Plano de Testes

Nome do sistema: Cálculo de Salário Líquido - Mobills

Versão: 1.0.

Histórico das alterações

Data	Versão Descrição	Autor
10/12/2022	Definição do modelo do plano de testes	Bruna Borges

1 – Introdução

Este documento é baseado no cronograma geral do projeto e apresenta o planejamento das atividades de testes da funcionalidade de cálculo de salário líquido do sistema Mobills, tem como objetivo: entregar ótimos resultados para o negócio, otimizando tempo e recursos disponíveis; fazer a análise e priorização dos conjuntos de testes e base para a sequência de trabalho a ser feito e consulta de ferramentas e processos a serem utilizados.

2 – Requisitos a testar

REQ.01. Deve haver máscara de validação dos campos
REQ.02. Inputs de salário, descontos e dependentes devem aceitar apenas valores positivos.
REQ.03. Deve ser possível compartilhar o resultado do cálculo via redes sociais ou cópia do link.
REQ.04. Deve ser possível imprimir o resultado do cálculo.
REQ.05. Deve ser possível salvar o resultado do cálculo no formato .XLS
REQ.06. O resultado do cálculo deve levar em consideração os cálculos de INSS e IRPF adequadamente.

3 – Tipos de testes

Testes de contrato (verificação e validação, requisitos)
Testes manuais funcionais (fluxo, smoketest e regressão)
Testes de usabilidade e layout
Testes automatizados
Testes de API
Testes de performance
Testes de acessibilidade

3.1 – Estratégia de teste

- Análise do contrato ou documentação dos critérios fornecidos pelo PO para validar se o que já foi desenvolvido está em conformidade com os requisitos da aplicação.
- Análise geral da aplicação para o levantamento das possibilidades de casos de teste
- Calcular a cobertura de testes
- Modelagem (Preparação, ambiente, massa de testes, ferramentas)
- Os requisitos a serem testados serão priorizados de acordo com as necessidades do negócio e interdependência entre cada funcionalidade
- Levantamento das possibilidades de casos de testes a serem realizados.
- Ao final de cada Sprint, deverão ser apresentados os relatórios com os casos de teste, a documentação das evidências e caso sejam encontrados bugs, será enviado um relatório ao Scrum Master para que ele direcione aos devs de forma que façam a correção.

4 – Equipe e Infraestrutura

- **Equipe:**
  - PO: William Rosendo
  - QA: Bruna Borges
  - Equipe de desenvolvimento
- **Ferramentas:**
  - X Mind para plano de testes e levantamento dos casos de teste
  - Excel e Word para relatórios e levantamento da massa de dados
  - IntelliJ ou VScode
  - Plugin Cucumber
  - Git e GitHub para armazenamento e versionamento da documentação e Scripts de teste

    - Navegador Google Chrome
- **Documentação:**
    - Contrato ou documentos com as regras de aceite da funcionalidade
    - Swagger com as informações das APIs
- **Fornecedores Externos:**
    - Não há
- **Equipamentos:**
    - Notebook
- **Ambientes:**
    - Computador pessoal
    - Ambiente de homologação

5 – Cronograma

- **Sprint1: (12/12 a 16/12):**

  12/12:

  - Escrita dos casos de teste
  - Execução dos casos de teste
  - Relatório de Execução

  14/12:

  - Automação dos casos de teste 1/2

  16/12:

  - Review e Retro

- **Sprint2: (19/12 a 23/12):**

  19/12:

  - Automação dos casos de teste 2/2
  - Relatório de execução

21/12:

- Envio da planilha com report de Bugs

23/12:

- Review e Retro

- **Backlog:**

Testes de API

- Validar que a arquitetura da API Rest está de acordo com as recomendações gerais do estilo de arquitetura Rest

Testes de performance

- Funcionalidade cálculo de salário líquido
- APIs

Testes de acessibilidade

6 – Documentação complementar

- Documentação da funcionalidade: (A enviar pelo P.O)
- Tabela com a priorização dos requisitos: (A enviar pela equipe de Q.A)
- Casos de testes manuais: (A enviar pela equipe de Q.A)
- Execução dos casos de testes: (A enviar pela equipe de Q.A)
- Planilha com report de bugs: (A enviar pela equipe de Q.A)
- Link de acesso ao Swagger para realizar testes de API: (A enviar pela equipe de desenvolvimento)

7 – Notas Importantes

- Testes devem ser executados nos breakpoints: 1280 e 320px
- Deverá ser utilizado Scenario Outline
- Deverá ser utilizado Cucumber
- Deverão ser realizados testes positivos e negativos
- Entregas deverão ser realizadas através do GitHub