## ALGORITMOS DE CLASIFICACIÓN INTERNA

		Pseudo-código	Lenguaje de programación C
Tipos de datos	Const MAX=1000		#define MAX 1000
	Tipos tipo_vector: vector[1MAX] de tipo_elemento		
	tipo_elemento: registro de		typedef struct{
	clave: tipo_clave		int clave;
	{otros campos}		/*resto de componentes*/
	Fin_registro		} tipo_elemento;
	tipo_clave: entero		<pre>typedef tipo_elemento tipo_vector[MAX];</pre>
recta	Procedimiento inserción directa(a: tipo vector(E/S), n: entero(E))		<pre>void insercion(tipo_vector a,int n) {</pre>
	Var	i,j: entero	int i,j;
		x: tipo_elemento	tipo_elemento x;
	Inicio	Desde i=2 Hasta n Hacer	
	x <b>←</b> a[i]		for (i=1;i <n;++i) th="" {<=""></n;++i)>
Di		j <b>←</b> i-1	x=a[i];
	Mientras (j≥1)y(x.clave <a[j].clave) hacer<="" th=""><th>j=i-1;</th></a[j].clave)>		j=i-1;
ón	a[j+1]←a[j] j←j-1		while $((j>=0) \&\& (x.clave < a[j].clave)) $ {
-			a[j+1]=a[j];
ŭ	Fin mientras		j=j-1;
W W		a[j+1] <b>←</b> x	}
Inserci		Fin desde	a[j+1]=x;
H	Fin procedimiento		}
	Tin_procec	TIMI CITEO	}

```
Procedimiento SelecciónDirecta(a: tipo vector (E/S), n: entero(E))
                                                                                  void seleccion(tipo vector a, int n) {
                   i, j, k: entero
      Var
                                                                                     int i,j;
                   x: tipo elemento
                                                                                     tipo elemento x;
recta
                   Desde i=1 Hasta n-1 Hacer
                                                                                     int \overline{k};
      Inicio
                                                                                     for (i=0; i< n-1; ++i) {
                                                                                        k=i;
                          Desde j=i+1 Hasta n Hacer
Di
                                                                                        for (j=i+1;j<n;++j)
                                Si (a[j].clave<a[k].clave)</pre>
ón
                                             k←j
                                                                                           if (a[j].clave < a[k].clave)
                                Entonces
                                Fin si
Selecci
                          Fin desde
                                                                                        x=a[k];
                                                                                        a[k]=a[i];
                         x \leftarrow a[k]
                                                                                        a[i]=x;
                          a[k] \leftarrow a[i]
                         a[i]←x
                   Fin desde
      Fin procedimiento
      Procedimiento Burbuja(a: tipo vector (E/S), n: entero (E))
                                                                                  void burbuja(tipo vector a,int n) {
cto
                   i, j: entero
      Var
                                                                                     int i, j;
                   x: tipo elemento
                                                                                     tipo elemento x;
      Inicio
                   Desde i=2 Hasta n Hacer
                          Desde j=n Hasta i paso=-1 Hacer
                                                                                     for (i=1;i<n;i++)
A
                                Si (a[j].clave<a[j-1].clave)</pre>
  (Burbuj
                                                                                        for (j=n-1; j>=i; j--)
0
                                             x←a[j]
                                                                                           if (a[j].clave<a[j-1].clave) {
                                Entonces
Intercambi
                                             a[j]←a[j-1]
                                                                                               x=a[j];
                                                                                               a[j]=a[j-1];
                                             a[j-1]←x
                                                                                               a[j-1]=x;
                                Fin si
                          Fin desde
                   Fin desde
      Fin procedimiento
```

## ALGORITMOS DE BÚSQUEDA BINARIA

		Pseudo-código	Lenguaje de programación C
	Const MAX=1000		#define MAX 1000
d de	Tipos tipo_vector: vector[1MAX] de tipo_elemento tipo_elemento: registro de clave: tipo_clave {otros campos}  Fin_registro tipo_clave: entero		
Tipos d datos			typedef struct{
os atc			int clave;
d h			/*resto de componentes*/
H			} tipo_elemento;
			<pre>typedef tipo_elemento tipo_vector[MAX];</pre>
	Procedimiento Búsqueda_binaria(a: tipo_vector (E/S),		<pre>void Busqueda_binaria(tipo_vector a, int n,</pre>
a S 🕣		<pre>n: entero (E), clave:tipo_clave(E),</pre>	<pre>int clave, int *encontrado, int *pos) {</pre>
de ntc nal		encontrado: lógico (S), pos: entero (S))	<pre>int izq,der,med;</pre>
encia de elemento o igual)	Var	izq,der,med: entero	
ii e ii	Inicio	izq <b>←</b> 1	izq=0;
ocurrencia :imer eleme mayor o ig		der <b>←</b> n	der=n-1;
		Mientras (izq≤der) Hacer	<pre>while(izq&lt;=der) {</pre>
ir		$med \leftarrow (izq+der)$ DIV 2	<pre>med=(izq+der)/2;</pre>
ocurr rimer mayor		Si (clave≤a[med].clave)	<pre>if(clave&lt;=a[med].clave)</pre>
o :I =		Entonces der←med-1	der=med-1;
ca ocurr (primer re mayor		Sino izq <b>←</b> med+1	else izq=med+1;
		Fin si	}
imera ave (E		Fin mientras	*pos=izq;
Primera clave (F e clave		pos <del>←</del> izq	*encontrado $\leftarrow$ (izq<=n) &&a[izq].clave==clave);
Prime clave de cla		encontrado←(izq≤n)y(a[izq].clave=clave)	}
	Fin_proce		

```
Procedimiento Búsqueda pos insercion(a: tipo vector (E/S),
                                                                                   void Busqueda pos insercion(tipo vector a, int n,
inserción
                                n: entero (E), clave:tipo clave(E),
                                                                                                          int clave, int *pos) {
  elemento de
    estrictamente
                                encontrado: lógico (S))
                                                                                         int izq,der,med;
                          izq,der,med: entero
            Var
                                                                                         izq=0;
             Inicio
                         izq←1
                                                                                         der=n-1;
                         der←n
                                                                                         while(izq<=der) {</pre>
                         Mientras (izq≤der) Hacer
                                                                                                med=(izq+der)/2;
                                med←(izq+der) DIV 2
de
                                                                                                if(clave<a[med].clave)</pre>
                                Si (clave < a [med].clave)
  (primer clave es
                                                                                                       der=med-1;
Posición
                                             der←med-1
                                Entonces
                                                                                                else izq=med+1;
                                             izq←med+1
                                Sino
                                Fin si
                                                                                         *pos=izq;
                          Fin mientras
                         pos←izq
            Fin procedimiento
```