



DEPARTAMENTO
DE COMPUTACION

Facultad de Ciencias Exactas y Naturales - UBA

Trabajo Práctico 1: especificación de TAD's

Algoritmos y Estructuras de Datos

17 de mayo de 2025

Materia de la carrera

Grupo6

Integrante	LU	Correo electrónico
Abizanda, Facundo	1332/21	facundoabizanda@gmail.com
Julian, Alvarez	1944/21	juroalwi@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>

Especificación de Berretacoin

Transaccion **ES** Struct $\langle id_tr : \mathbb{Z}, id_com : \mathbb{Z}, id_ven : \mathbb{Z}, monto : \mathbb{Z} \rangle$

Transacciones **ES** Seq $\langle Transaccion \rangle$

Bloque **ES** Struct $\langle id_bl : \mathbb{Z}, trs : Transacciones \rangle$

Cadena **ES** Seq $\langle Bloque \rangle$

TAD \$Berretacoin{

obs cadena : Cadena

aux balanceUsuario(id_usuario: \mathbb{Z} , c: Cadena) : \mathbb{Z} =

$$\left(\sum_{i=0}^{|c|-1} \left(\sum_{j=0}^{|c[i].trs|-1} \text{if } c[i].trs[j].id_ven = id_usuario \text{ then } c[i].trs[j].monto \text{ else } 0 \text{ fi} \right) \right) - \left(\sum_{i=0}^{|c|-1} \left(\sum_{j=0}^{|c[i].trs|-1} \text{if } c[i].trs[j].id_com = id_usuario \text{ then } c[i].trs[j].monto \text{ else } 0 \text{ fi} \right) \right)$$

pred transaccionValida(tr: Transaccion, es_creacion: Bool){

$(tr.id_tr \geq 0) \wedge (tr.id_ven > 0) \wedge (tr.id_com \neq tr.id_ven) \wedge (tr.monto > 0) \wedge$
 (
 $(es_creacion \rightarrow_L tr.id_com = 0 \wedge tr.monto = 1) \wedge$
 $(\neg es_creacion \rightarrow_L tr.id_com > 0)$
)
}

pred bloqueValido(bl : Bloque){

$(bl.id_bl \geq 0) \wedge$
 $(bl.id_bl < 3000 \rightarrow_L |bl.trs| \geq 1) \wedge$
 $(|bl.trs| \leq 50) \wedge$
 $(\forall i : \mathbb{Z})(0 \leq i < |bl.trs| \rightarrow_L$
 transaccionValida(bl.trs[i], esTrDeCreacion(bl.id_bl, bl.trs[i].id_tr))
) \wedge
 $(\forall i : \mathbb{Z})(0 \leq i < |bl.trs| - 1 \rightarrow_L$
 $bl.trs[i].id_tr < bl.trs[i+1].id_tr$
)
}

aux esTrDeCreacion(id_tr: \mathbb{Z} , id_bl: \mathbb{Z}) : Bool =

 if $id_bl < 3000 \wedge id_tr = 0$ then True else False fi

pred vendedoresDistintosEnCreacion(cadena : Cadena){

$(\forall i : \mathbb{Z})(0 \leq i < (\text{if } |cadena| \leq 3000 \text{ then } |cadena| \text{ else } 3000 \text{ fi}) \rightarrow_L$
 $(\forall j : \mathbb{Z})(0 \leq j < (\text{if } |cadena| \leq 3000 \text{ then } |cadena| \text{ else } 3000 \text{ fi}) \wedge i \neq j \rightarrow_L$
 $\text{head}(cadena[i].trs).id_ven \neq \text{head}(cadena[j].trs).id_ven$
)
)
}

pred esMaximoTenedor(tenedor: \mathbb{Z} , cadena : Cadena){

$(\forall usuario : \mathbb{Z})(\text{balanceUsuario}(usuario, cadena) \leq \text{balanceUsuario}(tenedor, cadena))$

}

aux sumaMontosTrs(trs: Transacciones, desde: \mathbb{Z}) : \mathbb{Z} =

$$\sum_{i=desde}^{|trs|-1} trs[i].monto$$

aux sumaMontosBloques(cadena: Cadena) : \mathbb{Z} =

$$\sum_{i=0}^{|cadena|-1} \left(\begin{array}{l} \text{if } cadena[i].id_bl < 3000 \\ \text{then sumaMontosTrs}(cadena[i].trs, 1) \\ \text{else sumaMontosTrs}(cadena[i].trs, 0) \text{ fi} \end{array} \right)$$

aux cantidadTrsCadena(cadena: Cadena) : \mathbb{Z} =

$$\sum_{i=0}^{|cadena|-1} |cadena[i].trs| - \text{if } cadena[i].id_bloque < 3000 \text{ then } 1 \text{ else } 0 \text{ fi}$$

```

proc nuevaBerretacoin() : Berretacoin{
  requiere {
    True
  }

  asegura {
    res.cadena = ⟨⟩
  }
}

proc maximosTenedores(in berretacoin : Berretacoin ) : seq ⟨ℤ⟩ {
  requiere {
    True
  }

  asegura {
    (∀ usuario : ℤ)(esMaximoTenedor(usuario, berretacoin.cadena) →L usuario ∈ res)
  }

  asegura {
    (∀ usuario : ℤ)(usuario ∈ res →L esMaximoTenedor(usuario, berretacoin.cadena))
  }

  asegura {
    (∀ i : ℤ)((∀ j : ℤ)(i ≠ j ∧ 0 ≤ i < |res| ∧ 0 ≤ j < |res| →L res[i] ≠ res[j]))
  }
}

proc montoMedio(in berretacoin : Berretacoin ) : ℤ{
  requiere {
    True
  }

  asegura {
    cantidadTrsCadena(berretacion.cadena) > 0 →L
    res = sumaMontosBloques(berretacion.cadena)/cantidadTrsCadena(berretacion.cadena)
  }

  asegura {
    cantidadTrsCadena(berretacion.cadena) = 0 →L res = 0
  }
}

```

```

proc cotizacionAPesos(in berretacoin : Berretacoin, in cotizaciones : seq ⟨ℤ⟩ ) : seq ⟨ℤ⟩ {
  requiere {
    |berretacoin.cadena| = |cotizaciones|
  }

  requiere {
    (∀cotizacion ∈ cotizaciones)(cotizacion > 0)
  }

  asegura {
    |res| = |cotizaciones|
  }

  asegura {
    (∀i : ℤ)(0 ≤ i < |res| →L res[i] = cotizaciones[i] * sumaMontosTrs(berretacoin.cadena[i].trs, 0))
  }
}

proc agregarBloque(inout berretacoin : Berretacoin, in bloque : Bloque ){
  requiere {
    berretacoin = B0
  }

  requiere {
    bloqueValido(bloque)
  }

  requiere {
    bloque.bl_id = |berretacoin.cadena|
  }

  requiere {
    (∀usuario : ℤ)(
      ((∀i : ℤ)(0 ≤ i < |bloque.trs| →L
        ( balanceUsuario(usuario, berretacoin.cadena) –
          
$$\sum_{j=0}^i \text{if } \text{bloque.trs}[j].\text{id\_comp} = \text{usuario} \text{ then } \text{bloque.trs}[j].\text{monto} \text{ else } 0 \text{ fi}$$

        ) ≥ 0
      )
    )
  )
  }

  asegura {
    berretacoin.cadena = B0.cadena ++ ⟨bloque⟩
  }

  asegura {
    vendedoresDistintosEnCreacion(berretacoin.cadena)
  }
}
}

```