

EJEMPLOS DE TIPO DE DATOS ABSTRACTOS

LISTA:

```
TipoLista Crear()
void Imprimir(TipoLista lista)
int ListaVacía(TipoLista lista)
void Insertar(TipoLista lista, TipoElemento elem)
void Eliminar(TipoLista lista, TipoElemento elem)
Opcionalmente, si la implementación lo requiere, puede definirse:
int ListaLlena(TipoLista lista)
```

COLA:

```
TipoCola Crear();
/* Crea y devuelve una cola vacía */
int ColaVacía(TipoCola Cola);
/* Devuelve 1 sólo si "cola" está vacía */ int ColaLlena(TipoCola Cola); /*
Devuelve 1 sólo si "cola" está llena */
int Sacar(TipoCola Cola, TipoElemento *elem);
/* Saca el siguiente elemento del frente y lo pone en "elem" */
int Meter(TipoCola Cola, TipoElemento elem);
/* Mete "elem" al final de la cola */
```

PILA:

```
TipoPila Crear(void); /* vacía */
/* Crea una pila vacía */
int PilaVacía(TipoPila pila);
/* Comprueba si la pila está vacía */
int PilaLlena(TipoPila pila);
/* Comprueba si la pila está llena */
void Sacar(TipoPila *pila, TipoElemento *elem);
/* Saca un elemento. No comprueba antes si la pila está vacía */
void Meter(TipoPila pila, TipoElemento elem);
/* Mete un elemento en la pila. No comprueba si está llena. */
```

ARBOL BINARIO:

CrearVacío(): TipoABin

(* Crea un árbol vacío *)

CrearRaíz(elem: TipoElemento): TipoABin

(* Crea un árbol de un único elemento *)

ArbolVacío(árbol: TipoABin): B

(* Comprueba si "árbol" está vacío *)

AsignarIzq(VAR árbol: TipoABin; izq: TipoABin)

(* Establece "izq" como subárbol izq. de "árbol" *)

AsignarDer(VAR árbol: TipoABin; der: TipoABin)

(* Establece "izq" como subárbol izq. de "árbol" *)

Info(árbol: TipoABin): TipoElemento

(* Devuelve el nodo raíz de "árbol" *)

Izq(árbol: TipoABin): TipoABin

(* Devuelve el subárbol izquierdo de "árbol" *)

Der(árbol: TipoABin): TipoABin

(* Devuelve el subárbol derecho de "árbol" *)

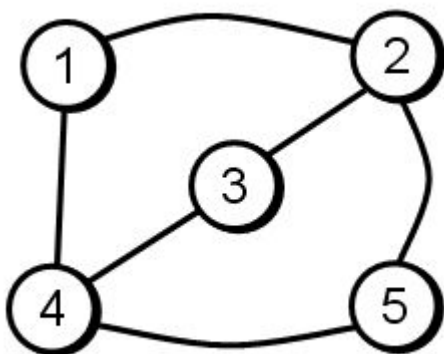
BorrarRaíz(VAR árbol, izq, der: TipoABin)

(* Devuelve en "izq" y "der" los subárboles de "árbol" *)

Imprimir(árbol: TipoABin)

(* Imprime en pantalla el contenido completo de "árbol" *)

GRAFO:



M	1	2	3	4	5
1	0	1	0	1	0
2	1	0	1	0	1
3	0	1	0	1	0
4	1	0	1	0	1
5	0	1	0	1	0

REFERENCIAS:

<http://www.lcc.uma.es/~jlleivao/algoritmos/t5.pdf>