

PLANO DE TESTE: <Nome do Projeto>

Lucas Ryu Maroka

Matheus Ferreira Santos

João Gabriel Oliveira Coelho

José Daniel da Silva de Queiroz

Centro Universitário Senac – Santo Amaro

Histórico de revisões

Versão	Data	Autor	Descrição
1.0.0	03/10/2023	Grupo	Plano de Teste

PLANO DE TESTE

O objetivo deste documento é descrever o Plano de Teste do Projeto de Software **XXXXXX**. Ele contém a definição do escopo do Projeto de Software, a especificação dos requisitos do sistema, a estratégia de teste e o esforço necessário para sua realização. Na estratégia de teste estão definidos as técnicas, níveis e tipos de teste que serão executados na iteração e os objetivos que devem ser atingidos. O documento apresenta ainda a definição do ambiente de testes, a matriz de responsabilidade e a relação dos casos de testes. Em seguida são listados os riscos de teste, os critérios para conclusão dos testes e o cronograma do projeto de teste.

<Para descrever cada capítulo/item deste plano, procurem relacioná-los aos conceitos discutidos na disciplina de Teste de Software. Para isso, utilizem referências de autores da área de engenharia e teste de software. Todas as obras pesquisadas e citadas no texto devem estar relacionadas no final deste documento.>

Sumário

1	ESCOPO DO PROJETO DE SOFTWARE.....	4
1.1	<i>Objetivo do Projeto.....</i>	4
1.2	<i>Metodologia de desenvolvimento.....</i>	4
2	Matriz de rastreabilidade de requisitos de sistema.....	5
2.1	<i>Requisitos Funcionais.....</i>	5
2.2	<i>Requisitos Não Funcionais.....</i>	6
3	ESTRATÉGIA DE TESTE.....	7
4	AMBIENTE de TESTE.....	8
5	MATRIZ DE RESPONSABILIDADE.....	9
6	RELAÇÃO DOS CASOS DE TESTES.....	10
7	RISCOS DO PROJETO DE TESTE DE SOFTWARE.....	11
8	CRITÉRIOS DE CONCLUSÃO.....	12
9	CRONOGRAMA DO PROJETO DE TESTE DE SOFTWARE.....	13
	REFERÊNCIAS.....	14

1 ESCOPO DO PRODUTO DE SOFTWARE

A declaração do escopo do produto de software descreve, em detalhes, as entregas do produto e o trabalho necessário para criar essas entregas. Fornece um entendimento comum do escopo do produto a todas as partes interessadas no projeto e descreve os principais objetivos do projeto. Além disso, permite que a equipe do projeto realize um planejamento mais detalhado, orientando a execução do projeto, fornecendo linha de base para avaliar solicitações de mudanças ou trabalho adicional.

1.2 *Objetivo do Projeto*

O objetivo deste projeto é implementar um sistema de Ponto Eletrônico eficiente e confiável, automatizando o registro de horas, eliminando erros manuais e garantindo conformidade com regulamentações trabalhistas. Além disso, busca-se proporcionar uma experiência positiva aos usuários, simplificando o processo de registro e oferecendo ferramentas para a gestão eficaz do tempo. O foco é estabelecer um ambiente de trabalho mais eficiente, transparente e em conformidade com as melhores práticas de gestão de recursos humanos.

1.3 *Metodologia de desenvolvimento*

Nós iremos utilizar a abordagem ágil de desenvolvimento de software. Isso significa que vamos trabalhar em pequenas partes do projeto de cada vez, fazendo melhorias frequentes. Teremos reuniões regulares e manteremos uma colaboração constante entre a equipe e os interessados. Essa maneira de trabalhar nos permite ser flexíveis e se adaptar rapidamente às mudanças, garantindo que o sistema de Ponto Eletrônico que estamos construindo atenda às necessidades da organização de forma eficiente, a cada iteração será apresentada o nosso plano de testes para aquela sprint específica.

2 MATRIZ DE RASTREABILIDADE DE REQUISITOS DE SISTEMA

A rastreabilidade dos requisitos do sistema desempenha um papel fundamental no planejamento de testes de software. Garantindo um entendimento abrangente dos requisitos, ela alinha os testes com os objetivos do negócio, detecta precocemente problemas, define o escopo adequado, facilita a adaptação a mudanças, assegura uma avaliação clara da cobertura de teste e permite a geração eficiente de relatórios. Essa prática essencial assegura que o software seja desenvolvido e testado de maneira eficaz, atendendo às expectativas dos usuários e às necessidades do negócio.

2.1 Requisitos Funcionais

Descreve as características do produto, serviço ou resultado para cuja criação o projeto será realizado. Essas características terão normalmente menos detalhes nas fases iniciais e mais detalhes nas fases posteriores, conforme as características do produto forem progressivamente elaboradas. Embora a forma e o conteúdo das características variem, as descrições dos requisitos funcionais devem sempre fornecer detalhes suficientes para dar suporte ao planejamento do projeto de software e teste.

REQUISITOS FUNCIONAIS (RF)		
ID	Nome do RF	Descrição
RF01	Registro de Ponto	Funcionários devem ser capazes de registrar seus pontos de entrada e saída.
RF02	Autenticação	O sistema deve autenticar os funcionários para garantir que apenas eles possam registrar o ponto.
RF03	Relatórios de Ponto	Deve ser possível gerar relatórios de ponto para cada funcionário, incluindo horas trabalhadas, atrasos e faltas.
RF04	Aprovação de Ponto	Supervisores devem ter a capacidade de aprovar ou rejeitar registros de ponto.
RF05	Histórico de Ponto	Manter um histórico completo de registros de ponto para cada funcionário, permitindo acessar registros anteriores.
RF06	Notificações	Notificar os funcionários sobre a necessidade de registrar o ponto e enviar alertas aos supervisores quando ocorrerem atrasos ou faltas.
RF07	Controle de Horas Extras	Calcular e registrar as horas extras trabalhadas, se aplicável, de acordo com as políticas da empresa.
RF08	Integração com Folha de Pagamento	Integrar o sistema com o software de folha de pagamento para facilitar o cálculo dos salários com base nas horas trabalhadas.
RF09	Gestão de Férias e Licenças	Permitir que os funcionários solicitem e os supervisores aprovem férias e licenças diretamente no sistema.

2.2. Requisitos Não Funcionais

Segundo Pressman (2016), requisitos não-funcionais são os requisitos relacionados ao uso da aplicação em termos de desempenho, usabilidade, confiabilidade, segurança, disponibilidade, manutenção e tecnologias envolvidas. Estes requisitos dizem respeito a como as funcionalidades serão entregues ao usuário do software.

REQUISITOS NÃO-FUNCIONAIS			
RNF	Categoria	Nome	Descrição
01	Segurança	Segurança	Garantir a segurança dos dados dos funcionários, implementando medidas como criptografia, autenticação forte e auditorias de segurança.
02	Desempenho	Desempenho	Garantir que o sistema seja rápido e responsivo, mesmo com um grande número de funcionários registrando o ponto simultaneamente.
03	Escalabilidade	Escalabilidade	O sistema deve ser capaz de lidar com um aumento no número de funcionários à medida que a empresa cresce.
04	Disponibilidade	Disponibilidade	Garantir que o sistema esteja disponível 24 horas por dia, 7 dias por semana, com um tempo de inatividade mínimo para manutenção.
05	Conformidade Legal	Conformidade Legal	Certificar-se de que o sistema esteja em conformidade com as leis e regulamentações trabalhistas locais.
06	Facilidade de Uso	Facilidade de Uso	O sistema deve ser intuitivo e de fácil utilização para os funcionários e supervisores, minimizando a necessidade de treinamento extensivo.
07	Compatibilidade	Compatibilidade	Certificar-se de que o sistema funcione em uma variedade de dispositivos e navegadores, para garantir o acesso de todos os funcionários.
08	Backup e Recuperação	Backup e Recuperação	Implementar rotinas de backup regulares e um plano de recuperação de desastres para proteger os dados em caso de falha do sistema.

3 ESTRATÉGIA DE TESTE

Para garantir a qualidade do sistema, adotaremos uma estratégia de testes abrangente. Iniciaremos com Testes de Unidade, implementando a abordagem TDD para validar cada parte do código. Em Testes de Integração, garantiremos uma integração eficiente entre módulos por meio de uma abordagem incremental. Os Testes de Sistema serão executados para avaliar o sistema como um todo, os Testes de Desempenho incluirão cenários de carga e estresse para verificar a capacidade do sistema em diferentes condições, testes de usabilidade serão realizados, incorporando testes de acessibilidade, A/B, navegação e entre outros para garantir uma interface amigável e de fácil uso. Em termos de portabilidade, serão feitos Testes de Navegadores e Dispositivos para assegurar acessibilidade e adaptabilidade.

→ *Segue uma tabela, na próxima página, para mais detalhes.*

Continuação...

ESTRATÉGIA DE TESTE				
NÍVEIS DE TESTE				
ID	Nome do Teste	Técnicas	Descrição	Requisitos de Sistemas
01	Teste Unitário	TDD	É uma abordagem de desenvolvimento de software em que os testes unitários são escritos antes da implementação do código.	
02	Teste de Integração	Abordagem de Testes incremental	É uma abordagem que integra e testa módulos da aplicação de forma progressiva.	
03	Teste de Sistema			
04	Teste de Aceitação	Validação	Valida se o sistema está de acordo com as expectativas do cliente.	
TIPOS DE TESTES				
ID	Tipo do Teste	Técnicas	Descrição	Requisitos de Sistemas
01	Desempenho		Iremos utilizar os testes de carga e estresse, ele, a grosso modo, coloca o sistema em condições extremas, com a finalidade de avaliar seu comportamento "sob pressão".	
02	Usabilidade		Testes de Acessibilidade, A/B, Navegação, e entre outros, para garantir que o sistema seja de fácil uso para todas as faixas etárias e grupos sociais e que nenhuma alteração seja feita para piorar a experiência do usuário final	
03	Portabilidade		Testes em múltiplos sistemas operacionais, em diferentes dispositivos/navegadores, diferentes resoluções de tela e entre outros, para garantir que o sistema possa ser executado com sucesso em diferentes ambientes	

4 AMBIENTE DE TESTE

O ambiente de testes é crucial no planejamento, oferecendo um espaço isolado e controlado para testes confiáveis e repetíveis. Permite a configuração precisa de condições, incluindo hardware, software e rede, facilitando a identificação de defeitos e a integração eficaz de componentes. Em resumo, é fundamental para garantir a qualidade e confiabilidade do software, o nosso plano de testes utiliza o seguinte ambiente:

AMBIENTE DE TESTE	
Recurso	Descrição
Computador cliente	Para os testes iremos utilizar máquinas com as seguintes especificações: 2 laptops, com o sistema operacional Ubuntu 22.04LTS e outro com o MacOS e um computador com Windows e outro com MacOS, onde, o computador mais fraco é o laptop com o Ubuntu instalado, o intermediário é o computador com o MacOS (iMac) e os mais fortes são o Windows e o laptop com MacOS (Macbook Pro)
Mobile	Para os testes nos aparelhos mobiles utilizaremos os seguintes smartphones: Zenfone 5 (android) e iPhone X (iOS).
Servidor de aplicação	Será utilizado o servidor Apache Tomcat.
Servidor de banco de dados	Será utilizado o SGBD MySQL.
Criação de relatórios	
Browser cliente	Utilizaremos os seguintes Browsers: Firefox, Chrome e Safari.
SO cliente	Usaremos os sistemas operacionais Windows, Ubuntu e MacOS.

5 MATRIZ DE RESPONSABILIDADE

<Descreva aqui a importância da definição da matriz de responsabilidade no planejamento de teste com base nos autores da área de engenharia e teste de software.>

MATRIZ DE RESPONSABILIDADE				
Fase do Projeto	Atribuição Técnica	Responsabilidade	Nome	Descrição da Atividade
Planejamento	Líder de Projeto de Teste	Técnico responsável pela liderança de um projeto de teste específico, geralmente relacionado a um sistema de desenvolvimento, seja um projeto novo ou uma manutenção.		
	Arquiteto de teste			
Especificação	Analista de Teste			
Execução	Testador			
Resultados				

6 RELAÇÃO E DESCRIÇÃO DOS CASOS DE TESTES

Em engenharia de software, o caso de teste é um conjunto de condições usadas para teste de software. Neste plano de teste consta a relação de todos os cenários e seus respectivos casos de testes a serem especificados pelos analistas de teste.

CASOS DE TESTES			
ID	Cenário de Teste	Caso de Teste	Descrição
01	CT_LOGIN	CT01_Novo Usuário	Caso de teste para validar o cadastramento de um novo usuário.
		CT02_Usuário Cadastrado	Caso de teste para validar o acesso de um novo usuário já cadastrado.
		CT03_Esqueci a senha	Caso de teste para validar a reinicialização de senha para o usuário.
02			
03			
04			
05			

6.1 Descrição dos principais cenários de testes

<Utilizando as técnicas de caixa preta descreva os principais cenários de testes do projeto (mínimo 3).>

7 RISCOS DO PROJETO DE TESTE DE SOFTWARE

Este capítulo contém um sumário dos riscos chaves do projeto de teste, que se acontecerem, poderão afetar os resultados esperados e o projeto de software.

RISCOS DO PROJETO DE TESTE				
ID	Risco	Descrição	Grau do Risco	Resposta
01	Hardware	Risco do equipamento (hardware) utilizado para execução da homologação do teste falhar.	Baixo	Mitigar – equipamento de reserva
03				
04				
05				

8 CRITÉRIOS DE CONCLUSÃO

Os critérios de conclusão/continuidade dos testes determinam qual a tolerância a falhas, ou seja, partir deste momento se os testes serão suspensos. Detalhando quais critérios serão utilizados para suspensão de um teste e retomada do desenvolvimento, ou mesmo continuidade apesar dos cenários de falha, ou seja, se após executado um determinado número de cenários de testes obtiver-se um número significativo de falhas que os testes são suspensos e a entrega deverá passar por uma revisão completa, ou mesmo se determinada falha ocorrer os testes não prosseguirão até que a mesma esteja corrigida. Aqui também se define as condições as quais o teste prosseguirá ou não se os dados de teste não estiverem em conformidade com a massa especificada anteriormente.

<Defina quais os critérios para aceitação, parada e retomada dos testes.>

Critério	Descrição
Cobertura	<ul style="list-style-type: none">▪ Todos os requisitos definidos no escopo deste projeto deverão ser testados.▪ Testar fluxos básico, alternativo e de exceção de cada requisito.▪ Todas as regras de negócio serão testadas.
Quando parar de testar	<ul style="list-style-type: none">▪ Quando nenhuma falha for encontrada após executar todos os casos de teste pelo menos uma vez.▪ Quando for encontrado um número excessivo de falhas logo no início dos testes.▪ Quando for encontrada alguma falha que inviabilize a continuidade dos testes.▪ Quando atingir no mínimo os seguintes critérios:<ul style="list-style-type: none">o Casos de Teste<ul style="list-style-type: none">▪ 100% dos casos de teste executadoso Defeitos<ul style="list-style-type: none">▪ 100% dos defeitos de severidade 1 corrigidos e verificados▪ 90% dos defeitos de severidade 2 corrigidos e verificados▪ 75% dos defeitos de severidade 3 corrigidos e verificados
Em que momento reportar as falhas	<ul style="list-style-type: none">▪ Reportar todas as falhas imediatamente após encontrá-las.
Teste de regressão	<ul style="list-style-type: none">▪ O teste de regressão será feito sobre a falha encontrada, sendo um fluxo básico e um fluxo alternativo do caso de uso.

9 CRONOGRAMA DO PROJETO DE TESTE DE SOFTWARE

<Descreva aqui a importância do cronograma do projeto no planejamento de teste com base nos autores da área de engenharia e teste de software, e insira o cronograma das atividades a serem executadas no projeto de teste.>

REFERÊNCIAS

PRESSMAN, Roger S; MAXIM, Bruce R. **Engenharia de Software: Uma Abordagem Profissional**. 8ª ed. Editora McGraw-Hill, 2016.

<Todos os autores citados no texto devem ser incluídos na lista de referências, que seguirão a formatação da NBR 6023:2002. A referência é composta pelos seguintes elementos: autor, título, edição, local, editora e data de publicação. Devem ser alinhadas à margem esquerda do texto com espaço simples e separadas umas das outras por um espaço simples. Usar o mesmo tipo e tamanho de fonte do texto do artigo, *Arial*, tamanho 12. Não deve ser numerada.>

EXEMPLOS E SUGESTÕES:

DELAMARO, Márcio Eduardo; MALDONADO, José Carlos; JINO, Mario. **Introdução ao Teste de Software**. São Paulo: Elsevier, 2007.

MOLINARI, Leonardo. **Inovação e Automação de Testes de Software**. Rio de Janeiro: Érica, 2010.

RIOS, Emerson. **Análise de riscos**: em projetos de teste de software. Rio de Janeiro: Alta Books, 2005.

RIOS, Emerson; MOREIRA FILHO, Trayahú. **Teste de Software**. Rio de Janeiro: Alta Books, 2006.

SOMMERVILLE, Ian. **Engenharia de software**. 9. ed. São Paulo: Pearson, 2011.