

## Teste de Caixa Branca: Validação de Preço (RF010):

### 1. Contexto e Lógica Interna Assumida

Para realizar um teste de **Caixa Branca**, precisamos *assumir* a lógica de programação subjacente que realiza a validação do preço do produto.

#### Lógica Assumida (Trecho de Código para o Mínimo):

```
// Código simplificado do backend para validação de preço
function validar_preco_produto(preco) {
  // Decisão 1: Verifica se o preço é zero ou negativo
  if (preco <= 0) {
    // RAMO A: Preço Inválido (deve ser bloqueado)
    return 'ERRO_PRECO_INVALIDO';
  } else {
    // RAMO B: Preço Válido (continua a validação)
    // ... Aqui o código segue para verificar o limite máximo (R$ 999,99) ...
    return 'OK';
  }
}
```

### 2. Objetivo da Cobertura de Decisão

O teste deve garantir que a Decisão 1 (if (preco <= 0)) seja exercida em ambos os caminhos possíveis:

1. Caminho 1 (TRUE / RAMO A): A condição é Verdadeira.

2. Caminho 2 (FALSE / RAMO B): A condição é Falsa.

### 3. Casos de Teste de Caixa Branca

D do Caso	Objetivo de Cobertura	Dado de Entrada (Preço)	Resultado Esperado	Caminho Coberto	Técnica
CT_CB_RF010_01	Cobrir o <b>RAMO A</b> (Decisão 1 = TRUE) e validar a correção do BUG-452.	R\$ 0,00	O sistema <b>deve retornar ERRO_PRECO_INVALIDO</b> (Bloqueado).	Decisão 1: TRUE	Cobertura de Decisão
CT_CB_RF010_02	Cobrir o <b>RAMO B</b> (Decisão 1 = FALSE) e permitir que o código avance.	R\$ 0,01	O sistema <b>deve retornar OK</b> ou avançar para a próxima validação (Limite Máximo).	Decisão 1: FALSE	Cobertura de Decisão