

## **Plano de Teste**

**Nome do Projeto:** Projeto Lira

### **Equipe:**

Adriana Lino de Macedo Fonseca

Cleber Fontes da Silva

Thalles Gomes de Melo

Yasmin Reis Tello

### **1. Introdução**

**Este plano de teste tem por objetivo identificar o que fará parte ou não do escopo de teste da versão, bem como as premissas e os riscos envolvidos.**

Fará parte dos testes os métodos e construtores criados nas classes: Aula, Endereço, Instrumento, Música e Pessoa. Escolhemos a criação de um objeto da classe pessoa chamada Maria, que está linkado ao endereço e as aulas. Assim também escolhemos a música Aquarela de Toquinho para participar dos testes dos métodos de música e instrumento.

### **2. Escopo de Teste**

**Especificar o que fará parte do escopo de teste (principais classes e métodos).**

As classes são: Aula, Endereço, Instrumento, Música e Pessoa.

Na classe Aula, temos os métodos: Assistir aula, e Dar aula.

Na classe Endereço temos apenas o construtor apresentado no ToString()

Na classe Instrumento temos os métodos: Vender, Afinar e Contar tempo.

Na classe Música temos os métodos: Executar, Pausar, Gravar, Compartilhar, Ler e Pesquisar

Na classe Pessoa temos apenas o construtor apresentado no ToString().

### **3. Fora do Escopo de Teste**

**Especificar o que não será testado. Exemplo: não será testar a Interface.**

Não será testado todos os getters e setters, apenas o necessário.

### **4. Considerações de Infraestrutura**

**Descrever quais as configurações necessárias para a realização dos testes. Exemplo.**

**Plataforma, computador.**

Será necessário a instalação da máquina virtual java em qualquer sistema operacional. No caso testamos no Windows, e fizemos na IDE NetBeans.

## **5. Premissas**

**Descrever as premissas (se houver). Exemplo: considerar que todas as variáveis foram declaradas como private.**

Todos os atributos foram setados como privados exceto Instrumento na classe Música e Endereço na classe Pessoa pois foram utilizadas na Agregação.

## **6. Riscos**

**Identificar os riscos do projeto que poderão impactar na realização dos testes. Exemplo: Este teste não valida as regras de negócios relacionadas à base de dados.**

Além desse teste não validar as regras de negócio relacionada a base de dados, ele apenas simula uma saída dos métodos que não é a real. Apenas mostra um texto na tela.

## **7. Conclusão**

**Avalie os resultados finais.**

No final dos testes podemos verificar como funcionaria o sistema do nosso aplicativo em termos concretos de herança e agregação, que acabou ficando diferente do que havíamos montado nos diagramas da UML. Fazendo os testes verificamos a verdadeira relação entre as classes e como poderia o usuário relacionar com o aplicativo.