

# Trabalho Extra – LINGUAGEM GHERKIN

## Exercício 5

Funcionalidade: Testar o Método somar da Classe  
MinhaClasseAvancado

Cenário: Somando elementos de uma lista

Dado: Uma lista de números [6, 0, -5]

Quando: Eu somar os elementos da lista com o método somar da classe MinhaClasseAvancado

Então: o resultado deve ser 1

Cenário: Somando elementos de uma lista com um número

Dado: Uma lista de números [2]

Quando: Eu somar o elemento da lista com o método somar da classe MinhaClasseAvancado

Então: o resultado deve ser o próprio número

Cenário: Somando elementos de uma lista com decimais

Dado: Uma lista de números [0.2, 0.9]

Quando: Eu somar os elementos da lista com o método somar da classe MinhaClasseAvancado

Então: o resultado deve ser 1.1

Cenário: Somando elementos de uma lista com decimais diferentes

Dado: Uma lista de números [0.100, 2]

Quando: Eu somar os elementos da lista com o método somar da classe MinhaClasseAvancado

Então: o resultado deve ser 2.1

Cenário: Somando elementos de uma lista vazia

Dado: Uma lista de números vazia

Quando: Eu somar os elementos da lista com o método somar da classe MinhaClasseAvancado

Então: o resultado deve ser "Array não deve ser vazia"

Cenário: Somando elementos de uma lista com argumentos inválidos

Dado: Uma lista de números [1, false]

Quando: Eu somar os elementos da lista com o método somar da classe MinhaClasseAvancado

Então: o resultado deve ser uma mensagem de erro

E: não deve retornar erro com boolean

Mas: uma mensagem de erro com "Array deve conter apenas números"

Dado: Uma lista de números [1, null]

Quando: Eu somar os elementos da lista com o método somar da classe MinhaClasseAvancado

Então: o resultado deve ser uma mensagem de erro

E: não deve retornar erro com null

Mas: uma mensagem de erro com "Array deve conter apenas números"

Dado: Uma lista de números [1, ( )]

Quando: Eu somar os elementos da lista com o método somar da classe MinhaClasseAvancado

Então: o resultado deve ser uma mensagem de erro

E: não deve retornar erro com array

Mas: uma mensagem de erro com "Array deve conter apenas números"

Dado: Uma lista de números [1, new stdClass()]

Quando: Eu somar os elementos da lista com o método somar da classe MinhaClasseAvancado

Então: o resultado deve ser uma mensagem de erro

E: não deve retornar erro com objeto

Mas: uma mensagem de erro com "Array deve conter apenas números"

Dado: Uma lista de números [1, "1"]

Quando: Eu somar os elementos da lista com o método somar da classe MinhaClasseAvancado

Então: o resultado deve ser uma mensagem de erro

E: não deve retornar erro com string numérica

Mas: uma mensagem de erro com "Array deve conter apenas números"

Funcionalidade: Testar o Método subtrair da Classe  
MinhaClasseAvancado

Cenário: Subtraindo elementos de uma lista

Dado: Uma lista de números [5, 5]

Quando: Eu subtrair os elementos da lista com o método subtrair da classe MinhaClasseAvancado

Então: o resultado deve ser 0

Cenário: Subtraindo elementos de uma lista com zeros

Dado: Uma lista de números [0, 0]

Quando: Eu subtrair os elementos da lista com o método subtrair da classe MinhaClasseAvancado

Então: o resultado deve ser 0

Cenário: Subtraindo elementos de uma lista com negativos

Dado: Uma lista de números [-5, -5]

Quando: Eu subtrair os elementos da lista com o método subtrair da classe MinhaClasseAvancado

Então: o resultado deve ser 0

E: não -10

Cenário: Subtraindo elementos de uma lista com positivo e negativo

Dado: Uma lista de números [5, -5]

Quando: Eu subtrair os elementos da lista com o método subtrair da classe MinhaClasseAvancado

Então: o resultado deve ser 10

E: não 0

Cenário: Subtraindo elementos de uma lista com um número

Dado: Uma lista de números [2]

Quando: Eu subtrair o elemento da lista com o método subtrair da classe MinhaClasseAvancado

Então: o resultado deve ser proprio número

Cenário: Subtraindo elementos de uma lista com decimais

Dado: Uma lista de números [2.2, 1.1]

Quando: Eu subtraio os elementos da lista com o método subtrair da classe MinhaClasseAvancado

Então: o resultado deve ser 1.1

Cenário: Subtraindo elementos de uma lista com decimais diferentes

Dado: Uma lista de números [3.2, 1.100]

Quando: Eu subtraio os elementos da lista com o método subtrair da classe MinhaClasseAvancado

Então: o resultado deve ser 2.1

Cenário: Subtraindo elementos de uma lista vazia

Dado: Uma lista de números vazia

Quando: Eu subtraio os elementos da lista com o método subtrair da classe MinhaClasseAvancado

Então: o resultado deve ser "Apenas números são permitidos"

Cenário: Subtraindo elementos de uma lista com argumentos inválidos

Dado: Uma lista de números [1, false]

Quando: Eu subtraio os elementos da lista com o método subtrair da classe MinhaClasseAvancado

Então: o resultado deve ser uma mensagem de erro

E: não deve retornar erro com boolean

Mas: uma mensagem de erro com "Apenas números são permitidos"

Cenário: Subtraindo elementos de uma lista com argumentos inválidos

Dado: Uma lista de números [1, null]

Quando: Eu subtrair os elementos da lista com o método subtrair da classe MinhaClasseAvancado

Então: o resultado deve ser uma mensagem de erro

E: não deve retornar erro com null

Mas: uma mensagem de erro com "Apenas números são permitidos"

Dado: Uma lista de números [1, () ]

Quando: Eu subtrair os elementos da lista com o método subtrair da classe MinhaClasseAvancado

Então: o resultado deve ser uma mensagem de erro

E: não deve retornar erro com array

Mas: uma mensagem de erro com "Apenas números são permitidos"

Dado: Uma lista de números [1, new stdClass()]

Quando: Eu subtrair os elementos da lista com o método subtrair da classe MinhaClasseAvancado

Então: o resultado deve ser uma mensagem de erro

E: não deve retornar erro com objeto

Mas: uma mensagem de erro com "Apenas números são permitidos"

Dado: Uma lista de números [1, "1"]

Quando: Eu subtrair os elementos da lista com o método subtrair da classe MinhaClasseAvancado

Então: o resultado deve ser uma mensagem de erro

E: não deve retornar erro com string numérica

Mas: uma mensagem de erro com "Apenas números são permitidos"