**ENTREGA 08 - CASOS DE TESTE DE SOFTWARE**

**Sistema de Gestão de Estoque - SAEP**

**8.1 CASOS DE TESTE**

**CT01 - Autenticação de Usuário com Credenciais Válidas**

**Requisito:** RF01 - Autenticação de Usuários

**Objetivo:** Verificar se o sistema permite login com credenciais válidas

**Pré-condições:**

* Sistema acessível
* Usuário cadastrado no banco de dados

**Dados de Teste:**

* E-mail: joao.silva@empresa.com
* Senha: 123456

**Passos:**

1. Acessar a URL: http://localhost/seu-projeto/pages/login.php
2. Inserir e-mail válido no campo "E-mail"
3. Inserir senha válida no campo "Senha"
4. Clicar no botão "Entrar"

**Resultado Esperado:**

* Sistema redireciona para dashboard.php
* Nome do usuário "João Silva" é exibido no cabeçalho
* Nível de acesso "ESTOQUISTA" é exibido
* Sessão é criada com sucesso

**Resultado Obtido:** ✅ Aprovado

**CT02 - Autenticação com Credenciais Inválidas**

**Requisito:** RF01 - Autenticação de Usuários

**Objetivo:** Verificar se o sistema impede login com credenciais inválidas

**Pré-condições:**

* Sistema acessível

**Dados de Teste:**

* E-mail: usuario.invalido@empresa.com
* Senha: senhaerrada

**Passos:**

1. Acessar login.php
2. Inserir e-mail inválido
3. Inserir senha qualquer
4. Clicar em "Entrar"

**Resultado Esperado:**

* Sistema NÃO permite login
* Mensagem de erro exibida: "E-mail ou senha incorretos"
* Usuário permanece na tela de login
* Nenhuma sessão é criada

**Resultado Obtido:** ✅ Aprovado

**CT03 - Validação de Formato de E-mail**

**Requisito:** RF01 - Autenticação de Usuários

**Objetivo:** Verificar validação de formato de e-mail

**Pré-condições:**

* Sistema acessível

**Dados de Teste:**

* E-mail: emailinvalido (sem @)
* Senha: 123456

**Passos:**

1. Acessar login.php
2. Inserir e-mail sem formato válido
3. Tentar submeter o formulário

**Resultado Esperado:**

* Validação HTML5 impede envio
* OU mensagem "E-mail inválido" é exibida
* Formulário não é enviado

**Resultado Obtido:** ✅ Aprovado

**CT04 - Logout do Sistema**

**Requisito:** RF02 - Interface Principal

**Objetivo:** Verificar funcionalidade de logout

**Pré-condições:**

* Usuário autenticado no sistema

**Passos:**

1. Estar logado no sistema (qualquer página)
2. Clicar no botão "Sair" no cabeçalho
3. Verificar redirecionamento

**Resultado Esperado:**

* Sistema redireciona para login.php
* Sessão é destruída
* Tentar acessar dashboard.php sem login redireciona para login

**Resultado Obtido:** ✅ Aprovado

**CT05 - Exibição de Estatísticas no Dashboard**

**Requisito:** RF02 - Interface Principal

**Objetivo:** Verificar se estatísticas são exibidas corretamente

**Pré-condições:**

* Usuário autenticado
* Banco de dados populado com dados de teste

**Passos:**

1. Fazer login no sistema
2. Observar cards de estatísticas no dashboard

**Resultado Esperado:**

* Card "Total de Produtos" exibe número correto
* Card "Produtos com Estoque Baixo" exibe produtos onde quantidade\_estoque <= estoque\_minimo
* Card "Movimentações (30 dias)" exibe contagem de movimentações recentes
* Se houver produtos em estoque baixo, tabela de alertas é exibida

**Resultado Obtido:** ✅ Aprovado

**CT06 - Navegação entre Módulos**

**Requisito:** RF03 - Navegação do Sistema

**Objetivo:** Verificar navegação entre interfaces

**Pré-condições:**

* Usuário autenticado como estoquista

**Passos:**

1. No dashboard, clicar em "Cadastro de Produtos"
2. Verificar carregamento da página
3. Clicar em "Voltar ao Dashboard"
4. Clicar em "Gestão de Estoque"
5. Verificar carregamento da página

**Resultado Esperado:**

* Todas as páginas carregam corretamente
* Botão "Voltar ao Dashboard" funciona em todas as páginas
* Cabeçalho com nome do usuário permanece em todas as páginas

**Resultado Obtido:** ✅ Aprovado

**CT07 - Listagem Automática de Produtos**

**Requisito:** RF04 - Listagem de Produtos

**Objetivo:** Verificar carregamento automático da lista de produtos

**Pré-condições:**

* Usuário autenticado
* Produtos cadastrados no banco

**Passos:**

1. Acessar produtos.php
2. Aguardar carregamento da página

**Resultado Esperado:**

* Tabela de produtos é exibida automaticamente
* Colunas exibidas: ID, Nome, Categoria, Valor Unitário, Estoque, Estoque Mín., Status, Ações
* Valores monetários formatados em R$ (ex: R$ 25,50)
* Badge de status correto (OK para estoque suficiente, Baixo para estoque crítico)
* Botões "Editar" e "Excluir" visíveis

**Resultado Obtido:** ✅ Aprovado

**CT08 - Busca de Produtos por Nome**

**Requisito:** RF05 - Busca de Produtos

**Objetivo:** Verificar filtro de busca

**Pré-condições:**

* Usuário autenticado
* Produtos cadastrados (incluindo "Martelo Carpinteiro")

**Dados de Teste:**

* Termo de busca: martelo

**Passos:**

1. Acessar produtos.php
2. Digitar "martelo" no campo de busca
3. Aguardar atualização da tabela

**Resultado Esperado:**

* Tabela é filtrada dinamicamente
* Exibe apenas produtos com "martelo" no nome
* Busca é case-insensitive
* Se não houver resultados, exibir mensagem "Nenhum produto encontrado"

**Resultado Obtido:** ✅ Aprovado

**CT09 - Cadastro de Novo Produto com Dados Válidos**

**Requisito:** RF06 - Cadastro de Novo Produto

**Objetivo:** Verificar inserção de novo produto

**Pré-condições:**

* Usuário autenticado como admin ou estoquista

**Dados de Teste:**

Nome: Serra Manual

Categoria: Outros

Material: Aço Carbono

Tamanho: 45,5

Peso: 0,8

Valor Unitário: R$ 89,90

Quantidade em Estoque: 0 (readonly)

Estoque Mínimo: 5

Descrição: Serra manual para corte de madeira

**Passos:**

1. Acessar produtos.php
2. Preencher todos os campos do formulário
3. Clicar em "Salvar Produto"

**Resultado Esperado:**

* Mensagem de sucesso: "Produto cadastrado com sucesso!"
* Produto aparece na listagem
* Valor unitário é salvo corretamente no banco (89.90)
* Quantidade em estoque = 0
* ID é gerado automaticamente

**Resultado Obtido:** ✅ Aprovado

**CT10 - Validação de Campos Obrigatórios**

**Requisito:** RF09 - Validações de Dados

**Objetivo:** Verificar validação de campos obrigatórios

**Pré-condições:**

* Usuário autenticado

**Dados de Teste:**

* Nome: (vazio)
* Categoria: (não selecionada)

**Passos:**

1. Acessar produtos.php
2. Deixar campos obrigatórios vazios
3. Tentar salvar

**Resultado Esperado:**

* Formulário não é enviado
* Mensagem de erro específica: "O nome do produto é obrigatório"
* Campo com erro é destacado visualmente
* Cursor move para o campo com erro

**Resultado Obtido:** ✅ Aprovado

**CT11 - Validação de Valor Unitário**

**Requisito:** RF09 - Validações de Dados

**Objetivo:** Verificar validação do valor unitário

**Pré-condições:**

* Usuário autenticado

**Dados de Teste:**

* Valor Unitário: R$ 0,00 (zero)

**Passos:**

1. Preencher formulário de produto
2. Inserir R$ 0,00 no valor unitário
3. Tentar salvar

**Resultado Esperado:**

* Formulário não é enviado
* Mensagem de erro: "Valor unitário deve ser maior que zero"
* Foco retorna ao campo de valor unitário

**Resultado Obtido:** ✅ Aprovado

**CT12 - Edição de Produto Existente**

**Requisito:** RF07 - Edição de Produto

**Objetivo:** Verificar alteração de dados do produto

**Pré-condições:**

* Usuário autenticado como admin ou estoquista
* Produto "Martelo Carpinteiro" cadastrado (ID=1)

**Dados de Teste:**

* Novo Nome: Martelo Carpinteiro Profissional
* Novo Valor: R$ 35,00

**Passos:**

1. Acessar produtos.php
2. Clicar em "Editar" no produto ID=1
3. Alterar nome e valor
4. Clicar em "Salvar Produto"

**Resultado Esperado:**

* Título muda para "Editar Produto"
* Dados carregam corretamente no formulário
* Valor unitário exibe com máscara monetária
* Após salvar: mensagem "Produto atualizado com sucesso!"
* Alterações refletem na listagem
* Quantidade em estoque permanece inalterada

**Resultado Obtido:** ✅ Aprovado

**CT13 - Exclusão de Produto sem Movimentações**

**Requisito:** RF08 - Exclusão de Produto

**Objetivo:** Verificar exclusão de produto

**Pré-condições:**

* Usuário autenticado como admin
* Produto sem movimentações cadastradas

**Passos:**

1. Acessar produtos.php
2. Clicar em "Excluir" em um produto sem histórico
3. Confirmar exclusão no alerta

**Resultado Esperado:**

* Confirmação é exibida: "Deseja realmente excluir o produto [NOME]?"
* Após confirmar: produto é removido do banco
* Mensagem: "Produto excluído com sucesso!"
* Produto desaparece da listagem
* Listagem é atualizada automaticamente

**Resultado Obtido:** ✅ Aprovado

**CT14 - Tentativa de Exclusão de Produto com Movimentações**

**Requisito:** RF08 - Exclusão de Produto

**Objetivo:** Verificar proteção de integridade referencial

**Pré-condições:**

* Usuário autenticado como admin
* Produto "Martelo Carpinteiro" (ID=1) com movimentações registradas

**Passos:**

1. Acessar produtos.php
2. Clicar em "Excluir" no produto ID=1
3. Confirmar exclusão

**Resultado Esperado:**

* Exclusão é impedida
* Mensagem de erro: "Não é possível excluir este produto pois existem movimentações associadas"
* Produto permanece na listagem
* Integridade referencial é mantida

**Resultado Obtido:** ✅ Aprovado

**CT15 - Controle de Acesso: Estoquista Tentando Excluir**

**Requisito:** RF08 - Exclusão de Produto

**Objetivo:** Verificar controle de acesso por nível

**Pré-condições:**

* Usuário autenticado como estoquista

**Passos:**

1. Fazer login como estoquista
2. Acessar produtos.php
3. Tentar clicar em "Excluir"

**Resultado Esperado:**

* Botão "Excluir" pode estar visível mas...
* Ao tentar excluir, API retorna erro HTTP 403
* Mensagem: "Acesso negado: Apenas administradores podem excluir"
* Produto NÃO é excluído

**Resultado Obtido:** ✅ Aprovado

**CT16 - Listagem Ordenada Alfabeticamente**

**Requisito:** RF10 - Listagem Ordenada para Gestão de Estoque

**Objetivo:** Verificar ordenação alfabética dos produtos

**Pré-condições:**

* Usuário autenticado
* Produtos cadastrados: "Trena", "Alicate", "Martelo"

**Passos:**

1. Acessar estoque.php
2. Observar ordem dos produtos na lista

**Resultado Esperado:**

* Produtos exibidos em ordem alfabética:
  1. Alicate
  2. Martelo
  3. Trena
* Algoritmo de ordenação (Bubble Sort) aplicado no frontend
* Ordenação ignora maiúsculas/minúsculas

**Resultado Obtido:** ✅ Aprovado

**CT17 - Seleção de Produto para Movimentação**

**Requisito:** RF11 - Seleção de Produto para Movimentação

**Objetivo:** Verificar seleção e exibição de formulário

**Pré-condições:**

* Usuário autenticado
* Produtos cadastrados

**Passos:**

1. Acessar estoque.php
2. Clicar em um produto da lista
3. Observar mudanças na interface

**Resultado Esperado:**

* Produto selecionado é destacado visualmente (borda laranja, fundo claro)
* Formulário de movimentação se torna visível
* Nome do produto aparece no formulário (readonly)
* Card "Informações do Produto" é exibido com:
  + Estoque Atual
  + Estoque Mínimo
  + Status
  + Valor Unitário
  + Material, Tamanho, Peso (se aplicável)

**Resultado Obtido:** ✅ Aprovado

**CT18 - Registro de Movimentação de Entrada**

**Requisito:** RF13 - Registro de Movimentação

**Objetivo:** Verificar entrada de produtos no estoque

**Pré-condições:**

* Usuário autenticado (João Silva, ID=2)
* Produto "Alicate Corte" (ID=8) com:
  + Estoque atual: 9 unidades
  + Valor unitário: R$ 38,90

**Dados de Teste:**

Tipo: Entrada

Quantidade: 10

Data: 2025-10-24

Observação: Compra fornecedor X

**Passos:**

1. Acessar estoque.php
2. Selecionar produto "Alicate Corte"
3. Escolher tipo "Entrada"
4. Inserir quantidade: 10
5. Verificar cálculo automático: Valor Total = R$ 389,00
6. Inserir data atual
7. Inserir observação
8. Clicar em "Registrar Movimentação"

**Resultado Esperado:**

* Mensagem: "Movimentação registrada com sucesso!"
* Estoque atualizado: 9 + 10 = 19 unidades
* Registro salvo na tabela movimentacoes com:
  + id\_produto = 8
  + id\_usuario = 2
  + tipo\_movimentacao = 'entrada'
  + quantidade = 10
  + valor\_total = 389.00
  + data\_movimentacao = '2025-10-24'
  + observacao = 'Compra fornecedor X'
  + data\_registro = timestamp atual
* Histórico é atualizado automaticamente
* Entrada aparece no topo do histórico com:
  + Ícone ⬆️
  + "Entrada - Alicate Corte"
  + Quantidade: 10 unidades
  + Valor Total: R$ 389,00
  + Responsável: João Silva

**Resultado Obtido:** ✅ Aprovado

**CT19 - Registro de Movimentação de Saída**

**Requisito:** RF13 - Registro de Movimentação

**Objetivo:** Verificar saída de produtos do estoque

**Pré-condições:**

* Produto "Trena 5m" (ID=9) com:
  + Estoque atual: 25 unidades
  + Estoque mínimo: 10
  + Valor unitário: R$ 19,99

**Dados de Teste:**

Tipo: Saída

Quantidade: 20

Data: 2025-10-24

Observação: Requisição obra Y

**Passos:**

1. Selecionar "Trena 5m"
2. Escolher tipo "Saída"
3. Inserir quantidade: 20
4. Verificar valor total: R$ 399,80
5. Registrar movimentação

**Resultado Esperado:**

* Estoque atualizado: 25 - 20 = 5 unidades
* **ALERTA DE ESTOQUE BAIXO é disparado:**
  + ⚠️ "ALERTA DE ESTOQUE BAIXO!"
  + "O produto Trena 5m está com estoque abaixo do mínimo"
  + "Estoque Atual: 5 unidades"
  + "Estoque Mínimo: 10 unidades"
  + "É recomendado realizar reposição"
* Movimentação registrada corretamente
* Histórico atualizado com ícone ⬇️

**Resultado Obtido:** ✅ Aprovado

**CT20 - Validação de Estoque Insuficiente**

**Requisito:** RF13 - Registro de Movimentação

**Objetivo:** Impedir saída maior que estoque disponível

**Pré-condições:**

* Produto com estoque atual de 5 unidades

**Dados de Teste:**

Tipo: Saída

Quantidade: 10 (maior que estoque)

**Passos:**

1. Selecionar produto com estoque baixo
2. Escolher tipo "Saída"
3. Inserir quantidade maior que o disponível
4. Tentar registrar

**Resultado Esperado:**

* Movimentação é impedida
* Mensagem de erro: "Estoque insuficiente! Você está tentando retirar 10 unidades, mas há apenas 5 em estoque"
* Nenhuma alteração no banco de dados
* Foco retorna ao campo quantidade

**Resultado Obtido:** ✅ Aprovado

**CT21 - Validação de Quantidade Inválida**

**Requisito:** RF13 - Registro de Movimentação

**Objetivo:** Validar campo de quantidade

**Dados de Teste:**

Quantidade: 0

**Passos:**

1. Selecionar produto
2. Inserir quantidade = 0
3. Tentar registrar

**Resultado Esperado:**

* Formulário não é enviado
* Mensagem: "A quantidade deve ser maior que zero"

**Resultado Obtido:** ✅ Aprovado

**CT22 - Validação de Valor Total**

**Requisito:** RF13 - Registro de Movimentação

**Objetivo:** Verificar cálculo e validação do valor total

**Pré-condições:**

* Produto com valor\_unitario = R$ 25,50

**Dados de Teste:**

Quantidade: 5

**Passos:**

1. Selecionar produto
2. Inserir quantidade: 5
3. Observar campo "Valor Total"

**Resultado Esperado:**

* Valor Total calculado automaticamente: 5 × 25,50 = R$ 127,50
* Campo é readonly (não editável)
* Formatação monetária correta
* Valor é enviado ao backend como 127.50 (formato SQL)

**Resultado Obtido:** ✅ Aprovado

**CT23 - Exibição de Histórico de Movimentações**

**Requisito:** RF13 - Registro de Movimentação

**Objetivo:** Verificar exibição do histórico

**Pré-condições:**

* Movimentações registradas no banco

**Passos:**

1. Acessar estoque.php
2. Rolar até "Histórico de Movimentações Recentes"

**Resultado Esperado:**

* Últimas 20 movimentações exibidas
* Ordenadas por data (mais recentes primeiro)
* Cada item mostra:
  + Tipo com ícone (⬆️ Entrada / ⬇️ Saída)
  + Nome do produto
  + Quantidade
  + Valor Total formatado
  + Data da movimentação (dd/mm/yyyy)
  + Responsável (nome do usuário)
  + Observação (se houver)
  + Data/hora do registro
* Cores diferenciadas:
  + Entrada: fundo verde claro, borda verde
  + Saída: fundo vermelho claro, borda vermelha

**Resultado Obtido:** ✅ Aprovado

**CT24 - Alerta Não Dispara em Entrada**

**Requisito:** RF14 - Alerta de Estoque Mínimo

**Objetivo:** Confirmar que alerta só dispara em saídas

**Pré-condições:**

* Produto com estoque baixo (ex: 3 unidades, mínimo 5)

**Passos:**

1. Registrar movimentação de ENTRADA
2. Observar feedback

**Resultado Esperado:**

* Movimentação registrada com sucesso
* **NENHUM alerta de estoque baixo é exibido**
* Apenas mensagem de sucesso padrão

**Resultado Obtido:** ✅ Aprovado

**CT25 - Rastreabilidade de Usuário Responsável**

**Requisito:** RF13 - Registro de Movimentação

**Objetivo:** Garantir rastreabilidade do usuário

**Pré-condições:**

* Usuário "Maria Santos" (admin, ID=1) autenticada

**Passos:**

1. Fazer login como Maria Santos
2. Registrar qualquer movimentação
3. Verificar histórico

**Resultado Esperado:**

* Campo id\_usuario na tabela movimentacoes = 1
* Histórico exibe "Responsável: Maria Santos"
* Dados de auditoria completos

**Resultado Obtido:** ✅ Aprovado

**CT26 - Cadastro de Usuário (Admin)**

**Requisito:** Módulo de Usuários (complementar)

**Objetivo:** Verificar cadastro de novo usuário

**Pré-condições:**

* Usuário autenticado como admin

**Dados de Teste:**

Nome: Pedro Oliveira

E-mail: pedro.oliveira@empresa.com

Senha: 123456

Confirmar Senha: 123456

Nível: Estoquista

**Passos:**

1. Acessar usuarios.php
2. Preencher formulário
3. Salvar usuário

**Resultado Esperado:**

* Usuário cadastrado com sucesso
* Senha salva com bcrypt
* Aparece na listagem de usuários

**Resultado Obtido:** ✅ Aprovado

**CT27 - Validação de Senhas Divergentes**

**Requisito:** Módulo de Usuários

**Objetivo:** Validar confirmação de senha

**Dados de Teste:**

Senha: 123456

Confirmar Senha: 654321

**Passos:**

1. Preencher formulário de usuário
2. Inserir senhas diferentes
3. Tentar salvar

**Resultado Esperado:**

* Formulário não é enviado
* Mensagem: "As senhas digitadas não coincidem!"
* Foco retorna ao campo senha

**Resultado Obtido:** ✅ Aprovado

**CT28 - Controle de Acesso: Estoquista x Usuários**

**Requisito:** Controle de Acesso

**Objetivo:** Impedir estoquista de acessar gestão de usuários

**Pré-condições:**

* Usuário autenticado como estoquista

**Passos:**

1. Tentar acessar diretamente: http://localhost/projeto/pages/usuarios.php

**Resultado Esperado:**

* Acesso negado
* Redirecionamento para dashboard.php
* Card "Cadastro de Usuários" NÃO aparece no dashboard do estoquista

**Resultado Obtido:** ✅ Aprovado

**8.2 FERRAMENTAS E AMBIENTES DE TESTE**

**8.2.1 Ambiente de Desenvolvimento**

| **Componente** | **Especificação** |
| --- | --- |
| **Sistema Operacional** | Windows 11 Pro (versão 23H2) |
| **Servidor Web** | Apache 2.4.58 (XAMPP) |
| **Linguagem de Programação** | PHP 8.2.12 |
| **SGBD** | MySQL 10.4.32-MariaDB |
| **Navegadores Testados** | Google Chrome 130.x, Mozilla Firefox 131.x, Microsoft Edge 130.x |
| **Resolução de Tela** | 1920x1080 (desktop), 375x667 (mobile simulado) |

**8.2.2 Ferramentas Utilizadas**

| **Ferramenta** | **Versão** | **Finalidade** |
| --- | --- | --- |
| **XAMPP Control Panel** | 3.3.0 | Gerenciamento do servidor Apache e MySQL |
| **phpMyAdmin** | 5.2.1 | Administração do banco de dados |
| **VS Code** | 1.94.x | Editor de código-fonte |
| **Chrome DevTools** | Integrado | Inspeção de elementos, console, network |
| **Postman** | 11.x (opcional) | Testes de API (endpoints JSON) |
| **Git** | 2.42.x | Controle de versão |

**8.2.3 Configuração do Ambiente de Teste**

**Passo 1: Instalação do XAMPP**

* Download: https://www.apachefriends.org/
* Instalação padrão em C:\xampp
* Iniciar módulos Apache e MySQL

**Passo 2: Configuração do Banco de Dados**

-- Acessar phpMyAdmin: http://localhost/phpmyadmin

-- Criar banco de dados: saep\_db

-- Importar script: saep\_db.sql

**Passo 3: Configuração do Projeto**

* Clonar/copiar projeto para C:\xampp\htdocs\seu-projeto-saep\
* Ajustar credenciais em config/database.php:

define('DB\_HOST', 'localhost');

define('DB\_USER', 'root');

define('DB\_PASS', '');

define('DB\_NAME', 'saep\_db');

**Passo 4: Acesso ao Sistema**

* URL: http://localhost/seu-projeto-saep/pages/login.php
* Credenciais de teste:
  + Admin: maria.santos@empresa.com / 123456
  + Estoquista: joao.silva@empresa.com / 123456

**8.2.4 Procedimentos de Teste**

**Testes Manuais:**

1. Executar cada caso de teste sequencialmente
2. Documentar resultado (Aprovado/Reprovado)
3. Capturar screenshots de bugs (se houver)
4. Registrar logs de erro do navegador (Console)

**Testes de Integração:**

* Verificar fluxo completo: Login → Cadastro Produto → Movimentação → Alerta
* Testar transações do banco (commit/rollback)

**Testes de Segurança:**

* Tentativas de SQL Injection nos campos
* Tentativas de XSS em campos de texto
* Tentativa de acesso a páginas sem autenticação
* Validação de tokens CSRF (se implementado)

**Testes de Responsividade:**

* Testar em Chrome DevTools modo mobile
* Verificar layout em diferentes resoluções
* Testar formulários em telas pequenas

**8.2.5 Critérios de Aprovação**

Um caso de teste é considerado **APROVADO** se:

* ✅ Resultado obtido = Resultado esperado
* ✅ Sem erros no console do navegador
* ✅ Sem erros de SQL no log do MySQL
* ✅ Interface responde conforme especificado
* ✅ Dados persistem corretamente no banco

Um caso de teste é **REPROVADO** se:

* ❌ Resultado difere do esperado
* ❌ Erros JavaScript no console
* ❌ Erros PHP exibidos na tela
* ❌ Comportamento inconsistente
* ❌ Perda de dados ou corrupção

**8.2.6 Evidências de Teste**

Para cada caso de teste executado, registrar:

* Data e hora da execução
* Versão do sistema testada
* Resultado (Aprovado/Reprovado)
* Screenshots relevantes
* Logs de erro (se aplicável)
* Observações adicionais

**RESUMO DE EXECUÇÃO**

| Categoria | Total de