

## TESTES UNITÁRIOS – Classe: Funcionario

### ◆ getDadosFuncionario()

Entrada	Saída	Saída Esperada
Objeto com todos os dados preenchidos	String com dados	"Nome: João, CPF: 123..."
Objeto com campos vazios	ERRO ou String incompleta	"Dados incompletos" (se validado)

### ◆ getNome() / setNome(String nome)

Entrada	Saída	Saída Esperada
"Ana"	OK	"Ana"
""	ERRO	"Nome não pode ser vazio"
null	ERRO	"Insira um nome"

### ◆ getCpf() / setCpf(String cpf)

Entrada	Saída	Saída Esperada
"123.456.789-00"	OK	"123.456.789-00"
""	ERRO	"CPF obrigatório"
"abc"	ERRO	"Formato de CPF inválido"
null	ERRO	"CPF obrigatório"

### ◆ getCargo() / setCargo(String cargo)

Entrada	Saída	Saída Esperada
"Analista"	OK	"Analista"
""	ERRO	"Cargo inválido"
null	ERRO	"Cargo obrigatório"

### ◆ getDataAdmissao() / setDataAdmissao(LocalDate data)

Entrada	Saída	Saída Esperada
2020-01-01	OK	2020-01-01
null	ERRO	"Data obrigatória"
2099-01-01	ERRO	"Data não pode ser futura"

- ◆ `getSalarioBruto() / setSalarioBruto(BigDecimal valor)`

Entrada	Saída	Saída Esperada
3500.00	OK	3500.00
-100.00	ERRO	"Salário não pode ser negativo"
null	ERRO	"Salário inválido"

- ◆ `getNumeroDependentes() / setNumeroDependentes(int qtd)`

Entrada	Saída	Saída Esperada
2	OK	2
-1	ERRO	"Número de dependentes inválido"
999999	ERRO	"Valor muito grande"

- ◆ `isRecebeAdicionalPericulosidade() / setRecebeAdicionalPericulosidade(boolean valor)`

Entrada	Saída	Saída Esperada
true	OK	true
false	OK	false

- ◆ `getGrauInsalubridade() / setGrauInsalubridade(enum grau)`

Entrada	Saída	Saída Esperada
NENHUM	OK	NENHUM
ALTO	OK	ALTO
"EXTREMO"	ERRO	"Grau inválido"
null	ERRO	"Grau obrigatório"

- ◆ `getValorDiarioValeAlimentacao() / setValorDiarioValeAlimentacao(BigDecimal valor)`

Entrada	Saída	Saída Esperada
20.00	OK	20.00
-5.00	ERRO	"Valor inválido"
null	ERRO	"Valor obrigatório"

◆ getValorValeTransporteEntregue() / setValorValeTransporteEntregue(BigDecimal valor)

Entrada	Saída	Saída Esperada
150.00	OK	150.00
-10.00	ERRO	"Valor inválido"
null	ERRO	"Valor obrigatório"

◆ getDiasTrabalhadosMes() / setDiasTrabalhadosMes(int dias)

Entrada	Saída	Saída Esperada
22	OK	22
-1	ERRO	"Dias inválidos"
35	ERRO	"Número de dias acima do permitido"

◆ getHorasSemanais() / setHorasSemanais(int horas)

Entrada	Saída	Saída Esperada
40	OK	40
-5	ERRO	"Horas inválidas"
200	ERRO	"Horas semanais acima do permitido"

## TESTES UNITÁRIOS – Classe: FolhaPagamento

◆ getMesReferencia() / setMesReferencia(String ref)

Entrada	Saída	Saída Esperada
"08/2025"	OK	"08/2025"
"2025-08"	ERRO	"Formato inválido"
""	ERRO	"Mês de referência obrigatório"
null	ERRO	"Mês de referência obrigatório"

◆ getSalarioHora() / setSalarioHora(BigDecimal valor)

Entrada	Saída	Saída Esperada
20.00	OK	20.00
-5.00	ERRO	"Salário-hora não pode ser negativo"
null	ERRO	"Valor obrigatório"

◆ getTotalProventos() / setTotalProventos(BigDecimal valor)

Entrada	Saída	Saída Esperada
5000.00	OK	5000.00
-10.00	ERRO	"Total de proventos inválido"
null	ERRO	"Valor obrigatório"

◆ getTotalDescontos() / setTotalDescontos(BigDecimal valor)

Entrada	Saída	Saída Esperada
1000.00	OK	1000.00
-50.00	ERRO	"Desconto inválido"
null	ERRO	"Valor obrigatório"

◆ getSalarioLiquido() / setSalarioLiquido(BigDecimal valor)

Entrada	Saída	Saída Esperada
4000.00	OK	4000.00
-100.00	ERRO	"Salário líquido inválido"
null	ERRO	"Valor obrigatório"

◆ `getValorAdicionalPericulosidade()` / `setValorAdicionalPericulosidade(BigDecimal valor)`

Entrada	Saída	Saída Esperada
500.00	OK	500.00
-1.00	ERRO	"Valor inválido"
null	ERRO	"Valor obrigatório"

◆ `getValorAdicionalInsalubridade()` / `setValorAdicionalInsalubridade(BigDecimal valor)`

Entrada	Saída	Saída Esperada
300.00	OK	300.00
-50.00	ERRO	"Valor inválido"
null	ERRO	"Valor obrigatório"

◆ `getValorValeAlimentacao()` / `setValorValeAlimentacao(BigDecimal valor)`

Entrada	Saída	Saída Esperada
200.00	OK	200.00
-5.00	ERRO	"Valor inválido"
null	ERRO	"Valor obrigatório"

◆ `getValorDescontoValeTransporte()` / `setValorValeTransporte(BigDecimal valor)`

Entrada	Saída	Saída Esperada
150.00	OK	150.00
-10.00	ERRO	"Desconto inválido"
null	ERRO	"Valor obrigatório"

◆ `getValorDescontoINSS()` / `setValorDescontoINSS(BigDecimal valor)`

Entrada	Saída	Saída Esperada
500.00	OK	500.00
-1.00	ERRO	"Valor inválido"
null	ERRO	"Valor obrigatório"

◆ `getValorDescontoFGTS()` / `setValorDescontoFGTS(BigDecimal valor)`

Entrada	Saída	Saída Esperada
400.00	OK	400.00
-1.00	ERRO	"Valor inválido"

null	ERRO	"Valor obrigatório"
------	------	---------------------

- ◆ `getValorDescontoIRRF()` / `setValorDescontoIRRF(BigDecimal valor)`

Entrada	Saída	Saída Esperada
300.00	OK	300.00
-5.00	ERRO	"Valor inválido"
null	ERRO	"Valor obrigatório"

- ◆ `getBaseCalculoINSS()` / `setBaseCalculoINSS(BigDecimal valor)`

Entrada	Saída	Saída Esperada
5000.00	OK	5000.00
-100.00	ERRO	"Base de cálculo inválida"
null	ERRO	"Valor obrigatório"

- ◆ `getBaseCalculoFGTS()` / `setBaseCalculoFGTS(BigDecimal valor)`

Entrada	Saída	Saída Esperada
5000.00	OK	5000.00
-100.00	ERRO	"Base de cálculo inválida"
null	ERRO	"Valor obrigatório"

- ◆ `getBaseCalculoIRRF()` / `setBaseCalculoIRRF(BigDecimal valor)`

Entrada	Saída	Saída Esperada
5000.00	OK	5000.00
-100.00	ERRO	"Base de cálculo inválida"
null	ERRO	"Valor obrigatório"

- ◆ `calcular()`

Entrada	Saída	Saída Esperada
Dados completos e válidos	OK	Cálculos de salário líquido, descontos e proventos corretamente preenchidos
Campos nulos ou inválidos	ERRO	"Não é possível calcular sem todos os valores obrigatórios"

## TESTES UNITÁRIOS – Classe: CalculadoraFolha

◆ calcularSalarioHora(salarioBruto: BigDecimal, horasSemanais: int): BigDecimal

Entrada	Saída	Saída Esperada
3200.00, 40	OK	20.00 (R\$3200 ÷ 160h mensais)
0.00, 40	OK	0.00
3200.00, 0	ERRO	"Horas semanais devem ser > 0"
-1000.00, 40	ERRO	"Salário inválido"
null, 40	ERRO	"Salário obrigatório"

◆ calcularAdicionalPericulosidade(salarioBase: BigDecimal): BigDecimal

Entrada	Saída	Saída Esperada
3000.00	OK	900.00 (30% de R\$3000)
0.00	OK	0.00
-1000.00	ERRO	"Salário inválido"
null	ERRO	"Salário obrigatório"

◆ calcularAdicionalInsalubridade(salarioMinimo: BigDecimal, grau: GrauInsalubridade): BigDecimal

Entrada	Saída	Saída Esperada
1412.00, NENHUM	OK	0.00
1412.00, BAIXO	OK	141.20 (10%)
1412.00, MEDIO	OK	282.40 (20%)
1412.00, ALTO	OK	423.60 (30%)
null, MEDIO	ERRO	"Salário mínimo obrigatório"
1412.00, null	ERRO	"Grau de insalubridade obrigatório"

◆ calcularDescontoValeTransporte(salarioBruto: BigDecimal, valorEntregue: BigDecimal): BigDecimal

Entrada	Saída	Saída Esperada
3000.00, 200.00	OK	200.00 (menor entre 6% do salário e entregue)
3000.00, 500.00	OK	180.00 (6% do salário)
3000.00, null	ERRO	"Valor entregue obrigatório"
null, 200.00	ERRO	"Salário obrigatório"

◆ calcularValeAlimentacao(valorDiario: BigDecimal, diasTrabalhados: int):

BigDecimal

Entrada	Saída	Saída Esperada
25.00, 22	OK	550.00
0.00, 22	OK	0.00
25.00, 0	OK	0.00
25.00, -5	ERRO	"Dias inválidos"
null, 20	ERRO	"Valor diário obrigatório"

◆ calcularINSS(salarioContribuicao: BigDecimal): BigDecimal

Entrada	Saída	Saída Esperada
3000.00	OK	330.00 (11%)
1000.00	OK	75.00 (7.5%)
-1000.00	ERRO	"Salário inválido"
null	ERRO	"Valor obrigatório"

◆ calcularFGTS(baseCalculoFGTS: BigDecimal): BigDecimal

Entrada	Saída	Saída Esperada
3000.00	OK	240.00 (8%)
0.00	OK	0.00
-1000.00	ERRO	"Valor inválido"
null	ERRO	"Valor obrigatório"

◆ calcularIRRF(salarioBruto: BigDecimal, descontoINSS: BigDecimal, numDependentes: int): BigDecimal

Entrada	Saída	Saída Esperada
5000.00, 500.00, 2	OK	420.00 (exemplo com base e tabela)
3000.00, 300.00, 0	OK	150.00
3000.00, -300.00, 0	ERRO	"Desconto INSS inválido"
null, 500.00, 1	ERRO	"Salário obrigatório"
5000.00, null, 1	ERRO	"Desconto INSS obrigatório"
5000.00, 500.00, -1	ERRO	"Número de dependentes inválido"



## TESTES UNITÁRIOS – Classe: Cargo

### ◆ getIdCargo() / setIdCargo(int id)

Entrada	Saída	Saída Esperada
1	OK	1
0	OK	0
-1	ERRO	"ID do cargo inválido"

### ◆ getIdUsuario() / setIdUsuario(int id)

Entrada	Saída	Saída Esperada
10	OK	10
0	OK	0
-5	ERRO	"ID de usuário inválido"

◆ `getNome()` / `setNome(String nome)`

Entrada	Saída	Saída Esperada
"Analista"	OK	"Analista"
""	ERRO	"Nome do cargo obrigatório"
null	ERRO	"Nome do cargo obrigatório"

◆ `getSalarioBase()` / `setSalarioBase(BigDecimal valor)`

Entrada	Saída	Saída Esperada
4000.00	OK	4000.00
0.00	OK	0.00
-1000.00	ERRO	"Salário base inválido"
null	ERRO	"Salário obrigatório"