

# Algoritmos y Estructuras de Datos II

Departamento de Computación  
Facultad de Ciencias Exactas y Naturales  
Universidad de Buenos Aires

## TP 1 - Broker System

### Grupo 20

Integrante	LU	Correo electrónico
Fernando Gasperi Jabalera	56/09	fgasperijabalera@gmail.com
Esteban Romero	659/06	estebantaborcias@gmail.com
Leandro Tozzi	-	leandro.tozzi@gmail.com
Alfredo Terrile Cendoya	022/11	freddy199_0@hotmail.com

### Reservado para la cátedra

Instancia	Docente	Nota
Primera entrega		
Segunda entrega		

## Índice

<b>1. Observaciones</b>	<b>3</b>
<b>2. TAD BROKER</b>	<b>4</b>
<b>3. TAD PROMESA</b>	<b>10</b>
<b>4. TAD TÍTULO</b>	<b>11</b>

## 1. Observaciones

Las aclaraciones que nos parecen pertinente y necesario hacer son las siguientes:

- el TAD broker al cumplir las promesas primero cumple las de venta y luego las de compra. Esto no impone orden alguno sobre el cumplimiento de las promesas, simplemente hace más simple las axiomatizaciones. Esto se debe a que las de compra pueden llegar a habilitarse sólo si las de venta liberan acciones.
- decidimos no aceptar promesas cuya cantidad de acciones sea mayor a la que el broker pueda manejar para ese título. Es decir, no aceptamos promesas que sabemos que nunca vamos a poder cumplir porque el máximo de acciones para un título determinado que maneja el broker es fijo.
- Los TADs cliente y nombre son un renombre de String.

## 2. TAD BROKER

### TAD BROKER

#### igualdad observacional

$$(\forall b, b' : \text{broker}) \left( b =_{\text{obs}} b' \iff \left( \begin{array}{l} \text{titulos}(b) =_{\text{obs}} \text{titulos}(b') \wedge \text{clientes}(b) =_{\text{obs}} \text{clientes}(b') \wedge \\ (\forall c : \text{cliente}) (\text{esta?}(c, \text{clientes}(b)) =_{\text{obs}} \text{esta?}(c, \text{clientes}(b'))) \\ \wedge \text{esta?}(c, \text{clientes}(b)) \Rightarrow (\text{promesas}(c, b) =_{\text{obs}} \text{promesas}(c, b')) \\ \wedge \text{acciones}(c, b) =_{\text{obs}} \text{acciones}(c, b')) \end{array} \right) \right)$$

**géneros** broker

**exporta** broker, observadores, enAlza

**usa** BOOL, NAT

#### observadores básicos

titulos	: broker	→ conj(titulo)	
promesas	: cliente $c \times$ broker	→ conj(promesas)	$\{c \in \text{clientes}(b)\}$
clientes	: broker	→ conj(clientes)	
acciones	: cliente $c \times$ nombre $n \times$ broker $b$	→ nat	$\{\text{estaDefinido?}(n, \text{titulos}(b)) \wedge c \in \text{clientes}(b)\}$

#### generadores

nuevo	: conj(titulo) $ts \times$ conj(cliente) $cs$	→ broker	$\{\neg \text{Vacio?}(cs) \wedge \neg \text{hayNombresRepetidos}(ts)\}$
CambiarCotizacion	: nombre $n \times$ cotizacion $\times$ broker $b$	→ broker	$\{\text{estaDefinido?}(n, \text{titulos}(b))\}$
AgPromesa	: cliente $c \times$ promesa $p \times$ broker $b$	→ broker	$\left\{ \begin{array}{l} c \in \text{clientes}(b) \wedge \text{estaDefinido?}(\text{nombreTitulo}(p), \text{titulos}(b)) \wedge_{\text{L}} \text{cantidad}(p) \leq \\ \text{maximoAcciones}(\text{nombreTitulo}(p), \text{titulos}(b)) \wedge \neg \text{HayPromMismoTipo}(p, \text{promesas}(c, b)) \\ \wedge_{\text{L}} (\text{vende?}(p) \Rightarrow \text{puedeVender?}(c, p, b)) \end{array} \right\}$

#### otras operaciones

hayNombresRepetidos	: conj(titulo) $ts$	→ bool	
maximoAcciones	: nombre $n \times$ conj(titulo) $ts$	→ nat	
enAlza	: nombre $n \times$ broker $b$	→ bool	$\{\text{estaDefinido?}(n, \text{titulos}(b))\}$
HayPromMismoTipo	: promesa $p \times$ conj(promesa) $ps$	→ bool	
cumplirPromesas	: cliente $c \times$ conj(promesa) $ps \times$ nombre $\times$ cotizacion $\times$ nat $\times$ broker $b$	→ conj(promesa)	
dameMaximo	: nombre $\times$ conj(titulo)	→ nat	
accionesLibres	: titulo $t \times$ broker $b$	→ nat	
accionesTomadas	: nombre $\times$ conj(cliente) $\times$ broker	→ nat	
cumplirVentas	: conj(promesa) $\times$ nombre $\times$ cotizacion	→ conj(promesa)	
existeVenta	: promesa $\times$ nombre $\times$ cotizacion	→ bool	
accLibresDspVenta	: conj(cliente) $\times$ nombre $\times$ cotizacion $\times$ nat $\times$ broker	→ nat	
vendioAcciones	: conj(promesa) $\times$ nombre $\times$ cotizacion	→ bool	
nuevasAccLibres	: conj(promesa) $\times$ nombre $\times$ cotizacion	→ nat	
cumplirCompras	: cliente $\times$ conj(promesa) $\times$ titulo $\times$ cotizacion $\times$ conj(clientes) $\times$ acciones $\times$ broker	→ conj(promesas)	
accDspDeComprasDeOtros	: conj(promesa) $\times$ nombre $\times$ cotizacion $\times$ nat	→ nat	

<code>promDspDeVerSiCompro</code>	: $\text{conj}(\text{promesa}) \times \text{nombre} \times \text{cotizacion} \times \text{nat}$	$\longrightarrow \text{conj}(\text{promesa})$
<code>puedeVender?</code>	: $\text{cliente } c \times \text{promesa } p \times \text{broker } b$	$\longrightarrow \text{bool}$
<code>estaDefinido?</code>	: $\text{nombre} \times \text{conj}(\text{titulo})$	$\longrightarrow \text{bool}$
<code>seCumple?</code>	: $\text{promesa} \times \text{broker}$	$\longrightarrow \text{bool}$
<code>dameCotizacion</code>	: $\text{nombre} \times \text{conj}(\text{titulo})$	$\longrightarrow \text{nat}$
<code>tituloNuevaCot</code>	: $\text{nombre} \times \text{cotizacion} \times \text{conj}(\text{titulo})$	$\longrightarrow \text{conj}(\text{titulo})$
<code>accClienteDspVenta</code>	: $\text{nat} \times \text{conj}(\text{promesa}) \times \text{nombre} \times \text{cotizacion}$	$\longrightarrow \text{nat}$
<code>accDspDeCompras</code>	: $\text{cliente} \times \text{nombre} \times \text{conj}(\text{promesas}) \times \text{conj}(\text{cliente}) \times \text{nat} \times$ $\text{nat} \times \text{broker}$	$\longrightarrow \text{nat}$
<code>accClienteDspCompras</code>	: $\text{conj}(\text{promesa}) \times \text{nombre} \times \text{cotizacion} \times \text{nat} \times \text{nat}$	$\longrightarrow \text{nat}$

<b>axiomas</b>	$\forall b: \text{broker}, \forall c, c_1: \text{cliente}, \forall cs: \text{conj}(\text{cliente}), \forall ps: \text{conj}(\text{promesa}), \forall a, av, ct: \text{nat}, \forall ts: \text{conj}(\text{titulo}), \forall n: \text{string}, \forall t: \text{titulo}$
$\text{promesas}(c, \text{nuevo}(ts, cs))$	$\equiv \emptyset$
$\text{promesas}(c, \text{AgPromesa}(c_1, p, b))$	$\equiv \text{if seCumple?}(p, b) \text{ then } \text{promesas}(c, b) \text{ else } \text{if } c = c_1 \text{ then } \text{Ag}(p, \text{promesas}(c, b)) \text{ else } \text{promesas}(c, b) \text{ fi fi}$
$\text{promesas}(c, \text{cambiarCot}(n, ct, b))$	$\equiv \text{cumplirPromesas}(c, \text{promesas}(c, b), n, ct, \text{accionesLibres}(n, b), b)$
$\text{titulos}(\text{nuevo}(ts, cs))$	$\equiv ts$
$\text{titulos}(\text{AgTitulo}(t, b))$	$\equiv \text{Ag}(t, \text{titulos}(b))$
$\text{titulos}(\text{cambiarCot}(n, ct, b))$	$\equiv \text{tituloNuevaCot}(n, ct, \text{titulos}(b))$
$\text{clientes}(\text{nuevo}(ts, cs))$	$\equiv cs$
$\text{clientes}(\text{cambiarCot}(n, ct, b))$	$\equiv \text{clientes}(b)$
$\text{clientes}(\text{AgPromesa}(c, p, b))$	$\equiv \text{clientes}(b)$
$\text{clientes}(\text{AgTitulo}(t, b))$	$\equiv \text{clientes}(b)$
$\text{acciones}(c, t, \text{nuevo}(ts, cs))$	$\equiv 0$
$\text{acciones}(c, t, \text{CambiarCot}(n, ct, b))$	$\equiv \text{if } t = n \text{ then } \text{accDspDeCompras}(c, n, \text{promesas}(c, b), \text{clientes}(b), \text{accClienteDspVenta}(\text{acciones}(c, t, b), \text{promesas}(c, b), n, ct), \text{accLibresDspVenta}(\text{clientes}(b), n, ct, \text{accionesLibres}(n, b), b), b) \text{ else } \text{acciones}(c, t, b) \text{ fi}$
$\text{acciones}(c, n, \text{AgPromesa}(c_1, p, b))$	$\equiv \text{if } c = c_1 \wedge \text{seCumple?}(p, b) \text{ then } \text{if vende?}(p) \text{ then } \text{acciones}(c, n, b) - \text{cantidad}(p) \text{ else } \text{acciones}(c, n, b) + \text{cantidad}(p) \text{ fi else } \text{acciones}(c, n, b) \text{ fi}$
$\text{hayNombresRepetidos?}(ts)$	$\equiv \text{if vacio?}(ts) \text{ then } \text{false} \text{ else } \text{if estaDefinido}(\text{nombre}(\text{dameUno}(ts)), \text{sinUno}(ts)) \text{ then } \text{true} \text{ else } \text{hayNombresRepetidos}(\text{sinUno}(ts)) \text{ fi fi}$
$\text{maximoAcciones}(n, ts)$	$\equiv \text{if } \text{nombre}(\text{dameUno}(ts)) = n \text{ then } \text{maximo}(\text{dameUno}(ts)) \text{ else } \text{maximoAcciones}(n, \text{sinUno}(ts)) \text{ fi}$
$\text{puedeVender?}(c, p, b)$	$\equiv \text{cantidad}(p) \leq \text{acciones}(c, \text{nombre}(p), b)$

HayPromMismoTipo( $p, ps$ )	<pre> ≡ if vacio?(ps) then     false else     if vende?(dameUno(ps)) = vende?(p) ∧       nombreTitulo(dameUno(ps)) = nombreTitulo(p) then         true     else         HayPromMismoTipo(p, sinUno(ps))     fi fi </pre>
estaDefinido?( $n, ts$ )	<pre> ≡ if vacio?(ts) then     false else     if nombre(dameUno(ts)) = n then         true     else         estaDefinido?(n, sinUno(ts))     fi fi </pre>
seCumple?( $p, b$ )	<pre> ≡ if vende?(p) then     dameCotizacion(nombreTitulo(p), titulos(b)) &lt; limite(p) else     if accionesLibres(nombreTitulo(p), b) &gt; cantidad(p) then         dameCotizacion(nombreTitulo(p), titulos(b)) &gt; limite(p)     else         false     fi fi </pre>
dameCotizacion( $n, ts$ )	<pre> ≡ if nombre(dameUno(ts)) = n then     cotizacion(dameUno(ts)) else     dameCotizacion(n, sinUno(ts)) fi </pre>
accionesLibres( $n, b$ )	<pre> ≡ dameMaximo(n, titulos(b)) - accionesTomadas(n, clientes(b), b) </pre>
dameMaximo( $n, ts$ )	<pre> ≡ if nombre(dameUno(ts)) = n then     maximo(dameUno(ts)) else     dameMaximo(n, sinUno(ts)) fi </pre>
accionesTomadas( $n, cs, b$ )	<pre> ≡ if vacio?(cs) then     0 else     acciones(dameUno(cs), n, b) + accionesTomadas(n,     sinUno(cs), b) fi </pre>
cumplirPromesas( $c, ps, t, ct, a, b$ )	<pre> ≡ cumplirCompras(c, cumplirVentas(ps, t, ct), t, ct, clientes(b), accLibresDespVenta(clientes(b), t, ct, a, b), b) </pre>
cumplirVentas( $ps, n, ct$ )	<pre> ≡ if vacio?(ps) then     ∅ else     if existeVenta(dameUno(ps), n, ct) then         sinUno(ps)     else         Ag(dameUno(ps), cumplirVentas(sinUno(ps), n, ct))     fi fi </pre>
existeVenta( $p, n, ct$ )	<pre> ≡ vende?(p) ∧ nombreTitulo(p) = n ∧ ct &lt; limite(p) </pre>

```

accLibresDspVenta(cs, n, ct, a, b)    ≡ if vacio?(cs) then
    a
  else
    if vendioAcciones(promesas(dameUno(cs), b), n, ct) then
      nuevasAccLibres(promesas(dameUno(cs), b), n, ct) +
      accLibresDspVenta(sinUno(cs), n, ct, a, b)
    else
      accLibresDspVenta(p, sinUno(cs), n, c, a)
    fi
  fi

vendioAcciones(ps, n, ct)             ≡ if vacio(ps) then
    false
  else
    if vende?(dameUno(p)) ∧ nombreTitulo(dameUno(ps)) = n ∧
    limite(dameUno(ps)) < ct then
      true
    else
      vendioAcciones(sinUno(ps), n, ct)
    fi
  fi

nuevasAccLibres(ps, n, ct)            ≡ if vacio(ps) then
    0
  else
    if vende?(dameUno(ps)) ∧ nombreTitulo(dameUno(ps)) = n
    ∧
    limite(dameUno(ps)) < ct then
      cantidad(dameUno(ps))
    else
      nuevasAccLibres(sinUno(p), n, ct)
    fi
  fi

cumplirCompras(c, ps, t, ct, cs, a, b) ≡ if dameUno(cs) = c then
    promDspDeVerSiCompro(ps, t, ct, a)
  else
    cumplirCompras(c, ps, t, ct, sinUno(cs),
    accDspDeComprasDeOtros(promesas(dameUno(cs), b), t, ct,
    a), b)
  fi

accDspDeComprasDeOtros(ps, n, ct, a) ≡ if vacio?(ps) then
    a
  else
    if ¬vende?(dameUno(ps)) ∧ nombreTitulo(dameUno(ps)) = n
    ∧ limite(dameUno(ps)) < ct then
      a - cantidad(p)
    else
      accDspDeComprasDeOtros(sinUno(p), n, ct, a)
    fi
  fi

promDspDeVerSiCompro(ps, n, ct, a)    ≡ if vacio?(ps) then
    ∅
  else
    if ¬vende?(dameUno(ps)) ∧ nombreTitulo(dameUno(ps)) =
    n ∧ limite(dameUno(ps)) < ct ∧ a ≥ cantidad(dameUno(ps))
    then
      sinUno(ps)
    else
      Ag(dameUno(ps), promDspDeVerSiCompro(sinUno(ps), n,
      ct, a))
    fi
  fi

```



```

tituloNuevaCot(n, ct, ts)           ≡ if nombre(dameUno(ts)) = n then
                                     Ag(cambiarValor(ct,dameUno(ts)),sinUno(ts))
                                     else
                                     Ag(dameUno(ts),tituloNuevaCot(n,ct,sinUno(ts))
                                     fi

accClienteDspVenta(a, ps, n, ct)    ≡ if vacio?(ps) then
                                     a
                                     else
                                     if existeVenta(dameUno(ps), t, ct) then
                                     a - cantidad(dameUno(ps))
                                     else
                                     accClienteDspVenta(a, sinUno(ps), n, ct)
                                     fi
                                     fi

accDspDeCompras(c, n, ps, cs, a, av, b) ≡ if dameUno(cs) = c then
                                     accClienteDspCompras(ps, t, ct, a, av)
                                     else
                                     accDspDeCompras(c,      n,      ps,      sinUno(cs),      ct,
                                     accDspDeComprasDeOtros(promesas(dameUno(cs),  b),  t,
                                     ct, av), b)
                                     fi

accClienteDspCompras(ps, n, ct, a, av) ≡ if vacio(ps) then
                                     a
                                     else
                                     if ¬vende?(dameUno(ps)) ∧ nombreTitulo(dameUno(ps)) =
                                     n ∧ limite(dameUno(ps)) < c ∧ av ≤ cantidad(dameUno(ps))
                                     then
                                     a + cantidad(dameUno(ps))
                                     else
                                     promDspDeVerSiCompro(sinUno(ps), n, ct, a, av)
                                     fi
                                     fi

```

**Fin TAD**

## TAD PROMESA

$$(\forall p, p' : \text{promesa}) \left( p =_{\text{obs}} p' \iff \begin{pmatrix} \text{nombreTitulo}(p) =_{\text{obs}} \text{nombreTitulo}(p') \wedge \text{vende?}(p) =_{\text{obs}} \text{vende?}(p') \\ \text{cantidad}(p) =_{\text{obs}} \text{cantidad}(p') \wedge \\ \text{limite}(p) =_{\text{obs}} \text{limite}(p') \end{pmatrix} \right)$$
**Fin TAD**

