



Trabajo Práctico 1: \$BerretaCoin  
Cómo funcionan las Blockchains

Grupo 30

Integrante	LU	Correo electrónico
Rankov, Jorge	714/23	jrankov@dc.uba.ar
Falbo, Tiziana	nnn/nn	nnn@gmail.com
Facundo	nnn/nn	nnn@gmail.com
Bautista	nnn/nn	nnn@gmail.com



Facultad de Ciencias Exactas y Naturales  
Universidad de Buenos Aires  
Ciudad Universitaria – (Pabellon I/Planta Baja)  
Intendente Guiraldes 2610 – C1428EGA  
Ciudad Autónoma de Buenos Aires – Rep. Argentina  
Tel/Fax: (++54 +11) 4576–3300  
<http://www.exactas.uba.ar>

# TAD \$BerretaCoin {

**obs blockchain:** Seq<Struct(transacciones: Seq<Struct(id, idComprador, idVendedor, monto), idBloque>)>

```
proc nuevoBerretaCoin(): BerretaCoin {
  asegura { res.blockchain = <> }
}
```

```
proc agregarBloque (inout bc: $BerretaCoin, in bloque: Seq< $\mathbb{Z} \times \mathbb{Z} \times \mathbb{Z} \times \mathbb{Z}$ >) {
  requiere { ( $|bc| < 3000$ )  $\wedge_L$  ( $|bloque| < 50$ ) }

  requiere { esTransaccionCreacion (bloque[0]) }

  requiere { ( $\forall i: \mathbb{N}$ ) ( $0 \leq i < |bloque| \rightarrow_L$  (bloque[i][1]  $\neq$  bloque[i][2])) }

  asegura { ( $\exists$  idMontos: Seq< $\mathbb{Z} \times \mathbb{Z}$ >) (sinRepetirId (idMontos)  $\wedge_L$  esTransaccionValida (bloque, idMontos)) }

  asegura { bc = concat (bc0, <bloque>) }

  asegura {  $|bc| = |bc_0| + 1$  }
}
```

```
proc agregarBloque2.0 (inout B: BerretaCoin, in S: bloque) {
  requiere {  $B = B_0 \wedge 0 < |S.transacciones| \leq 50 \wedge$  bloqueValido(B, S)  $\wedge$  sonTransaccionesValidas(S, B) }

  asegura { b.blockchain = b0.blockchain + {S} }
}
```

```
proc agregarBloque3.0 (inout cripto: BerretaCoin, in bloque: Seq< $\mathbb{Z} \times \mathbb{Z} \times \mathbb{Z} \times \mathbb{Z}$ >) {
  requiere { length (cripto.blockchain) < 50 }

  requiere { cripto = cripto0 }

  asegura { cripto.blockchain = concat (cripto0.blockchain, <bloque>) }
}
```

```
proc montosDeUsuarios {
  asegura {  $\forall id \in \text{sinRepetidos (Usuarios (Cripto.blockchain))} \rightarrow id \in \text{res}$ 
     $\longleftrightarrow$  (esMaximo (MontoDeUsuario (Cripto.blockchain, id)); Montos (Usuarios (Cripto.blockchain))) }
}
```

```
proc maximosTenedores (in bc: BerretaCoin): Seq< $\mathbb{Z}$ > {
  asegura { ( $\forall i: \mathbb{Z}$ ) ( $0 \leq i < |res|$ )  $\rightarrow$  esUsuario ( $res_{[i]}$ , bc.blockchain) }

  asegura { ( $\forall id: \mathbb{Z}$ ) ( $id \in res$ )  $\longleftrightarrow \neg$  ( $\exists$  otro:  $\mathbb{Z}$ ) (esUsuario (otro; bc.blockchain))  $\wedge$ 
    (montoDeUsuario (otro, bc)  $\geq$  MontoDeUsuario (id, bc)) }
}
```

```
proc montoMedio (S: Seq<Seq< $\mathbb{Z} \times \mathbb{Z} \times \mathbb{Z} \times \mathbb{Z}$ >>):  $\mathbb{Z}$  {
  requiere {  $\forall$  bloque  $\in S$ ,  $|bloque| > 0$  }

  asegura {  $\text{res} = \frac{\sum_{j=0}^{|S|-1} \sum_{i=1}^{|S_{[j]}|-1} S_{[j][i][3]}}{\sum_{j=0}^{|S|-1} (|S_{[j]}|-1)}$  }
}
```

```
proc Usuarios (in bc: BerretaCoin): Seq< $\mathbb{Z}$ > {
  asegura { ( $\forall i: \mathbb{Z}$ ) ( $0 \leq i < |bc.blockchain|$ ) }

  asegura { ( $\forall j: \mathbb{Z}$ ) ( $0 \leq i < |bc.blockchain_{[i]}|$ ) }

  asegura { ( $bc.blockchain_{[i][j][1]} \wedge bc.blockchain_{[i][j][2]} \in res \wedge$  ningunOtroElem  $\in res$  ) }
}
```

```
proc cotizacionAPesos (in cotizaciones: Seq< $\mathbb{Z}$ >, in B: BerretaCoin): Seq< $\mathbb{Z}$ > {
  requiere { ( $\forall i: \mathbb{Z}$ ) ( $0 \leq i < |cotizaciones|$ )  $\rightarrow$  cotizaciones[i] > 0 }

  requiere {  $|cotizaciones| = |B.blockchain|$  }
}
```

```
asegura { |res| = |cotizaciones| }  
  
asegura { (∀i :ℤ)(0 ≤ i < |res|) → res[i] = cotizarBloque (i, cotizaciones, b.bloque) }
```

# Predicados

**pred esTransaccionCreacion** (t: Seq< $\mathbb{Z} \times \mathbb{Z} \times \mathbb{Z} \times \mathbb{Z}$ >) {  
 $t_{[1]} = 0$   
}

**pred sinRepetirId** (ids: Seq< $\mathbb{Z} \times \mathbb{Z}$ >) {  
 $(\forall i, j: \mathbb{N}) ((0 \leq i < |ids| \wedge_L (0 \leq j < |ids| \wedge_L (j \neq i)) \rightarrow_L id_{[i][0]} \neq id_{[j][0]}))$  }

**pred esMaximo** (Monto:  $\mathbb{Z}$ , Montos: Seq< $\mathbb{Z}$ >) {  
 $(\forall i \in Montos) \rightarrow_L Monto \geq i$   
}

**pred esUsuario** (id:  $\mathbb{Z}$ , b: BerretaCoin) {  
 $(\exists i: \mathbb{Z}) (\exists j: \mathbb{Z}) (0 < i \leq |b.blockchain| \wedge_L 0 < j < |b.blockchain_{[i]}.transacciones|) \wedge$   
 $(id = b.blockchain_{[i]}.transacciones_{[j]}.idVendedor)$   
}

**pred bloqueValido** (B: cadenaDeBloques, S: bloque){  
 $(|B| < 3000 \rightarrow S.transacciones_{[0]}.idComprador = 0) \wedge (|B| \geq 3000 \rightarrow S.transacciones_{[0]}.idComprador \neq 0) \wedge$   
 $(|S.transacciones| \leq 50)$   
  
 $\wedge$   
  
 $[(\forall i \in S.transacciones) \rightarrow (S.transacciones_{[i]}.idComprador \neq S.transacciones_{[i]}.idVendedor)] \wedge$   
 $[(\forall j \in S.transacciones) \rightarrow (S.transacciones_{[j]}.id \geq 0 \wedge S.transacciones_{[j]}.idComprador \geq 0 \wedge$   
 $S.transacciones_{[j]}.idVendedor \geq 0 \wedge S.transacciones_{[j]}.idMontos > 0) \wedge$   
 $estaOrdenado(S) \wedge idOrdenado(B, S)]$   
}

**pred esTransaccionValida** (b: Seq< $\mathbb{Z} \times \mathbb{Z} \times \mathbb{Z} \times \mathbb{Z}$ >, ids: Seq< $\mathbb{Z} \times \mathbb{Z}$ >) {  
 $(\forall i, j: \mathbb{N}) (j \leq i < |b| \wedge_L (0 \leq j < |ids|) \wedge_L (b_{[i][2]} = ids_{[j][0]})) \rightarrow_L (b_{[i][3]} \leq ids_{[j][1]})$   
}

**pred sonTransaccionesValidas** (S: bloque, B: blockchain) {  
 $(\forall id: \mathbb{Z}) (\forall j: \mathbb{Z}) (0 \leq j < |S.transacciones|) (esUsuario(id, blockchain) \vee [esUsuarioDeBloque(id, S.transacciones)$   
 $\rightarrow montoDeUsuarioHastaTransferencia(id, S, B, j) \geq 0])$   
}

**pred estaOrdenado** (S: bloque) {  
 $(\forall i: \mathbb{Z}) (0 \leq i < |S.transacciones|) \rightarrow (S.transacciones_{[i]}.id = i)$   
}

# Auxiliares

**aux sinRepetidos** (S: Seq<ℤ>): Seq<ℤ>=

$$[S_{[0]}] + \sum_{i=1}^{|s|-1} ifThenElse(S_{[i]} \in SubSeq(S, 0, i - 1); \emptyset; [S_{[i]}])$$

**aux Usuarios** (S:Seq<Seq<ℤ×ℤ×ℤ×ℤ>>): Seq<ℤ>=

$$\sum_{i=0}^{|s|-i}|s_{[i]}|-1 \sum_{j=0} (S_{[i][j][1]}, S_{[i][j][2]})$$

**aux MontoDeUsuario** (id:ℤ; B:BerretaCoin): ℤ=

$$\sum_{i=0}^{|b.blockchain|-1} - \left( \sum_{j=0}^{|b.blockchain_{[i]}|-1} ifThenElse(id = b.blockchain_{[i][j]_1}, b.blockchain_{[i][j]_3}, 0) \right) + \left( \sum_{j=0}^{|b.blockchain_{[i]}|-1} ifThenElse(id = b.blockchain_{[i][j]_1}, b.blockchain_{[i][j]_2}, 0) \right)$$

**aux Montos** (S: Seq<ℤ>): Seq<ℤ>=

$$\sum_{j=0}^{|s|-1} (MontoDeUuario(S_{[i]}))$$

**aux cotizarBloque** (in posicion. cotizaciones: Seq): ℤ=

$$\left( \sum_{i=0}^{|blockchain[posicion]}|-1 blockchain[posicion]_{[i]}.montos \right) \cdot cotizaciones[posicion]$$

}

# anotaciones

cadenaDeBloques = Seq<Seq<ℤ×ℤ×ℤ×ℤ>>  
bloque = Seq<ℤ×ℤ×ℤ×ℤ>