TesteCorrelacaoDePearson.hs

```
1 module TesteCorrelacaoDePearson(main) where
2
3 import Test.HUnit
4 import CorrelacaoDePearson
5
6 testaMediaListaVazia :: Test
7 testaMediaListaVazia = TestCase (assertEqual "MÃ©dia de lista vazia" 0
(calculaMedia []))
8
9 testaMediaLista :: Test
10 testaMediaLista = TestCase (assertEqual "MÃ©dia de uma lista" 3 (calculaMedia
[3,3,3]))
11
12 testaVetorXVazia :: Test
13 testaVetorXVazia = TestCase (assertEqual "Vetor X" 0 (length(vetorX [])) )
14
15 testaVetorX :: Test
16 testaVetorX = TestCase (assertEqual "Vetor X" [1,2,3] (vetorX [1,2,3,4]))
17
18 testaVetorYVazia :: Test
19 testaVetorYVazia = TestCase (assertEqual "Vetor Y" 0 (length(vetorY [])) )
20
21 testaVetorY :: Test
22 testaVetorY = TestCase (assertEqual "Vetor Y" [1,2,3] (vetorY [0,1,2,3]))
23
24 testaSomaQuadradoVetorVazia :: Test
25 testaSomaQuadradoVetorVazia = TestCase (assertEqual "Soma quadrÃ¡tica de uma
lista vazia" 0 (somaQuadradoVetor []))
26
27 testaSomaQuadradoVetor :: Test
28 testaSomaQuadradoVetor = TestCase (assertEqual "Soma quadrÃ¡tica de uma lista
[1,2,3]" 14 (somaQuadradoVetor [1,2,3]))
29
30 suiteDeTeste :: Test
31 suiteDeTeste = TestList [testaMediaListaVazia,testaMediaLista,testaVetorX,
testaVetorXVazia, testaVetorY, testaVetorYVazia,
testaSomaQuadradoVetor,testaSomaQuadradoVetorVazia]
32
33 main :: IO Counts
```

```
34 main = runTestTT suiteDeTeste
```

TesteFibonacci.hs

```
1 module TesteFibonacci(main) where
2
3 import Test.HUnit
4 import Fibonacci
5
6 testaSuporte :: Test
7 testaSuporte = TestCase (assertEqual "Suporte de uma lista Comum" 0 (suporte
[0,1,2,3,4]))
8
9 testaResistencia :: Test
10 testaResistencia = TestCase (assertEqual "ResistÃªncia de uma lista Comum" 4
(resistencia [0,1,2,3,4]))
11
12 testaRetracao :: Test
13 testaRetracao = TestCase (assertEqual "RetraÃ§Ã£o Simples" 1 (retracao 1 1 0))
14
15 suiteDeTeste :: Test
16 suiteDeTeste = TestList [testaRetracao,testaSuporte,testaResistencia]
17
18 main :: IO Counts
19 main = runTestTT suiteDeTeste
```

TesteMinimosQuadrados.hs

```
1 module TesteMinimosQuadrados(main) where
2
3 import Test.HUnit
4 import MinimosQuadrados
5
6 testaNumeradorVazio :: Test
7 testaNumeradorVazio = TestCase (assertEqual "Numerador com lista vazia" 0
(numerador [] []))
8
9 testaDenominadorVazio :: Test
10 testaDenominadorVazio = TestCase (assertEqual "Denominador com lista vazia" 0
```

```
(denominador []))
11
12 suiteDeTeste :: Test
13 suiteDeTeste = TestList [testaNumeradorVazio, testaDenominadorVazio]
14
15 main :: IO Counts
16 main = runTestTT suiteDeTeste
```