

# Algoritmos y Estructuras de Datos II

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

## Trabajo Práctico 1 “Algebra Relacional”

Integrante	LU	Correo electrónico
Luis Ricardo Bustamante	43/18	luisbustamante097@gmail.com
Leandro Emanuel Rodriguez	521/17	leandro21890000@gmail.com
Ramiro Augusto Ciruzzi	228/17	ramiro.ciruzzi@gmail.com
David Alejandro Venegas	783/18	venegasr.david@gmail.com

Reservado para la cátedra

Instancia	Docente	Nota
Primera entrega		
Segunda entrega		

# 1. Tad's Utilizados

## TAD BASEDEDATOS

**géneros**      base\_de\_datos

**usa**            REGISTRO, TABLA, CONSULTA, BOOL, NAT, CONJUNTO, NOMBRECAMPO, NOMBRETABLA, VALOR, TIPOCONSULTA

**exporta**      generadores, observadores, agregarRegistro, sacarRegistro, sacarTabla, ejecutarConsulta

**igualdad observacional**

$$(\forall bd1, bd2 : \text{base\_de\_datos}) \left( bd1 =_{\text{obs}} bd2 \iff \left( \begin{array}{l} \text{tablas}(bd1) =_{\text{obs}} \text{tablas}(bd2) \wedge_L (\forall t : \text{tabla}) (t \in \text{tablas}(bd1) \wedge t \in \text{tablas}(bd2) \Rightarrow_L \text{nombreTabla}(bd1, t) = \text{nombreTabla}(bd2, t)) \end{array} \right) \right)$$

### generadores

nuevaBD :  $\longrightarrow$  base\_de\_datos

agregarTabla : base\_de\_datos  $\times$  tabla  $\times$  nombre\_tabla  $\longrightarrow$  base\_de\_datos

### observadores básicos

tablas : base\_de\_datos  $\longrightarrow$  conj(tabla)

nombreTabla : base\_de\_datos  $bd \times$  tabla  $t \longrightarrow$  nombre\_tabla  $\{t \in \text{tablas}(bd)\}$

### otras operaciones

pertenece : base\_de\_datos  $\times$  nombre\_tabla  $\longrightarrow$  bool

reemplazarTabla : base\_de\_datos  $\times$  nombre\_tabla  $\times$  tabla  $\longrightarrow$  base\_de\_datos

tablaCorrespondiente : base\_de\_datos  $bd \times$  nombre\_tabla  $nt \longrightarrow$  tabla  $\{\text{pertenece}(bd, nt)\}$

sacarTabla : base\_de\_datos  $bd \times$  nombre\_tabla  $nt \longrightarrow$  base\_de\_datos  $\{\text{pertenece}(bd, nt)\}$

agregarRegistro : base\_de\_datos  $bd \times$  nombre\_tabla  $nt \times$  registro  $\longrightarrow$  base\_de\_datos  $\{\text{pertenece}(bd, nt)\}$

sacarRegistro : base\_de\_datos  $bd \times$  nombre\_tabla  $nt \times$  valor  $\longrightarrow$  base\_de\_datos  $\{\text{pertenece}(bd, nt)\}$

consultaFrom : base\_de\_datos  $\times$  consulta  $q \longrightarrow$  conj(registro)  $\{\text{tipo\_consulta}(q) \in \{\text{FROM}\}\}$

consultaSelect : base\_de\_datos  $\times$  consulta  $q \longrightarrow$  conj(registro)  $\{\text{tipo\_consulta}(q) \in \{\text{SELECT}\}\}$

filtrarTabla : conj(registro)  $cr \times$  nombre\_campo  $nc \longrightarrow$  conj(registro)  $\{(\forall r : \text{registro})(r \in cr \Rightarrow_L nc \in \text{campos}(r))\}$

consultaMatch : base\_de\_datos  $\times$  consulta  $q \longrightarrow$  conj(registro)  $\{\text{tipo\_consulta}(q) \in \{\text{MATCH}\}\}$

encontrarRepetidos : conj(registro)  $cr \times$  nombre\_campo  $c1 \times$  nombre\_campo  $c2 \longrightarrow$  conj(registro)  $\{(\forall r : \text{registro})(r \in cr \Rightarrow_L c1 \in \text{campos}(r) \wedge c2 \in \text{campos}(r))\}$

consultaProj : base\_de\_datos  $\times$  consulta  $q \longrightarrow$  conj(registro)  $\{\text{tipo\_consulta}(q) \in \{\text{PROJ}\}\}$

filtroProyeccion : conj(registro)  $\times$  conj(nombre\_campo)  $\longrightarrow$  conj(registro)

regFiltradoPorCampos : registro  $r \times$  conj(nombre\_campo)  $\longrightarrow$  registro

consultaRename : base\_de\_datos  $\times$  consulta  $q \longrightarrow$  conj(registro)  $\{\text{tipo\_consulta}(q) \in \{\text{RENAME}\}\}$

renombrar : conj(registro)  $nc \times$  nombre\_campo  $nc1 \times$  nombre\_campo  $nc2 \longrightarrow$  registro

renombrarRegistro : registro  $r \times$  conj(nombre\_campo)  $cnc \times$  nombre\_campo  $c1 \times$  nombre\_campo  $c2 \longrightarrow$  registro  $\{\text{campos}(r) = cnc \wedge ((c1 \in cnc) \wedge \neg(c2 \in cnc))\}$

consultaInter : base\_de\_datos  $\times$  consulta  $q \longrightarrow$  conj(registro)  $\{\text{tipo\_consulta}(q) \in \{\text{INTER}\}\}$

union : conj(registro)  $\times$  conj(registro)  $\longrightarrow$  conj(registro)

consultaUnion : base\_de\_datos  $\times$  consulta  $q \longrightarrow$  conj(registro)  $\{\text{tipo\_consulta}(q) \in \{\text{UNION}\}\}$

inter : conj(registro)  $\times$  conj(registro)  $\longrightarrow$  conj(registro)

consultaProduct	: base_de_datos $\times$ consulta $q$	$\longrightarrow$ conj(registro) $\{\text{tipo\_consulta}(q) \in \{\text{PRODUCT}\}\}$
prodCart	: conj(registro) $\times$ conj(registro)	$\longrightarrow$ conj(registro)
auxProdCart	: conj(registro) $\times$ conj(registro) $\times$ conj(registro)	$\longrightarrow$ conj(registro)
mergeReg	: registro $r1 \times$ registro $r2 \times$ conj(nombre_campo)	$\longrightarrow$ conj(registro)
	$cnc2$	$\{(cnc2 = \text{campos}(r2))\}$
ejecutarConsulta	: base_de_datos $\times$ consulta $q$	$\longrightarrow$ conj(registro)
	$\{\text{tipo\_consulta}(q) \in \{\text{FROM, SELECT, MATCH, PROJ, RENAME, INTER, UNION, PRODUCT}\}\}$	

**axiomas**  $\forall bd : \text{base\_de\_datos}, \forall t : \text{tabla}, \forall nt : \text{nombre\_tabla}$

$\text{tablas}(\text{nuevaBD}) \equiv \emptyset$

$\text{tablas}(\text{agregarTabla}(bd, t, nt)) \equiv \text{if } (\neg \text{pertenece}(bd, nt)) \text{ then } \text{Ag}(t, \text{tablas}(bd))$   
 $\text{else}$   
 $\text{tablas}(\text{reemplazarTabla}(bd, nt, t))$   
 $\text{fi}$

$\text{nombreTabla}(\text{agregarTabla}(bd, t, nt), t') \equiv \text{if } (t = t') \text{ then } nt \text{ else } \text{nombreTabla}(bd, t') \text{ fi}$

$\text{pertenece}(\text{nuevaBD}, nt) \equiv nt$

$\text{pertenece}(\text{agregarTabla}(bd, t, nt), nt') \equiv \text{if } (nt = nt') \text{ then true else } \text{pertenece}(bd, nt') \text{ fi}$

$\text{reemplazarTabla}(bd, nt, t) \equiv \text{agregarTabla}(t, \text{sacarTabla}(bd, nt))$

$\text{tablaCorrespondiente}(\text{agregarTabla}(bd, t, nt), nt') \equiv \text{if } (nt = nt') \text{ then } t$   
 $\text{else}$   
 $\text{tablaCorrespondiente}(bd, nt')$   
 $\text{fi}$

$\text{sacarTabla}(bd, t, nt), nt' \equiv \text{if } (nt = nt') \text{ then } bd \text{ else } \text{agregarTabla}(\text{sacarTabla}(bd, nt'), t, nt) \text{ fi}$

$\text{agregarRegistro}(bd, nt, r) \equiv \text{agregarTabla}(\text{sacarTabla}(bd, nt), \text{insertar}(\text{tablaCorrespondiente}(bd, nt), r), nt)$

$\text{sacarRegistro}(bd, nt, v) \equiv \text{agregarTabla}(\text{sacarTabla}(bd, nt), \text{borrar}(\text{tablaCorrespondiente}(bd, nt), v), nt)$

$\text{consultaFrom}(bd, q) \equiv \text{registros}(\text{tablaCorrespondiente}(bd, \text{nombreTabla}(q)))$

$\text{consultaSelect}(bd, q) \equiv \text{filtrarTabla}(\text{ejecutarConsulta}(bd, \text{subconsulta1}(q)), \text{campo1}(q), \text{valor}(q))$

$\text{filtrarTabla}(cr, nc, v) \equiv \text{if } (\emptyset?(cr)) \text{ then } \emptyset$   
 $\text{else}$   
 $\text{if } (\text{dameUno}(cr)[nc] = v) \text{ then } \text{Ag}(\text{dameUno}(cr), \text{filtrarTabla}(\text{sinUno}(cr, nc, v)))$   
 $\text{else}$   
 $\text{filtrarTabla}(\text{sinUno}(cr), nc, v)$   
 $\text{fi}$

$\text{consultaMatch}(bd, q) \equiv \text{encontrarRepetidos}(\text{ejecutarConsulta}(bd, \text{subconsulta1}(q)), \text{campo1}(q), \text{campo2}(q))$

$\text{encontrarRepetidos}(cr, c1, c2) \equiv \text{if } (\emptyset?(cr)) \text{ then } \emptyset$   
 $\text{else}$   
 $\text{if } (\text{dameUno}(cr)[c1] = \text{dameUno}(cr)[c2]) \text{ then } \text{Ag}(\text{dameUno}(cr), \text{encontrarRepetidos}(\text{sinUno}(cr), c1, c2))$   
 $\text{else}$   
 $\text{encontrarRepetidos}(\text{sinUno}(cr), c1, c2)$   
 $\text{fi}$

$\text{consultaProj}(bd, q) \equiv \text{filtroProyeccion}(\text{ejecutarConsulta}(bd, \text{subconsulta1}(q)), \text{conj\_campos}(q))$

```

filtroProjeccion(cr, cnc)  $\equiv$  if ( $\emptyset?(cnc)$ ) then
     $\emptyset$ 
else
    if ( $\emptyset?(cr)$ ) then
         $\emptyset$ 
    else
        Ag(regFiltradoPorCampos(dameUno(cr), cnc),
            filtroProjeccion(sinUno(cr), cnc))
    fi
fi

regFiltradoPorCampos(r, cnc)  $\equiv$  if ( $\emptyset?(cnc)$ ) then
     $\emptyset$ 
else
    if (dameUno(cnc)  $\in$  campos(r)) then
        definir( regFiltradoPorCampos(r, sinUno(cnc)),
            dameUno(cnc), registro[dameUno(cnc)] )
    else
        regFiltradoPorCampos(r, sinUno(cnc))
    fi
fi

consultaRename(bd, q)  $\equiv$  renombrar(ejecutarConsulta(bd, subconsulta1(q)), campo1(q), campo2(q))

renombrar(cr, c1, c2)  $\equiv$  if ( $\emptyset?(cr)$ ) then
     $\emptyset$ 
else
    Ag(renombrarRegistro(dameUno(cr), campos(dameUno(cr)), c1, c2),
        renombrar(sinUno(cr), c1, c2))
    fi

renombrarRegistro(r, cnc, c1, c2)  $\equiv$  if ( $\emptyset?(cnc)$ ) then
     $\emptyset$ 
else
    if (dameUno(cnc) = c1) then
        definir(renombrarRegistro(r, sinUno(cnc), c1, c2), c2, r[c1])
    else
        renombrarRegistro(r, sinUno(cnc) c1, c2)
    fi
fi

consultaUnion(bd, q)  $\equiv$  union(ejecutarConsulta(bd, subconsulta1(q)), ejecutarConsulta(bd, subconsulta2(q)))

union(cr1, cr2)  $\equiv$  cr1  $\cup$  cr2

consultaInter(bd, q)  $\equiv$  interseccion(ejecutarConsulta(bd, subconsulta1(q)), ejecutarConsulta(bd,
    subconsulta2(q)))

interseccion(cr1, cr2)  $\equiv$  cr1  $\cap$  cr2

consultaProduct(bd, q)  $\equiv$  prodCart(ejecutarConsulta(bd, subconsulta1(q)), ejecutarConsulta(bd, subconsulta2(q)))

prodCart(cr1, cr2)  $\equiv$  auxProdCart(cr1, cr2, cr2)

auxProdCart(cr1, cr2, cr2')  $\equiv$  if ( $\emptyset?(cr1)$ ) then
     $\emptyset$ 
else
    if ( $\emptyset?(cr2)$ ) then
        auxProdCart(sinUno(cr1), cr2', cr2')
    else
        Ag(mergeReg(dameUno(cr1), dameUno(cr2)),
            campos(dameUno(cr2)), auxProdCart(cr1, sinUno(cr2), cr2'))
    fi
fi

mergeReg(r1, r2, cnc2)  $\equiv$  if ( $\emptyset?(cc2)$ ) then
    r1
else
    definir(mergeReg(r1, r2, sinUno(cnc2)), dameUno(cnc2), r2[dameUno(cnc2)])
fi

```

```

ejecutarConsulta(bd, q) ≡ if (tipo_consulta(q) ∈ {FROM}) then
    consultaFrom(bd, q)
else
    if (tipo_consulta(q) ∈ {SELECT}) then
        consultaSelect(bd, q)
    else
        if (tipo_consulta(q) ∈ {MATCH}) then
            consultaMatch(bd, q)
        else
            if (tipo_consulta(q) ∈ {PROJ}) then
                consultaProj(bd, q)
            else
                if (tipo_consulta(q) ∈ {RENAME}) then
                    consultaRename(bd, q)
                else
                    if (tipo_consulta(q) ∈ {INTER}) then
                        consultaInter(bd, q)
                    else
                        if (tipo_consulta(q) ∈ {UNION}) then
                            consultaUnion(bd, q)
                        else
                            consultaProduct(bd, q)
                        fi
                    fi
                fi
            fi
        fi
    fi
fi

```

**Fin TAD**

## 2. Modulo BaseDeDatos

El modulo BaseDeDatos provee operaciones básicas a realizarse sobre una base de datos. Permite crear una base de datos, agregar o eliminar una tabla, agregar o eliminar registros a una tabla, y también permite ejecutar consultas para obtener registros de una tabla.

### Interfaz

**se explica con:** BASEDEDATOS

**usa:** REGISTRO, TABLA, CONSULTA, BOOL, NAT, CONJUNTOLINEAL

**género:** BASE\_DE\_DATOS

Trabajo Práctico 1 Operaciones básicas de BaseDeDatos

NUEVABD() → *res* : BaseDeDatos

**Pre** ≡ {true}

**Post** ≡ {*res* = nuevaBD}

**Descripción:** Crea una nueva base de datos.

AGREGARTABLA(**in/out** *b*: BaseDeDatos, **in** *t*: Tabla, **in** *nt*: NombreTabla)

**Pre** ≡ {*b* = *b0* ∧ ¬TablasRepetidasEnBd(*b0*)}

**Post** ≡ {¬TablasRepetidasEnBd(*b*) ∧ *b* = AgregarTabla(*b0*, *t*, *nt*)}

**Descripción:** Inserta una tabla a una base de datos. En caso de que la tabla ya existe en la base, la sobrescribe.

ELIMINARTABLA(**in/out** *b*: BaseDeDatos, **in** *nt*: NombreTabla)

**Pre** ≡ {*b* = *b0*}

**Post** ≡ {(*nt* ∉ tablas(*b0*) ∧ *b*=*b0*) ∨ (*nt* ∈ tablas(*b0*) ∧ *b*=sacarTabla(*b0*, *nt*))}

**Descripción:** Elimina una tabla de una base de datos. En caso de que la tabla no exista en la base, la operacion

queda sin efecto.

AGREGARREGISTRO(**in/out**  $b: BaseDeDatos$ , **in**  $nt: NombreTabla$ , **in**  $r: Registro$ )

**Pre**  $\equiv \{b = b0 \wedge (\text{campos}(\text{tablaCorrespondiente}(nt)) = \text{campos}(r))\}$

**Post**  $\equiv \{\neg \text{TablasRepetidasEnBd}(b) \wedge b = \text{AgregarRegistro}(b0, \text{tablaCorrespondiente}(nt), r)\}$

**Descripción:** Inserta una registro en una tabla en una base de datos dándole valor a todos (y solamente a) los campos de la tabla. En caso de que el registro ya existe en la base, lo sobrescribe.

ELIMINARREGISTRO(**in/out**  $b: BaseDeDatos$ , **in**  $nt: NombreTabla$ , **in**  $r: Registro$ )

**Pre**  $\equiv \{b = b0 \wedge \text{ExisteTablaEnBd}(b0, nt)\}$

**Post**  $\equiv \{(b = b0 \wedge r \notin \text{registros}(\text{tablaCorrespondiente}(nt))) \vee (r \in \text{registros}(\text{tablaCorrespondiente}(nt)) \wedge b = \text{sacarRegistro}(b0, \text{registros}(\text{tablaCorrespondiente}(nt)))\}$

**Descripción:** Elimina un registro de una tabla en una base de datos. En caso de que el registro no pertenezca a la tabla, esta queda sin efecto

TABLAS(**in**  $b: BaseDeDatos$ )  $\rightarrow res: Conj(nombreTabla)$

**Pre**  $\equiv \{true\}$

**Post**  $\equiv \{res = \text{tablas}(b)\}$

**Descripción:** Devuelve el conjunto de tablas de la base de datos.

NOMBRETABLA(**in**  $b: BaseDeDatos$ , **in**  $t: Tabla$ )  $\rightarrow res: nombreTabla$

**Pre**  $\equiv \{t \in \text{tablas}(b)\}$

**Post**  $\equiv \{res = \text{nombretabla}(b, t)\}$

**Descripción:** Dados una tabla y una base de datos, devuelve el nombre de la tabla en dicha base.

REGISTROS(**in**  $b: BaseDeDatos$ , **in**  $t: Tabla$ )  $\rightarrow res: Conj(Registros)$

**Pre**  $\equiv \{\text{ExisteTablaEnBd}(b, t)\}$

**Post**  $\equiv \{res = \text{registros}(t)\}$

**Descripción:** Devuelve el conjunto de registros de una tabla perteneciente a la base de datos.

EJECUTARCONSULTA(**in**  $b: BaseDeDatos$ , **in**  $q: Consulta$ )  $\rightarrow res: conj(Registro)$

**Pre**  $\equiv \{true\}$

**Post**  $\equiv \{$

$(\text{TipoConsulta}(q) \in \{FROM\} \wedge \text{ExisteTablaEnBd}(b, \text{NombreTabla}(q)) \wedge_L res = \text{ejecutarConsulta}(b, q)) \vee$

$(\text{TipoConsulta}(q) \in \{FROM\} \wedge \neg \text{ExisteTablaEnBd}(b, \text{NombreTabla}(q)) \wedge res = \emptyset) \vee$

$(\text{TipoConsulta}(q) \in \{SELECT\} \wedge \text{RegistrosTienenCampo}(\text{ejecutarConsulta}(b, \text{subconsulta1}(q)), \text{campo1}(q)) \wedge_L$

$res = \text{ejecutarConsulta}(b, q)) \vee$

$(\text{TipoConsulta}(q) \in \{SELECT\} \wedge \neg \text{RegistrosTienenCampo}(\text{ejecutarConsulta}(b, \text{subconsulta1}(q)), \text{campo1}(q)) \wedge$

$res = \text{ejecutarConsulta}(b, \text{subconsulta1}(q)) \vee$

$(\text{TipoConsulta}(q) \in \{MATCH\} \wedge \text{RegistrosTienenCampo}(\text{ejecutarConsulta}(b, \text{subconsulta1}(q)), \text{campo1}(q)) \wedge$

$\text{RegistrosTienenCampo}(\text{ejecutarConsulta}(b, \text{subconsulta1}(q)), \text{campo2}(q)) \wedge_L res = \text{ejecutarConsulta}(b, q) \vee$

$(\text{TipoConsulta}(q) \in \{MATCH\} \wedge \neg (\text{RegistrosTienenCampo}(\text{ejecutarConsulta}(b, \text{subconsulta1}(q)), \text{campo1}(q))$

$\wedge \text{RegistrosTienenCampo}(\text{ejecutarConsulta}(b, \text{subconsulta1}(q)), \text{campo2}(q))) \wedge res = \text{ejecutarConsulta}(b, \text{subcon}$

$sulta1(q)) \vee$

$(\text{TipoConsulta}(q) \in \{RENAME\} \wedge \text{RegistrosTienenCampo}(\text{ejecutarConsulta}(b, \text{subconsulta1}(q)), \text{campo1}(q))$

$\wedge \neg \text{RegistrosTienenCampo}(\text{ejecutarConsulta}(b, \text{subconsulta1}(q)), \text{campo2}(q)) \wedge_L res = \text{ejecutarConsulta}(b, q) \vee$

$(\text{TipoConsulta}(q) \in \{RENAME\} \wedge \neg (\text{RegistrosTienenCampo}(\text{ejecutarConsulta}(b, \text{subconsulta1}(q)), \text{campo1}(q))$

$\wedge \neg \text{RegistrosTienenCampo}(\text{ejecutarConsulta}(b, \text{subconsulta1}(q)), \text{campo2}(q))) \wedge res = \text{ejecutarConsulta}(b, \text{subcon}$

$sulta1(q)) \vee$

$(\text{TipoConsulta}(q) \in \{PRODUCT\} \wedge \neg \text{HayCamposEnComun}(\text{ejecutarConsulta}(b, \text{subconsulta1}(q)), \text{ejecutarCon}$

$sulta}(b, \text{subconsulta2}(q))) \wedge_L res = \text{ejecutarConsulta}(b, q) \vee$

$(\text{TipoConsulta}(q) \in \{PRODUCT\} \wedge \text{HayCamposEnComun}(\text{ejecutarConsulta}(b, \text{subconsulta1}(q)), \text{ejecutarCon}$

$sulta}(b, \text{subconsulta2}(q))) \wedge res = \emptyset) \vee$

$(\text{TipoConsulta}(q) \in \{PROJ, INTER, UNION\} \wedge res = \text{ejecutarConsulta}(b, q))$

$\}$

**Descripción:** Dados una consulta  $c$  y una base de datos  $b$ , devuelve los registros resultantes luego de haber ejecutado la consulta  $c$  en la base  $b$ , añadiendo las siguientes particularidades dependiendo del  $\text{TipoConsulta}(c)$ :

Si  $\text{TipoConsulta}(c) \in \{\text{FROM}\}$ , devuelve todos los registros contenidos en la tabla correspondiente de  $b$  cuyo nombre es  $\text{nombre\_tabla}(c)$ . Si dicha tabla no se encuentra en  $b$ , devolvemos el conjunto vacío.

Si  $\text{TipoConsulta}(c) \in \{\text{SELECT}\}$ , devuelve todos los registros resultantes al ejecutar la consulta  $\text{subconsulta1}(c)$  en  $b$ , tales que el campo  $\text{campo1}(c)$  tiene valor  $\text{valor}(c)$ . Si hay registros que no contienen el campo  $\text{campo1}(c)$ , devolvemos el conjunto de registros resultantes al ejecutar la  $\text{subconsulta1}(c)$  sin modificar.

Si  $\text{TipoConsulta}(c) \in \{\text{MATCH}\}$ , devuelve todos los registros resultantes al ejecutar la consulta  $\text{subconsulta1}(c)$  en  $b$ , tales que los campos  $\text{campo1}(c)$  y  $\text{campo2}(c)$  tienen el mismo valor. Si hay registros que no contienen el campo  $\text{campo1}(c)$  o el campo  $\text{campo2}(c)$ , devolvemos el conjunto de registros resultantes al ejecutar la  $\text{subconsulta1}(c)$  sin modificar.

Si  $\text{TipoConsulta}(c) \in \{\text{PROJ}\}$ , devuelve todos los registros resultantes al ejecutar la consulta  $\text{subconsulta1}(c)$  en  $b$ , pero que incluyen solamente los campos del conjunto  $\text{conj\_campos}(c)$ .

Si  $\text{TipoConsulta}(c) \in \{\text{RENAME}\}$ , devuelve todos los registros resultantes al ejecutar la consulta  $\text{subconsulta1}(c)$  en  $b$ , tales que el campo  $\text{campo1}(c)$  será reemplazado por el campo  $\text{campo2}(c)$ . Si hay registros que no tienen definido el campo  $c1$  o que ya tienen definido el campo  $c2$ , devolvemos el conjunto de registros resultantes al ejecutar la  $\text{subconsulta1}(c)$  sin modificar.

Si  $\text{TipoConsulta}(c) \in \{\text{INTER}\}$ , devuelve la intersección entre los registros resultantes de la ejecución de  $\text{subconsulta1}(c)$  y  $\text{subconsulta2}(c)$

Si  $\text{TipoConsulta}(c) \in \{\text{UNION}\}$ , devuelve la unión entre los registros resultantes de la ejecución de  $\text{subconsulta1}(c)$  y  $\text{subconsulta2}(c)$

Si  $\text{TipoConsulta}(c) \in \{\text{PRODUCT}\}$ , devuelve el producto cartesiano entre los registros resultantes de la ejecución de  $\text{subconsulta1}(c)$  y  $\text{subconsulta2}(c)$ . Si hay registros provenientes de la consulta  $\text{subconsulta1}(c)$  que tienen campos en común con registros provenientes de la  $\text{subconsulta2}(c)$ , devolvemos el conjunto vacío.

Trabajo Práctico 1Especificación de las operaciones auxiliares utilizadas en la interfaz

$\text{TablasRepetidasEnBd} : \text{BaseDeDatos} \rightarrow \text{bool}$

$$\text{TablasRepetidasEnBd}(b) \equiv (\exists t1: \text{Tabla})(\exists t2: \text{Tabla})(t1 \neq t2 \wedge t1 \in \text{Tablas}(b) \wedge t2 \in \text{Tablas}(b)) \wedge_L \text{nombreTabla}(b, t1) = \text{nombreTabla}(b, t2))$$

$\text{ExisteTablaEnBd} : \text{BaseDeDatos} \times \text{NombreTabla} \rightarrow \text{bool}$

$$\text{ExisteTablaEnBd}(b, nt) \equiv (\exists t: \text{Tabla})(t \in \text{Tablas}(b) \wedge_L \text{NombreTabla}(b, t) = nt)$$

$\text{RegistrosTienenCampo} : \text{conj}(\text{Registro}) \times \text{NombreCampo} \rightarrow \text{bool}$

$$\text{RegistrosTienenCampo}(c, nc) \equiv (\forall r: \text{Registro})(r \in c \Rightarrow_L nc \in \text{campos}(r))$$

$\text{HayCamposEnComun} : \text{conj}(\text{Registro}) \times \text{conj}(\text{Registro}) \rightarrow \text{bool}$

$$\text{HayCamposEnComun}(cr1, cr2) \equiv (\exists r1: \text{Registro})(\exists r2: \text{Registro})(r1 \in cr1 \wedge r2 \in cr2 \wedge (\exists nc: \text{nombreCampo})(nc \in \text{campos}(r1) \wedge (nc \in \text{campos}(r2))))$$

### 3. Modulo Consulta

El modulo Consulta provee operaciones básicas para generar una consulta. Permite construir una consulta a partir de una tabla.

#### Interfaz

**se explica con:** CONSULTA

**usa:** REGISTRO, TABLA, BOOL, NAT

**género:** CONSULTA

Trabajo Práctico 1 Operaciones básicas de Consulta

**FROM**(*in nt : NombreTabla*)  $\rightarrow res : Consulta$

**Pre**  $\equiv \{true\}$

**Post**  $\equiv \{res = FROM(nt)\}$

**Descripción:** Crea una consulta *c* a partir de un nombreTabla *nt*, que tiene las siguientes particularidades:

TipoConsulta(*c*): FROM, NombreTabla(*c*): *nt*

**SELECT**(*in c : Consulta, in nc : nombreCampo, in v : Valor*)  $\rightarrow res : Consulta$

**Pre**  $\equiv \{true\}$

**Post**  $\equiv \{res = SELECT(c, nc, v)\}$

**Descripción:** Dada una consulta *c*, un nombreCampo *nc* y un valor *v*, SELECT es una nueva consulta que contiene las siguientes particularidades: TipoConsulta(*res*) : SELECT, PrimerCampo(*res*): nombre, Valor(*res*): *v*, SubconsultaUno(*res*): *c*

**MATCH**(*in c : Consulta, in nc1 : nombreCampo, in nc2 : nombreCampo*)  $\rightarrow res : Consulta$

**Pre**  $\equiv \{true\}$

**Post**  $\equiv \{res = MATCH(c0, nc1, nc2)\}$

**Descripción:** Dada una consulta *c* y dos nombreCampo *nc1* y *nc2*, MATCH es una nueva consulta que contiene las siguientes particularidades: TipoConsulta(*res*) : MATCH, PrimerCampo(*res*): *nc1*, SegundoCampo(*res*): *nc2*, SubconsultaUno(*res*): *c*

**PROJ**(*in c : Consulta, in cnc : conj(nombreCampo)*)  $\rightarrow res : Consulta$

**Pre**  $\equiv \{true\}$

**Post**  $\equiv \{c = PROJ(c, cnc)\}$

**Descripción:** Dada una consulta *c* y un conj(nombreCampo) campos, PROJ es una nueva consulta que contiene las siguientes particularidades: TipoConsulta(*nc*) : PROJ, Conj\_Campos(*nc*): campos, SubconsultaUno(*nc*): *c*

**RENAME**(*in c : Consulta, in nc1 : nombreCampo, in nc2 : nombreCampo*)  $\rightarrow res : Consulta$

**Pre**  $\equiv \{true\}$

**Post**  $\equiv \{c = RENAME(c, nc1, nc2)\}$

**Descripción:** Dada una consulta *c* y dos nombresCampo *nc1* y *nc2*, RENAME es una nueva consulta que contiene las siguientes particularidades: TipoConsulta(*res*) : RENAME, PrimerCampo(*res*) : *nc1*, SegundoCampo(*res*) : *nc2*, SubconsultaUno(*res*): *c*

**INTER**(*in q1 : consulta, in q2 : Consulta*)  $\rightarrow res : Consulta$

**Pre**  $\equiv \{true\}$

**Post**  $\equiv \{c = INTER(q1, q2)\}$

**Descripción:** Dada una consulta *c* y dos nombresCampo *nc1* y *nc2*, INTER es una nueva consulta que contiene las siguientes particularidades: TipoConsulta(*res*) : INTER, SubconsultaUno(*res*): *q1* SubconsultaDos(*res*): *q2*

**UNION**(*in q1 : consulta, in q2 : Consulta*)  $\rightarrow res : Consulta$

**Pre**  $\equiv \{true\}$

**Post**  $\equiv \{c = UNION(q1, q2)\}$

**Descripción:** Dada una consulta *c* y dos nombresCampo *nc1* y *nc2*, UNION es una nueva consulta que contiene las siguientes particularidades: TipoConsulta(*res*) : UNION, SubconsultaUno(*res*): *q1* SubconsultaDos(*res*): *q2*

**PRODUCT**(*in q1 : consulta, in q2 : Consulta*)  $\rightarrow res : Consulta$

**Pre**  $\equiv \{true\}$

**Post**  $\equiv \{c = PRODUCT(q1, q2)\}$

**Descripción:** Dada una consulta *c* y dos nombresCampo *nc1* y *nc2*, UNION es una nueva consulta que contiene las siguientes particularidades: TipoConsulta(*res*) : PRODUCT, SubconsultaUno(*res*): *q1* SubconsultaDos(*res*): *q2*

**TIPOCONSULTA**(*in c : Consulta*)  $\rightarrow res : tipoConsulta$

**Pre**  $\equiv \{true\}$

**Post**  $\equiv \{res = tipo_consulta(c)\}$

**Descripción:** Dada una consulta, devuelve su tipo

**NOMBRETABLA**(*in c : Consulta*)  $\rightarrow res : nombreTabla$



**Pre**  $\equiv \{\text{tipo.consulta}(c) \in \{FROM\}\}$

**Post**  $\equiv \{\text{res} = \text{nombre.tabla}(c)\}$

**Descripción:** Dada una consulta de tipo FROM, devuelve la tabla asociada a dicha consulta

PRIMERCAMPO(**in**  $c: Consulta$ )  $\rightarrow res: \text{nombre\_campo}$

**Pre**  $\equiv \{\text{tipo.consulta}(c) \in \{SELECT, MATCH, RENAME\}\}$

**Post**  $\equiv \{\text{res} = \text{campo1}(c)\}$

**Descripción:** Dada una consulta de tipo FROM, MATCH O RENAME, devuelve el primer parámetro de tipo nombreCampo asociado a dicha consulta

SEGUNDOCAMPO(**in**  $c: Consulta$ )  $\rightarrow res: \text{nombre\_campo}$

**Pre**  $\equiv \{\text{tipo.consulta}(c) \in \{MATCH, RENAME\}\}$

**Post**  $\equiv \{\text{res} = \text{campo2}(c)\}$

**Descripción:** Dada una consulta de tipo MATCH O RENAME, devuelve el segundo parámetro de tipo nombreCampo asociado a dicha consulta

VALOR(**in**  $c: Consulta$ )  $\rightarrow res: \text{valor}$

**Pre**  $\equiv \{\text{tipo.consulta}(c) \in \{SELECT\}\}$

**Post**  $\equiv \{\text{res} = \text{valor}(c)\}$

**Descripción:** Dada una consulta de tipo SELECT(consulta, nombreCampo, valor), devuelve el valor asociado a dicha consulta

CONJCAMPOS(**in**  $c: Consulta$ )  $\rightarrow res: \text{Conj}(\text{nombreCampo})$

**Pre**  $\equiv \{\text{tipo.consulta}(c) \in \{PROJ\}\}$

**Post**  $\equiv \{\text{res} = \text{conj.campos}(c)\}$

**Descripción:** Dada una consulta de tipo PROJ(consulta, conjCampos), devuelve el conjunto de campos asociado a dicha consulta

SUBCONSULTAUNO(**in/out**  $c: Consulta$ )

**Pre**  $\equiv \{c = c0 \wedge \text{tipo.consulta}(c0) \in \{SELECT, MATCH, PROJ, RENAME, INTER, UNION, PRODUCT\}\}$

**Post**  $\equiv \{c = \text{subconsulta1}(c0)\}$

**Descripción:** Dada una consulta de tipo SELECT, MATCH, PROJ, RENAME, INTER, UNION O PRODUCT, devuelve la primer subconsulta asociada a dicha consulta

SUBCONSULTADOS(**in/out**  $c: Consulta$ )

**Pre**  $\equiv \{c = c0 \wedge \text{tipo.consulta}(c0) \in \{INTER, UNION, PRODUCT\}\}$

**Post**  $\equiv \{c = \text{subconsulta2}(c0)\}$

**Descripción:** Dada una consulta de tipo INTER, UNION, PRODUCT, devuelve la segunda subconsulta asociada a dicha consulta

Trabajo Práctico 1Especificación de las operaciones auxiliares utilizadas en la interfaz

TablasRepetidasEnBd : BaseDeDatos  $\rightarrow$  bool

$$\text{TablasRepetidasEnBd}(b) \equiv (\exists t1: \text{Tabla})(\exists t2: \text{Tabla})(t1 \neq t2 \wedge t1 \in \text{Tablas}(b) \wedge t2 \in \text{Tablas}(b) \wedge \text{nombreTabla}(b, t1) = \text{nombreTabla}(b, t2))$$

## 4. Modulo Registro

El modulo Registro provee operaciones básicas a realizarse sobre un registro. Permite crear un nuevo registro, obtener los campos de un registro, y obtener el valor de uno de los campos de un registro.

### Interfaz

se explica con: REGISTRO

usa: REGISTRO, TABLA, CONSULTA, BOOL, NAT

género: REGISTRO

Trabajo Práctico 1Operaciones básicas de Registro

NUEVO()  $\rightarrow res : Registro$

**Pre**  $\equiv \{true\}$

**Post**  $\equiv \{res = Registro\}$

**Descripción:** Crea un nueva registro vacio.

DEFINIR(**in/out**  $r : Registro$ , **in**  $nc : NombreCompleto$ , **in**  $v : Valor$ )

**Pre**  $\equiv \{r = r0 \wedge (nc \notin campos(r))\}$

**Post**  $\equiv \{res = definir(r0, nc, v)\}$

**Descripción:** Define un nuevo campo y su valor correspondiente dentro de un registro.

CAMPOS(**in**  $r : Registro$ )  $\rightarrow res : Conj(nombreCampo)$

**Pre**  $\equiv \{true\}$

**Post**  $\equiv \{res = campos(r)\}$

**Descripción:** Devuelve las campos de un registro ( $conj(nombreCampo)$ ).

**• [•]**(**in**  $r : Registro$ , **in**  $nc : NombreCampo$ , **in**  $v : Valor$ )  $\rightarrow res : Registro$

**Pre**  $\equiv \{nc \in campos(r)\}$

**Post**  $\equiv \{res = campos(r)\}$

**Descripción:** Devuelve el valor del campo de un registro.

## 5. Modulo Tabla

El modulo Tabla provee operaciones básicas a realizarse sobre una tabla. Permite crear una tabla, agregar o eliminar un registro de una tabla, y brinda operaciones para obtener la clave de una tabla, obtener las columnas( $conj(nombreCampo)$ ) y las filas( $conj(Registro)$ ) de una tabla.

### Interfaz

**se explica con:** TABLA

**usa:** REGISTRO, CONSULTA, BOOL, NAT, CONJUNTOLINEAL

**género:** tabla.

Trabajo Práctico 1Operaciones básicas de Tabla

NUEVATABLA(**in**  $c : conj(nombreTabla)$ , **in**  $nt : nombreTabla$ )  $\rightarrow res : Tabla$

**Pre**  $\equiv \{nt \in c\}$

**Post**  $\equiv \{res = nueva(t, nt)\}$

**Descripción:** Crea una tabla a una base de datos.

INSERTAR(**in/out**  $t : Tabla$ , **in**  $r : Registro$ )

**Pre**  $\equiv \{t = t0 \wedge campos(t) = campos(r)\}$

**Post**  $\equiv \{t = insertar(t0, r)\}$

**Descripción:** Dados una tabla y un registro, inserta el registro en dicha tabla. En caso de que el registro ya exista en la tabla, lo sobrescribe

BORRAR(**in/out**  $t : Tabla$ , **in**  $v : Valor$ )

**Pre**  $\equiv \{t = t0\}$

**Post**  $\equiv \{(\neg ValorEnTabla(t, v) \wedge t=t0) \vee (ValorEnTabla(t, v) \wedge t=borrar(t, v))\}$

**Descripción:** Dados una tabla y un valor, se fija si dicho valor coincide con alguno de los valores del campo clave de la tabla. En caso afirmativo borra el registro que contenga dicho valor, de lo contrario devuelve la tabla sin modificaciones.

CAMPOS(**in**  $t : Tabla$ )  $\rightarrow res : Conj(nombreCampo)$

**Pre**  $\equiv \{true\}$

**Post**  $\equiv \{res = campos(t)\}$

**Descripción:** Devuelve las columnas( $conj(nombreCampo)$ ) de la tabla.

REGISTROS(**in**  $t : Tabla$ )  $\rightarrow res : Conj(Registro)$

**Pre**  $\equiv \{true\}$

**Post**  $\equiv \{\text{res} = \text{registros}(t)\}$

**Descripción:** Devuelve las filas( $\text{conj}(\text{Registro})$ ) de la tabla.

**CLAVE**(**in**  $t: \text{Tabla}$ )  $\rightarrow \text{res} : \text{nombreCampo}$

**Pre**  $\equiv \{\text{true}\}$

**Post**  $\equiv \{\text{res} = \text{clave}(t)\}$

**Descripción:** Devuelve el nombre del campo declarado como clave de la tabla.

Trabajo Práctico 1Especificación de las operaciones auxiliares utilizadas en la interfaz

$\text{ValorEnTabla} : \text{Tabla} \times \text{Valor} \longrightarrow \text{bool}$

$\text{ValorEnTabla}(t, v) \equiv (\exists r: \text{Registro})(r \in \text{registros}(t) \wedge (\exists \text{nc}: \text{NombreCampo})(\text{nc} = \text{clave}(t)) \wedge (r[\text{nc}] = v))$