

# Tema 1. Archivos.

## **Ejemplo: creación del archivo indexado**

Se desea indexar el archivo directo de almacén. Para ello se debe leer secuencialmente el archivo de organización directa e incluir el código de producto y la posición de todas las ranuras ocupadas en un array de índices. Al acabar el proceso se debe ordenar el array y almacenarlo de forma temporal en un archivo secuencial para las siguientes ocasiones en que se desee gestionar el índice.

```
//Creación del archivo indexado a partir del archivo de productos
algoritmo CreaciónIndexado
const
    MaxReg = 120
    //El índice sólo tiene 100 posiciones ya que sólo hay 100 productos distintos
    numElemIndice = 100
tipos
    registro = rProducto
        entero : código
        cadena: desc
        entero : stock
        entero : estado
    fin_registro
    archivo_d de rProducto = aProducto
    registro = RIndice
        entero : clave
        entero : NRR
    fin_registro
    array[0..numElemIndice] de RIndice = vIndice
var
    aProducto : A
    rProducto : R
    vIndice : Ind
    entero : n, NRR
inicio
    abrir(A, 'PRODUCTOS.DAT', lectura/escritura)
    n ← 0
    NRR ← 0
    leer(A, R)
    mientras no fda(A) hacer
        NRR ← NRR + 1
        si R.estado = 1 entonces
            n ← n + 1
            ind[n].clave ← R.código
            ind[n].NRR ← NRR
        fin_si
        leer(A, R)
    fin_mientras
    cerrar(A)
    Ordenar(Ind, n)
    GuardarIndice(Ind, n)
fin

//Para no perder el array de índices, el procedimiento GuardarIndice
//lo vuelca en un archivo secuencial
procedimiento GuardarIndice(valor vIndice : v; valor entero : n)
var
    archivo_s de RIndice : A
    entero : i
inicio
    crear('CODIGO.IDX')
    abrir(A, 'CODIGO.IDX', escritura)
    desde i ← 1 hasta n hacer
        escribir(A, v[i])
```

```

    fin_si
    cerrar (A)
fin_procedimiento

```

## Ejemplo: Gestión del archivo indexado de productos

### Programa principal

Para los algoritmos que aparecen a continuación, se suponen las siguientes declaraciones en el cuerpo del programa. En el programa principal, la variable entera n guarda el total de elementos ocupados del índice. El procedimiento CargarIndice, coge el archivo secuencial creado en el ejemplo anterior y lo vuelca sobre un array.

```

algoritmo GestiónArchivoIndexado
const
    MaxReg = 120
    //El índice sólo tiene 100 posiciones ya que sólo hay 100 productos distintos
    numElemIndice = 100
tipos
    registro = rProducto
    ...
    fin_registro
    archivo_d de rProducto = aProducto
    registro = RIndice
    ...
    fin_registro
    array[0.. numElemIndice] de RIndice = vIndice
var
    aProducto : A
    rProducto : R
    vIndice : Ind
    entero : n
inicio
    abrir (A, 'PRODUCTOS.DAT', lectura/escritura)
    CargarIndice (Ind,n)
    ...
    GuardarIndice (Ind,n)
    cerrar (A)
fin

//Carga el archivo secuencial con el índice en un array
//Devuelve el número de elementos ocupados del índice (n)
procedimiento CargarIndice(ref vIndice : v; ref entero : n)
var
    archivo_s de RIndice : A
    RIndice : R
inicio
    abrir (A, 'CODIGO.IDX', escritura)
    n ← 0
    leer (A,R)
    mientras no fda(A) hacer
        n ← n + 1
        v[n] ← R
        leer (A,R)
    fin_mientras
    cerrar (A)
fin_procedimiento

```

### Mantenimiento del archivo

El mantenimiento incluirá los procedimientos para dar altas, bajas y modificaciones en al archivo

```

//El procedimiento Alta añade el registro R en una posición libre del área de datos
//e inserta de forma ordenada la clave y la posición en el área de índices
procedimiento Alta(ref AProductos:A;valor RProducto:R; ref vIndice:Ind; ref entero:n)
var

```

```

RProducto : RAux
entero : NRR,i
inicio
//La función buscar realiza una búsqueda binaria entre los n
//primeros elementos del array Ind y devuelve la posición o 0
si (buscar(Ind,R,n) <> 0) o (numElemIndice = n) entonces
//Ya está o índice lleno
si_no
//Alta en área de datos
//Buscar hueco libre mediante una función HASH
NRR ← hash(R.código)
leer(A, RAux, NRR)
mientras RAux.estado = 1 hacer
    NRR ← NRR mod MaxReg + 1
    leer(A, RAux, NRR)
fin_mientras
//Graba el registro en el área de datos,
//marcándolo como ocupado (R.estado = 1)
R.estado ← 1
escribir(A, NRR,R,)
//Alta en el índice
Ind[0].clave ← R.ISBN
Ind[0].NRR ← NRR
i ← n
mientras Ind[i].clave > Ind[0].clave hacer
    Ind[i+1] ← Ind[i]
    i ← i - 1
fin_mientras
Ind[i+1] ← Ind[0]
n ← n + 1
fin_si
fin_procedimiento

//El procedimiento Baja da una baja lógica en el área de datos
//y elimina la entrada en el área de índices
//El argumento R contiene el código del producto a dar de baja
procedimiento Baja(ref AProductos:A; valor RProducto:R; ref vIndice:Ind;
ref entero:n)

var
    entero : p,i
inicio
p ← buscar(Ind,R,n)
si p = 0 entonces
//No está
si_no
//Dar una baja lógica en el área de datos
leer(A, Ind[p].NRR,R)
R.estado ← 2
escribir(A, Ind[p].NRR,R)
//Eliminar del índice
desde i ← p hasta n-1 hacer
    Ind[i] ← Ind[i+1]
fin_desde
n ← n - 1
fin_si
fin_procedimiento

//El procedimiento Modificar, modifica un registro en el área de datos.
//El índice no cambia, ya que el código del producto no se puede modificar
//El argumento R contiene el código del producto a modificar y los campos modificados
procedimiento Modificar(ref AProductos:A; valor RProducto:R; valor vIndice:Ind;
valor entero:n)

var
    entero : p

```

```

inicio
  p ← buscar(Ind,R,n)
  si p = 0 entonces
    //No está
  si_no
    escribir (A, Ind[p].NRR,R)
  fin_si
fin_procedimiento

```

## Acceso a los registros

El **acceso directo** (por ejemplo para una consulta) se realiza mediante una búsqueda binaria en el área de índices y accediendo de forma directa al registro a partir de su NRR.

```

procedimiento Consulta(ref AProductos:A; valor RProducto:R; valor vIndice:Ind;
                      valor entero:n)
var
  entero : p
inicio
  p ← buscar(Ind,R,n)
  si p = 0 entonces
    //No está
  si_no
    leer (A, Ind[p].NRR,R)
    //Procesar el registro (por ejemplo, presentarlo por pantalla)
    escribir (R.código,R.desc,R.stock)
  fin_si
fin_procedimiento

```

El **acceso secuencial** se realiza accediendo secuencialmente a la tabla de índices y leyendo de forma directa el registro en el área de datos a partir de su NRR.

```

procedimiento Listado(ref AProductos:A; valor RProducto:R; valor vIndice:Ind;
                      valor entero:n)
var
  entero : i
inicio
  desde i ← 1 hasta n hacer
    leer (A, Ind[i].NRR,R)
    //Procesar el registro (por ejemplo, presentarlo por pantalla)
    escribir (R.código,R.desc,R.stock)
  fin_desde
fin_procedimiento

```