

Algoritmos y Estructuras de Datos II

Primer Cuatrimestre de 2016

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

Trabajo Practico 1

Especificacion

Integrante	LU	Correo electrónico
Delmagro Agustin	596/14	agustin.delmagro@gmail.com
Lopez Valiente Patricio	457/15	patricio454@gmail.com
Vercinsky Ivan	141/15	ivan9074@gmail.com
Zar Abad Ciro	128/15	ciromanzar@gmail.com

Reservado para la cátedra

Instancia	Docente	Nota
Primera entrega		
Segunda entrega		

Índice

1. Renombres de TADs	3
2. Operaciones Adicionales de TIPOS BASICOS	3
3. TAD Campo	4
4. TAD Registro	5
5. TAD Tabla	7
6. TAD Base de Datos	9

1. Renombres de TADs

TAD NOMBRE es *String*

TAD TYPE es $\{N, S\}$ (*Funciona como Bool*)

TAD JOIN es $TUPLA(Conjunto(Nombre), Campo)$

2. Operaciones Adicionales de TIPOS BASICOS

otras operaciones

$\bullet = \bullet$: $\text{conj}(\alpha) \times \text{conj}(\alpha) \longrightarrow \text{bool}$

$\bullet = \bullet$: $\text{multiconj}(\alpha) \times \text{multiconj}(\alpha) \longrightarrow \text{bool}$

$\bullet = \bullet$: $\text{tupla}(\alpha \times \beta) \times \text{tupla}(\alpha \times \beta) \longrightarrow \text{bool}$

multiIncluido? : $\text{multiconj}(\alpha) \times \text{multiconj}(\alpha) \longrightarrow \text{bool}$

quitarN : $\alpha \times \text{nat} \times \text{multiconj}(\alpha) \longrightarrow \text{multiconj}(\alpha)$

axiomas $\forall c_1, c_2: \text{conj}(\alpha) \forall mc_1, mc_2: \text{multiconj}(\alpha) \forall ta_1, ta_2: \text{tupla}(\alpha, \beta) \forall a: \alpha \forall n: \text{nat}$

$c_1 = c_2$ \equiv **if** $(\#(c_1) \equiv \#(c_2))$ **then**
 if $(\emptyset?(c_1))$ **then**
 $true$
 else
 $c_1 \subseteq c_2$
 fi
else
 $false$
fi

$mc_1 = mc_2$ \equiv **if** $(\#(mc_1) \equiv \#(mc_2))$ **then**
 $\text{multiIncluido?}(mc_1, mc_2)$
else
 $false$
fi

$ta_1 = ta_2$ \equiv $(\pi_1(ta_1) = \pi_1(ta_2)) \wedge (\pi_2(ta_1) = \pi_2(ta_2))$

$\text{multiIncluido?}(mc_1, mc_2)$ \equiv **if** $(\emptyset?(mc_1))$ **then**
 $true$
else
 if $(\#(\text{dameUno}(mc_1), mc_1) \equiv \#(\text{dameUno}(mc_1), mc_2))$ **then**
 $\text{multiIncluido?}(\text{quitarN}(\text{dameUno}(mc_1), \#(\text{dameUno}(mc_1), mc_1), mc_1), mc_2)$
 else
 $false$
 fi

$\text{quitarN}(a, n, mc_1)$ \equiv **if** $(n \neq 0)$ **then**
 $\text{quitarN}(a, (n - 1), (mc_1 - \{a\}))$
else
 mc_1
fi

3. TAD Campo

TAD CAMPO

géneros campo

exporta CrearCampo, Nombre, Tipo, =

usa TYPE, NOMBRE, BOOL

igualdad observacional

$(\forall c_1, c_2 : \text{campo}) \ (c_1 =_{\text{obs}} c_2 \iff (\text{Nombre}(c_1) =_{\text{obs}} \text{Nombre}(c_2) \wedge \text{Tipo}(c_1) =_{\text{obs}} \text{Tipo}(c_2)))$

generadores

CrearCampo : nombre $n \times$ type \longrightarrow campo

$\{n \neq \emptyset\}$

observadores básicos

Nombre: campo \longrightarrow nombre

Tipo : campo \longrightarrow type

otras operaciones

$\bullet = \bullet : \text{campo} \times \text{campo} \longrightarrow \text{bool}$

axiomas $\forall n: \text{nombre} \ \forall t: \text{type}$

$\text{Nombre}(\text{CrearCampo}(n, t)) \equiv n$

$\text{Tipo}(\text{CrearCampo}(n, t)) \equiv t$

$c_1 = c_2 \equiv (\text{Nombre}(c_1) \equiv \text{Nombre}(c_2)) \wedge (\text{Tipo}(c_1) \equiv \text{Tipo}(c_2))$

Fin TAD

4. TAD Registro

TAD REGISTRO

géneros registro

exporta CrearRegistro, AgregarCDNat, AgregarCDString, Campos, DatoNat, DatoString, agregarDatos, =

usa CAMPO, CONJUNTO(α), NAT, STRING, BOOL

igualdad observacional

$$(\forall r_1, r_2 : \text{registro}) \left(r_1 =_{\text{obs}} r_2 \iff \left(\begin{array}{l} \text{Campos}(r_1) =_{\text{obs}} \text{Campos}(r_2) \wedge_{\text{L}} \\ (\forall c : \text{campo}, c \in \text{Campos}(r_1)) \\ (\text{Tipo}(c) =_{\text{obs}} N \wedge_{\text{L}} \text{DatoNat}(r_1, c) =_{\text{obs}} \text{DatoNat}(r_2, c)) \\ \vee \\ (\text{Tipo}(c) =_{\text{obs}} S \wedge_{\text{L}} \text{DatoString}(r_1, c) =_{\text{obs}} \text{DatoString}(r_2, c)) \end{array} \right) \right)$$

generadores

CrearRegistro : \longrightarrow registro

AgregarCDNat : registro $r \times$ campo $c \times$ nat \longrightarrow registro

AgregarCDString : registro $r \times$ campo $c \times$ string \longrightarrow registro

$\{\text{Tipo}(c) \equiv N \wedge c \notin \text{Campos}(r)\}$

$\{\text{Tipo}(c) \equiv S \wedge c \notin \text{Campos}(r)\}$

observadores básicos

Campos : registro \longrightarrow conj(campo)

DatoNat : registro $r \times$ campo $c \longrightarrow$ nat

DatoString : registro $r \times$ campo $c \longrightarrow$ string

$\{c \in \text{Campos}(r) \wedge \text{Tipo}(c) \equiv N\}$

$\{c \in \text{Campos}(r) \wedge \text{Tipo}(c) \equiv S\}$

otras operaciones

agregarDatos : registro $r_1 \times$ conj(campo) $cc \times$ registro $r_2 \longrightarrow$ registro

$\bullet = \bullet$: registro \times registro \longrightarrow bool

IgualDatos : registro \times registro \times conj(campo) \longrightarrow bool

$\{cc \subseteq \text{campos}(r_2)\}$

axiomas $\forall r : \text{registro} \forall cc : \text{conj}(\text{campo}) \forall c_1, c_2 : \text{campo} \forall n : \text{nat} \forall s : \text{string}$

Campos(CrearRegistro) $\equiv \emptyset$

Campos(AgregarCDNat(r, c_1, n)) $\equiv \text{Ag}(c_1, \text{Campos}(r))$

Campos(AgregarCDString(r, c_1, s)) $\equiv \text{Ag}(c_1, \text{Campos}(r))$

DatoNat(AgregarCDNat(r, c_1, n), c_2) \equiv if ($c_1 = c_2$) then n else DatoNat(r, c_2) fi

DatoNat(AgregarCDString(r, c_1, s), c_2) \equiv DatoNat(r, c_2)

DatoString(AgregarCDString(r, c_1, s), c_2) \equiv if ($c_1 = c_2$) then s else DatoString(r, c_2) fi

DatoString(AgregarCDNat(r, c_1, n), c_2) \equiv DatoString(r, c_2)

agregarDatos(r_1, cc, r_2) \equiv if ($\emptyset?(cc)$) then

r_1

else

if ($\text{dameUno}(cc) \in \text{Campos}(r_1)$) then

agregarDatos($r_1, \text{sinUno}(cc), r_2$)

else

if ($\text{Tipo}(\text{dameUno}(cc)) \equiv N$) then

AgregarCDNat($\text{agregarDatos}(r_1, \text{sinUno}(cc), r_2)$,

$\text{dameUno}(cc), \text{DatoNat}(r_2, \text{dameUno}(cc))$)

else

AgregarCDString($\text{agregarDatos}(r_1, \text{sinUno}(cc), r_2)$,

$\text{dameUno}(cc), \text{DatoString}(r_2, \text{dameUno}(cc))$)

fi

fi

$r_1 = r_2 \equiv$ if ($\text{Campos}(r_1) = \text{Campos}(r_2)$) then

IgualDatos($r_1, r_2, \text{Campos}(r_1)$)

else

false

fi

IgualDatos(r_1, r_2, cc)

```
≡ if (∅?(cc)) then
    true
  else
    if (Tipo(dameUno(cc)) ≡ N) then
      if (DatoNat( $r_1$ , dameUno(cc)) ≡
DatoNat( $r_2$ , dameUno(cc))) then
        IgualDatos( $r_1, r_2$ , sinUno(cc))
      else
        false
    fi
  else
    if (DatoString( $r_1$ , dameUno(cc)) ≡
DatoString( $r_2$ , dameUno(cc))) then
      IgualDatos( $r_1, r_2$ , sinUno(cc))
    else
      false
    fi
  fi
fi
```

Fin TAD

$\text{AuxColumnaNat}(cr, c)$	\equiv	if $(\emptyset?(cr))$ <i>then</i> \emptyset else $\text{Ag}(\text{DatoNat}(\text{dameUno}(cr), c),$ $\text{AuxColumnaNat}(\text{sinUno}(cr), c))$
$\text{AuxColumnaString}(cr, c)$	\equiv	fi if $(\emptyset?(cr))$ <i>then</i> \emptyset else $\text{Ag}(\text{DatoString}(\text{dameUno}(cr), c),$ $\text{AuxColumnaString}(\text{sinUno}(cr), c))$
$\text{NoRepiteDatosClave}(r, t)$	\equiv	$\text{AuxNoRepite}(\text{Claves}(t), r, \text{Registros}(t))$
$\text{AuxNoRepite}(cc, r, cr)$	\equiv	if $(\emptyset?(cc))$ <i>then</i> true else if $(\text{Tipo}(\text{dameUno}(cc)) \equiv N)$ <i>then</i> if $\neg(\text{DatoNat}(r, \text{dameUno}(cc)) \in$ $\text{AuxColumnaNat}(cr, \text{dameUno}(cc)))$ <i>then</i> $\text{AuxNoRepite}(\text{sinUno}(cc), r, cr)$ else false fi else if $\neg(\text{DatoString}(r, \text{dameUno}(cc)) \in$ $\text{AuxColumnaString}(cr, \text{dameUno}(cc)))$ <i>then</i> $\text{AuxNoRepite}(\text{sinUno}(cc), r, cr)$ else false fi fi
$t_1 = t_2$	\equiv	fi $(\text{Nombre}(t_1) \equiv \text{Nombre}(t_2)) \wedge$ $(\text{Campos}(t_1) = \text{Campos}(t_2)) \wedge$ $(\text{Claves}(t_1) = \text{Claves}(t_2)) \wedge$ $(\text{Registros}(t_1) = \text{Registros}(t_2))$

Fin TAD

6. TAD Base de Datos

TAD BASE DE DATOS

géneros bdd

exporta CrearBase, AgTabla, AgRegistro, BorrarRegistrosNat, BorrarRegistrosString, CrearJoin, CrearTrigger, BorrarJoin, BorrarTrigger, Tablas, Joins, Triggers, # Usos, VisualizarJoin, TablaMasUsada

usa TABLA, REGISTRO, CAMPO, JOIN, CONJUNTO(α), NOMBRE, NAT, STRING

observadores básicos

Tablas : bdd \longrightarrow conj(tabla)
 Joins : bdd \longrightarrow conj(join)
 Triggers : bdd $b \times$ nombre $nt \longrightarrow$ conj(nombre) $\{nt \in DameNombres(Tablas(b))\}$
 # Usos : bdd $b \times$ tabla $t \longrightarrow$ nat $\{t \in Tablas(b)\}$

igualdad observacional

$$(\forall b_1, b_2 : \text{bdd}) \left(b_1 =_{\text{obs}} b_2 \iff \left(\begin{array}{l} Tablas(b_1) =_{\text{obs}} Tablas(b_2) \wedge \\ Joins(b_1) =_{\text{obs}} Joins(b_2) \wedge_L \\ (\forall t : \text{tabla}, t \in Tablas(b_1)) \\ Triggers(b_1, Nombre(t)) =_{\text{obs}} Triggers(b_2, Nombre(t)) \wedge \\ \#Usos(b_1, t) =_{\text{obs}} \#Usos(b_2, t) \end{array} \right) \right)$$

generadores

CrearBase : \longrightarrow bdd
 AgTabla : bdd $b \times$ tabla $t \longrightarrow$ bdd
 AgRegistro : bdd $b \times$ tabla $t \times$ registro $r \longrightarrow$ bdd $\{Nombre(t) \notin DameNombres(Tablas(b))\}$
 BorrarRegistrosNat : bdd $b \times$ tabla $t \times$ campo $c \times$ nat $d \longrightarrow$ bdd $\{Campos(r) = Campos(t) \wedge_L NoRepiteDatosClaves(r, t)\}$
 BorrarRegistrosString : bdd $b \times$ tabla $t \times$ campo $c \times$ string $d \longrightarrow$ bdd $\{c \in Campos(t) \wedge Tipo(c) \equiv N\}$
 CrearJoin : bdd $b \times$ join $j \longrightarrow$ bdd $\{c \in Campos(t) \wedge Tipo(c) \equiv S\}$
 CrearTrigger : bdd $b \times$ nombre $nt_1 \times$ nombre $nt_2 \longrightarrow$ bdd $\{EsJoinValido(b, j)\}$
 BorrarJoin : bdd $b \times$ join $j \longrightarrow$ bdd $\{nt_1 \in DameNombres(Tablas(b)) \wedge nt_2 \in DameNombres(Tablas(b)) \wedge \neg(nt_1 = nt_2) \wedge \{Claves(DameTabla(nt_2)) \subseteq Claves(DameTabla(nt_1)) \wedge_L nt_2 \notin Triggers(b, nt_1)\}\}$
 BorrarTrigger : bdd $b \times$ nombre $nt_1 \times$ nombre $nt_2 \longrightarrow$ bdd $\{j \in Joins(b)\}$
 BorrarTrigger : bdd $b \times$ nombre $nt_1 \times$ nombre $nt_2 \longrightarrow$ bdd $\{nt_2 \in Triggers(b, nt_1)\}$

otras operaciones

EsJoinValido : bdd \times join \longrightarrow bool
 Actualizar : bdd \times conj(nombre) \times registro \longrightarrow conj(tabla)
 CompletarRegistro : conj(campo) \times registro \longrightarrow registro
 NomxTablas : conj(tabla) $ct \times$ conj(nombre) $cn \longrightarrow$ conj(tabla) $\{(\forall nt : \text{nombre}, nt \in cn), nt \in DameNombres(ct)\}$
 DameTabla : conj(tabla) $ct \times$ nombre $nt \longrightarrow$ tabla $\{nt \in DameNombres(ct)\}$
 DameNombres : conj(tabla) \longrightarrow conj(nombre)
 VisualizarJoin : bdd $b \times$ join $j \longrightarrow$ conj(registro) $\{j \in Joins(b)\}$
 ArmarJoin : conj(registro) \times conj(registro) \times campo \longrightarrow conj(registro)
 UnirRegCoinc : registro $r \times$ conj(registro) $cr \times$ campo $c \longrightarrow$ registro $\{(c \in Campos(r)) \wedge ((\forall r_1 : \text{registro}, r_1 \in cr) c \in Campos(r_1))\}$
 CoincideAlgunDato? : registro $r \times$ conj(registro) $cr \times$ campo $c \longrightarrow$ bool $\{(c \in Campos(r)) \wedge ((\forall r_1 : \text{registro}, r_1 \in cr) c \in Campos(r_1))\}$
 TablaMasUsada : bdd $b \longrightarrow$ tabla $\{-\emptyset?(Tablas(b))\}$
 MayorUso : conj(tabla) \times nat \longrightarrow nat
 TablasConUso : conj(tabla) \times nat \longrightarrow conj(tabla)

axiomas $\forall b : \text{bdd} \forall ct : \text{conj}(\text{tabla}) \forall cn : \text{conj}(\text{nombre}) \forall cr, cr_1, cr_2 : \text{conj}(\text{registro}) \forall cc : \text{conj}(\text{campo}) \forall t, t_1 : \text{tabla}$
 $\forall j : \text{join} \forall r : \text{registro} \forall c : \text{campo} \forall nt, nt_1, nt_2 : \text{nombre} \forall s : \text{string} \forall i, m, u : \text{nat}$

Tablas(CrearBase()) $\equiv \emptyset$
 Tablas(AgTabla(b, t)) $\equiv Ag(t, Tablas(b))$
 Tablas(AgRegistro(b, t, r)) $\equiv Ag(AgregarRegistro(t, r), ((Tablas(b) - NomxTablas(Tablas(b), Triggers(b, Nombre(t)))) \cup Actualizar(b, Triggers(b, Nombre(t)), r)) - \{t\})$

$\text{Tablas}(\text{BorrarRegistrosNat}(b, t, c, i))$	$\equiv \text{Ag}(\text{BorrarRegistrosNat}(t, c, i), (\text{Tablas}(b) - \{t\}))$
$\text{Tablas}(\text{BorrarRegistrosString}(b, t, c, s))$	$\equiv \text{Ag}(\text{BorrarRegistrosString}(t, c, s), (\text{Tablas}(b) - \{t\}))$
$\text{Tablas}(\text{CrearJoin}(b, j))$	$\equiv \text{Tablas}(b)$
$\text{Tablas}(\text{BorrarJoin}(b, j))$	$\equiv \text{Tablas}(b)$
$\text{Tablas}(\text{CrearTrigger}(b, nt_1, nt_2))$	$\equiv \text{Tablas}(b)$
$\text{Tablas}(\text{BorrarTrigger}(b, nt_1, nt_2))$	$\equiv \text{Tablas}(b)$
$\text{Joins}(\text{CrearBase}())$	$\equiv \emptyset$
$\text{Joins}(\text{AgTabla}(b, t))$	$\equiv \text{Joins}(b)$
$\text{Joins}(\text{AgRegistro}(b, t, r))$	$\equiv \text{Joins}(b)$
$\text{Joins}(\text{BorrarRegistroNat}(b, t, c, i))$	$\equiv \text{Joins}(b)$
$\text{Joins}(\text{BorrarRegistroString}(b, t, c, s))$	$\equiv \text{Joins}(b)$
$\text{Joins}(\text{CrearJoin}(b, j))$	$\equiv \text{Ag}(j, \text{Joins}(b))$
$\text{Joins}(\text{BorrarJoin}(b, j))$	$\equiv \text{Joins}(b) - \{j\}$
$\text{Joins}(\text{CrearTrigger}(b, nt, nt_2))$	$\equiv \text{Joins}(b)$
$\text{Joins}(\text{BorrarTrigger}(b, nt, nt_2))$	$\equiv \text{Joins}(b)$
$\text{Triggers}(\text{AgTabla}(b, t), nt)$	$\equiv \text{if } (nt \equiv \text{Nombre}(t)) \text{ then } \emptyset \text{ else } \text{Triggers}(b, nt) \text{ fi}$
$\text{Triggers}(\text{AgRegistro}(b, t, r), nt)$	$\equiv \text{Triggers}(b)$
$\text{Triggers}(\text{BorrarRegistrosNat}(b, t, c, i), nt)$	$\equiv \text{Triggers}(b, nt)$
$\text{Triggers}(\text{BorrarRegistrosString}(b, t, c, s), nt)$	$\equiv \text{Triggers}(b, nt)$
$\text{Triggers}(\text{CrearJoin}(b, j))$	$\equiv \text{Triggers}(b, nt)$
$\text{Triggers}(\text{BorrarJoin}(b, j))$	$\equiv \text{Triggers}(b, nt)$
$\text{Triggers}(\text{CrearTrigger}(b, nt_1, nt_2), nt)$	$\equiv \text{if } (nt \equiv nt_1) \text{ then } \text{Ag}(nt_2, \text{Triggers}(b, nt)) \text{ else } \text{Triggers}(b, nt) \text{ fi}$
$\text{Triggers}(\text{BorrarTrigger}(b, nt_1, nt_2), nt)$	$\equiv \text{if } (nt \equiv nt_1) \text{ then}$ $\quad \text{Triggers}(b, nt_1) - \{nt_2\}$ else $\quad \text{Triggers}(b, nt)$ fi
$\# \text{Usos}(\text{AgTabla}(b, t_1), t)$	$\equiv \text{if } (t = t_1) \text{ then } 0 \text{ else } \# \text{Usos}(b, t) \text{ fi}$
$\# \text{Usos}(\text{AgRegistro}(b, t_1, r), t)$	$\equiv \text{if } (t = t_1) \text{ then}$ $\quad 1 + \# \text{Usos}(b, t)$ else $\quad \beta(\text{Nombre}(t) \in \text{Triggers}(b, t_1)) + \# \text{Usos}(b, t)$ fi
$\# \text{Usos}(\text{BorrarRegistrosNat}(b, t_1, c, i), t)$	$\equiv \text{if } (t = t_1) \text{ then}$ $\quad \# \text{Usos}(b, t) + \#(i, \text{AuxColumnaNat}(\text{Registros}(t), c))$ else $\quad \# \text{Usos}(b, t)$ fi
$\# \text{Usos}(\text{BorrarRegistrosString}(b, t, c, s), t)$	$\equiv \text{if } (t = t_1) \text{ then}$ $\quad \# \text{Usos}(b, t) + \#(d, \text{AuxColumnaString}(\text{Registros}(t), c))$ else $\quad \# \text{Usos}(b, t)$ fi
$\# \text{Usos}(\text{CrearJoin}(b, j), t)$	$\equiv \# \text{Usos}(b, t)$
$\# \text{Usos}(\text{BorrarJoin}(b, j), t)$	$\equiv \# \text{Usos}(b, t)$
$\# \text{Usos}(\text{CrearTrigger}(b, nt_1, nt_2), t)$	$\equiv \# \text{Usos}(b, t)$
$\# \text{Usos}(\text{BorrarTrigger}(b, nt_1, nt_2), t)$	$\equiv \# \text{Usos}(b, t)$
$\text{EsJoinValido}(b, j)$	$\equiv (\# \pi_1(j) \equiv 2) \wedge$ $(\pi_1 \subseteq \text{DameNombres}(\text{Tablas}(b))) \wedge$ $(j \notin \text{Joins}(b)) \wedge_L$ $(\pi_2(j) \in \text{Claves}(\text{DameTabla}(\text{dameUno}(\pi_1(j)))) \wedge$ $(\pi_2(j) \in \text{Claves}(\text{DameTabla}(\text{dameUno}(\text{SinUno}(\pi_1(j)))))$

Actualizar(cn, r)	\equiv if ($\emptyset?(cn)$) <i>then</i> \emptyset else if ($NoRepiteDatosClaves(r, DameTabla(Tablas(b), dameUno(cn)))$) <i>then</i> $Ag(AgregarRegistro(DameTabla(Tablas(b), dameUno(cn)),$ $CompletarRegistro(Campos(DameTabla(Tablas(b), dameUno(cn))), r),$ $Actualizar(sinUno(cn), r))$ else $Actualizar(sinUno(cn), r)$ fi
CompletarRegistro(cc, r)	\equiv if ($\emptyset?(cc)$) <i>then</i> $CrearRegistro()$ else if ($dameUno(cc) \in Campos(r)$) <i>then</i> if ($Tipo(dameUno(cc)) \equiv N$) <i>then</i> $AgregarCDNat(CompletarRegistro(sinUno(cc), r),$ $dameUno(cc), DatoNat(r, dameUno(r, dameUno(cc)))$ else $AgregarCDString(CompletarRegistro(sinUno(cc), r),$ $dameUno(cc), DatoString(r, dameUno(r, dameUno(cc)))$ fi else if ($Tipo(dameUno(cc)) \equiv N$) <i>then</i> $AgregarCDNat(CompletarRegistro(sinUno(cc), r),$ $dameUno(cc), 0)$ else $AgregarCDString(CompletarRegistro(sinUno(cc), r),$ $dameUno(cc), "NN")$ fi fi
NomxTablas(ct, cn)	\equiv if ($\emptyset?(cn)$) <i>then</i> \emptyset else $Ag(DameTabla(ct, dameUno(cn)), NomxTablas(ct, sinUno(cn)))$ fi
DameTabla(ct, n)	\equiv if ($Nombre(dameUno(ct)) \equiv n$) <i>then</i> $dameUno(ct)$ else $DameTabla(sinUno(ct), n)$ fi
DameNombres(ct)	\equiv if ($\emptyset?(ct)$) <i>then</i> \emptyset else $Ag(Nombre(dameUno(ct)), DameNombres(sinUno(ct)))$ fi
VisualizarJoin(b, j)	\equiv $ArmarJoin(Registros(DameTabla(Tablas(b), dameUno(\pi_1(j))))$ $, Registros(DameTabla(Tablas(b), dameUno(sinUno(\pi_1(j))))$ $, \pi_2(j))$

ArmarJoin(cr_1 , cr_2 , c)	\equiv if ($\emptyset?(cr_1)$) <i>then</i> \emptyset else if (<i>CoincideAlgunDato?</i> (<i>dameUno</i> (cr_1), cr_2 , c)) <i>then</i> $Ag(UnirRegCoinc(dameUno(cr_1), cr_2, c),$ $ArmarJoin(sinUno(cr_1), cr_2, c))$ else $ArmarJoin(sinUno(cr_1), cr_2, c)$ fi
UnirRegCoinc(r , cr , c)	\equiv fi if (<i>Tipo</i> (c) $\equiv N$) <i>then</i> if (<i>DatoNat</i> (r , c) $\equiv DatoNat(dameUno(cr), c)$) <i>then</i> $agregarDatos(r, Campos(dameUno(cr)), dameUno(cr))$ else $UnirRegCoinc(r, sinUno(cr), c)$ fi else if (<i>DatoString</i> (r , c) $\equiv DatoString(dameUno(cr), c)$) <i>then</i> $agregarDatos(r, Campos(dameUno(cr)), dameUno(cr))$ else $UnirRegCoinc(r, sinUno(cr), c)$ fi
CoincideAlgunDato?(r , cr , c)	\equiv fi if (<i>Tipo</i> (c) $\equiv N$) <i>then</i> $DatoNat(r, c) \in AuxColumnaNat(cr, c)$ else $DatoString(r, c) \in AuxColumnaString(cr, c)$ fi
TablaMasUsada(b)	\equiv fi $dameUno(TablasConUso(Tablas(b),$ $MayorUso(Tablas(b), \#Uso(dameUno(Tablas(b))))$ fi
TablasConUso(ct , u)	\equiv if ($\emptyset?(ct)$) <i>then</i> \emptyset else if ($\#Uso(dameUno(ct)) \equiv u$) <i>then</i> $Ag(dameUno(ct), TablasConUso(sinUno(ct), u))$ else $TablasConUso(sinUno(ct), u)$ fi
MayorUso(ct , m)	\equiv fi if ($\emptyset?(ct)$) <i>then</i> m else $MayorUso(sinUno(ct), max(\#Uso(dameUno(ct)), m))$ fi

Fin TAD