Testes Funcionais - Evolução de Regras de Negócio

João Paulo Pereira de Medeiros[1]

[1]Instituto Metrópole Digital – Universidade Federal do Rio Grande do Norte (UFRN)
59.078.970 – Natal – RN – Brazil

joao.pereira.118@ufrn.edu.br

Abstract. *This paper addresses the impact of business rules updates into automated tests regarding UFRN's approval use case in 2024.1.*

Resumo. *O presente trabalho descreve o impacto que evoluções em regras de negócio têm sobre casos de testes automatizados. Para tanto, usou-se como caso de uso, a regra para aprovação por nota na UFRN em vigor a partir de 2024.1.*

1. Contextualização

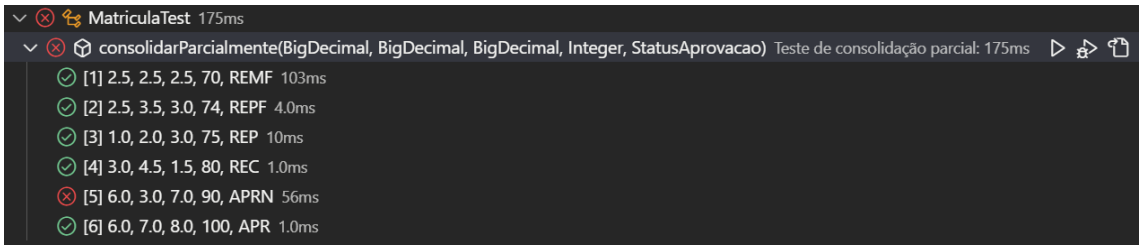
A resolução nº 016/2023-CONSEPE, de 04 de julho de 2023, instaurou a nova regra para aprovação por nota na UFRN em vigor a partir de 2024.1. O artigo 114 considera aprovado o estudante com média parcial igual ou superior a 6,0 (seis) e com notas iguais ou superiores a 4,0 (quatro) em cada unidade.

Por essa razão, houve a alteração da regra de negócio no código fonte do projeto "Matrícula" e, por conseguinte, dos valores para o teste parametrizável. Com isso, tornou-se nítido o impacto que evoluções em regras de negócio possuem acerca dos casos de testes automatizados.

2. Implicações em casos de testes automatizados

A primeira implicação foi a quebra do caso de teste com status "APRN", isto é, aprovado por nota (Figura 01) no arquivo MatriculaTest.java visto que este caso não existiria mais.

Figura 01– Execução inadequada do caso de teste



Fonte: o autor (2023)

A segunda implicação da mudança da regra de negócios para os testes foi a remoção da quinta linha do @CsvSource (Figura 02).

Figura 02 – Mudança em CsvSource

```
@ParameterizedTest
@CsvSource({
    "2.5, 2.5, 2.5, 70, REMF",
    "2.5, 3.5, 3.0, 74, REPF",
    "1.0, 2.0, 3.0, 75, REP",
    "3.0, 4.5, 1.5, 80, REC",
    "6.0, 5.0, 8.0, 100, APR"
})
```

Fonte: o autor (2023)

3. Implicações em código-fonte

A classe Matricula.java teve como alteração o método estaAprovado(). Este passou a considerar diferentes *thresholds*. Além disso, foram adicionados *static final atributes* contendo os valores quatro e seis como *BigDecimal* no início do arquivo.

Figura 03 – Mudança em método de Matricula.java

```
/**
 * É considerado aprovado, quanto à avaliação do rendimento acadêmico, o estudante que tem média parcial igual ou superior a 6,0 (seis)
 * com rendimento acadêmico igual ou superior a 4,0(quatro) em todas as unidades.
 * @return Boolean
 */
public Boolean estaAprovado()
{
    return this.mediaParcial.compareTo(SEIS) >= 0 &&
    (this.nota1.compareTo(QUATRO) >= 0 && this.nota2.compareTo(QUATRO) >= 0 && this.nota3.compareTo(QUATRO) >= 0);
}
```

Fonte: o autor (2023)

Além disso, removeu-se a atribuição do status no arquivo, bem como o campo “APRN” (aprovado por nota) do enum presente em StatusAprovacao.java.

Repare

References

CONSEPE. *Resolução nº 0166/2023-CONSEPE, de 04 de julho de 2023*. UFRN. 2023.