

Tema 1. Archivos.

Ordenación por mezcla natural

```
algoritmo Ordenar
tipos
    registro = tipoReg
        entero : clave
        //Resto de datos
    fin_registro
archivo_s de tipoReg = tipoArch
var
    entero : n
inicio
    repetir
        //Parte el archivo ARCHIVO.DAT en dos particiones AUX1 y AUX2
        //Cada partición estará formada por las secuencias ordenadas
        //de ARCHIVO.DAT
        partición('ARCHIVO.DAT','AUX1','AUX2')
        //Mezcla las secuencias ordenadas de AUX1 y AUX2 en el
        //archivo ARCHIVO.DAT. La variable n indica el número
        //de secuencias que se han mezclado
        mezcla('AUX1','AUX2','ARCHIVO.DAT',n)
        //Cuando el número de secuencias mezcladas es 1
        //el archivo está ordenado
    hasta_que n = 1
fin

procedimiento partición(valor cadena : NomArch,NomArch1,NomArch2)
var
    entero : i
    lógico : GrabarA1
    tipoReg : R,RAux
    tipoArch : A,A1,A2
inicio
    crear(NomArch1)
    crear(NomArch2)
    abrir(A1,NomArch1,escritura)
    abrir(A2,NomArch2,escritura)
    abrir(A,NomArch,lectura)
    GrabarA1 ← verdad
    leer(A,R)
    mientras no fda(A) hacer
        si GrabarA1 entonces
            escribir(A1,R)
        si_no
            escribir(A2,R)
        fin_si
        RAux ← R
        leer(A,R)
        si R.clave < RAux.clave entonces
            GrabarA1 ← no GrabarA1
        fin_si
    fin_mientras
    cerrar(A)
    cerrar(A1)
    cerrar(A2)
fin_procedimiento
```

```

procedimiento mezcla(valor cadena:NomArch1,NomArch2,NomArch;ref entero: n)
var
    tipoArch : A,A1,A2
    tipoReg : R1,R2,RAux1,RAux2
inicio
    crear (NomArch)
    abrir (A,NomArch,escritura)
    abrir (A1,NomArch1,lectura)
    abrir (A2,NomArch2,lectura)
    leer (A1,R1)
    leer (A2,R2)
    n ← 0
    mientras no fda(A1) o no fda(A2) hacer
        RAux1 ← R1
        RAux2 ← R2
        mientras no fda(A1) y no fda(A2) y (R1.clave>=RAux1.clave)
            (R2.clave >=RAux2.clave) hacer
                si r1.clave < r2.clave entonces
                    escribir (A,R1)
                    RAux1.clave ← R1
                    leer (A1,R1)
                si_no
                    escribir (A,R2)
                    RAux2 ← R2
                    leer (A2,R2)
                fin_si
            fin_mientras
        mientras no fda(A1) y (R1.clave >= RAux1.clave) hacer
            escribir (A,R1)
            RAux1 ← R1
            leer (A1,R1)
        fin_mientras
        mientras no fda(A2) y (R2.clave >= RAux2.clave) hacer
            escribir (A,R2)
            RAux2 ← R2
            leer (A2,R2)
        fin_mientras
    n ← n + 1
    fin_mientras
    cerrar (A,A1,A2)
fin_procedimiento

```