



Especificación de TADs

\$Berretacoin

23 de abril 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⟨id : ℤ, seq⟨Struct⟨transaccion : ℤ, comprador : ℤ, vendedor : ℤ, monto : ℤ⟩⟩⟩⟩

obs montosUsuarios : dict(ℤ, ℤ)

obs montosTrans : seq⟨ℤ⟩

obs montosPorBloque : seq⟨ℤ⟩

obs recibieron : seq⟨ℤ⟩

pred esTransaccionValida (in t: Struct⟨transaccion : ℤ, comprador : ℤ, vendedor : ℤ, monto : ℤ⟩) {
    (t.transaccion ≥ 0) ∧ (t.comprador > 0) ∧ (t.vendedor > 0) ∧ (t.monto > 0)
}

pred esTransaccionCreacion (in t: Struct⟨transaccion : ℤ, comprador : ℤ, vendedor : ℤ, monto : ℤ⟩, in r: seq⟨ℤ⟩) {
    (t.transaccion ≥ 0) ∧ (t.comprador = 0) ∧ (t.vendedor ∉ r) ∧ (t.monto = 1)
}

pred esMax (in d : dict(ℤ, ℤ), in k: ℤ) {
    (k ∈ d) ∧ ((n ∈ d) → d[k] ≥ d[n])
}

}

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

proc agregarBloque (inout b : $Berretacoin, in s : seq⟨Struct⟨transaccion : ℤ, comprador : ℤ, vendedor : ℤ, monto : ℤ⟩⟩) :
    requiere {b = B0}
    requiere {|s| ≤ 50}
    requiere {(∀i : ℤ) (0 ≤ i < |s| →L s[i].comprador ≠ s[i].vendedor)}
    requiere {(∀j : ℤ) ((0 ≤ j < |s| ∧L s[j].comprador ≠ 0) →L (s[j].comprador ∈ B0.montosUsuarios ∧ s[j].monto ≤ B0.montosUsuarios(s[j].comprador)))}
    requiere {|B0.cadena| ≤ 3000 → (esTransaccionCreacion(s[0], B0.recibieron) ∧ (∀k : ℤ) (0 < k < |s| →L esTransaccionValida(s[k])))}
    requiere {|B0.cadena| > 3000 → (∀k : ℤ) (0 < k < |s| →L esTransaccionValida(s[k]))}
    asegura {b.cadena = B0.cadena ++< (|B0.cadena|, s) >}
    asegura {b.montosPorBloque = B0.montosPorBloque ++< (∑i=0|s|-1 (s[i].monto)) >}

```

```

asegura  $\{(\forall i : \mathbb{Z}) (0 \leq i < |s| \longrightarrow_L \text{IfThenElse}(s[i].vendedor \in B_0.montosUusuarios, \\ \text{setKey}(b.montosUusuarios, s[i].vendedor, B_0.montosUusuarios(s[i].vendedor) + s[i].monto), \\ \text{setKey}(b.montosUusuarios, s[i].vendedor, s[i].monto)))\}$ 

asegura  $\{(\forall j : \mathbb{Z}) (((0 \leq j < |s|) \wedge_L s[j].comprador \neq 0) \longrightarrow_L \\ \text{setKey}(b.montosUusuarios, s[j].comprador, B_0.montosUusuarios(s[j].comprador) - s[j].monto) \wedge (b.montosTrans = \\ B_0.montosTrans ++ \langle s[j].monto \rangle))\}$ 

asegura  $\{b.recibieron = B_0.recibieron ++ \langle s[0].vendedor \rangle\}$ 

proc maximosTenedores (in b : $Berretacoin) : seq<mathbb{Z}>

  requiere  $\{k \in b.montosUusuarios\}$ 

  asegura  $\{k \in res \iff esMax(b.montosUusuarios, k)\}$ 

proc montoMedio (in b : $Berretacoin) : R

  requiere  $\{True\}$ 

  asegura  $\{ifThenElse(|b.montosTrans| > 0, res = (\frac{\sum_{i=0}^{|b.montosTrans|-1} (b.montosTrans[i])}{|b.montosTrans|}), res = 0)\}$ 

proc cotizaciónAPesos (in l : seq<mathbb{Z}>, in b : $Berretacoin) : seq<mathbb{Z}>

  requiere  $\{|l| = |b.cadena|\}$ 

  requiere  $\{(\forall i : \mathbb{Z}) (0 \leq i \leq |l| \longrightarrow_L 0 < l[i])\}$ 

  asegura  $\{(\forall i : \mathbb{Z}) (0 \leq i \leq |res| \longrightarrow_L res[i] = b.montosPorBloque[i] * l[i])\}$ 

}

```