Desenvolvedoras podem ser atualizados corretamente;
- Verificar se hierarquias dos usuários podem ser atualizados corretamente;
- Verificar se jogos podem ser associados ou desassociados do carrinho de usuários;
- Verificar se jogos nas plataformas podem ser gerenciadas (Steam, epic games, etc.)
- Verificar se mensagens podem ser inseridas, removidas entre os usuário;
- Verificar se atividades de gerenciamento podem ser criadas, modificadas, realocadas e removidas (ADM);
- Verificar se o Carrinho dos usuários são atualizados e criados, removidos e desativados :

- Verificar se arquivos(imagens dos jogos) podem ser inseridos, editados e removidos dos repositórios;
- Verificar se a pesquisa de jogos é funcional.

#### 3.2. TESTE FUNCIONAL

#### Descrição

O teste funcional tem como objetivo verificar a aceitação dos dados, o processamento, a resposta a esse processamento e a correta implementação das regras de negócio. Esse tipo de teste é baseado em técnicas de caixa-preta, ou seja, verifica o sistema e seu processo interno por meio da interação com a Interface Gráfica do Usuário (GUI) e pela análise das saídas ou resultados.

- Verificar se o usuário consegue se cadastrar e descadastrar no sistema;
- Verificar se o usuário consegue acessar sua conta através de login e senha;
- Verificar se o usuário consegue alterar suas informações pessoais na sua conta (após o cadastro);
- Verificar se o usuário consegue acompanhar o pedido de um jogo que tenha adquirido;
- Verificar se o usuário gerente do projeto consegue acompanhar o andamento do sistema (vendas e gestão);
- Verificar se o usuário adm pode compartilhar arquivos no repositório online( subir imagens);
- Verificar se um usuário consegue alterar seu status : usuário comum / adm;
- Verificar se o usuário consegue gerenciar sua lista de desejos;
- Verificar se o usuário consegue pesquisar, através do sistema de recomendações, outros usuários que se encaixem em um dado perfil por ele especificado;
- Verificar se o usuário consegue enviar e receber mensagens através do formulário de contato;
- Verificar se um usuário consegue adicionar ou remover um jogo da sua lista de desejo;
- Verificar se um usuário consegue usar sistema de follow para seguir as novidades do site;
- Verificar se o usuário consegue visualizar as notícias divulgadas no site;

- Verificar se o usuário consegue realizar compras na loja virtual;
- Verificar se o usuário consegue dar sugestões para o melhor funcionamento do sistema (área de feedback ou contato);

### 3.3. TESTE DE CICLO DE NEGÓCIO

#### Descrição

O teste de ciclo de negócio é realizado para garantir que os alvos de teste e os processos do módulo funcionem de acordo com os modelos de negócio e os cronogramas exigidos. Os testes funcionais podem ser usados, e é possível simular vários usuários para verificar as regras de negócio.

#### Testes executados no projeto

- Verificar se os campos obrigatórios estão sendo preenchidos no momento em que o cadastro é realizado:
- Verificar se os campos destinados a edição de informações (exemplo: descrição, desenvolvedora, código) estão sendo preenchidos no formato correto;
- Verificar se usuários com tipos de conta diferentes possuem as restrições e direitos inerentes a seu tipo;
- Verificar se o sistema de liberação das keys ocorre segundo os critérios dos requisitos;
- Verificar se jogos desativados aparecem para usuários comuns;

de inscrições e exibem os resultados nas datas previstas;

- Verificar se os usuários presentes nas listas de usuario têm seu acesso restringido em relação aos administradores;
- Verificar o funcionamento do sistema de follow (acompanhamento de algum jogo), ou seja, se os games que contém interesse do usuário realmente funciona;.
- Verificar se quando um usuário coloca um status associado a algum jogo, automaticamente ele fica como "não disponível" para todos.

### 3.4. TESTE DA INTERFACE DO USUÁRIO

#### Descrição

O teste de interface verifica se a interface do usuário oferece acesso adequado às funções do sistema e uma navegação adequada. Além disso, esse teste garante que os objetos dentro da interface do usuário funcionem de acordo com os padrões definidos pelo sistema. Também deve verificar a facilidade com que o software pode ser compreendido e operado pelos usuários.

#### Testes executados no projeto

- Navegar por todos os casos de uso, verificando se cada interface do usuário pode ser facilmente acessada;
- Verificar se todas as funções do site funcionam corretamente;
- Verificar se as interfaces gráficas condizem com o que foi especificado nos respectivos requisitos;
- Verificar se as interfaces gráficas são de fácil uso;
- Verificar se os links contidos nas interfaces apontam para os locais/documentos certos;
- Verificar se todas as palavras e frases das interfaces estão de acordo com as normas sintáticas e gramaticais.

#### 3.5. TESTE DE PERFORMANCE

#### Descrição

O teste de performance mede e avalia o tempo de resposta, o número de transações, usuários e outros requisitos sensíveis ao tempo.

- Verificar o tempo de resposta da troca de informações entre o servidor e os terminais é aceitável:
- Verificar o tempo de resposta ao consultar/inserir/remover/atualizar no banco de

#### dados é aceitável:

- Verificar se o tempo de resposta para operações que envolvam dados multimídia não ultrapassa 30 segundos (ex: adicionar imagens).

#### 3.6. TESTE DE CARGA

#### Descrição

O teste de carga submete o sistema a diferentes níveis de carga de trabalho para medir e avaliar o desempenho e a capacidade de continuar funcionando adequadamente sob cargas de trabalho diferentes e grandes quantidades de dados. Esse teste determina se os limites que causam falhas no software são alcançados. Também avalia as características de desempenho, como tempos de resposta, taxas de transação e outros casos sensíveis ao tempo, e identifica a carga ou o volume máximo persistente que o sistema pode suportar por um determinado período.

#### Testes executados no projeto

- Acompanhar o comportamento do sistema com 10 usuários;
- Acompanhar o comportamento do sistema com 20 usuários;
- Acompanhar o comportamento do sistema com 30 usuários;
- Acompanhar o comportamento do sistema com 100 usuários.

#### 3.7. TESTE DE ESTRESSE

#### Descrição

O teste de estresse é um tipo de teste de desempenho realizado para entender o comportamento do sistema durante condições extremas ou fora das tolerâncias esperadas. Geralmente envolve recursos limitados ou a competição por recursos. Recursos limitados revelam falhas que não são aparentes em condições normais.

#### Testes executados no projeto

o Verificar o comportamento do sistema ao serem inseridos e/ou modificados vários

dados, fazendo diversas operações na base de dados num período de tempo pequeno;

- o Verificar o funcionamento do sistema quando usado num computador com baixa capacidade de memória principal e processamento;
- o Verificar o desempenho do sistema usando o maior número de produtos do mesmo simultaneamente.

#### 3.8. TESTE DE SEGURANÇA E CONTROLE DE ACESSO

#### Descrição

O teste de controle de segurança e acesso concentra-se em duas áreas principais de segurança:

- Segurança em nível de aplicação, incluindo o acesso a dados ou funções de negócio.
- Segurança em nível de sistema, incluindo o acesso ao sistema localmente ou remotamente.
- Segurança em nível de aplicação garante que os usuários sejam restritos a funções específicas ou casos de uso, de acordo com a segurança desejada, ou limitados aos dados disponibilizados a eles conforme o perfil definido.
- Segurança em nível de sistema garante que apenas os usuários com permissão de acesso sejam capazes de acessar o sistema.

- Verificar se usuários comuns não podem inserir, remover ou modificar jogos;
- Verificar se usuários comuns não podem inserir, remover ou modificar jogos;
- Verificar se usuários comuns não podem inserir, remover ou modificar compras e históricos de compras;
- Verificar se usuários comuns não podem inserir, remover ou modificar compras e históricos de outros usuários;
- Verificar se usuários comuns não podem inserir, remover ou modificar descrição e dados de jogos;
- Verificar se usuários comuns não podem inserir, remover ou desativar produtos do site;
- Verificar se usuários comuns não podem inserir, remover ou modificar listas de desejos de outros usuários:

- Verificar se administradores estão restritos a realizar alguma operação dentro do sistema.

### 3.9. TESTE DE FALHA/RECUPERAÇÃO

#### Descrição

O teste de falha e recuperação garante que o sistema seja capaz de recuperar com sucesso os dados após uma falha no hardware, software ou rede, quando ocorrer perda ou comprometimento dos dados.

#### Testes executados no projeto

- Provocar uma falha crítica no sistema ao realizar operações não permitidas e verificar a integridade das informações após a falha.

### **4.ESTRATÉGIAS DE TESTES**

#### 4.1. TESTE DE BANCO DE DADOS

OBJETIVO DO TESTE	O objetivo é garantir a integridade dos dados e o correto funcionamento dos métodos e processos de acesso ao banco de dados.
TÉCNICA	Chamar todos os métodos e processos de acesso ao banco de dados, inserindo ou solicitando dados válidos e inválidos. Verificar se os dados válidos foram inseridos corretamente. Verificar se, ao adicionar dados inválidos, a mensagem de erro apropriada é retornada ao usuário e o dado não é inserido. Verificar se todos os eventos do banco de dados ocorrem conforme o esperado.
CRITÉRIO DE FINALIZAÇÃO	Todos os métodos e processos de acesso ao banco de dados funcionam como esperado, e os dados são mantidos consistentes.
CONSIDERAÇÕES ESPECIAIS	O teste requer um ambiente de desenvolvimento de SGBD (Sistema de Gerenciamento de Banco de Dados) para inserir, modificar ou remover os dados diretamente no banco de dados. As invocações dos testes serão realizadas manualmente.

### **4.2 TESTE FUNCIONAL**

OBJETIVO DO TESTE	Assegurar a correta funcionalidade de cada caso de uso testado.
TÉCNICA	Realizar a execução de cada caso de uso, percorrendo seus fluxos, utilizando tanto dados válidos (para verificar se o resultado esperado ocorre) quanto dados inválidos (para verificar se são retornadas mensagens de erro adequadas).
CRITÉRIO DE FINALIZAÇÃO	Garantir a execução de todos os testes planejados e tratar os erros identificados.
CONSIDERAÇÕES ESPECIAIS	Não há considerações especiais.

### **4.3 TESTE DE CICLO DE NEGÓCIO**

OBJETIVO DO TESTE	Verificar a correta implementação das regras de negócio, a fim de manter a integridade do sistema.
TÉCNICA	Executar os casos de uso, inserindo dados válidos para verificar o funcionamento adequado, e inserindo dados inválidos para verificar se são retornadas mensagens de erro apropriadas. Avaliar se as regras de negócio propostas pelo sistema estão sendo seguidas.
CRITÉRIO DE FINALIZAÇÃO	Certificar-se de que todas as informações inseridas estão de acordo com o padrão especificado, sem violações das regras de negócio.
CONSIDERAÇÕES ESPECIAIS	Não há considerações especiais.

### 4.4. TESTE DA INTERFACE DO USUÁRIO

0	BJETIVO DO TESTE	Garantir a correta localização dos objetos e interface, assim como o acesso adequado às funções e requisitos do negócio. Verificar as funcionalidades da interface do usuário e sua facilidade de uso.
	TÉCNICA	Realizar testes da interface com vários usuários para avaliar a navegabilidade em cada tela, além de verificar o nível de usabilidade do site. Testar cada seção da interface para garantir seu funcionamento correto e consistente de acordo com suas funcões.
	CRITÉRIO DE	Verificar se o usuário consegue utilizar o site facilmente, e se todas as janelas funcionam corretamente e de forma consistente.
	CONCLUSÃO CONSIDERAÇÕES	
	ESPECIAIS	A navegação em algumas janelas do site é restrita a determinados tipos de usuários (usuário comum, MENU administrador).

### 4.5. TESTE DE PERFORMANCE

OBJETIVO DO TESTE	Analisar o tempo de resposta do sistema (em diferentes condições de hardware, software e rede) para obter, atualizar e inserir dados, além de executar as funcionalidades do sistema.
TÉCNICA	Realizar testes da interface com vários usuários para avaliar a navegabilidade em cada tela, além de verificar o nível de usabilidade do site. Testar cada seção da interface para garantir seu funcionamento correto e consistente de acordo com suas funções.
CRITÉRIO DE CONCLUSÃO	alizar operações fornecidas pelo sistema com um ou vários usuários simultaneamente. Realizar operações fornecidas pelo sistema em diferentes condições de rede, hardware e software.
CONSIDERAÇÕES ESPECIAIS	Garantir que todas as operações sejam realizadas dentro de intervalos de tempo aceitáveis.

### 4.6. TESTE DE CARGA

OBJETIVO DO TESTE	Verificar o funcionamento do sistema sob diferentes condições de carga de trabalho.
TÉCNICA	Vários usuários realizarão testes simultâneos de cada função do sistema.
CRITÉRIO DE CONCLUSÃO	Certificar-se de que o sistema funcione corretamente e as operações sejam realizadas dentro de tempos aceitáveis, de acordo com as condições executadas.
CONSIDERAÇÕES ESPECIAIS	O teste de carga deve ser realizado em um computador sem a execução simultânea de outros programas para obter medidas mais precisas.

### **4.7. TESTE DE ESTRESSE**

OBJETIVO DO TESTE	Observar o comportamento do sistema em situações de alta demanda ou em computadores com recursos limitados.
TÉCNICA	Realizar diversos testes de função simultaneamente. Utilizar vários programas ao mesmo tempo enquanto o sistema está em uso.
CRITÉRIO DE CONCLUSÃO	Verificar se o sistema funciona corretamente, mesmo que com processamento lento, e se os tempos de resposta são aceitáveis dadas as condições.
CONSIDERAÇÕES ESPECIAIS	Não há considerações especiais.

# 4.8. TESTE DE SEGURANÇA E CONTROLE DE ACESSO

OBJETIVO DO TESTE	Verificar se as restrições especificadas para os tipos de usuários estão sendo devidamente cumpridas.
TÉCNICA	Identificar e listar cada tipo de usuário e suas permissões correspondentes. Testar as funções permitidas para cada usuário e verificar se estão corretas. Testar as funções não permitidas para cada usuário e verificar se a operação é interrompida e se uma mensagem de erro adequada é retornada ao usuário.
CRITÉRIO DE CONCLUSÃO	Certificar-se de que as operações permitidas para cada usuário estejam disponíveis e de que o sistema se comporte conforme o esperado diante de tentativas de realização de operações não permitidas para determinados usuários.
CONSIDERAÇÕES ESPECIAIS	Não há considerações especiais.

# 4.9. TESTE DE FALHA/RECUPERAÇÃO

OBJETIVO DO TESTE	Observar o comportamento do sistema e sua recuperação diante de falhas, garantindo que a execução e as funções do sistema não sejam prejudicadas.
TÉCNICA	Forçar o sistema a falhar em diversas funcionalidades, inserindo dados que violem as restrições ou realizando operações ilegais. Simular a queda na rede quando os usuários estão ativos no sistema.
CRITÉRIO DE CONCLUSÃO	Verificar se o sistema se recupera corretamente das falhas e se nenhuma função ou execução do sistema é prejudicada.
CONSIDERAÇÕES ESPECIAIS	Não há considerações especiais.

### 5. RECURSOS NECESSÁRIOS

#### **5.1. FERRAMENTAS**

- Google Docs
- Excel
- QaSe.io
- PHPMyAdmin 5.2.1
- XAMPP 8.2.4-0

#### **5.2. RECURSOS HUMANOS**

Gerente de testes: Lucas Claudiano

Equipe de testes:

- André Augusto de Souza Soares
- Erick Goes Sampaio
- Pedro Henrique Garcia Rocha
- Yohan Gabriel E.Q dos Santos Ferreira Aquino

#### **5.3. RECURSO DO SISTEMA**

- Sistema de Gerenciamento de Bancos de Dados: MySQL
- Sistema de acesso MySQL pela Internet: PHPMyAdmin.
- Terminais de usuários: Dispositivos Conectados à Internet

### 6. EXECUÇÃO DOS TESTES

À medida que os desenvolvedores da AELY Games implementam os diferentes componentes do sistema, eles também realizam testes preliminares para avaliar o funcionamento de cada um deles. Ao finalizar suas tarefas, eles entregam o trabalho para a equipe de testes, que irá validar e realizar os testes planejados ao longo do período. Essa abordagem permite que o desenvolvimento dos casos de uso para a próxima iteração aconteça simultaneamente aos testes. Semanalmente, são criadas builds do sistema, buscando integrar os módulos e proporcionar maior consistência e precisão ao sistema.

Se durante os testes forem identificados quaisquer erros, eles serão comunicados aos desenvolvedores conforme os critérios estabelecidos pela equipe. Em seguida, os desenvolvedores procederão com a correção dos erros e submeterão novamente os componentes para serem testados, assegurando que cada parte do sistema seja cuidadosamente examinada até que todos os resultados desejados sejam alcançados.

### 6.2. MODELO DE REPORTAÇÃO DE ERRO

Na descrição do erro no QaSe.io, deve-se preencher os campos da seguinte forma:

Test Case: [BUG] < Nome do erro, ex: Falha no Login>

**Description**: Neste caso, é importante identificar o método de teste utilizado, permitindo que o desenvolvedor saiba os passos exatos que foram seguidos para reproduzir o erro e, assim, poder reproduzi-lo novamente.

**Pre-conditions/ Pos-conditions**: Qualquer observação relevante, como suspeitas sobre a origem do erro, deve ser registrada - porém, isso só deve ser feito quando houver uma suspeita válida.

Properties: Severity / Blocker / Status / Actual / Priority / Behavior / Type / Tags / Layer