Ejercicio 5. Especifique los TADs indicados a continuación pero utilizando los observadores propuestos:

b) Conjunto(T) observado con funciones

TAD conjunto<T> {

obs está(elem: T): Bool → no hace falta definir la f. del obs.

· proc nuevoConjunto(): conjunto<T> {

rep {True}

asig {estáVacío}}

· proc vacío(in c: conjunto<T>): Bool {

rep {True}

asig {res <=> estáVacío(c)}}

· proc pertenece(in c: conjunto<T>, in elem<T>): Bool {

rep {True}

asig {c.está(elem)}}

· proc agregar(in elem<T>, inout c: conjunto<T>) {

rep {c = c_0}

rep {¬c_0.está(elem)}

asig {difDeUnElem(c, c_0, elem) ∧ c.está(elem)}}

· pred difDeUnElem(in c_1: Conjunto<T>, in c_2: conjunto<T>, in elem: T): Bool {

(∀e: T)(e ≠ elem → (c_1.está(e) <=> c_2.está(e)))}

· proc quitar(in elem<T>, inout c: conjunto<T>) {

rep {c = c_0}

rep {¬c_0.está(elem)}

asig {difDeUnElem(c, c_0, elem) ∧ ¬c.está(elem)}}

```
proc unión(in c1: Conjunto<T>, in c2: Conjunto<T>): Conjunto<T>{
    req {True}
    aseg {(∀ elem: T)(enUnión(c1, c2, e) ∧ res.está(elem))}}.

pred enUnión(in c1: Conjunto<T>, in c2: Conjunto<T>, in e: T){
    c1.está(e) ∨ c2.está(e)}

proc Intersección(in c1: Conjunto<T>, in c2: Conjunto<T>): Conjunto<T>{
    req {True}
    aseg {(∀ elem: T)(enInt(c1, c2, e) ∧ res.está(elem))}}.

pred enInt(in c1: Conjunto<T>, in c2: Conjunto<T>, in e: T){
    c.está(e) ∧ c2.está(e)}

proc diferencia(in c1: Conjunto<T>, in c2: Conjunto<T>): Conjunto<T>{
    req {True}
    aseg {(∀ elem: T)(endiferencia(c1, c2, e) ⟺ res.está(elem))}}.

pred endiferencia(in c1: conjunto<T>, c2: conjunto<T>, in e: T){
    c1.está(e) ∧ ¬c2.está(e)}.

}
```