



# Especificación de TADs

\$Berretacoin

18 de mayo de 2025

Algoritmos y Estructuras de Datos / Algoritmos y Estructuras de Datos II

LVJM03

Integrante	LU	Correo electrónico
Cattanio, Mateo	1244/23	mateocattanio@gmail.com
Cellerino, Juan Bautista	697/22	jcellerino@gmail.com
Ruiz Díaz González, Lucio Tadeo	162/24	luciotadeo02@gmail.com
Villanueva, Valentin	1925/21	valentincordoba00@gmail.com



**Facultad de Ciencias Exactas y Naturales**  
Universidad de Buenos Aires

Ciudad Universitaria - (Pabellón I/Planta Baja)

Intendente Güiraldes 2610 - C1428EGA

Ciudad Autónoma de Buenos Aires - Rep. Argentina

Tel/Fax: (++54 +11) 4576-3300

<http://www.exactas.uba.ar>

```

TAD $Berretacoin {

obs  cadena : seq⟨Struct⟨idBloque : ℤ, bloque : seq⟨Struct⟨transaccion : ℤ, comprador : ℤ, vendedor : ℤ, monto : ℤ⟩⟩⟩⟩

pred esTransaccionValida (t: Struct⟨transaccion : ℤ, comprador : ℤ, vendedor : ℤ, monto : ℤ⟩) {

    (t.transaccion ≥ 0) ∧ (t.comprador ≥ 0) ∧ (t.vendedor > 0) ∧ (t.comprador ≠ t.vendedor) ∧ (t.monto > 0)

}

pred esTransaccionCreacion (b: $Berretacoin, t: Struct⟨transaccion : ℤ, comprador : ℤ, vendedor : ℤ, monto : ℤ⟩) {

    (t.transaccion ≥ 0) ∧ (t.comprador = 0) ∧ (∀i : ℤ) (0 ≤ i < |b| ∧L b[i].bloque[0].vendedor ≠ t.vendedor) ∧ (t.monto = 1)

}

pred transaccionesOrdenadas (ts: seq⟨Struct⟨transaccion : ℤ, comprador : ℤ, vendedor : ℤ, monto : ℤ⟩⟩) {

    (∀i : ℤ) (0 ≤ i < |ts| - 1 →L ts[i].transaccion < ts[i + 1].transaccion)

}

pred esUsuario (m : ℤ) {

    (∀i : ℤ) (0 ≤ i < |b.cadena| →L (∀j : ℤ) (0 ≤ j < |b.cadena[i].bloque| →L ((m = b.cadena[i].bloque[j].vendedor) ∨
    (m = b.cadena[i].bloque[j].comprador))))

}

pred sinRepetidos (s: seq⟨ℤ⟩) {

    (∀i : ℤ) (0 ≤ i < |s| →L (∀j : ℤ) (0 ≤ j < |s| ∧ j ≠ i →L s[j] ≠ s[i]))

}

pred cotizo (l: seq⟨ℤ⟩, b: $Berretacoin, res: seq⟨ℤ⟩) {

    (∀i : ℤ) (0 ≤ i < |l| →L res[i] = l[i] * sumaBloque(b.cadena[i].bloque))

}

pred esPromedio (b: $Berretacoin, res: ℝ) {


$$res = \frac{\sum_{i=0}^{|b.cadena|-1} (\sum_{j=1}^{|b.cadena[i].bloque|-1} b.cadena[i].bloque[j].monto)}{|b.cadena|}$$


}

aux SaldoUsuario (in b : $Berretacoin, in u : ℤ) : ℤ =
(∀i : ℤ) (0 ≤ i < |b.cadena| →L
res = (

$$\sum_{j=0}^{|b.cadena[i].bloque|-1} IfThenElse(u = b.cadena[i].bloque[j].vendedor, b.cadena[i].bloque[j].valor, 0)) -$$


$$(\sum_{j=0}^{|b.cadena[i].bloque|-1} IfThenElse(u = b.cadena[i].bloque[j].comprador, b.cadena[i].bloque[j].valor, 0))) ;$$


aux sumaBloque (in bloque: seq⟨transacciones : ℤ, comprador : ℤ, vendedor : ℤ, monto : ℤ⟩) : ℤ =
res = 
$$\sum_{i=0}^{|bloque|-1} bloque[i].monto ;$$


```

```

proc creacion () : $Berretacoin
    requiere {True}
    asegura {res.cadena = []}

proc agregarBloque (inout b : $Berretacoin, in s : seq⟨Struct⟨transaccion : ℤ, comprador : ℤ, vendedor : ℤ, monto : ℤ⟩⟩) :
    requiere {b = B0}
    requiere {|s| ≤ 50}
    requiere {transaccionesOrdenadas(s)}
    requiere {(∀i : ℤ) (0 < i < |s| →L esTransaccionValida(s[i]))}
    requiere {( |B0.cadena| ≤ 3000 ∧ esTransaccionCreacion(B0, s[0]) )
    ∨ ( |B0.cadena| > 3000 ∧ esTransaccionValida(s[0]) )}
    asegura {b.cadena = B0.cadena ++ < ( |B0.cadena|, s ) >}

proc maximosTenedores (in b : $Berretacoin) : seq⟨ℤ⟩
    requiere {True}
    asegura {k ∈ res ⇔ (esUsuario(m) →L SaldoUsuario(b, k) ≥ SaldoUsuario(b, m))}
    asegura {sinRepetidos(res)}

proc montoMedio (in b : $Berretacoin) : ℝ
    requiere {True}
    asegura {( |b| = 0 ∧ res = 0 ) ∨ ( |b| > 0 ∧ esPromedio(b, res) )}

proc cotizaciónAPesos (in l : seq⟨ℤ⟩, in b : $Berretacoin) : seq⟨ℤ⟩
    requiere {|l| = |b.cadena|}
    requiere {(∀i : ℤ) (0 ≤ i < |l| →L 0 < l[i])}
    asegura {( |res| = |l| ) ∧L (cotizo(l, b, res))}
}

```