

Algoritmos Elementares de Ordenação

```
algoritmo "Bolha"
var
    numeros: vetor[1..20] de inteiro
    cont, aux: inteiro
    troca: logico
inicio
// Seção de Comandos
cont <- 1
enquanto cont <= 20 faca
    leia(numeros[cont])
    cont <- cont + 1
fimenquanto

repita
    troca <- falso
    cont <- 1
    enquanto cont < 20 faca
        se numeros[cont] > numeros[cont + 1] entao
            aux <- numeros[cont]
            numeros[cont] <- numeros[cont + 1]
            numeros[cont + 1] <- aux

            troca <- verdadeiro
        fimse
        cont <- cont + 1
    fimenquanto
ate troca = falso

cont <- 1
enquanto cont <= 20 faca
    escreval(numeros[cont])
    cont <- cont + 1
fimenquanto
fimalgoritmo
```

```

algoritmo "SelecaoDireta"
var
    numeros: vetor[1..20] de inteiro
    cont1, cont2, menor, aux: inteiro
inicio
// Seção de Comandos
    cont1 <- 1
    enquanto cont1 <= 20 faca
        leia(numeros[cont1])
        cont1 <- cont1 + 1
    fimenquanto

    cont1 <- 1
    enquanto cont1 <= 20 faca
        menor <- cont1
        cont2 <- cont1 + 1
        enquanto cont2 <= 20 faca
            se numeros[cont2] < numeros[menor] entao
                menor <- cont2
            fimse
            cont2 <- cont2 + 1
        fimenquanto

        aux <- numeros[cont1]
        numeros[cont1] <- numeros[menor]
        numeros[menor] <- aux

        cont1 <- cont1 + 1
    fimenquanto

    cont1 <- 1
    enquanto cont1 <= 20 faca
        escreval(numeros[cont1])
        cont1 <- cont1 + 1
    fimenquanto
fimalgoritmo

```

```

algoritmo "InsercaoDireta"
var
    numeros: vetor[1..20] de inteiro
    cont, pivo, aux: inteiro
inicio
// Seção de Comandos
cont <- 1
enquanto cont <= 20 faca
    leia(numeros[cont])
    cont <- cont + 1
fimenquanto

pivo <- 2
enquanto pivo <= 20 faca
    aux <- numeros[pivo]
    cont <- pivo - 1
    enquanto numeros[cont] > aux faca
        numeros[cont + 1] <- numeros[cont]

        cont <- cont - 1
        se cont < 1 entao
            interrompa
        fimse
    fimenquanto

    numeros[cont + 1] <- aux
    pivo <- pivo + 1
fimenquanto

cont <- 1
enquanto cont <= 20 faca
    escreval(numeros[cont])
    cont <- cont + 1
fimenquanto
fimalgoritmo

```

Algoritmos de Pesquisa

```
algoritmo "PesquisaSequencial"
var
    c, v: inteiro
    A : vetor [1..10] de inteiro
inicio
    // Seção de Comandos
    c <- 1
    enquanto c <= 10 faca
        A[c] <- c * 10
        c <- c + 1
    fimenquanto

    c <- 1
    enquanto c <= 10 faca
        escreva(A[c])
        c <- c + 1
    fimenquanto

    escreva("Informe o valor a pesquisar:")
    leia(v)

    c <- 1
    enquanto (c <= 10) faca
        se A[c] = v entao
            interrompa
        fimse
        c <- c + 1
    fimenquanto

    se c > 10 entao
        escreva("Valor nao encontrado no vetor")
    senao
        escreva("Valor encontrado no elemento ", c)
    fimse
fimalgoritmo
```

```

algoritmo "PesquisaBinaria"
var
    c, v, x, i, d: inteiro
    A : vetor [1..10] de inteiro
inicio
    (** Assuma que nesta parte inicial estão as instruções que
    preenchem o vetor A com valores inteiros em ordem crescente
    **)

    escreva("Informe o valor a pesquisar:")
    leia(v)

    i <- 1
    d <- 10
    repita
        x <- (i + d ) \ 2
        se v < A[x] entao
            d <- x - 1
        senao
            i <- x + 1
        fimse
    ate (v = A[x]) ou (i > d)

    se v = A[x] entao
        escreva("Valor encontrado no elemento ", x)
    senao
        escreva("Valor nao encontrado no vetor")
    fimse
finalgoritmo

```