

TESTES UNITÁRIOS -> FOLHA DE PAGAMENTO

autenticar(login: String, senha: String) : boolean

Possíveis testes da classe:

- Login: "usuarioValido", Senha: "senhaValida" → Esperado: true
 - Login: "usuarioInvalido", Senha: "senhaValida" → Esperado: false
 - Login: "usuarioValido", Senha: "senhaInvalida" → Esperado: false
 - Login: "", Senha: "" → Esperado: false
-

ajustarSalario() : void

Possíveis testes da classe:

- Salário atual: 3000, Novo salário: 3500 → Esperado: salário atualizado para 3500
 - Salário atual: 3000, Novo salário: 0 → Esperado: operação aceita ou rejeitada conforme regra
 - Salário atual: 3000, Novo salário: -500 → Esperado: operação rejeitada/erro
 - Salário atual: 3000, Novo salário: 3000.75 → Esperado: salário atualizado com precisão decimal
-

adicionarFuncionario() : void

Possíveis testes da classe:

- Funcionário: João → Esperado: funcionário adicionado à lista
- Funcionário: João (duplicado) → Esperado: sistema rejeita duplicação
- Funcionário com dados incompletos → Esperado: validação de erro

removerFuncionario() : void

Possíveis testes da classe:

- Remover João existente → Esperado: funcionário removido da lista
 - Remover Maria que não existe → Esperado: nenhuma alteração, sem erro
 - Remover funcionário nulo → Esperado: erro tratado
-

listarFuncionario() : void

Possíveis testes da classe:

- Lista contém João e Maria → Esperado: retorna João e Maria
 - Lista vazia → Esperado: retorna lista vazia
 - Após adicionar/remover → Esperado: lista atualizada corretamente
-

exibirInformações() : String

Possíveis testes da classe:

- Funcionário: João → Esperado: retorna informações corretas de João
 - Sem funcionários → Esperado: retorno vazio ou mensagem apropriada
-

exibirDados() : String

Possíveis testes da classe:

- Funcionário: João → Esperado: retorna dados completos de João
 - Sem funcionários → Esperado: retorno vazio ou mensagem apropriada
-

temDireitoInsalubridade() : boolean

Possíveis testes da classe:

- Funcionário com direito → Esperado: true
 - Funcionário sem direito → Esperado: false
 - Dados inconsistentes → Esperado: false ou erro tratado
-

calcularTempo() : int

Possíveis testes da classe:

- Tempo de trabalho correto → Esperado: retorna valor correto
 - Dados inválidos → Esperado: retorna 0 ou erro tratado
-

adicionarBeneficio() : void

Possíveis testes da classe:

- Benefício: Vale Alimentação → Esperado: benefício adicionado à lista
 - Benefício duplicado → Esperado: rejeitado ou não duplicado
 - Benefício inválido → Esperado: erro ou rejeição
-

List<String> ListarBeneficios() : String

Possíveis testes da classe:

- Benefícios: Vale Alimentação, Vale Transporte → Esperado: retorna os dois benefícios
 - Sem benefícios → Esperado: lista vazia
-

calcular() : void

Possíveis testes da classe:

- Dados completos → Esperado: todos os valores de proventos e descontos atualizados corretamente
 - Dados faltando → Esperado: valores inconsistentes tratados
-

adicionarDesconto() : void

Possíveis testes da classe:

- Desconto: 200 → Esperado: desconto adicionado
 - Desconto duplicado → Esperado: não duplicar ou rejeitar
 - Desconto negativo → Esperado: erro ou rejeição
-

calcularTotalDesconto() : BigDecimal

Possíveis testes da classe:

- Descontos: 100, 50 → Esperado: 150
 - Sem descontos → Esperado: 0
-

calcularTotalProventos() : BigDecimal

Possíveis testes da classe:

- Proventos: 2000, 500 → Esperado: 2500
 - Sem proventos → Esperado: 0
-

calcularSalarioLiquido() : BigDecimal

Possíveis testes da classe:

- Salário bruto: 3000, Descontos: 500, Proventos: 200 → Esperado: 2700
 - Sem descontos → Esperado: igual ao salário bruto
 - Sem proventos → Esperado: salário líquido = salário bruto - descontos
-

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

Possíveis testes da classe:

- Salário bruto: 3000, Horas: 40 → Esperado: valor correto por hora
 - Horas: 0 → Esperado: 0
 - Horas negativas → Esperado: erro ou tratamento
-

**calcularAdicionalPericulosidade(salarioBase: BigDecimal) :
BigDecimal**

Possíveis testes da classe:

- Salário base: 2000 → Esperado: 20% de adicional (400)
 - Salário base: 0 → Esperado: 0
 - Salário base negativo → Esperado: erro tratado
-

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

Possíveis testes da classe:

- Salário mínimo: 1212, Grau: baixo → Esperado: valor adicional correto
- Grau médio → Esperado: valor correspondente

- Grau alto → Esperado: valor correspondente
 - Grau inválido → Esperado: erro ou 0
-

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

Possíveis testes da classe:

- Salário bruto: 3000, Valor entregue: 150 → Esperado: desconto correto
 - Valor entregue maior que limite → Esperado: desconto máximo aplicado
 - Salário bruto: 0 → Esperado: 0
-

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

Possíveis testes da classe:

- Valor diário: 20, Dias: 22 → Esperado: 440
 - Dias: 0 → Esperado: 0
 - Dias negativos → Esperado: erro ou rejeição
-

calcularINSS(salarioContribuicao: BigDecimal) : BigDecimal

Possíveis testes da classe:

- Salário: 3000 → Esperado: INSS calculado corretamente
 - Salário: 0 → Esperado: 0
 - Salário negativo → Esperado: erro ou rejeição
-

calcularFGTS(baseCalculoFGTS: BigDecimal) : BigDecimal

Possíveis testes da classe:

- Base: 3000 → Esperado: 8% de FGTS = 240
 - Base: 0 → Esperado: 0
 - Base negativa → Esperado: erro ou rejeição
-

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

Possíveis testes da classe:

- Salário bruto: 5000, INSS: 500, Dependentes: 2 → Esperado: valor correto do IRRF
 - Sem dependentes → Esperado: cálculo correto sem dedução
 - INSS: 0 → Esperado: efeito correto no IRRF
 - Salário bruto: 0 → Esperado: 0
-

exibirDemonstrativo(FolhaPagamento) : void

Possíveis testes da classe:

- Folha completa → Esperado: demonstrativo com todos os campos preenchidos corretamente
- Folha vazia → Esperado: retorno vazio ou mensagem adequada
- Dados inconsistentes → Esperado: erros tratados ou valores zerados