

# Especificacion de TADs

## Ejercicio 1

```
TAD NumeroRac{
  obs num:  $\mathbb{Z}$ 
  obs den:  $\mathbb{Z}$ 

  proc nuevoRac (in  $n, d: \mathbb{Z}$ ):NumeroRac {
    requiere  $\{d \neq 0\}$ 
    asegura  $\{res.num = n \wedge res.den = d\}$ 
  }

  proc suma (inout  $a: NumeroRac, in\ b: NumeroRac$ ):NumeroRac {
    requiere  $\{a = A_0\}$ 
    asegura  $\{a.num = A_0.num * b.den + A_0.den * b.num\}$ 
  }

  proc resta (inout  $a: NumeroRac, in\ b: NumeroRac$ ):NumeroRac {
    requiere  $\{a = A_0\}$ 
    asegura  $\{a.num = A_0.num * b.den - A_0.den * b.num\}$ 
  }

  proc multiplicacion (inout  $a: NumeroRac, in\ b: NumeroRac$ ):NumeroRac {
    requiere  $\{a = A_0\}$ 
    asegura  $\{a.num = A_0.num * b.num \wedge a.den = A_0.den * b.den\}$ 
  }

  proc division (inout  $a: NumeroRac, in\ b: NumeroRac$ ):NumeroRac {
    requiere  $\{a = A_0\}$ 
    asegura  $\{a.num = A_0.num * b.den \wedge a.den = A_0.den * b.num\}$ 
  }

  proc iguales (in  $a, b: NumeroRac$ ):Bool {
    requiere  $\{True\}$ 
    asegura  $\{res = True \leftrightarrow a.num * b.den = a.den * b.num\}$ 
  }
}
```

## Ejercicio 2

```
TAD Rectangulo2D{
  obs vsi: tupla( $\mathbb{R}, \mathbb{R}$ )
  obs vsd: tupla( $\mathbb{R}, \mathbb{R}$ )
  obs vii: tupla( $\mathbb{R}, \mathbb{R}$ )
  obs vid: tupla( $\mathbb{R}, \mathbb{R}$ )
  aux resta (in  $a, b: tupla(\mathbb{R}, \mathbb{R})$ ):tupla( $\mathbb{R}, \mathbb{R}$ )= $(a_0 - b_0, a_1 - b_1)$ 
  aux prod (in  $a, b: tupla(\mathbb{R}, \mathbb{R})$ ): $\mathbb{R}=a_0 * b_0 + a_1 * b_1$ 
  pred noRepes (in  $t: seq(\mathbb{R})$ ){
     $(\forall i, j: \mathbb{Z})(0 \leq i, j < |t| \wedge i \neq j \rightarrow_L t[i] \neq t[j])$ 
  }
}
```

```

proc nuevoRectangulo (in a,b,c,d:ℝ):Rectangulo2D {
  requiere {noRepes(⟨a,b,c,d⟩)}
  asegura {prod(resta(a.vsi,b.vsd),resta(a.vsi,c.vii))=0}
  asegura {prod(resta(d.vid,b.vsd),resta(d.vid,c.vii))=0}
  asegura {prod(resta(c.vii,a.vsi),resta(c.vii,d.vid))=0}
  asegura {prod(resta(b.vsd,a.vsi),resta(b.vsd,d.vid))=0}
}

proc mover (inout r:Rectangulo2D,in dx,dy:ℝ):Rectangulo2D {
  requiere {r = R0}
  asegura {r.vsi = (R0.vsi0 + dx, R0.vsi1 + dy)}
  asegura {r.vsi = (R0.vsd0 + dx, R0.vsd1 + dy)}
  asegura {r.vsi = (R0.vii0 + dx, R0.vii1 + dy)}
  asegura {r.vsi = (R0.vid0 + dx, R0.vid1 + dy)}
}

```

### Ejercicio 3

a) TAD Cola $\langle T \rangle$

```

obs s: seq⟨T⟩

proc nuevaCola (in s:seq⟨T⟩):cola⟨T⟩ {
  requiere {True}
  asegura {|res.s|=0}
}

proc estaVacía (in c:cola⟨T⟩):Bool {
  requiere {True}
  asegura {res = True ↔ |c.s| = 0}
}

proc encolar (inout c:cola⟨T⟩,e:T):cola⟨T⟩ {
  requiere {c = C0}
  asegura {c.s = concat(C0.s,{e})}
}

proc desencolar (inout c:cola⟨T⟩):T {
  requiere {c = C0}
  asegura {c.s = tail(C0.s) ∧ res = head(C0.s)}
}

TAD Pila⟨T⟩{
  obs s: seq⟨T⟩

  proc nuevaPila (in s:seq⟨T⟩):pila⟨T⟩ {
    requiere {True}
    asegura {|res.s|=0}
  }
}

```

```

proc estaVacía (in  $c : \text{pila}\langle T \rangle$ ):Bool {
  requiere {True}
  asegura { $res = \text{True} \leftrightarrow |c.s| = 0$ }
}

proc apilar (inout  $c : \text{pila}\langle T \rangle, e : T$ ): $\text{pila}\langle T \rangle$  {
  requiere { $c = C_0$ }
  asegura { $c.s = \text{concat}(\{e\}, C_0.s)$ }
}

proc desapilar (inout  $c : \text{pila}\langle T \rangle$ ):T {
  requiere { $c = C_0$ }
  asegura { $c.s = \text{tail}(C_0.s) \wedge res = \text{head}(C_0.s)$ }
}

```