



**UNIVERSIDADE FEDERAL DO CEARÁ**  
**Campus de Quixadá**

# **Plano de Medição de Qualidade**

Gabriel Maia Gondim - 478943

**Responsável:** Camilo Almendra

# ÍNDICE

GLOSSÁRIO	3
HISTÓRICO DE REVISÕES	3
1. INTRODUÇÃO	4
1.1. Descrição dos produtos a serem avaliados	4
1.2. Objetivos da avaliação	4
2. MÉTODO	4
2.1. Participantes (caso necessite / depende da característica escolhida)	4
2.2. Ambiente de avaliação	4
2.3. Procedimentos da Avaliação	4
2.4. Medidas de Software	4
2.4.1. Eficácia	4
2.4.2. Eficiência	5
2.5. Procedimentos de Interpretação	5
3. REFERÊNCIAS	5

## GLOSSÁRIO

[Inclua definições para siglas citadas no Plano de Medição. Veja exemplos abaixo]

Siglas	Definição
UFC	Universidade Federal do Ceará

## HISTÓRICO DE REVISÕES

[Inclua o histórico das versões do Plano de Medição.]

Data	Versão	Descrição	Responsável
18/10/2022	1.0	Criação do documento.	Gabriel Maia Gondim
25/10/2022	1.1	Preenchido documento e adicionadas métricas.	Gabriel Maia Gondim

## 1. INTRODUÇÃO

Este documento se destina aos responsáveis pela avaliação e garantia de qualidade do produto e também para os responsáveis pela execução das avaliações manuais deste.

### 1.1. Descrição dos produtos a serem avaliados

Esse plano busca avaliar as principais funcionalidades do aplicativo NewHome, um sistema de adoção de animais. Para todas as métricas, todas as funcionalidades do sistema serão avaliadas.

### 1.2. Objetivos da avaliação

O objetivo da avaliação é validar a eficiência, para garantir que o sistema atende os padrões de usabilidade, para garantir que o sistema satisfaz os usuários, manutenibilidade, para garantir a qualidade do código e uma alta facilidade de mudanças deste e confiabilidade, para garantir as principais funcionalidades do produto NewHome. Usuários representativos serão solicitados a realizar tarefas típicas da aplicação.

Analisar	NewHome
Para o propósito de	melhorar e avaliar a qualidade
Com respeito a	usabilidade, manutenibilidade e confiabilidade
Do ponto de vista	do usuário
No contexto de	dispositivos móveis Android

## 2. MÉTODO

### 2.1. Ambiente de avaliação

- Em todas as métricas, todas as funcionalidades do sistema serão avaliadas;
- Para os dados coletados automaticamente, a coleta será realizada pelo SonarQube, uma plataforma que inspeciona e coleta dados de qualidade de código automaticamente. Será usado um plugin do Jenkins no SonarQube para permitir disparar a análise do código automaticamente na pipeline do Jenkins;
- Para os dados coletados manualmente, a coleta será realizada em um laboratório, em um ambiente real usando um celular Samsung Galaxy A30.

### 2.2. Procedimentos da Avaliação

Ao chegar, os participantes foram informados de que a usabilidade do produto NewHome seria avaliada para descobrir se o produto satisfaz suas necessidades. Os usuários foram informados de que não era um teste de suas habilidades e sim da usabilidade da aplicação. Em seguida, o avaliador explicou para os usuários como seria a avaliação, apresentando o dispositivo a ser utilizado, o laboratório e informou que sua interação seria registrada. Os usuários foram convidados a assinar um formulário de liberação.

### 2.3. Medidas de Software

#### 2.3.1. Usabilidade

Nome	Descrição	Função de Medição	Método
Taxa de Sucesso da Tarefa	Qual a razão entre tarefas realizadas com	$X = (B/A) * 100$ A = Número total de tarefas tentadas	Teste com usuários

	sucesso e tarefas tentadas?	B = Número de tarefas realizadas com sucesso	
Taxa de erro do usuário	Qual a razão entre o número de erros obtidos e o número total de possíveis erros?	$X = (A / (B * C)) * 100$ A = Total de erros ocorridos B = Total de tentativas C = Número de possíveis erros	Teste com usuários

### 2.3.2. Manutenibilidade

Nome	Descrição	Função de Medição	Método
Razão de débito técnico	Razão entre o custo para corrigir software e o custo para desenvolver este.	$X = A/(B*C)$ A = Custo de remediação B = Custo para desenvolver 1 linha de código (0.06 dias) C = Número de linhas de código	Coletado automaticamente pelo SonarQube
Code smells	Número de code smells no código	X = Número de code smells	Coletado automaticamente pelo SonarQube

### 2.3.3. Confiabilidade

Nome	Descrição	Função de Medição	Método
Razão de bugs	Número de bugs por funcionalidade do sistema	$X = A/B$ A = Número de bugs B = Número de funcionalidades implementadas no sistema	Coletado automaticamente pelo SonarQube

## 2.4. Procedimentos de Interpretação

- Para taxa de sucesso da tarefa, o valor é uma porcentagem, onde 0% indica que nenhuma tarefa foi realizada com sucesso e 100% indica que todas as tarefas foram realizadas com sucesso. Para essa métrica, o intervalo de valores considerados aceitáveis é de mais de 95%;
- Para taxa de erro do usuário, o valor é uma porcentagem que indica a quantidade média de erros ocorridos em comparação com o total possível de erros, onde 0% indica que, em média, nenhum dos erros possíveis ocorre, e 100% indica que, em média, todos os erros possíveis sempre ocorrem. Para essa métrica, o intervalo de valores considerados aceitáveis é de menos de 10%;
- Para razão de débito técnico, o valor é um índice que indica a taxa de retrabalho, indica uma razão entre o custo gasto com retrabalho e o custo do software em si, onde 0 indica que não há custos com retrabalho, 1 que o mesmo tempo gasto desenvolvendo foi também gasto com retrabalho em correções do software e maior que 1 indica que houve mais tempo corrigindo o software do que realmente desenvolvendo novas funcionalidades. Para essa métrica, o intervalo de valores aceitáveis é de menos que 0.3 (indica que menos de 30% do tempo desenvolvendo foi gasto com retrabalho);
- Para code smells, o valor indica a quantidade de code smells no código. Para essa métrica, o intervalo de valores aceitáveis é de menos que 20;
- Para razão de bugs, o valor indica a quantidade média de bugs para cada funcionalidade do sistema, sendo 0 nenhum bug por funcionalidade, 1 um bug por funcionalidade, assim por diante. Para essa métrica, o intervalo de valores aceitáveis é de menos que 3;

### **3. REFERÊNCIAS**

ISO/IEC 25000. Software Engineering - Software Product Quality Requirements and Evaluation (SQuaRE) – Guide to SQuaRE. v. 2005, 2005.

ISO/IEC 9126. Software Engineering – Product Quality – Part 1. 2001