

## Relatório – Trabalho III

André Cidade Irie e Herich Gabriel de Campos

1. **Objetivo dos Testes:** O objetivo principal é garantir que os métodos de acesso (getters e setters) e o construtor das classes Professor e Estudante funcionem corretamente.

### 2. Testes Realizados:

#### Para a classe Professor:

- `testGetProfessorID()`: Verifica se o método `getProfessorID()` retorna o ID correto. Uso de `assertEquals(1, professor.getProfessorID())`.
- `testSetProfessorID()`: Verifica se o método `setProfessorID()` altera corretamente o ID. A asserção foi `assertNotEquals(1, professor.getProfessorID())` para garantir que o ID foi modificado.
- `testProfessorConstructor()`: Testa o construtor da classe. Cria um novo objeto Professor e assegura que ele não seja null com `assertNotNull(professor1)`.
- `testToString()`: Verifica o método `toString()`. Compara a saída com a string esperada usando `assertEquals(expected, professor.toString())`.

#### Para a classe Estudante:

- `testGetAndSetEstudanteID()`: Verifica o funcionamento correto dos métodos `getEstudanteID()` e `setEstudanteID()`. Utiliza as asserções `assertEquals(1, estudante.getEstudanteID())` para o valor inicial e `assertEquals(2, estudante.getEstudanteID())` após a alteração.
- `testGetAndSetNome()`: Valida os métodos `getNome()` e `setNome()`. A asserção inicial é `assertEquals("Carlos Silva", estudante.getNome())` e, após a alteração, `assertEquals("Ana Pereira", estudante.getNome())`.
- `testGetAndSetIdade()`: Testa os métodos `getIdade()` e `setIdade()`. O valor inicial é verificado com `assertEquals(20, estudante.getIdade())`, e o valor após a alteração com `assertEquals(25, estudante.getIdade())`.
- `testToString()`: Avalia o método `toString()` da classe Estudante. Compara a saída com a string esperada usando `assertEquals(expected, estudante.toString())`.

Esses testes abrangem os principais comportamentos esperados das classes Professor e Estudante, garantindo a confiabilidade dos métodos implementados.