**PROJETO ORGANIZAI**

**Requisitos da disciplina Teste Qualidade e DevOps**

São Paulo

2024

**INTEGRANTES DO PROJETO e RA’S**

**Matheus Emiliano - 23025470**

**Guilherme Silva - 22023647**

**Matheus Moura - 23025570**

**Giovane Estevan - 23025240**

**Sumário**

[**1.**](#_30j0zll) **INTRODUÇÃO 3**

[**2.**](#_1fob9te) **Teste de Software. 3**

[**2.1.**](#_3znysh7) **Plano de Teste. 3**

[**2.2.**](#_2et92p0) **Apresentar 2 testes unitários. 3**

[**2.3.**](#_tyjcwt) **Apresentar 2 testes de componentes. 4**

[**2.4.**](#_3dy6vkm) **Apresentar um teste de sistema. 4**

[**3.**](#_1t3h5sf) **Qualidade de Software. 4**

[**3.1.**](#_4d34og8) **Indicar 4 atributos de qualidade de software e informar como foi aplicado no projeto integrador (PI)**.

[**3.2.**](#_2s8eyo1) **Apresentar um Modelo de qualidade de software. 6**

[**3.3.**](#_17dp8vu) **Apresentar um Processo (plano) de gerenciamento de qualidade de software. 7**

[**4. REFERÊNCIAS BIBLIOGRÁFICAS**](#_3rdcrjn) **7**

# INTRODUÇÃO

Com o OrganizAi, você tem todas as ferramentas necessárias para controlar suas finanças de forma simples e eficiente. Ao se cadastrar, você informa alguns dados básicos, como sua renda, o número de pessoas na sua casa e se você mora em área rural ou urbana. Com essas informações, o app cria um plano financeiro personalizado, te ajuda a cadastrar suas receitas e despesas e te mostra quais benefícios sociais você tem direito.

# Teste de Software.

# Plano de Teste.

### Objetivo.

Verificar a funcionalidade do aplicativo, garantindo que ele calcule corretamente os benefícios, registre as receitas de forma precisa e ofereça uma interface intuitiva para o usuário.

### Escopo.

* **Funcionalidades:** Cadastro de usuário, cálculo de benefícios, registro de receitas, interface do usuário.
* **Dados de teste:** Dados válidos e inválidos para os campos de cadastro, diferentes perfis de usuários (renda, número de dependentes, área rural/urbana), diversos valores de receitas..

### Critérios de Aceitação.

* O usuário deve conseguir se cadastrar com sucesso, fornecendo informações válidas.
* O cálculo dos benefícios deve ser preciso, considerando as informações fornecidas pelo usuário.
* As receitas devem ser registradas corretamente e estar disponíveis para consulta.
* A interface do usuário deve ser intuitiva e fácil de navegar.

# Apresentar 2 testes unitários.

* **Teste 1:** Verificar se o cálculo do benefício é correto para um usuário com renda mínima e sem dependentes, em área urbana.
* **Teste 2:** Verificar se o sistema rejeita o cadastro de um usuário com idade inferior a 18 anos.

# Apresentar 2 testes de componentes.

* **Teste 1:** Verificar se a funcionalidade de cadastro funciona corretamente, validando todos os campos obrigatórios.
* **Teste 2:** Verificar se a funcionalidade de registro de receitas permite adicionar múltiplas receitas em diferentes datas.

# Apresentar um teste de sistema.

* **Teste 1:** Simular o fluxo completo de um usuário, desde o cadastro até a consulta dos benefícios e histórico de receitas.

# Qualidade de Software.

# Indicar 4 atributos de qualidade de software e informar como foi aplicado no projeto integrador (PI).

### Funcionalidade:

* + **Descrição:** O software executa as funções para as quais foi projetado de forma correta e completa.
  + **Aplicação no projeto:**
    - **Cálculo preciso de benefícios:** O aplicativo calcula corretamente os benefícios a que o usuário tem direito, considerando sua renda, número de dependentes e situação (rural ou urbana).
    - **Registro preciso de receitas:** As receitas são registradas de forma correta e podem ser consultadas pelo usuário.
    - **Cadastro de usuários:** O sistema permite o cadastro de novos usuários, validando os dados informados e armazenando-os de forma segura.
    - **Interface intuitiva:** A interface do usuário é fácil de usar e permite que o usuário navegue pelo aplicativo de forma simples e eficiente.

### Confiabilidade:

* + **Descrição:** O software não falha com frequência e pode ser usado por longos períodos sem interrupções.
  + **Aplicação no projeto:**
    - **Estabilidade:** O aplicativo deve funcionar de forma estável, sem apresentar crashes ou erros inesperados.
    - **Teste extensivo:** A realização de testes unitários, de componentes e de sistema garante a qualidade e a confiabilidade do software.

### Usabilidade:

* + **Descrição:** O software é fácil de aprender e usar.
  + **Aplicação no projeto:**
    - **Interface intuitiva:** A interface do usuário é projetada para ser fácil de entender e usar, mesmo para usuários com pouca experiência em tecnologia.
    - **Ajuda e documentação:** O aplicativo deve fornecer ajuda e documentação claras para auxiliar o usuário.
    - **Feedback:** O sistema deve fornecer feedback claro ao usuário sobre suas ações, como mensagens de confirmação e alertas de erro.

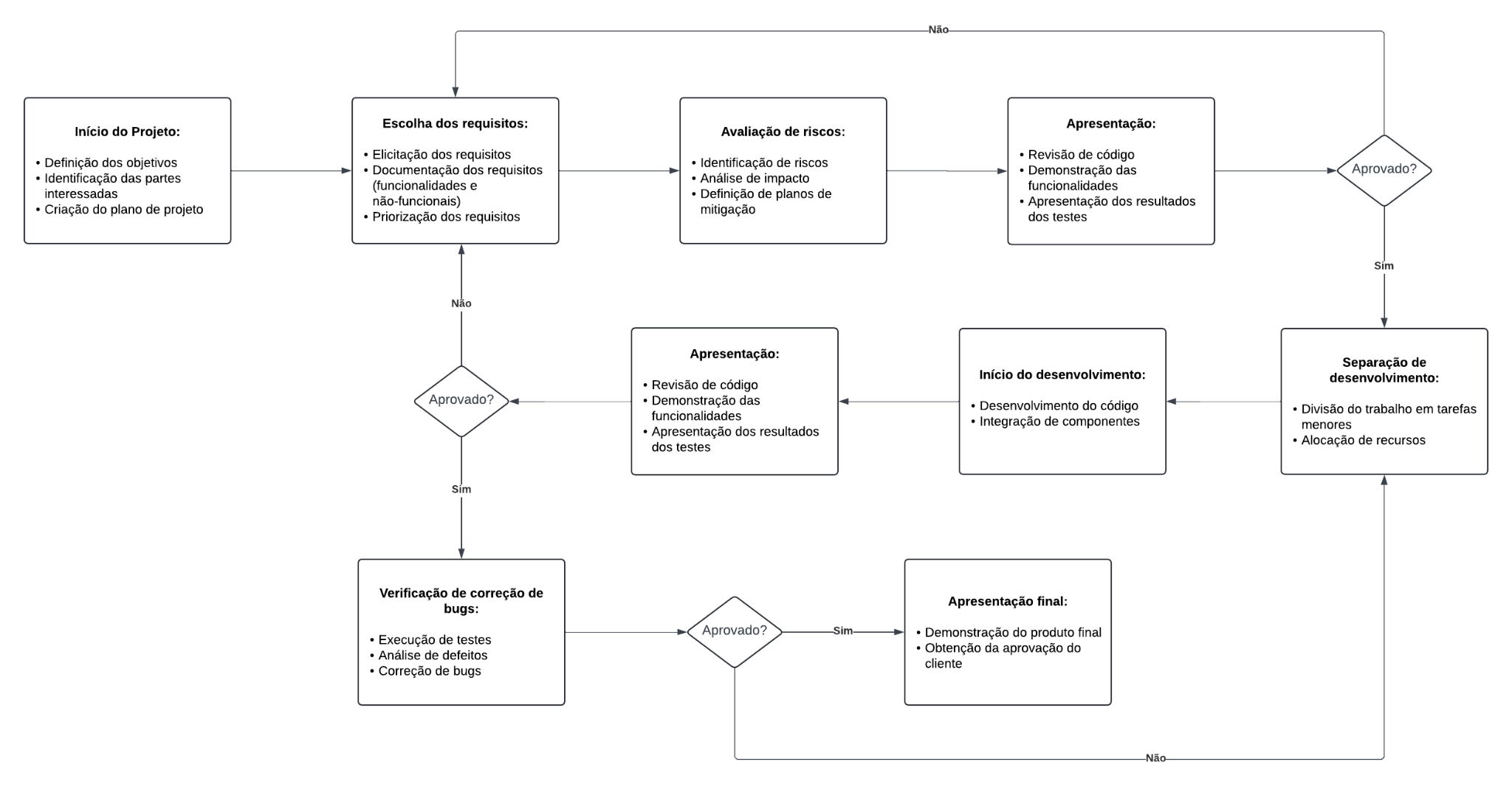
### Eficiência:

* + **Descrição:** O software responde às solicitações do usuário de forma rápida e utiliza os recursos do sistema de forma eficiente.
  + **Aplicação no projeto:**
    - **Desempenho:** O aplicativo deve responder às solicitações do usuário de forma rápida, mesmo com um grande volume de dados.
    - **Otimização do código:** O código deve ser otimizado para garantir o melhor desempenho possível.
    - **Gerenciamento de recursos:** O aplicativo deve utilizar os recursos do sistema de forma eficiente, evitando o consumo excessivo de memória e processamento.

# 

# Apresentar um Modelo de qualidade de software.

# Apresentar um Processo (plano) de gerenciamento de qualidade de software



# REFERÊNCIAS BIBLIOGRÁFICAS

SOMMERVILLE, I. **Engenharia de Software.** 11ª Edição. São Paulo: Pearson Addison-Wesley, 2017.