

```
--|:.....:
--| MODULO DE IMPLEMENTACION DEL TAD 'conjunto generico'
--|:.....:
--| Version: 2.0
--| Fecha: 31-X-96; corregido 16-10-03
--| Autor: Javier Campos; corregido por Elvira Mayordomo
--| FICHERO: conjuntos.adb
```

package body conjuntos is

procedure vacio (A:out conjunto) **is**

-- Post: A=vacío

-- coste en tiempo $O(\text{número de datos de tipo elemento})$

begin

 A.card:=0;

for e **in** elemento **loop** A.elmto(e):=false; **end loop**;

end vacio;

function esVacio (A: conjunto) **return** boolean **is**

-- Post: esVacio(A)=esvacío(A)

-- coste en tiempo $O(1)$

begin

return A.card=0;

end esVacio;

function pertenece (e: elemento; A: conjunto) **return** boolean **is**

-- Post: pertenece(e,A)= $e \in A$

-- coste en tiempo $O(1)$

begin

return A.elmto(e);

end pertenece;

procedure poner (e:in elemento; A:in out conjunto) **is**

-- Pre: $A=A_0$

-- Post: $A=\text{añadir}(e,A_0)$

-- coste en tiempo $O(1)$

begin

if not pertenece(e,A) **then** A.elmto(e):=true; A.card:=A.card+1; **end if**;

end poner;

procedure quitar (e:in elemento; A:in out conjunto) **is**

-- Pre: $A=A_0$

-- Post: $A=\text{quitar}(e,A_0)$

-- coste en tiempo $O(1)$

begin

if pertenece(e,A) **then** A.elmto(e):=false; A.card:=A.card-1; **end if**;

end quitar;

procedure union (A,B:**in** conjunto; C:**out** conjunto) **is**

-- Post: $C = A \cup B$

-- coste en tiempo $O(\text{número de datos de tipo elemento})$

SOL:conjunto;

begin

 SOL.card:=0;

for e **in** elemento **loop**

 SOL.elmto(e):=A.elmto(e) or B.elmto(e);

if SOL.elmto(e) **then** SOL.card:=SOL.card+1; **end if**;

end loop;

 C:=SOL;

end union;

procedure interseccion (A,B:**in** conjunto; C:**out** conjunto) **is**

-- Post: $C = A \cap B$

-- coste en tiempo $O(\text{número de datos de tipo elemento})$

SOL:conjunto;

begin

 SOL.card:=0;

for e **in** elemento **loop**

```
        SOL.elmto(e):=A.elmto(e) and B.elmto(e);  
        if SOL.elmto(e) then SOL.card:=SOL.card+1; end if;  
    end loop;  
    C:=SOL;  
end interseccion;
```

```
function cardinal (A: conjunto) return integer is  
-- Post: cardinal(A)=cardinal(A)  
-- coste en tiempo O(1)  
begin  
    return A.card;  
end cardinal;
```

```
end conjuntos;
```