

 Serviço Nacional de Aprendizagem Industrial Santa Catarina	SA - Sistema para empreendimento indispensável na indústria	Desempenho
	Data: 18/11	
	Docente: <i>Júlia Caroline Pereira</i>	
	Curso Técnico em <i>Desenvolvimento de Sistemas</i>	
	Unidade Curricular: <i>Lógica de programação</i>	
	Turma: <i>DS 24 m7</i>	
	Grupo: guilherme alvim, caio alves, nathan e gabriel	

Desafio:

Cada estudante do grupo deverá desenvolver um cenário com um problema que deverá ser resolvido utilizando a linguagem portugal.

Setores (Após terminar o código de setor, coloque um verificado no setor realizado):

- ☒ RH: ~~Gestão de folha de pagamento, recrutamento e seleção de novos funcionários.~~
- ☒ Financeiro: ~~Controle de despesas, previsão de receitas, análise de fluxo de caixa.~~
- ☒ Infraestrutura: ~~Manutenção de equipamentos, alocação de recursos, melhoria de processos produtivos.~~
- ☒ Marketing: ~~Desenvolvimento de campanhas publicitárias, análise de mercado, estratégia de vendas.~~

Códigos o que vamos ter que usar obrigatoriamente pelo menos uma vez (Aos que foram marcados com verificado o item já foi usado e não há necessidade de uso e virou objetiva, caso queiram pergunte a professora):

- ☒ ~~se, então e senão~~
- ☒ ~~e e ou~~
- ☒ ~~falso e verdadeiro~~
- ☒ ~~para faca, enquanto ou repita~~
- ☒ ~~escola caso~~
- ☒ ~~vetores~~
- ☒ ~~registro~~
- ☒ ~~procedimento e função~~

Códigos:

Rh - Nathan

Setor RH

Problema: A empresa Nidec está com problemas de administrar o seu setor RH, isso inclui gestão de folha de pagamento, recrutamento e seleção de novos funcionários. Atual sistema do RH da Nidec é muito difícil, complexo e confuso de mexer.

Resolução: Sistema feito por um pseudocódigo portugal no visualg, facilitando o usuário administrar o setor, desde gestão de folha de pagamento, recrutamento e seleção de novos funcionários.

Código do sistema:

Algoritmo "Setor RH"

//

//

// Gestão de folha de pagamento, recrutamento e seleção de novos funcionarios

//

//

Var

opcao, op1, op2, op3, i, num_candidatos, cp: inteiro

n, salar: real

car: caractere

candidatos: vetor[1..150] de caractere

cpfs: vetor[1..150] de caractere

nomes: vetor[1..150] de caractere

cargos: vetor[1..150] de caractere

salarios: vetor[1..150] de real

telefonesFuncionarios: vetor[1..150] de inteiro

cnpj: vetor[1..150] de caractere

percentual_contri, contribuicaoSindical, inss, fgts: real

procedimento registrarNovosFuncionarios()

var

salario: real

inicio

escreval("Digite nome completo do funcionário: ")

leia(nomes[i])

escreval("Digite cargo atual do funcionário: ")

leia(cargos[i])

escreval("Digite salário atual do funcionário: ")

leia(salarios[i])

salario <- salarios[i]

```

se salario <= 1412 então
    inss <- salario * (8/100)
senao
    se salario <= 2666 então
        inss <- salario * (9/100)
    senao
        se salario <= 4000 então
            inss <- salario * (12/100)
        senao
            se salario <= 7786 então
                inss <- salario * (14/100)
            senao
                inss <- salario * (16/100)
        fimse
    fimse
fimse

```

```

escreval("Valor do INSS a ser descontado: ", inss)

```

```

escreval("Digite o percentual da contribuição de 2% ate 8%: ")
leia(percentual_contri)

```

```

se (percentual_contri <= 2) ou (percentual_contri >= 8) então
    escreval("Percentual inválido. A contribuição sindical deve ser entre 2% e 8%.")
    percentual_contri <- 2
fimse

```

```

contribuicaoSindical <- salario*(percentual_contri/100)
escreval("Valor da contribuição sindical a ser descontado: ", contribuicaoSindical:0:2)

```

```

fgts <- salario *(8/100)
escreval("Valor do FGTS: ", fgts)

```

```

escreval("Digite CPF do funcionário: ")
leia(cpfs[i])
escreval("Digite número de telefone do funcionário: ")
leia(telefonesFuncionarios[i])
escreval("Digite CNPJ do funcionário: ")
leia(cnpj[i])

```

```

i <- i + 1
escreval("Funcionário cadastrado com sucesso.")
fimprocedimento

```

```

procedimento funcionariosRegistrados()
var
    j: inteiro

```

```

inicio
    escreval("Funcionários registrados:")
    para j de 1 ate i-1 faca
        escreval("Nome: ", nomes[j])
        escreval("CNPJ: ", cnpj[j])
        escreval("Cargo: ", cargos[j])
        escreval("Salario: ", salarios[j])
        escreval("CPF: ", cpfs[j])
        escreval("Telefone: ", telefonesFuncionarios[j])
    fimpara
fimprocedimento

```

```

procedimento listaCandidatos()
var
    j: inteiro
inicio
    escreval("Lista de Candidatos:")
    para j de 1 ate num_candidatos faca
        escreval("Candidato ", j, ": ", candidatos[j])
    fimpara
fimprocedimento

```

```

procedimento marcarEntrevista()
var
    nome_candidato, data_entrevista: caractere
inicio
    escreval("Digite o nome do candidato para marcar a entrevista: ")
    leia(nome_candidato)
    escreval("Digite a data da entrevista (data/mês/ano): ")
    leia(data_entrevista)
    escreval("Entrevista marcada para ", nome_candidato, " na data ", data_entrevista, ".")
fimprocedimento

```

```

procedimento selecionarCandidato()
var
    nome_candidato: caractere
    j: inteiro
inicio
    escreval("Digite o nome do candidato a ser selecionado: ")
    leia(nome_candidato)
    para j de 1 ate num_candidatos faca
        se candidatos[j] = nome_candidato então
            escreval("Candidato ", nome_candidato, " selecionado com sucesso!")
            pare
        senao
            escreval("Candidato não encontrado.")
        fimse
    fimpara

```

fimprocedimento

Inicio

i <- 1

num_candidatos <- 0

opcao <- 0

enquanto opcao <> 4 faca

 escreval("Menu do RH - Nidec")

 escreval("[1] - Gestão de Folha de Pagamento")

 escreval("[2] - Recrutamento")

 escreval("[3] - Seleção de novos funcionários")

 escreval("[4] - Sair")

 leia(opcao)

 escolha opcao

 caso 1

 enquanto op1 <> 4 faca

 escreval("Menu de Gestão da Folha de Pagamento")

 escreval("[1] - Procurar dados de funcionários")

 escreval("[2] - Registrar novo funcionário")

 escreval("[3] - Dados dos funcionários")

 escreval("[4] - Voltar")

 leia(op1)

 escolha op1

 caso 1

 escreval("Menu Procuração de Dados de Funcionario")

 escreval("Digite o CPF do funcionario: ")

 leia(cp)

 para j de 1 ate i - 1 faca

 se cp = cpfs[j] então

 escreval("nome: ", nomes[j])

 escreval("CNPJ: ", cnpj[j])

 escreval("cargo: ", cargos[j])

 escreval("salario: ", salarios[j])

 escreval("telefone: ", telefonesFuncionarios[j])

 senao

 escreval("Funcionario nao foi encontrado")

 fimse

 fimpara

 caso 2

 registrarNovosFuncionarios()

 caso 3

 funcionariosRegistrados()

 fimescolha

 fimenquanto

 caso 2

```

    enquanto op2 <> 3 faca
        escreval("Menu de recrutamento - Nidec")
        escreval("[1] - Informações dos candidatos")
        escreval("[2] - Marcar entrevista")
        escreval("[3] - Sair")
        leia(op2)
        escolha op2
            caso 1
                listaCandidatos()
            caso 2
                marcarEntrevista()
        fimescolha
    fimenquanto

    caso 3
        enquanto op3 <> 4 faca
            escreval("Menu de seleção de candidatos")
            escreval("[1] - Adicionar candidato")
            escreval("[2] - Selecionar candidato")
            escreval("[3] - Lista de candidatos")
            escreval("[4] - Sair")
            leia(op3)
            escolha op3
                caso 1
                    se num_candidatos <= 150 então
                        escreval("Digite o nome do candidato: ")
                        leia(candidatos[num_candidatos+1])
                        num_candidatos <- num_candidatos+1
                        escreval("Candidato adicionado com sucesso")
                    senao
                        escreval("Foi atingido o limite de candidatos")
                    fimse
                caso 2
                    selecionarCandidato()
                caso 3
                    listaCandidatos()
            fimescolha
        fimenquanto
    fimescolha
fimenquanto

finalgoritmo

```

Algoritmo "semnome"

//

// Rh: Gestão de folha de pagamento, recrutamento e seleção de novos funcionários

Var

opcao, op1, op2, op3, i, num_candidatos, cp: inteiro

n, salar: real

car: caractere

candidatos: vetor[1 .. 150] de caractere

cpfs: vetor[1 .. 150] de caractere

nomes: vetor[1 .. 150] de caractere

cargos: vetor[1 .. 150] de caractere

salarios: vetor[1 .. 150] de real

telefonesFuncionarios: vetor[1 .. 150] de inteiro

cnpj: vetor[1 .. 150] de caractere

procedimento registrarNovosFuncionarios()

var

inss, salario, contr_sindical, fgts, percen_contri: real

inicio

inss <- 8/100

escreval("Digite nome completo do funcionário: ")

leia(nomes[i])

escreval("Digite cargo atual do funcionário: ")

leia(cargos[i])

escreval("Digite salário atual do funcionário: ")

leia(salarios[i])

escreval("Digite CPF do funcionario: ")

leia(cpfs[i])

escreval("Digite numero de telefone do funcionario: ")

leia(telefonesFuncionarios[i])

escreval("Digite CNPJ do funcionário: ")

leia(cnpj[i])

i <- i + 1

escreval("Funcionario cadastrado com sucesso")

fimprocedimento

procedimento funcionariosRegistrados()

var

j: inteiro

inicio

escreval("Funcionários registrados:")

para j de 1 ate i faca

escreval("Nome: ", nomes[j])

```

        escreval("CNPJ: ", cnpj[j])
        escreval("Cargo: ", cargos[j])
        escreval("Salário: ", salarios[j])
        escreval("CPF: ", cpfs[j])
        escreval("Telefone: ", telefonesFuncionarios[j])
    fimpara
fimprocedimento

```

```

procedimento listaCandidatos()
    var
        j: inteiro
    inicio
        escreval("Lista de Candidatos:")
        para j de 1 ate num_candidatos faca
            escreval("Candidato ", j, ": ", candidatos[j])
        fimpara
fimprocedimento

```

```

procedimento marcarEntrevista()
    var
        nome_candidato: caractere
        data_entrevista: caractere
    inicio
        escreval("Digite o nome do candidato para marcar a entrevista: ")
        leia(nome_candidato)
        escreval("Digite a data da entrevista(data/mês/ano): ")
        leia(data_entrevista)
        escreval("Entrevista marcada para ", nome_candidato, " na data ",
dataEntrevista, ".")
fimprocedimento

```

```

procedimento selecionarCandidato()
    var
        nome_candidato: caractere
        j: inteiro
    inicio
        escreval("Digite o nome do candidato a ser selecionado: ")
        leia(nome_candidato)
        para j de 1 ate num_candidatos faca
            se candidatos[j] = nome_candidato entao
                escreval("Candidato ", nome_candidato, " selecionado com sucesso!")
            senao
                escreval("Candidato não encontrado.")
            fimse
        fimpara
fimprocedimento

```

Inicio


```

i <- 1
num_candidatos <- 0
opcao <- 0

enquanto opcao <> 4 faca
  escreval("Menu do RH - Nidec")
  escreval("[1] - Gestão de Folha de Pagamento")
  escreval("[2] - Recrutamento")
  escreval("[3] - Seleção de novos funcionários")
  escreval("[4] - Sair")
  leia(opcao)

  escolha opcao
  caso 1
    enquanto op1 <> 4 faca
      escreval("Menu de Gestão da Folha de Pagamento")
      escreval("[1] - Procurar dados de funcionários")
      escreval("[2] - Registrar novo funcionário")
      escreval("[3] - Dados dos funcionários")
      escreval("[4] - Voltar")
      leia(op1)

    escolha op1
    caso 1
      escreval("Menu Procuraç o de Dados de Funcionario")
      escreval("Digite o CPF do funcionario: ")
      leia(cp)
      para i de 1 ate i-1 faca
        se cp = cpfs[j] entao
          escreval("nome: ", nomes[j])
          escreval("CNPJ: ", cnpj[j])
          escreval("cargo: ", cargos[j])
          escreval("salario: ", salarios[j])
          escreval("telefone: ", telefonesFuncionarios[j])
        senao
          escreval("Funcionario nao foi encontrado")
        fimse
      fimpara
    caso 2
      registrarNovosFuncionarios()
    caso 3
      funcionariosRegistrados()
    fimsecolha
  fimenquanto
  caso 2
    enquanto op2 <> 3 faca
      escreval("Menu de recrutamento - Nidec")
      escreval("[1] - Informa  es dos candidatos")

```

```

        escreval("[2] - Marcar entrevista")
        escreval("[3] - Sair")
        leia(op2)
        escolha op2
            caso 1
                listaCandidatos()
            caso 2
                marcarEntrevista()
        fimescolha
    fimenquanto

    caso 3
        enquanto op3 <> 4 faca
            escreval("Menu de seleção de candidatos")
            escreval("[1] - Adicionar candidato")
            escreval("[2] - Selecionar candidato")
            escreval("[3] - Lista de candidatos")
            escreval("[4] - Sair")
            leia(op3)
            escolha op3
                caso 1
                    se numCandidatos <= 150 entao
                        escreval("Digite o nome do candidato: ")
                        leia(candidatos[numCandidatos + 1])
                        numCandidatos <- numCandidatos + 1
                        escreval("candidato adicionado com sucesso")
                    senao
                        escreval("foi atingido limite de candidatos")
                    fimse
                caso 2
                    selecionarCandidato()
                caso 3
                    listaCandidatos()
            fimescolha
        fimenquanto
    fimescolha
fimenquanto

finalgoritmo

```

Da infraestrutura - Guilherme Alvim

Algoritmo "semnome"

```

// Os motores precisam de uma manutenção verificar a peca
// identificar o tipo e problemas no motor enviar para

```

```
// os setores corretos.
```

```
var
```

```
motoresEletricos: caracter
```

```
tipos: inteiro
```

```
peca: inteiro
```

```
a: inteiro
```

```
tip1, tip2, tip3: inteiro
```

```
destino: inteiro
```

```
verificar: caracter
```

```
acerto: logico
```

```
i: inteiro
```

```
m: inteiro
```

```
procedimento informacoesTipo(t1, t2 , t3: inteiro): inteiro
```

```
var
```

```
inicio
```

```
escreval ("Motor elétrico de corrente alternada há um rotor, circuito  
magnético estático ou bobina.")
```

```
escreval ("Motor elétrico de corrente contínua contém uma Série,  
Paralelo, Composto, Excitação independente")
```

```
escreval ("Motor universal tem um eixo, conjunto de bobinas ou  
enrolamentos de fio enrolados no eixo, rotor, estator, O estator ou  
(indutor)")
```

```
escreval ("")
```

```
fimprocedimento
```

```
inicio
```

```
acerto <- falso
```

```
escreval ("Um motor vai ser caminhado a você, até acabar seu turno")
```

```
// 0 randi(9) Dependendo do número que cair aleatoriamente
```

```
// ele vai selecionar um dos 3 tipos de máquina:
```

```
// - motores de corrente contínua
```

```
// - motores de corrente alternada
```

```
// - motores universais.
```

```
peca <- randi(9)
```

```
enquanto acerto <> verdadeiro faca
```

```
escreval ("Analise o tipo dela pela tais descrições das peças")
```

```
escreval ("Aperte [0] para mostrar as descrições de cada tipo")
```

```
se peca = 1 então
```

```
escreval ("A peça contém um retor e circuito magnético")
```

```
escreval ("")
```

```
fimse
```

```
se peca = 2 então
```

```
escreval ("A peça contém um retor, tem Paralelo, Excitação  
independente e etc")
```

```
escreval ("")
```

```
fimse
```

```
se peca = 3 então
```

```
escreval ("A peça contém um rotor, um eixo, conjunto de bobinas e  
indutor")
```

escreval ("")

fimse

se peca = 4 então

escreval ("A peça contém um retor e circuito magnético")

escreval ("")

fimse

se peca = 5 então

escreval ("A peça contém um retor, Série, Composto e etc.")

escreval ("")

fimse

se peca = 6 então

escreval ("A peça contém um rotor, um eixo, enrolamentos de fio enrolados no eixo e estator")

escreval ("")

fimse

se peca = 7 então

escreval ("A peça contém um retor, série, circuito magnético e paralelo")

escreval ("")

fimse

se peca = 8 então

escreval ("A peça contém um retor, composto, Excitação independente e etc")

escreval ("")

fimse

se peca = 9 então

escreval: ("A peça contém um rotor, indutor, eixo, motor")

```
escreval ("")
```

```
fimse
```

```
Escreval ("[1] - tipo 1 [2] - tipo 2 [3] - tipo 3 [4] - Com defeito  
[5] - informações")
```

```
Leia (destino)
```

```
escolha destino
```

```
caso 1
```

```
    escreval ("Você a destinou como uma peça para o motor de corrente  
contínua")
```

```
    se (peca = 1) ou (peca = 4) entao
```

```
        acerto <- verdadeiro
```

```
        escreval ("você acertou")
```

```
    senao
```

```
        escreval ("você errou, tente novamente")
```

```
    fimse
```

```
caso 2
```

```
    escreval ("Você a destinou como uma peça para o motor de corrente  
alternada")
```

```
    se (peca = 2) ou (peca = 5) então
```

```
        acerto <- verdadeiro
```

```
        escreval ("você acertou")
```

```
    senao
```

```
        escreval ("você errou, tente novamente")
```

```
    fimse
```

caso 3

escreval ("Você a destinou como uma peça para o motor universal")

se (peca = 3) ou (peca = 6) então

acerto <- verdadeiro

escreval ("você acertou")

senao

escreval ("você errou, tente novamente")

fimse

caso 4

escreval ("Está com defeito ou faltando peças")

se (peca = 7) ou (peca = 8) ou (peca = 9) então

acerto <- verdadeiro

escreval ("você acertou")

senao

escreval ("você errou, tente novamente")

fimse

caso 5

informacoesTipo(tip1, tip2 , tip3)

Escreval ("_____")

fimescolha

fimenquanto

```
acerto <- falso
```

```
escreval ("Você terminou seu turno, parabéns")
```

Fimalgoritmo

Caio

marketing: problema estratégia de vendas

Concorrência Agressiva: Concorrentes estão oferecendo melhores preços e promoções, além de uma experiência de compra superior. A empresa Nidec não está se destacando em análises de preços e propostas de valor.

Considerar a adoção de uma estratégia de preços competitiva, como penetração de mercado, onde a empresa oferece preços mais baixos inicialmente para atrair consumidores. criando um código de programação que simule uma estratégia de penetração de mercado, analisando a eficácia dessa abordagem em termos de atração de consumidores e simulação de vendas.


```

// corpo do algoritmo
var
  preco_base, preco_penetracao: real
  vendas_base, vendas_com_penetracao: inteiro
  elasticidade_demanda: real
  numero_consumidores: inteiro
  novos_consumidores: inteiro
inicio
  escreval("Resultados da Simulacao de Penetracao de Mercado:")
  escreval("-----")
  escreval("Preco padrao do produto: R$" )
  escreval("Preco de penetracao: R$")
  escreval("Vendas com preço padrao: " )
  escreval("Vendas com preço de penetração: ",)
  escreval("Número de novos consumidores atraídos: ")

  se vendas_com_penetracao > vendas_base entao
    escreval("A estrategia de penetracao foi eficaz, aumentando as vendas!")
  senao
    escreval("A estrategia de penetracao nao foi eficaz, as vendas nao aumentaram.")
  fimse

finalgoritmo

```

Financeiro - Gabriel

Setor financeiro

Problema: A empresa precisa controlar o fluxo de caixa diário. Esse controle é essencial para tomar decisões financeiras, como a realização de investimentos ou a negociação de dívidas.

Solução: Um sistema feito em portugol que calcula o saldo de caixa diário ,registrando todas as entradas e saídas .

Algoritmo "calculadora de saldo de caixa diário"

tipo

mudanca = registro

fonte: caractere

valor,porcentagem: real

dia_m:inteiro

fimregistro

Var

saldo_total,porcentagem_total,valor_total,valor_inicial,porcentagem: real
r,i,sd:inteiro
rc: caractere
dia: inteiro
sair: logico
Saldo_diario: vetor [0..100] de mudanca

Inicio

Escreva("Informe o valor total atual da empresa: ")
leia(valor_total)
valor_inicial <- valor_total

limpatela

dia <- 1

repita

Escreval("Dia: ",dia)

Escreval()

Escreval("Valor da empresa inicial: R\$",valor_inicial)

Escreval()

Escreval("fluxo de caixa: ")

para i de 0 ate 20 faca

se saldo_diario[i].dia_m = dia entao

Escreval("Fonte: ",saldo_diario[i].fonte,"| Valor: R\$",saldo_diario[i].valor,"|

Mudança em porcentagem: ",saldo_diario[i].porcentagem,"%")

Escreval()

fimse

fimpara

Escreval("Saldo total: R\$",saldo_total," Mudança de porcentagem total: ",porcentagem_total,"%")

Escreval()

Escreval("Valor da empresa total: R\$",valor_total)

Escreval()

Escreval("[1] Registrar entrada de dinheiro para esse dia.")

Escreval("[2] Registrar saída de dinheiro para esse dia. ")

Escreval("[3] Próximo dia. ")

Escreval("[4] Ver os registros de um dia específico")

Escreval("[5] Sair.")

leia(r)

escolha r

caso 1

limpatela

Escreva("Digite qual a fonte desse dinheiro: ")

leia(rc)

```
saldo_diario[sd].fonte <- rc
Escreva("Digite o valor ganho: ")
leia(r)
saldo_diario[sd].valor <- r
saldo_diario[sd].porcentagem <- 100*(saldo_diario[sd].valor/valor_inicial)
saldo_diario[sd].dia_m <- dia
valor_total <- valor_total + saldo_diario[sd].valor
sd <- sd+1
limpatela
```

caso 2

```
limpatela
Escreva("Digite qual a fonte dessa saída : ")
leia(rc)
saldo_diario[sd].fonte <- rc
Escreva("Digite o valor perdido: ")
leia(r)
saldo_diario[sd].valor <- r*-1

saldo_diario[sd].porcentagem <- 100*(saldo_diario[sd].valor/valor_inicial)
saldo_diario[sd].dia_m <- dia
```

```
valor_total <- valor_total + saldo_diario[sd].valor
sd <- sd+1
limpatela
```

caso 3

```
dia <- dia + 1
valor_inicial <- valor_total
limpatela
```

caso 4

```
limpatela
Escreval("Digite o dia que quer ver:")
leia(r)
Escreval("Dia: ",r)
```

para i de 0 ate 20 faca

```
se saldo_diario[i].dia_m = r entao
  Escreval("Fonte: ",saldo_diario[i].fonte,"| Valor: R$",saldo_diario[i].valor,"|
Mudança em porcentagem: ",saldo_diario[i].porcentagem,"%")
  Escreval()
```

```
fimse  
fimpara
```

caso 5

```
sair <- verdadeiro  
outrocaso  
limpatela  
Escreval("")  
Escreval("**Resposta inválida**")  
Escreval("")
```

fimescolha

```
saldo_total <- 0  
para i de 0 ate 20 faca
```

```
se saldo_diario[i].dia_m = dia entao  
    saldo_total <- saldo_total + saldo_diario[i].valor  
fimse
```

```
fimpara  
porcentagem_total <- 100*(saldo_total/valor_inicial)
```

ate sair = verdadeiro

Fimalgoritmo

Resultados esperados:

- Documento contendo os quatro problemas identificados.
- Documento contendo prints dos códigos-fonte e das execuções.

Entrega:

Coloque a atividade em um repositório no GitHub e compartilhe o link no Classroom.