**Algoritmo A23Libros**

Una biblioteca requiere almacenar la información de sus libros con los siguientes datos:

Autor(es), Título, Editorial, Número de edición, País de origen, ISBN, Fecha de registro, Género, Número de copias.

Cada título tiene un ID consecutivo.

El género sólo puede ser: Lenguajes de programación, Redes y telecomunicaciones, Bases de Datos, y Seguridad informática.

**Algoritmo A23Libros**

Const

MAX\_LIBROS = 100

MAX\_AUTORES = 5

Tipos

TrFecha = Registro

eDia, eMes, eAnio : entero

Fin Registro // TrFecha

TeGenero = Enumerado {LP, TELE, BD, SEGURIDAD}

TrLibro = Registro

aAutores = arreglo[1..MAX\_AUTORES] de entero

cTitulo, cEditorial, cPais, cISBN, : cadena

eNumEdicion, eAnioEdicion, eCopias, eId : entero

rFecRegistro : TrFecha

Fin Registro // TrLibro

TaLibros = Arreglo [1..MAX\_LIBROS] de TrLibro

Var

eTotal, eOpcion, eId : entero

rLibro : TrLibro

cTitulo : cadena

aLibros : TaLibros

eGenero : TeGenero

**Funcion LeerDatosLibro() : TrLibro**

Var

cOtro : carácter

rLibro : TrLibro

eAutor : entero

Inicio

Escribir (“Ingrese los datos del libro:”)

Escribir (“Título: “)

Leer (rLibro.cTitulo)

Hacer eAutor = 1

Repetir

Escribir (“Ingrese el autor”, eAutor,”:”)

Leer (rLibro.aAutores[eAutor])

Escribir (“¿Desea ingresar otro autor?”)

Leer (cOtro)

Hacer eAutor = eAutor + 1

Hasta que (cOtro = ‘n’ o cOtro = ‘N’)

Escribir (“Ingrese la editorial:”)

Leer (rLibro.cEditorial)

Escribir (“Número de edición: “)

Leer (rLibro.eNumEdicion)

Escribir (“Anio de edición: “)

Leer (rLibro.eAnioEdicion)

Escribir (“País de origen: “)

Leer (rLibro.cPais)

Escribir (“ISBN: “)

Leer (rLibro.cISBN)

Escribir (“Fecha de registro: “)

Escribir (“Día:”)

Leer (rLibro.FecRegistro.eDia)

Escribir (“Mes:”)

Leer (rLibro.FecRegistro.eMes)

Escribir (“Año:”)

Leer (rLibro.FecRegistro.eAnio)

Escribir (“Copias: “)

Leer (rLibro.eCopias)

Escribir (“Genero: “)

Leer (rLibro.eGenero)

Hacer rLibro = GenerarIdLibro ()

Regresar rLibro

Fin Función // LeerDatosLibro

**Procedimiento InsertarLibro (E/S aLibros : TaLibros, E/S eTotal: entero, E rLibro: TrLibro)**

Var

eRecorre: entero

Inicio

Si (eTotal = MAX\_LIBROS) entonces

Escribir (“No se pueden agregar más libros.”)

De lo contrario

Si (eTotal = 0 o rLibro.eId > aLibros[eTotal].eId) entonces

Hacer eTotal = eTotal + 1

Hacer aLibros[eTotal] = rLibro

De lo contrario

Si (BuscarLibroPorId (aLibros, eTotal, rLibro.eId) <> 0) entonces

Escribir (“Ya existe el ID”, rLibro.eId)

De lo contrario

Hacer eRecorre = eTotal

Mientras (eRecorre >= 1 y aLibros[eRecorre].eId > rLibro.eId) hacer  
 Hacer aLibros[eRecorre + 1] = aLibros[eRecorre]

Hacer eRecorre = eRecorre - 1

Fin Mientras

Hacer aLibros[eRecorre + 1] = rLibro

Hacer eTotal = eTotal + 1

Fin Si

Fin Si

Fin Si

Fin Procedimiento // InsertarLibro

**Función BuscarLibroPorId(E aLibros: TaLibros, E eTotal: entero, E eId: entero): entero**

Var

eMenor, eMayor, eMedio: entero

Inicio

Si ( eTotal = 0 ) entonces

Regresar 0

De lo contrario

Hacer eMenor = 1

Hacer eMayor = eTotal

Hacer eMedio = (eMayor + eMenor) DIV 2

Mientras (eMenor < eMayor y aLibros[eMedio].eId <> eId)

Si (eId > aLibros[eMedio].eId) entonces

Hacer eMenor = eMedio + 1

De lo contrario

Hacer eMayor = eMedio - 1

Fin Si

Hacer eMedio = (eMenor + eMayor) DIV 2

Fin Mientras

Si (aLibros [eMedio].eId = eId) entonces

Regresar eMedio

De lo contrario

Regresar 0

Fin Si  
Fin Si

Fin Función// BuscarLibroPorId

**Procedimiento BorrarLibroPorId(E/S aLibros: TaLibros, E/S eTotal: entero, E eId: entero)**

Var

ePos, eRecorre: entero

Inicio

Si (eTotal = 0) entonces

Escribir (“No hay libros almacenados.”)

De lo contrario

Hacer ePos = BuscarLibroPorId (aLibros, eTotal, eId)

Si (ePos = 0) entonces

Escribir (“No existe el ID”, eId)

De lo contrario

Desde eRecorre = ePos hasta eTotal – 1 hacer  
 Hacer aLibros[eRecorre + 1] = aLibros[eRecorre + 1]

Hacer eRecorre = eRecorre + 1

Fin Desde

Fin Si

Fin Si

Fin // Procedimiento BorrarLibroPorId

**Procedimiento ListarLibros(E aLibros: TaLibros, E eTotal: entero)**

Var

eAutor: entero

Inicio

Si (eTotal <> 0) entonces

Ejecutar ListarLibros(aLeibros, eTotal)

Escribir (aLibros[eTotal].eId, aLibros[eTotal].eTitulo)

Desde (eAutor = 1 hasta MAX\_AUTORES) hacer

Escribir (ObtenerNombreAutor (aLibros[eTotal], aAutores[eAutor])

Fin Desde

Escribir (aLibros[eTotal].eNumEdición,

aLibros[eTotal].eAnioEdicion,

aLibros[eTotal].cEditorial,

aLibros[eTotal].cPais,

aLibros[eTotal].cISBN,

aLibros[eTotal].rFecRegistro.eDia, “/”, aLibros[eTotal].rFecRegistro.eMes, “/”, aLibros[eTotal].rFecRegistro.eAnio,

ObtenerGenero(aLibros[eTotal].eGenero),

aLibros[eTotal].eCopias)

De lo contrario

Escribir (“Registro vacío.”)

Fin Si

Fin Procedimiento // ListarLibros

**Función ObtenerGenero(E eGenero: entero): cadena**

Var

cGenero: cadena

Inicio

En caso de (eGenero) hacer

LP: Hacer cGenero = “Lenguajes de Programación”

TELE: Hacer cGenero = “Redes y telecomunicaciones”

BD: Hacer cGenero = “Bases de Datos”

SEGURIDAD: Hacer cGenero = “Seguridad Informática”

De lo contrario

Hacer cGenero = “Indefinido”

Fin En caso de

Regresar cGenero

Fin Función // ObtenerGenero