

**Sistema de Gestão de Folha de Pagamento – PagueAqui**

**PUC Minas**

**Sistema de Gestão de Folha de Pagamento –  
PagueAqui**

Histórico da Revisão

Data	Versão	Descrição	Autor	Área/Setor
11/09/2025	1.0	Criação inicial do Plano de Testes	Mariana Tavares Milena Cardoso	Engenharia de Software

# Sistema de Gestão de Folha de Pagamento – PagueAqui

**Identificação do Plano de Testes:** PT-PAGUEAQUI-001

Plano de Testes do Sistema de Gestão de Folha de Pagamento (*PagueAqui*)

## 1. Introdução

### 1.1. Objeto

O objeto do plano de testes é o Sistema de Gestão de Folha de Pagamento – PagueAqui, responsável por gerenciar dados de colaboradores e empresas, calcular salários, proventos, descontos e adicionais, além de gerar contracheques de acordo com a legislação trabalhista vigente.

### 1.2. Objetivo

Garantir que todas as funcionalidades críticas do sistema, tanto funcionais quanto não funcionais, sejam validadas, incluindo cálculos corretos, usabilidade, desempenho, confiabilidade e segurança.

## 2. Escopo

O presente plano de testes abrange os seguintes tipos de testes:

- Teste Funcional: validar cadastro de colaboradores e empregadores, cálculos de salários, descontos, adicionais, geração de contracheques..
- Teste de Banco de Dados: garantir integridade dos registros e consistência das informações gravadas.
- Teste de Interface do Usuário: verificar usabilidade, navegação e clareza das mensagens de erro.
- Teste de Ciclos de Negócio: simulação de processamento mensal da folha de pagamento.
- Teste de Desempenho e Carga: avaliar se o sistema atende ao requisito de resposta inferior a 2 segundos e suporta 100 usuários simultâneos.
- Teste de Segurança e Controle de Acesso: validar permissões de acesso e restrição a funcionalidades.
- Teste de Recuperação: simular falhas de energia/rede e avaliar a retomada do sistema.
- Teste de Configuração: verificar compatibilidade em diferentes navegadores (Chrome, Firefox, Edge, Safari) e dispositivos (desktop, tablet, smartphone).

Itens de Software a serem testados:

### **Módulo de Cadastro de Colaboradores (Alta prioridade)**

Funcionalidades testadas:

- Inserção de dados obrigatórios (nome, CPF, PIS, cargo, salário, data de admissão). Edição e exclusão de registros.

## **Sistema de Gestão de Folha de Pagamento – PagueAqui**

- Validação de formatos (CPF, PIS, data).

Funcionalidades não testadas:

- Integração com bases externas de validação de documentos (fora do escopo).

### **Módulo de Cadastro da Empresa (Alta prioridade)**

Funcionalidades testadas:

- Registro de razão social, CNPJ, endereço, CNAE.
- Edição e exclusão de dados da empresa.

Funcionalidades não testadas:

- Emissão de relatórios fiscais integrados à Receita Federal (fora do escopo).

### **Módulo de Proventos (Alta prioridade)**

Funcionalidades testadas:

- Cálculo de salário base proporcional aos dias trabalhados.
- Adicionais: hora extra (50% e 100%), noturno, insalubridade e periculosidade.
- Benefícios: VA/VR, plano de saúde, transporte.

Funcionalidades não testadas:

- Comissões vinculadas a metas comerciais (não implementadas nesta versão).

### **Módulo de Descontos (Alta prioridade)**

Funcionalidades testadas:

- INSS, IRRF, faltas e atrasos, adiantamentos, pensão alimentícia.

Funcionalidades não testadas:

- Contribuições sindicais facultativas (não implementadas).

### **Módulo de Jornada de Trabalho (Média prioridade)**

Funcionalidades testadas:

- Registro de horas extras, faltas justificadas e injustificadas.
- Cálculo do Descanso Semanal Remunerado (DSR).

Funcionalidades não testadas:

- Controle de ponto eletrônico integrado (não implementado).

### **Módulo de Contracheque (Alta prioridade)**

Funcionalidades testadas:

- Geração de contracheque em PDF com dados do colaborador, proventos, descontos, salário líquido e FGTS.

Funcionalidades não testadas:

- Exportação para outros formatos (XML, CSV).

## **3. Abordagem**

A realização dos testes seguirá uma abordagem sistemática, combinando testes manuais e automatizados, com foco em garantir a conformidade do sistema de folha de pagamento (PagueAqui) com os requisitos funcionais, não funcionais e legais.

### **3.1 Técnicas Utilizadas**

- Caixa-preta: validação das funcionalidades a partir das entradas e saídas esperadas, sem considerar a implementação interna.

## Sistema de Gestão de Folha de Pagamento – PagueAqui

- Partição de equivalência e valores-limite: testes em cenários de cálculo de salários, descontos e adicionais, explorando dados válidos e inválidos.
- Testes exploratórios: identificação de falhas de usabilidade e mensagens de erro.
- Testes automatizados de unidade (JUnit): validação de métodos críticos de cálculo (ex.: salário líquido, descontos de INSS, IRRF, DSR).
- Testes de regressão automatizados (JUnit + Selenium): execução repetitiva de cenários críticos após correções ou novas versões.
- Testes de desempenho e carga (JMeter): simulação de até 100 usuários simultâneos.
- Testes de segurança: verificação de perfis de acesso e tentativas de acesso indevido.

### 3.2 Ferramentas de Apoio

- **JUnit (Java)** → testes unitários dos métodos de cálculo (salário, descontos, adicionais).
- **JMeter** → testes de desempenho, carga e stress.
- **MySQL Workbench / SQL scripts** → verificação de consistência do banco de dados.
- **Excel** → documentação de casos de teste e relatórios de resultados.

### 3.3 Restrições

- Integrações externas (eSocial, sistemas bancários, Receita Federal) não fazem parte do escopo e não serão testadas nesta fase.
- Ambiente de testes poderá ter limitações de infraestrutura em relação ao ambiente de produção (ex.: volume de dados reduzido).
- Massa de dados usada será simulada, não real.

## 4. Missão de Avaliação e Motivação dos Testes

### 4.1 Fundamentos

O sistema PagueAqui foi concebido para atender a uma necessidade recorrente em empresas de pequeno e médio porte: o gerenciamento da folha de pagamento de maneira eficiente, segura e em conformidade com a legislação trabalhista. Historicamente, o processo de folha nessas organizações costuma ser manual ou apoiado em planilhas, o que gera alta propensão a erros de cálculo, inconsistências em descontos e riscos legais.

O esforço de testes justifica-se pelo fato de que falhas no cálculo da folha de pagamento podem trazer sérios impactos, como passivos trabalhistas, problemas fiscais e insatisfação dos colaboradores. Garantir a qualidade do sistema significa assegurar que proventos, descontos, adicionais e benefícios sejam aplicados corretamente, respeitando limites e fórmulas estabelecidas pela lei.

A solução foi planejada com uma arquitetura modular, composta por cadastros (empresa e colaboradores), cálculos de proventos e descontos, controle de jornada de trabalho e geração de

## Sistema de Gestão de Folha de Pagamento – PagueAqui

contracheques. Essa modularidade permite isolar e testar cada parte individualmente, além de facilitar a regressão em futuras evoluções do sistema.

Portanto, os fundamentos do presente plano estão alicerçados na mitigação de riscos técnicos e legais, na melhoria da confiabilidade da aplicação e na entrega de um sistema robusto que atenda às expectativas dos usuários finais e dos gestores.

### 4.2 Missão de Avaliação

A missão dos testes nesta iteração é identificar o maior número possível de falhas que possam comprometer os cálculos de folha de pagamento e a geração de contracheques, avaliar riscos de qualidade perceptíveis e fornecer informações claras sobre a confiabilidade do sistema para apoiar a tomada de decisão quanto à sua liberação.

### 4.3 Motivadores dos Testes

Os testes desta iteração são motivados principalmente por:

- Riscos de qualidade: possibilidade de cálculos incorretos no salário líquido, descontos ou adicionais.
- Riscos técnicos: falhas no desempenho do sistema sob carga de usuários simultâneos e inconsistências no banco de dados.
- Riscos do projeto: atraso na entrega do sistema caso bugs críticos sejam identificados tardiamente.
- Requisitos funcionais: cálculo de proventos, descontos, adicionais e geração de contracheque.
- Requisitos não funcionais: desempenho (resposta < 2s), segurança (controle de acessos), confiabilidade (não perder dados em falhas de rede).
- Solicitações de mudança: ajustes previstos no cálculo de adicionais e descontos conforme atualizações de legislação trabalhista.

## 5. Itens de Teste-Alvo

A listagem a seguir identifica os principais itens de **software**, **hardware** e **elementos de suporte** que serão objeto de teste no sistema de folha de pagamento (*PagueAqui*). Cada item é classificado de acordo com sua importância relativa para o funcionamento e sucesso do sistema.

### 5.1 Itens de Software

- **Módulo de Cadastro de Colaboradores** (*Alta importância*)  
Responsável por armazenar dados pessoais, contratuais e salariais dos empregados.  
Impacto direto na confiabilidade das informações usadas nos cálculos da folha.
- **Módulo de Cadastro da Empresa** (*Média importância*)  
Centraliza dados cadastrais das organizações (CNPJ, razão social, endereço, CNAE).

Essencial para relatórios institucionais e geração de contracheques.

## Sistema de Gestão de Folha de Pagamento – PagueAqui

- **Módulo de Proventos** (*Alta importância*)  
Realiza o cálculo do salário base proporcional e adicionais (hora extra, noturno, insalubridade, periculosidade).  
Fundamental para a conformidade trabalhista.
- **Módulo de Descontos** (*Alta importância*)  
Aplica descontos obrigatórios (INSS, IRRF), faltas/atrasos, pensão alimentícia, entre outros.  
Um dos pontos de maior risco legal em caso de falha.
- **Módulo de Jornada de Trabalho** (*Média importância*)  
Registra horas extras, atrasos e faltas, calculando impacto sobre a remuneração.  
Suporte importante, mas não bloqueante para emissão da folha.
- **Módulo de Contracheque (Holerite)** (*Alta importância*)  
Consolida proventos, descontos e salário líquido em documento PDF.  
Saída final mais relevante para o usuário.
- **Banco de Dados Relacional (MySQL)** (*Alta importância*)  
Armazena todas as informações de colaboradores, empresas, proventos, descontos e registros da folha.  
Ponto crítico para integridade e consistência dos dados.

### 5.2 Itens de Hardware

- **Servidor de Aplicação** (*Alta importância*)  
Responsável por hospedar o sistema *PagueAqui*.  
Testado em relação a desempenho e capacidade de processamento.
- **Servidor de Banco de Dados** (*Alta importância*)  
Hospeda o banco de dados relacional utilizado pelo sistema.  
Crítico para integridade, performance e disponibilidade.
- **Estações de Trabalho de Teste** (*Média importância*)  
Máquinas clientes (Windows/Linux) utilizadas para simulação do uso do sistema.  
Servem para validar compatibilidade e desempenho no ambiente do usuário final.

### 5.3 Elementos de Suporte

- **Sistemas Operacionais** (*Média importância*)  
Windows 10/11 e Linux Ubuntu.  
Necessário validar compatibilidade mínima.
- **Navegadores Web** (*Alta importância*)  
Google Chrome, Mozilla Firefox, Microsoft Edge, Safari.  
Testes garantem acessibilidade e consistência de interface.
- **Ferramentas de Teste** (*Média importância*)  
**JUnit (Java)**: automação de testes unitários.

# Sistema de Gestão de Folha de Pagamento – PagueAqui

**JMeter:** simulação de carga e stress.

## 6. Resumo dos Testes Planejados

### 6.1 Inclusões dos Testes

Os seguintes testes serão executados:

- Testes Funcionais
  - Cadastro de Colaboradores: inserção, edição e exclusão de dados obrigatórios.
  - Cadastro de Empresa: registro de dados essenciais (CNPJ, razão social, endereço).
  - Proventos: cálculo de salário base, adicionais (hora extra, insalubridade, periculosidade, noturno), benefícios (VT, VR, plano de saúde).
  - Descontos: INSS, IRRF, adiantamentos, faltas/atrasos, pensão alimentícia.
  - Jornada: registro de horas extras, faltas justificadas/injustificadas, DSR.
  - Contracheque: geração de documento em PDF com valores brutos, descontos e salário líquido.
- Testes de Banco de Dados
  - Integridade de dados após inserções/alterações.
  - Verificação de consistência em cálculos armazenados.
- Testes de Usabilidade e Interface
  - Navegação entre telas.
  - Feedback de erros claros em casos de entradas inválidas (CPF, PIS, datas).
- Testes de Desempenho e Carga
  - Resposta < 2 segundos em operações principais.
  - Suporte a 100 usuários simultâneos sem degradação perceptível.
- Testes de Segurança e Acesso
  - Restrição de funcionalidades conforme tipo de usuário.
  - Bloqueio de acessos não autorizados via URL direta.
- Testes de Recuperação
  - Simulação de falhas de rede e energia com retomada do sistema sem perda de



## Sistema de Gestão de Folha de Pagamento – PagueAqui

dados.

- Testes de Configuração e Compatibilidade
  - Execução em diferentes navegadores (Chrome, Firefox, Edge, Safari).
  - Testes em dispositivos desktop, tablet e smartphone.
- Testes de Instalação
  - Instalação em ambiente limpo.
  - Atualização de versão existente.

### 6.2 Outros candidatos a possível inclusão

Estes testes poderão ser incluídos em versões futuras, mas não serão validados nesta iteração:

- Integração com eSocial para envio automático das folhas de pagamento.
- Integração com sistemas bancários para crédito direto em conta.
- Relatórios contábeis avançados para auditorias externas.
- Exportação de contracheques em XML e CSV.

### 6.3 Exclusões dos Testes

Os seguintes testes não serão executados neste plano, com justificativas:

- Contribuições sindicais facultativas → não implementadas.
- Controle de ponto eletrônico integrado → fora do escopo do sistema atual.
- Integrações externas (eSocial, Receita Federal, bancos) → dependem de APIs externas e não fazem parte desta versão.
- Relatórios fiscais completos → responsabilidade de sistemas contábeis especializados, não do PagueAqui.

## 7. Abordagem dos Testes

A estratégia de testes para o sistema **PagueAqui** será baseada em uma combinação de testes manuais e automatizados, priorizando a integridade dos cálculos da folha de pagamento, a consistência dos dados armazenados no banco de dados e a validação dos requisitos funcionais e não funcionais.

O foco será garantir conformidade com a legislação trabalhista e fiscal, segurança dos dados sensíveis dos colaboradores e confiabilidade no processamento da folha.

### 7.1 Catálogos Iniciais de Ideias de Teste e Outras Fontes de Referência

## Sistema de Gestão de Folha de Pagamento – PagueAqui

Fontes que servirão de apoio para derivar casos de teste:

- **Requisitos funcionais e não funcionais** documentados (seção 2 do doc).
- **Cartões CRC** (seção 3) para validar responsabilidades e interações entre classes.
- **Leis trabalhistas e tributárias vigentes** (cálculo de INSS, IRRF, FGTS).
- **Catálogo de Ideias de Teste (RUP)** como guia para identificar cenários genéricos (validação de campos obrigatórios, limites, erros de entrada etc.).
- **Boas práticas de testes de software** (ISTQB, IEEE 829/29119).
- **Casos de uso típicos de folha de pagamento:** cadastro de funcionários, geração de contracheque, descontos, adicionais e benefícios.

### 7.2 Tipos e Técnicas de Teste

#### 7.2.1 Teste de Integridade de Dados e de Banco de Dados

<b>Objetivo da Técnica:</b>	Experimentar processos e métodos de acesso ao banco de dados de forma independente da interface do usuário, observando e registrando comportamentos incorretos, falhas de processamento ou presença de dados corrompidos relacionados às operações de folha de pagamento (cadastro de funcionário, cálculo de salários, descontos, adicionais, benefícios e geração de contracheques).
<b>Técnica:</b>	<ul style="list-style-type: none"><li>- Disparar cada processo e método de acesso ao banco de dados com dados <b>válidos e inválidos</b>, simulando cenários reais (salário mínimo, teto de INSS, múltiplos dependentes) e situações anômalas (CPF duplicado, dados bancários inconsistentes).</li><li>- Inspeccionar diretamente o banco de dados para assegurar que os registros foram gravados, atualizados ou excluídos corretamente.</li><li>- Revisar os dados retornados (consultas de relatórios, contracheques) para garantir que a recuperação foi fiel ao planejado.</li><li>- Validar transações críticas (ex.: cálculo de IRRF e INSS) assegurando atomicidade e rollback em caso de falha.</li></ul>

## Sistema de Gestão de Folha de Pagamento – PagueAqui

<b>Estratégias:</b>	<ul style="list-style-type: none"><li>- Comparação automática entre valores calculados no banco e valores esperados (tabelas de alíquotas, regras trabalhistas).</li><li>- Criação de <b>scripts de verificação pós-processamento</b>, conferindo consistência entre tabelas relacionadas (ex.: funcionário vinculado corretamente à folha e empresa).</li><li>- Uso de <b>massa de dados mínima e controlada</b> para facilitar a identificação de inconsistências.</li><li>- Sempre que possível, utilização de testes automatizados para autoavaliação de êxito ou falha, reduzindo riscos de erros manuais.</li><li>- Observação de logs de execução e triggers para identificar falhas silenciosas.</li></ul>
<b>Ferramentas Necessárias:</b>	<ul style="list-style-type: none"><li>- Ferramenta de automação de scripts de teste (ex.: Selenium para disparo indireto, JUnit/PyTest com conexão ao banco).</li><li>- Restaurador e reproduzidor de imagem da configuração básica do DB.</li><li>- Ferramentas de backup e recuperação do banco de dados.</li><li>- Ferramentas de monitoramento de recursos (CPU, memória, disco, transações).</li><li>- Utilitários SQL (Workbench, pgAdmin, DBeaver).</li><li>- Geradores de massa de dados de teste.</li></ul>
<b>Critérios de Êxito:</b>	<ul style="list-style-type: none"><li>- Todos os principais processos e métodos de acesso a dados são validados.</li><li>- Nenhum dado inválido é persistido no banco sem tratamento adequado.</li><li>- Dados retornados são consistentes com os requisitos funcionais (salário bruto, descontos, salário líquido, benefícios).</li><li>- Em falhas, o sistema garante rollback, sem deixar registros parciais ou inconsistentes.</li></ul>

## Sistema de Gestão de Folha de Pagamento – PagueAqui

<b>Considerações Especiais:</b>	<ul style="list-style-type: none"><li>- Poderão ser necessários drivers ou ambientes de desenvolvimento DBMS para inserir ou modificar dados diretamente no banco.</li><li>- Parte dos processos poderá ser disparada manualmente em ambiente controlado.</li><li>- Recomenda-se o uso de <b>bancos de dados de teste pequenos</b>, com registros limitados, para aumentar a visibilidade de anomalias.</li><li>- É importante isolar dados sensíveis (como informações bancárias e CPF) durante os testes, utilizando <b>máscaras ou dados fictícios</b> para respeitar privacidade.</li></ul>
---------------------------------	---

### 7.2.2 Teste de Funcionamento

<b>Objetivo da Técnica:</b>	Validar se o sistema atende corretamente os requisitos funcionais, garantindo que cada funcionalidade (cadastro de funcionário, cálculo de folha, geração de contracheque, relatórios) opere conforme especificado.
<b>Técnica:</b>	Executar casos de teste funcionais baseados nos requisitos, cobrindo cenários normais, alternativos e de exceção.
<b>Estratégias:</b>	<ul style="list-style-type: none"><li>- Derivar testes diretamente dos requisitos funcionais.</li><li>- Usar particionamento de equivalência e análise de valor-limite para entradas (ex.: salário mínimo e teto de INSS).</li><li>- Validar saídas esperadas (contracheque, relatórios) contra resultados calculados manualmente.</li></ul>
<b>Ferramentas Necessárias:</b>	<ul style="list-style-type: none"><li>- Ferramentas de gerenciamento de testes (ex.: TestLink, Zephyr).</li><li>- Planilhas de validação de cálculos.</li></ul>
<b>Critérios de Êxito:</b>	Todas as funcionalidades críticas passam com sucesso em cenários esperados e alternativos.

## Sistema de Gestão de Folha de Pagamento – PagueAqui

<b>Considerações Especiais:</b>	Testes devem estar alinhados à legislação trabalhista vigente.
---------------------------------	--

### 7.2.3 Teste de Ciclos de Negócios

<b>Objetivo da Técnica:</b>	Assegurar que os fluxos completos de negócio (ex.: admissão → folha → pagamento → relatório fiscal) funcionem de ponta a ponta.
<b>Técnica:</b>	Executar cenários completos de processo real, simulando um ciclo mensal de folha de pagamento.
<b>Estratégias:</b>	<ul style="list-style-type: none"><li>- Criar cenários de usuário real (ex.: funcionário contratado, férias, desligamento).</li><li>- Verificar cálculos acumulados e históricos.</li></ul>
<b>Ferramentas Necessárias:</b>	<ul style="list-style-type: none"><li>- Ferramentas de automação de processos (ex.: Selenium).</li><li>- Geradores de massa de dados.</li></ul>
<b>Critérios de Êxito:</b>	Ciclos de negócios são executados corretamente do início ao fim, sem interrupções.
<b>Considerações Especiais:</b>	Deve-se validar dependências externas (ex.: exportação para órgãos fiscais).

### 7.2.4 Teste de Interface do Usuário

<b>Objetivo da Técnica:</b>	Verificar usabilidade, navegabilidade e responsividade da UI, garantindo clareza de mensagens e facilidade de uso.
<b>Técnica:</b>	Avaliar a interface via testes exploratórios e automatizados em diferentes dispositivos e navegadores.

## Sistema de Gestão de Folha de Pagamento – PagueAqui

<b>Estratégias:</b>	<ul style="list-style-type: none"><li>- Testar cenários de interação com teclado, mouse e dispositivos móveis.</li><li>- Validar acessibilidade (uso de leitores de tela, contraste de cores).</li></ul>
<b>Ferramentas Necessárias:</b>	<ul style="list-style-type: none"><li>- Selenium, Cypress.</li><li>- Ferramentas de acessibilidade (ex.: WAVE, Axe).</li></ul>
<b>Critérios de Êxito:</b>	UI atende aos requisitos de usabilidade e acessibilidade.
<b>Considerações Especiais:</b>	Validar consistência visual entre web e mobile

### 7.2.5 Determinação do Perfil de Desempenho

<b>Objetivo da Técnica:</b>	Definir benchmarks de desempenho do sistema em condições normais e de pico.
<b>Técnica:</b>	Executar medições de tempo de resposta, uso de CPU/memória e consumo de rede em cenários variados.
<b>Estratégias:</b>	<ul style="list-style-type: none"><li>- Estabelecer cenários de carga média e máxima.</li><li>- Definir métricas de comparação (tempo &lt; 2s, 100 usuários simultâneos).</li></ul>
<b>Ferramentas Necessárias:</b>	<ul style="list-style-type: none"><li>- JMeter, Locust.</li><li>- Ferramentas de monitoramento de recursos.</li></ul>
<b>Critérios de Êxito:</b>	Sistema cumpre os requisitos de performance definidos no documento.

## Sistema de Gestão de Folha de Pagamento – PagueAqui

<b>Considerações Especiais:</b>	Monitorar gargalos de banco de dados e rede.
---------------------------------	--

### 7.2.6 Teste de Carga

<b>Objetivo da Técnica:</b>	Verificar o comportamento do sistema sob carga crescente de usuários simultâneos.
<b>Técnica:</b>	Simular múltiplos acessos simultâneos às principais funções (cálculo de folha, geração de relatórios)
<b>Estratégias:</b>	Aumentar progressivamente o número de usuários virtuais até atingir o limite esperado.
<b>Ferramentas Necessárias:</b>	JMeter, Gatling.
<b>Critérios de Êxito:</b>	Sistema mantém tempo de resposta aceitável e sem falhas até o limite definido.
<b>Considerações Especiais:</b>	Usar massa de dados realista para simulação.

### 7.2.7 Teste de Stress

<b>Objetivo da Técnica:</b>	Avaliar o sistema em condições extremas além da carga esperada, identificando pontos de falha.
<b>Técnica:</b>	Submeter o sistema a volumes muito acima da capacidade prevista.
<b>Estratégias:</b>	<ul style="list-style-type: none"><li>- Subir gradualmente até quebra.</li><li>- Observar comportamento pós-falha (mensagens de erro, estabilidade).</li></ul>

## Sistema de Gestão de Folha de Pagamento – PagueAqui

<b>Ferramentas Necessárias:</b>	JMeter, Locust.
<b>CrITÉRIOS de Êxito:</b>	Sistema falha de forma controlada, sem perda de dados.
<b>Considerações Especiais:</b>	Foco em resiliência, não apenas em performance.

### 7.2.8 Teste de Volume

<b>Objetivo da Técnica:</b>	Validar funcionamento do sistema com grandes volumes de dados armazenados (ex.: histórico de anos de folhas).
<b>Técnica:</b>	Popular o banco com grandes quantidades de registros e executar consultas e relatórios.
<b>Estratégias:</b>	Verificar impacto em consultas, relatórios e exportações.
<b>Ferramentas Necessárias:</b>	Scripts SQL de carga
<b>CrITÉRIOS de Êxito:</b>	Sistema mantém resposta aceitável e dados íntegros sob volume elevado.
<b>Considerações Especiais:</b>	Massa de dados deve representar cenários reais (milhares de funcionários e folhas).

### 7.2.9 Teste de Segurança e de Controle de Acesso

<b>Objetivo da Técnica:</b>	Assegurar que dados sensíveis (ex.: CPF, dados bancários) sejam protegidos e acessíveis apenas a perfis autorizados.
<b>Técnica:</b>	Executar testes de autenticação, autorização, criptografia e vulnerabilidades conhecidas.
<b>Estratégias:</b>	<ul style="list-style-type: none"><li>- Testar acesso restrito via interface e via URL direta.</li><li>- Validar senhas fortes e criptografia em trânsito/repouso.</li></ul>



## Sistema de Gestão de Folha de Pagamento – PagueAqui

<b>Ferramentas Necessárias:</b>	<ul style="list-style-type: none"><li>- OWASP ZAP, Burp Suite.</li><li>- Ferramentas de gerenciamento de usuários e papéis</li></ul>
<b>Critérios de Êxito:</b>	Acesso indevido não é permitido em nenhum cenário.
<b>Considerações Especiais:</b>	Testes devem evitar exposição de dados reais → usar dados mascarados.

### 7.2.10 Teste de Tolerância a Falhas e de Recuperação

<b>Objetivo da Técnica:</b>	Avaliar capacidade do sistema de se recuperar após falhas (queda de rede, falha no DB).
<b>Técnica:</b>	Simular falhas e verificar se o sistema restaura seu estado de forma íntegra.
<b>Estratégias:</b>	<ul style="list-style-type: none"><li>- Testar rollback de transações.</li><li>- Interromper conexão de rede durante cálculos críticos.</li></ul>
<b>Ferramentas Necessárias:</b>	<ul style="list-style-type: none"><li>- Ferramentas de simulação de falhas (Chaos Monkey).</li><li>- Logs e backups.</li></ul>
<b>Critérios de Êxito:</b>	Sistema recupera operações sem perda ou corrupção de dados.
<b>Considerações Especiais:</b>	Recomenda-se banco de teste pequeno para fácil rastreamento de erros.

### 7.2.11 Teste de Configuração

<b>Objetivo da Técnica:</b>	Validar funcionamento em diferentes ambientes, navegadores, SOs e dispositivos.
-----------------------------	---

## Sistema de Gestão de Folha de Pagamento – PagueAqui

<b>Técnica:</b>	Executar testes em configurações variadas (Chrome, Firefox, Edge, Android, iOS)
<b>Estratégias:</b>	<ul style="list-style-type: none"><li>- Verificar responsividade em múltiplas resoluções.</li><li>- Testar parâmetros de configuração (ex.: fuso horário, moeda).</li></ul>
<b>Ferramentas Necessárias:</b>	BrowserStack, Sauce Labs.
<b>CrITÉrios de Êxito:</b>	Sistema é compatível com os ambientes definidos nos requisitos não funcionais.
<b>Considerações Especiais:</b>	Garantir consistência entre web e mobile.

### 7.2.12 Teste de Instalação

<b>Objetivo da Técnica:</b>	Verificar se o sistema pode ser instalado, atualizado e desinstalado corretamente.
<b>Técnica:</b>	Executar instalação em ambientes de teste, seguindo instruções oficiais.
<b>Estratégias:</b>	<ul style="list-style-type: none"><li>- Validar setup inicial, dependências e permissões.</li><li>- Testar reinstalações e upgrades.</li></ul>
<b>Ferramentas Necessárias:</b>	<ul style="list-style-type: none"><li>- Scripts de deploy.</li><li>- Ferramentas de virtualização (Docker, VM).</li></ul>
<b>CrITÉrios de Êxito:</b>	Instalação concluída sem erros e sistema funcional após o setup
<b>Considerações Especiais:</b>	Verificar rollback em caso de falha na instalação.

# Sistema de Gestão de Folha de Pagamento – PagueAqui

## 8. Critérios de Entrada e de Saída

### 8.1 Plano de Teste

#### 8.1.1 Critérios de Entrada de Plano de Teste

- Documento de requisitos funcionais e não funcionais aprovado.
- Ambiente de teste configurado e estável (DBMS, aplicação web e mobile).
- Massa de dados de teste definida (funcionários, empresas, proventos, descontos).
- Ferramentas de automação, monitoramento e gerenciamento de defeitos instaladas.
- Equipe de teste treinada e com papéis atribuídos.

#### 8.1.2 Critérios de Saída de Plano de Teste

- Todos os casos de teste planejados foram executados.
- Defeitos críticos e altos corrigidos e revalidados.
- Taxa mínima de cobertura de requisitos atendida ( $\geq 90\%$ ).
- Relatórios de teste revisados e aprovados pelos stakeholders.
- Nenhum incidente em aberto que comprometa cálculos da folha ou segurança de dados

#### 8.1.3 Critérios de Suspensão e de Reinício

- **Suspensão:** Testes serão suspensos em caso de falhas críticas no ambiente (queda de servidor, indisponibilidade do banco, build instável).
- **Reinício:** Retomada ocorrerá após correção do problema, restauração do ambiente e execução de smoke test validando condições mínimas de estabilidade.

### 8.2 Ciclos de Teste

#### 8.2.1 Critérios de Entrada de Ciclo de Teste

- Build estável entregue pela equipe de desenvolvimento.
- Registro de defeitos pendentes analisados e priorizados.
- Ambiente de teste atualizado e limpo para nova execução.

## **Sistema de Gestão de Folha de Pagamento – PagueAqui**

### **8.2.2 Critérios de Saída de Ciclo de Teste**

- Casos de teste do ciclo atual executados e registrados.
- Defeitos críticos e altos reportados, corrigidos e revalidados.
- Métricas de cobertura de testes e qualidade registradas.

### **8.2.3 Término Anormal do Ciclo de Teste**

- Ciclo poderá ser encerrado antecipadamente se:
  - O build apresentar instabilidade severa.
  - Volume de defeitos críticos inviabilizar a continuidade.
  - Problemas de infraestrutura impossibilitarem execução confiável.
- Reinício será considerado em um novo build corrigido.

## **9. Produtos Liberados**

### **9.1 Sumários de Avaliação de Testes**

- Relatórios executivos contendo status dos testes, defeitos encontrados, cobertura de casos e avaliação de qualidade.
- Gerados semanalmente e ao final de cada ciclo.

### **9.2 Geração de Relatórios sobre Cobertura de Teste**

- Relatórios automáticos sobre % de casos executados, aprovados e falhos.
- Métricas de cobertura de requisitos funcionais e não funcionais.
- Ferramentas: TestLink, Jira + XRay, planilhas Excel.
- Emitidos ao final de cada ciclo.

### **9.3 Relatórios da Qualidade Perceptível**

- Relatórios qualitativos baseados em feedback dos testadores e stakeholders.
- Incluem usabilidade, clareza de mensagens, responsividade e estabilidade percebida.
- Gerados a cada ciclo de teste completo.
- 

### **9.4 Registros de Incidentes e Solicitações de Mudança**

## Sistema de Gestão de Folha de Pagamento – PagueAqui

- Rastreabilidade de defeitos no Jira ou ferramenta equivalente.
- Incidentes categorizados por severidade, prioridade e status.
- Solicitações de mudança ligadas a requisitos e builds afetados

### 9.5 Conjunto de Testes de Regressão e Scripts de Teste de Suporte

- Scripts de automação de regressão mantidos em repositório Git.
- Executados a cada novo build para garantir não regressão.

### 9.6 Produtos de Trabalho Adicionais

#### 9.6.1 Resultados Detalhados dos Testes

- Relatórios em Excel e logs de execução armazenados em repositório de testes.

#### 9.6.2 Scripts de Teste Funcionais Automatizados Adicionais

- Scripts automatizados em Selenium/Python para UI e em JMeter para desempenho.

#### 9.6.3 Guia de Teste

- Documento com boas práticas, catálogo de ideias de teste, modelos de falhas e padrões de automação.

#### 9.6.4 Matrizes de Rastreabilidade

- Planilha de rastreabilidade de requisitos × casos de teste × defeitos.

## 10. Fluxo de Trabalho de Teste

Esta seção fornece um resumo do fluxo de trabalho a ser seguido pela equipe de teste no desenvolvimento e na execução deste Plano de Teste para o projeto

Sistema de Gestão de Folha de Pagamento (PagueAqui). O fluxo de trabalho detalhado e customizado para o projeto está documentado separadamente no Caso de Desenvolvimento do projeto, sendo este um resumo de alto nível.

O processo de teste será dividido em fases distintas e iterativas, garantindo que cada etapa do desenvolvimento do software seja acompanhada por atividades de validação e verificação de qualidade. As tarefas detalhadas de cada fase serão gerenciadas em uma ferramenta central de gerenciamento de projetos para garantir o acompanhamento dinâmico e evitar a duplicação de informações.

O fluxo de trabalho pode ser visualizado como um processo sequencial de fases, que se repete a

## Sistema de Gestão de Folha de Pagamento – PagueAqui

cada ciclo de desenvolvimento: Planejamento -> Projeto e Implementação -> Execução e Avaliação.  
Fases do Fluxo de Trabalho de Teste:

### 10.1. Fase de Planejamento e Estratégia de Teste

- **Atividades:** Esta fase inicial envolve o planejamento geral dos testes. As principais tarefas incluem a identificação dos requisitos de teste com base na documentação do software PagueAqui, a avaliação de riscos associados às funcionalidades (ex: cálculos de INSS/IRRF, segurança de dados do colaborador), o desenvolvimento de uma estratégia de teste, a identificação dos recursos necessários e a criação de um cronograma.
- **Resultados:** O principal produto liberado desta fase é a versão final deste Plano de Teste, que servirá como guia para toda a equipe.

### 10.2. Fase de Projeto e Implementação dos Testes

- **Atividades:** Com o plano definido, a equipe irá detalhar como os testes serão realizados. As tarefas incluem a análise da carga de trabalho, a identificação e descrição de casos de teste para cobrir os requisitos funcionais e não funcionais, a estruturação dos procedimentos de teste e a implementação de scripts de automação. Serão criados conjuntos de dados de teste (massa de dados) para simular diferentes cenários de funcionários e cálculos.
- **Resultados:** Os produtos desta fase incluem a documentação dos Casos de Teste, Procedimentos de Teste, Scripts de Teste automatizados e a massa de dados para a execução.

### 10.3. Fase de Execução e Avaliação dos Testes

- **Atividades:** Esta é a fase de execução prática dos testes planejados. A equipe executará os procedimentos de teste (manuais e automatizados), avaliará os resultados, investigará comportamentos inesperados e registrará os defeitos encontrados em uma ferramenta de controle.
- **Resultados:** Ao final desta fase, serão gerados os produtos liberados do esforço de teste, como sumários de avaliação, relatórios de cobertura de teste, relatórios sobre a qualidade do produto e registros de incidentes. A equipe também analisará os defeitos e avaliará se os critérios de conclusão dos testes foram alcançados.

## 11. Necessidades Ambientais

Esta seção apresenta os recursos não humanos necessários para a execução do Plano de Teste do sistema PagueAqui.

### 11.1 Hardware Básico do Sistema

Os conjuntos de tabelas a seguir apresentam os recursos do sistema necessários ao esforço de teste descrito neste Plano de Teste.

Recurso	Quantidade	Nome e Tipo
Servidor de Banco de Dados	1	Servidor dedicado para hospedar o SGBD MySQL

## Sistema de Gestão de Folha de Pagamento – PagueAqui

Servidor de Aplicação	1	Servidor para hospedar a aplicação web PagueAqui
PCs de Teste Cliente	3	Estações de trabalho com sistemas Windows 10/11 e Linux Ubuntu para simulação do usuário final
Repositório de Teste	1	Servidor de rede para armazenamento de scripts, massa de dados e resultados de testes.
PCs de Desenvolvimento de Teste	2	Estações de trabalho para desenvolvimento de scripts de testes automatizados.

### 11.2 Elementos de Software Básicos do Ambiente de Teste

São necessários os seguintes elementos de software básicos no ambiente de teste deste *Plano de Teste*.

Nome do Elemento de Software	Versão	Tipo e Outras Observações
<b>Windows</b>	<b>10/11</b>	<b>Sistema Operacional</b>
Linux Ubuntu	22.04 LTS ou superior	Sistema Operacional
Google Chrome		Navegador da Internet
Mozilla Firefox		Navegador da Internet
Microsoft Edge		Navegador da Internet
Safari		Navegador da Internet
MySQL Server	8.0 ou superior	Sistema de Gerenciamento de Banco de Dados

### 11.3 Ferramentas de Produtividade e de Suporte

Serão utilizadas as seguintes ferramentas para suportar o processo de teste deste *Plano de Teste*.

Categoria ou Tipo de Ferramenta	Nome da Marca da Ferramenta	Fornecedor ou Desenvolvida Internamente	Versão
Gerenciamento de Teste	Excel	Microsoft	Office

## Sistema de Gestão de Folha de Pagamento – PagueAqui

			365
Controle de Defeitos	A ser definido	A ser definido	
Ferramenta ASQ para teste funcional	Selenium	Open Source	Última estável
Ferramenta ASQ para teste de desempenho	JMeter	Apache	Última estável
Gerador de Perfil ou Monitor de Cobertura de Teste	JUnit (com JaCoCo)	Open Source	JUnit 5
Gerenciamento de Projeto	A ser definido	A ser definido	
Ferramentas DBMS	MySQL Workbench	Oracle	Última estável

### 11.4 Configurações do Ambiente de Teste

Devem ser fornecidas e suportadas as seguintes Configurações de Ambiente de Teste para este projeto.

Nome da Configuração	Descrição	Implementada na Configuração Física
Configuração do usuário comum	Simula o ambiente mais comum de um usuário final: Desktop com Windows 11 e navegador Google Chrome	PCs de Teste Cliente 1 e 2.
Mínima configuração suportada	Valida o funcionamento em um ambiente com Linux Ubuntu e navegador Firefox	PC de Teste Cliente 3.
Configuração de Compatibilidade	Focada em validar a interface e funcionalidades nos demais navegadores suportados (Edge e Safari).	Emulação/Virtualização nos PCs de Teste.
Sistema Operacional Internacional de Dois Bytes	Testes de responsividade e usabilidade em dispositivos móveis para validar a compatibilidade com tablets e smartphones.	Emulação via ferramentas de navegador e/ou dispositivos físicos.

## 12. Responsabilidades, Perfil da Equipe e Necessidades de Treinamento



## Sistema de Gestão de Folha de Pagamento – PagueAqui

### 12.1 Pessoas e Papéis

Esta tabela mostra as suposições referentes ao perfil da equipe do esforço de teste.

Recursos Humanos		
Papel	Recursos Mínimos Recomendáveis (número de papéis alocados em tempo integral)	Responsabilidades ou Comentários Específicos
Gerente de Testes  Coordenação geral do esforço de testes	1 em tempo integral	Supervisiona o gerenciamento. Entre as responsabilidades estão incluídas: <ul style="list-style-type: none"><li>• planejamento e logística</li><li>• combinar missão</li><li>• identificar motivadores</li><li>• adquirir recursos apropriados</li><li>• apresentar relatórios de gerenciamento</li><li>• defender os interesses do teste</li><li>• avaliar a eficiência do esforço de teste</li></ul>
Analista de Teste  Definição de testes a serem conduzidos	1 em tempo integral	Identifica e define os teste específicos a serem conduzidos. Entre as responsabilidades estão incluídas: <ul style="list-style-type: none"><li>• identificar ideias de teste</li><li>• definir detalhes dos testes</li><li>• determinar os resultados dos testes</li><li>• documentar solicitações de mudança</li><li>• avaliar a qualidade do produto</li></ul>

## Sistema de Gestão de Folha de Pagamento – PagueAqui

<p>Designer de Teste</p> <p>Definição da abordagem técnica</p>	<p>1 em tempo parcial</p>	<p>Define a abordagem técnica referente à implementação do esforço de teste. Entre as responsabilidades estão incluídas:</p> <ul style="list-style-type: none"> <li>• definir a abordagem dos testes</li> <li>• definir a arquitetura de automação de teste</li> <li>• verificar as técnicas de teste</li> <li>• definir os elementos de testabilidade</li> <li>• estruturar a implementação dos testes</li> </ul>
<p>Testador</p> <p>Execução dos testes planejados</p>	<p>≥ 2 em tempo integral</p>	<p>Implementa e executa os testes. Entre as responsabilidades estão incluídas:</p> <ul style="list-style-type: none"> <li>• implementar os testes e os conjuntos de testes</li> <li>• executar os conjuntos de testes</li> <li>• registrar os resultados</li> <li>• analisar as falhas dos testes e possibilitar a recuperação posterior</li> <li>• documentar incidentes</li> </ul>
<p>Administrador do Sistema de Teste</p> <p>Suporte ao ambiente de testes</p>	<p>1 em tempo parcial</p>	<p>Assegura a manutenção e o gerenciamento dos recursos e do ambiente do teste. Entre as responsabilidades estão incluídas:</p> <ul style="list-style-type: none"> <li>• administrar o sistema de gerenciamento de teste</li> <li>• instalar e suportar o acesso às configurações do ambiente de teste e aos laboratórios de teste, bem como a recuperação deles</li> </ul>

## Sistema de Gestão de Folha de Pagamento – PagueAqui

Administrador do Banco de Dados, Gerente do Banco de Dados  Suporte ao gerenciamento de dados de teste	1 em tempo parcial	Assegurar o gerenciamento e a manutenção dos recursos e do ambiente dos dados de teste (banco de dados). Entre as responsabilidades estão incluídas: <ul style="list-style-type: none"><li>• suportar a administração dos dados de teste e das plataformas de teste (banco de dados)</li></ul>
Designer  Identificação de operações e atributos das classes de teste	1 em tempo parcial	Identifica e define as operações, os atributos e as associações das classes de teste. Entre as responsabilidades estão incluídas: <ul style="list-style-type: none"><li>• define as classes de teste necessárias para suportar os requisitos de estabilidade conforme definido pela equipe de teste</li></ul>
Implementador  Criação e manutenção de componentes de teste	1 em tempo parcial	Implementa as classes de teste e os pacotes de teste e efetua testes unitários nos mesmos. Entre as responsabilidades estão incluídas: <ul style="list-style-type: none"><li>• cria os componentes de teste necessários para suportar os requisitos de estabilidade conforme definido pelo designer</li></ul>

### 12.2 Perfil da Equipe e Necessidades de Treinamento

O perfil da equipe de teste deve contemplar conhecimentos técnicos e de domínio:

- Domínio de negócio: Legislação trabalhista e tributária (INSS, IRRF, FGTS), rotinas de folha de pagamento.
- Técnico: SQL, automação de testes funcionais (Selenium, Cypress), testes de desempenho (JMeter), ferramentas de gestão de defeitos (Jira).
- Boas práticas de QA: Técnicas de teste, análise de risco, cobertura de requisitos e métricas de qualidade.

#### Necessidades de Treinamento

## Sistema de Gestão de Folha de Pagamento – PagueAqui

- **Ferramentas de automação:** Treinamento em Selenium/Cypress para a equipe de testadores e implementadores.
- **Ferramentas de performance:** Capacitação em JMeter e Locust para testes de carga e stress.
- **Banco de dados:** Treinamento em SQL avançado para analistas e administradores de BD, garantindo validação direta dos cálculos de folha.
- **Normas e compliance:** Workshop sobre legislação trabalhista, LGPD e segurança de dados.
- **Metodologias ágeis:** Treinamento rápido em Scrum/Kanban para alinhamento com o time de desenvolvimento.

### Estratégia de Treinamento

- Treinamentos devem ocorrer **próximo ao início da execução dos testes correspondentes**, evitando desperdício e esquecimento.
- Sempre que possível, combinar aquisição de ferramentas com **treinamentos oficiais do fornecedor**.
- Realizar **treinamentos internos** (on-the-job) para acelerar a curva de aprendizado em automação e banco de dados.
- Garantir envolvimento de **outros perfis de apoio** (DBAs, sysadmins, desenvolvedores) durante os testes mais técnicos.

Observações:

O plano de teste é um dos documentos produzidos na condução de um projeto. Ele funciona como:

- Um “integrador” entre diversas atividades de testes no projeto;
- Mecanismo de comunicação para os stakeholders (isto é a equipe de testes e outros interessados);
- Guia para execução e controle das atividades de testes.

O plano de teste, que pode ser elaborado pelo gerente de projeto ou gerente de testes, visa planejar as atividades a serem realizadas, definir os métodos a serem empregados, planejar a capacidade necessária, estabelecer métricas e formas de acompanhamento do processo.

**Estas são as tarefas relacionadas a teste:**

<b>1.Planejar Teste</b>
<b>2.Identificar os Requisitos de Teste</b>

## Sistema de Gestão de Folha de Pagamento – PagueAqui

<b>3.Avaliar o Risco</b>
<b>4.Desenvolver a Estratégia de Teste</b>
<b>5.Identificar os Recursos de Teste</b>
<b>6.Criar Programação</b>
<b>7.Gerar Plano de Teste</b>
<b>8.Projetar Teste</b>
<b>9.Análise da Carga de Trabalho</b>
<b>10.Identificar e Descrever Casos de Teste</b>
<b>11.Identificar e Estruturar Procedimentos de Teste</b>
<b>12.Revisar e Acessar a Cobertura de Teste</b>
<b>13.Implementar Teste</b>
<b>14.Registrar ou Programar Scripts de Teste</b>
<b>15.Identificar a funcionalidade específica de Teste no modelo de design e de implementação</b>
<b>16.Estabelecer Conjuntos de Dados Externos</b>
<b>17.Executar Teste</b>
<b>18.Executar os Procedimentos de Teste</b>
<b>19.Avaliar a Execução do Teste</b>
<b>20.Recuperar-se de uma Interrupção de Teste</b>

## Sistema de Gestão de Folha de Pagamento – PagueAqui

<b>21.Verificar os Resultados</b>
<b>22.Investigar os Resultados Inesperados</b>
<b>23.Registrar Defeitos</b>
<b>24.Avaliar Teste</b>
<b>25.Avaliar Cobertura de Caso de Teste</b>
<b>26.Avaliar Cobertura de Código</b>
<b>27.Analisar os Defeitos</b>
<b>28.Determinar se os Critérios de Conclusão e os Critérios de Sucesso do Teste foram obedecidos</b>