

Especificacion de TADs auxiliares de Base de Datos

Trabajo Práctico 2 - 2^{do} cuatrimestre de 2017

Algoritmos y Estructuras de Datos II, DC, UBA.

Versión: 1.0b del 2 de octubre de 2017 (ver CHANGELOG.md)

Especificación

1. TAD DATO

TAD TIPO es BOOL

TAD DATO

g�neros	dato
---------	------

usa string, nat, tipo

exporta generadores, observadores básicos y otras operaciones

igualdad observacional

$$(\forall d_1, d_2 : \text{dato}) \left(d_1 =_{\text{obs}} d_2 \iff \left(\begin{array}{l} \text{tipo?}(d_1) \equiv \text{tipo?}(d_2) \wedge_{\text{L}} \\ ((\text{Nat?}(d_1) \Rightarrow_{\text{L}} \text{valorNat}(d_1) =_{\text{obs}} \text{valorNat}(d_2)) \wedge \\ (\text{String?}(d_1) \Rightarrow_{\text{L}} \text{valorStr}(d_1) =_{\text{obs}} \text{valorStr}(d_2))) \end{array} \right) \right)$$

generadores

$$\text{datoString} : \text{string} \longrightarrow \text{dato}$$
$$\text{datoNat} : \text{nat} \longrightarrow \text{dato}$$

observadores básicos

$$\text{tipo?} \quad : \text{ dato} \quad \longrightarrow \text{ tipo}$$
$$\text{valorNat} : \text{dato } d \longrightarrow \text{nat} \quad \{\text{Nat?}(d)\}$$
$$\text{valorStr} : \text{dato } d \longrightarrow \text{string} \quad \{\text{String?}(d)\}$$

otras operaciones

$$\text{String?} \quad : \text{ dato} \quad \longrightarrow \text{ bool}$$
$$\text{Nat?} \quad : \text{dato} \quad \longrightarrow \text{bool}$$

axiomas

$$(\forall s: \text{string}, \forall n: \text{nat}, \forall d: \text{dato})$$
$$\text{tipo?}(\text{datoNat}(n)) \quad \equiv \quad \text{true}$$

```
tipo?(datoString(s))    ≡  false
```

$$\text{valorNat}(\text{datoNat}(n)) \quad \equiv \quad n$$

$\text{valorStr}(\text{datoString}(s)) \equiv s$
 $\text{Nat?}(d) \equiv \text{tipo?}(d)$
 $\text{String?}(d) \equiv \neg \text{Nat?}(d)$

Fin TAD

2. TAD REGISTRO

TAD CAMPO es STRING

TAD REGISTRO

TAD REGISTRO extiende a DICCIONARIO(CAMPO, DATO)

géneros registro

usa string, dato, campo, dicc

exporta otras operaciones

otras operaciones

$\text{campos} : \text{registro} \rightarrow \text{conj}(\text{campo})$

$\text{valor} : \text{campo } c \times \text{registro } r \rightarrow \text{dato} \quad \{c \in \text{campos}(r)\}$

$\text{nuevoRegistro} : \text{secu}(\text{campo}) \text{ cs} \times \text{secu}(\text{dato}) \text{ ds} \rightarrow \text{registro} \quad \{ \text{long}(\text{cs}) = \text{long}(\text{ds}) \}$

axiomas

$(\forall c: \text{campo}, \forall r: \text{registro})$

$\text{campos}(r) \equiv \text{claves}(r)$

$\text{valor}(c, r) \equiv \text{obtener}(c, r)$

$\text{nuevoRegistro}(\text{cs}, \text{ds}) \equiv \text{if } \text{vacía?}(\text{cs}) \text{ then}$
 vacío
 else
 $\text{definir}(\text{prim}(\text{cs}), \text{prim}(\text{ds}), \text{nuevoRegistro}(\text{fin}(\text{cs}), \text{fin}(\text{ds})))$
 fi

Fin TAD

3. TAD TABLA

TAD TABLA

géneros tabla

usa campo, clave, registro, conj, string

exporta Generadores, observadores, otras operaciones

igualdad observacional

$(\forall t_1, t_2 : \text{tabla}) \left(t_1 =_{\text{obs}} t_2 \iff \left(\begin{array}{l} \text{campos}(t_1) =_{\text{obs}} \text{campos}(t_2) \wedge_{\text{L}} (\forall c: \text{campo}) (c \in \text{campos}(t_1) \implies c \in \text{campos}(t_2)) \\ \text{campos}(t_1) \Rightarrow_{\text{L}} \text{tipoCampo}(c, t_1) =_{\text{obs}} \text{tipoCampo}(c, t_2) \wedge \\ \text{claves}(t_1) =_{\text{obs}} \text{claves}(t_2) \wedge \text{registros}(t_1) =_{\text{obs}} \text{registros}(t_2) \end{array} \right) \right)$

observadores básicos

claves	: tabla	\longrightarrow conj(campo)	
campos	: tabla	\longrightarrow conj(campo)	
tipoCampo	: campo $c \times$ tabla t	\longrightarrow bool	$\{c \in \text{campos}(t)\}$
registros	: tabla	\longrightarrow conj(registro)	

generadores

nuevaTabla	: conj(campo) $claves \times$ registro $columnas$	\longrightarrow tabla	$\{claves \neq \emptyset \wedge claves \subseteq \text{campos}(columnas)\}$
agregarRegistro	: registro $r \times$ tabla t	\longrightarrow tabla	$\{\text{campos}(r) =_{\text{obs}} \text{campos}(t) \wedge \text{puedoInsertar?}(r, t)\}$

otras operaciones

puedoInsertar?	: registro $r \times$ tabla t	\longrightarrow bool	
compatible	: registro \times tabla	\longrightarrow bool	
hayCoincidencia	: registro $r \times$ conj(campo) $claves \times$ conj(registro) cr	\longrightarrow bool	$\{(\forall r': \text{registro})(r' \in \{r\} \cup cr) \Rightarrow_L claves \subseteq \text{campos}(r')\}$
filtrarRegistros	: registro $r \times$ campo $clave \times$ conj(registro) rs	\longrightarrow conj(registro)	$\{(\forall r': \text{registro})(r' \in \{r\} \cup cr) \Rightarrow_L clave \in \text{campos}(r')\}$
misimosTipos	: conj(campo) $cs \times$ registro $r \times$ tabla t	\longrightarrow bool	$\{cs \subseteq \text{campos}(r) \wedge cs \subseteq \text{campos}(t)\}$
estructura	: tabla t	\longrightarrow registro	

axiomas

$(\forall claves: \text{conj}(\text{campo}), \forall t: \text{tabla}, \forall r: \text{registro}, \forall regs: \text{conj}(\text{registro}) \forall nombre: \text{string})$

claves(nuevaTabla($claves, r$))	\equiv $claves$
claves(agregarRegistro(r, t))	\equiv $claves(t)$
campos(nuevaTabla($claves, r$))	\equiv $\text{campos}(r)$
campos(agregarRegistro(r, t))	\equiv $\text{campos}(t)$
tipoCampo($c, nuevaTabla(claves, r)$)	\equiv $\text{tipo?}(\text{valor}(c, r))$
tipoCampo($c, agregarRegistro(r, t)$)	\equiv $\text{tipoCampo}(c, t)$
registros(nuevaTabla($claves, r$))	\equiv \emptyset
registros(agregarRegistro(r, t))	\equiv $\text{Ag}(r, \text{registros}(t))$
puedoInsertar?(r, t)	\equiv $\text{compatible}(r, t) \wedge \neg \text{hayCoincidencia}(r, \text{claves}(t), \text{registros}(t))$
compatible(r, t)	\equiv $\text{campos}(r) = \text{campos}(t) \wedge_L \text{misimosTipos}(\text{campos}(t), r, t)$
hayCoincidencia($r, claves, regs$)	\equiv $(\emptyset?(claves) \wedge \neg \emptyset?(regs)) \vee_L (\neg \emptyset?(regs) \wedge_L \text{hayCoincidencia}(r, \text{sinUno}(claves), \text{filtrarRegistros}(r, \text{dameUno}(c), regs)))$
filtrarRegistros($r, c, regs$)	\equiv if $\emptyset?(regs)$ then $regs$ else if $\text{valor}(c, r) = \text{valor}(c, \text{dameUno}(regs))$ then $\text{Ag}(\text{dameUno}(regs), \text{filtrarRegistros}(r, c, \text{sinUno}(regs)))$ else $\text{filtrarRegistros}(r, c, \text{sinUno}(regs))$ fi fi
misimosTipos(cs, r, t)	\equiv $\emptyset?(cs) \vee_L (\text{misimosTipos}(\text{sinUno}(cs), r, t) \wedge \text{tipo?}(\text{valor}(\text{dameUno}(cs), r)) = \text{tipoCampo}(\text{dameUno}(cs), t))$

$\text{estructura}(\text{nuevaTabla}(\text{claves}, r)) \equiv r$
 $\text{estructura}(\text{agregarRegistro}(r, t)) \equiv \text{estructura}(t)$

Fin TAD

4. TAD CRITERIO

TAD CRITERIO es CONJUTO(RESTRICCION)

TAD RESTRICCION

TAD RESTRICCION

géneros restricción

usa string, dato, bool

exporta otras operaciones

igualdad observacional

$$(\forall r_1, r_2 : \text{restriccion}) \left(r_1 =_{\text{obs}} r_2 \iff \left(\text{campo}(r_1) = \text{campo}(r_2) \wedge \text{dato}(r_1) = \text{dato}(r_2) \wedge \right) \right)$$

observadores básicos

$\text{campo} : \text{restriccion} \longrightarrow \text{campo}$

$\text{dato} : \text{restriccion} \longrightarrow \text{dato}$

$\text{porIgual} : \text{restriccion} \longrightarrow \text{bool}$

generadores

$\text{nueva} : \text{string } \text{campo} \times \text{dato } \text{valor} \times \text{bool } \text{porIgual} \longrightarrow \text{restriccion}$

axiomas

$(\forall r : \text{restriccion})$

$\text{campos}(\text{nueva}(c, v, i)) \equiv c$

$\text{dato}(\text{nueva}(c, v, i)) \equiv v$

$\text{porIgual}(\text{nueva}(c, v, i)) \equiv i$

Fin TAD

5. TAD BASEDEDATOS

TAD BASEDEDATOS

géneros base

usa NAT, STRING, TABLA, REGISTRO, CAMPO, DATO, CRITERIO, RESTRICCION, TIPO, BOOL, CONJ(α)

exporta generadores, observadores básicos, buscar, criterioValido, join y criteriosMasUsados

igualdad observacional

$$(\forall b_1, b_2 : \text{base}) \left(b_1 =_{\text{obs}} b_2 \iff \left(\begin{array}{l} \text{tablas}(b_1) =_{\text{obs}} \text{tablas}(b_2) \wedge_L \\ (\forall t: \text{string}) (t \in \text{tablas}(b_1)) \Rightarrow_L \\ (\text{dameTabla}(t, b_1) =_{\text{obs}} \text{dameTabla}(t, b_2)) \wedge_L \\ (\forall t: \text{string})(\forall c: \text{campo}) (t \in \text{tablas}(b_1) \wedge c \in \\ \text{campos}(\text{dameTabla}(t, b_1))) \Rightarrow_L \\ (\text{tieneIndice?}(t, c, d_1) =_{\text{obs}} \text{tieneIndice?}(t, c, d_2)) \wedge \\ (\forall \text{crit}: \text{criterio}) \\ \text{usoCriterio}(\text{crit}, d_1) =_{\text{obs}} \text{usoCriterio}(\text{crit}, d_2)) \end{array} \right) \right)$$

observadores básicos

<code>tablas</code>	: <code>base db</code>	\longrightarrow <code>conj(string)</code>	
<code>dameTabla</code>	: <code>string t × base db</code>	\longrightarrow <code>tabla</code>	$\{t \in \text{tablas}(db)\}$
<code>usoCriterio</code>	: <code>Criterio c × base db</code>	\longrightarrow <code>Nat</code>	
<code>tieneIndice?</code>	: <code>string t × string c × base db</code>	\longrightarrow <code>bool</code>	$\{t \in \text{tablas}(db) \wedge c \in \text{campos}(\text{dameTabla}(t, db))\}$

generadores

<code>nuevaDB</code>	:	\longrightarrow <code>base</code>	
<code>agregarTabla</code>	: <code>tabla tbl × string t × base db</code>	\longrightarrow <code>base</code>	$\{\emptyset?(\text{registros}(tbl)) \wedge t \notin \text{tablas}(db)\}$
<code>insertarEntrada</code>	: <code>registro reg × string t × base db</code>	\longrightarrow <code>base</code>	$\{t \in \text{tablas}(db) \wedge_L \text{puedoInsertar?}(reg, \text{dameTabla}(t, db))\}$
<code>crearIndice</code>	: <code>string t × campo c × base db</code>	\longrightarrow <code>base</code>	$\{t \in \text{tabla}(db) \wedge c \in \text{campos}(\text{dameTabla}(t, db))\}$
<code>realizarBusqueda</code>	: <code>string t × criterio crit × base db</code>	\longrightarrow <code>base</code>	$\{\text{criterioValido}(crit, t, db)\}$

otras operaciones

<code>buscar</code>	: <code>crit criterio × string t × base db</code>	\longrightarrow <code>tabla</code>	$\{t \in \text{tablas}(db) \wedge \text{criterioValido}(crit, t, db)\}$
<code>criterioValido</code>	: <code>criterio crit × string t × base db</code>	\longrightarrow <code>bool</code>	$\{t \in \text{tablas}(db)\}$
<code>restriccionValida</code>	: <code>restriccion rest × string t × base db</code>	\longrightarrow <code>bool</code>	$\{t \in \text{tablas}(db)\}$
<code>join</code>	: <code>string t × string t2 × campo c × base db</code>	\longrightarrow <code>conj(registro)</code>	$\left\{ \begin{array}{l} t \in \text{tablas}(db) \wedge t2 \in \text{tablas}(db) \wedge c \in \text{campos}(\text{dameTabla}(t, db)) \wedge \\ c \in \text{campos}(\text{dameTabla}(t2, db)) \wedge (\text{tieneIndice?}(t, c, db) \vee \text{tieneIndice?}(t2, c, db)) \end{array} \right\}$
<code>generarJoin</code>	: <code>campo c × conj(registro) regs × conj(registro) regs2</code>	\longrightarrow <code>conj(registro)</code>	$\left\{ \begin{array}{l} \forall r: \text{conj(registro)} \ r \in \text{regs} \Rightarrow c \in \text{campos}(r) \wedge \\ \forall r2: \text{conj(registro)} \ r2 \in \text{regs2} \Rightarrow c \in \text{campos}(r2) \end{array} \right\}$
<code>generarRegistrosDelJoin</code>	: <code>campo c × registro reg × conj(registro) regs2</code>	\longrightarrow <code>conj(registro)</code>	$\{c \in \text{campos}(reg) \wedge \forall r: \text{conj(registro)} \ r \in \text{regs2} \Rightarrow c \in \text{campos}(r)\}$
<code>combinarRegistros</code>	: <code>conj(campo) cc × registro reg × registro reg2</code>	\longrightarrow <code>registro</code>	$\{cc \subseteq \text{campos}(reg)\}$
<code>criterios</code>	: <code>base db</code>	\longrightarrow <code>conj(criterio)</code>	
<code>criteriosMasUsados</code>	: <code>base db</code>	\longrightarrow <code>conj(criterio)</code>	
<code>mayorUsoDeCriterios</code>	: <code>conj(criterio) cc × base db</code>	\longrightarrow <code>Nat</code>	
<code>filtrarCriteriosUsadosNVeces</code>	: <code>conj(criterio) cc × Nat n × base db</code>	\longrightarrow <code>conj(criterio)</code>	
<code>filtrarRegistrosSegunCriterio</code>	: <code>criterio crit × conj(registro) regs</code>	\longrightarrow <code>conj(registro)</code>	$\left\{ \begin{array}{l} \forall \text{rest}: \text{restriccion} \ (\text{rest} \in \text{crit} \Rightarrow (\forall r: \text{registro} \ (r \in \text{regs} \Rightarrow (\text{campo}(\text{rest}) \in \text{campos}(r) \wedge \\ \text{tipo?}(\text{dato}(\text{rest})) == \text{tipo?}(\text{valor}(\text{campo}(\text{rest})), r)))) \end{array} \right\}$
<code>filtrarRegistrosSegunRestriccion</code>	: <code>restriccion rest × conj(registro) regs</code>	\longrightarrow <code>conj(registro)</code>	$\left\{ \begin{array}{l} \forall r: \text{registro} \ (r \in \text{regs} \Rightarrow (\text{campo}(\text{rest}) \in \text{campos}(r) \wedge \\ \text{tipo?}(\text{dato}(\text{rest})) == \text{tipo?}(\text{valor}(\text{campo}(\text{rest})), r))) \end{array} \right\}$

agregarTodosLosRegistros : tabla $tbl \times \text{conj}(\text{registro}) \text{ regs} \longrightarrow \text{tabla}$
 $\{\forall r: \text{registro} (r \in \text{regs} \Rightarrow \text{puedoInsertar?}(r, tbl))\}$

axiomas

($\forall db: \text{base}, \forall t, t2: \text{string}, \forall tbl: \text{tabla}, \forall reg: \text{registro}, \forall crit, crit2: \text{criterio}, \forall c, c2: \text{campo}, \forall r: \text{restriccion}, \forall \text{regs}, \text{regs2}: \text{conj}(\text{registro}), \forall cc: \text{conj}(\text{campo}), \forall n: \text{nat}$)

tablas(nuevaDB())	$\equiv \emptyset$
tablas(agregarTabla(tbl, t, db))	$\equiv \text{Ag}(t, \text{tablas}(db))$
tablas(insertarEntrada(reg, t, db))	$\equiv \text{tablas}(db)$
tablas(crearIndice(t, c, db))	$\equiv \text{tablas}(db)$
tablas(realizarBusqueda($t, crit, db$))	$\equiv \text{tablas}(db)$
dameTabla($t, \text{agregarTabla}(tbl, t2, db)$)	$\equiv \text{if } t == t2 \text{ then } tbl \text{ else dameTabla}(t, db) \text{ fi}$
dameTabla($t, \text{insertarEntrada}(reg, t2, db)$)	$\equiv \text{if } t == t2 \text{ then}$ $\text{agregarRegistro}(reg, \text{dameTabla}(t, db))$ else $\text{dameTabla}(t, db)$ fi
dameTabla($t, \text{crearIndice}(t2, c, db)$)	$\equiv \text{dameTabla}(t, db)$
dameTabla($t, \text{realizarBusqueda}(t2, crit, db)$)	$\equiv \text{dameTabla}(t, db)$
usoCriterio($crit, \text{nuevaDB}()$)	$\equiv 0$
usoCriterio($crit, \text{agregarTabla}(tbl, t2, db)$)	$\equiv \text{usoCriterio}(db)$
usoCriterio($crit, \text{insertarEntrada}(reg, t2, db)$)	$\equiv \text{usoCriterio}(db)$
usoCriterio($crit, \text{crearIndice}(t, c, db)$)	$\equiv \text{usoCriterio}(db)$
usoCriterio($crit, \text{realizarBusqueda}(t, crit2, db)$)	$\equiv \text{usoCriterio}(db) + \text{if } crit == crit2 \text{ then } 1 \text{ else } 0 \text{ fi}$
tieneIndice?($t, c, \text{agregarTabla}(tbl, t2, db)$)	$\equiv \text{if } t == t2 \text{ then false else tieneIndice?}(t, c, db) \text{ fi}$
tieneIndice?($t, c, \text{insertarEntrada}(reg, t2, db)$)	$\equiv \text{tieneIndice?}(t, c, db)$
tieneIndice?($t, c, \text{crearIndice}(t2, c2, db)$)	$\equiv (\text{if } t == t2 \wedge c == c2) \text{ then}$ true else $\text{tieneIndice?}(t, c, db)$ fi
tieneIndice?($t, c, \text{realizarBusqueda}(t2, crit, db)$)	$\equiv \text{tieneIndice?}(t, c, db)$
criterioValido($crit, t, db$)	$\equiv \emptyset?(crit) \vee_L$ $(\text{restriccionValida}(\text{dameUno}(crit), t, db) \wedge \text{criterioValido}(\text{sinUno}(crit), t, db))$
restriccionValida($rest, t, db$)	$\equiv \text{campo}(rest) \in \text{campos}(\text{dameTabla}(t, db)) \wedge_L$ $\text{tipo?}(\text{dato}(rest)) = \text{tipoCampo}(\text{campo}(rest), \text{dameTabla}(t, db))$
Join(t_1, t_2, c, db)	$\equiv \text{generarJoin}(c, \text{registros}(\text{dameTabla}(t_1, db)), \text{registros}(\text{dameTabla}(t_2, db)))$
generarJoin($c, \text{regs}, \text{regs2}$)	$\equiv \text{if } \emptyset?(\text{regs}) \text{ then}$ \emptyset else $\text{generarRegistrosDelJoin}(c, \text{dameUno}(\text{regs}), \text{regs2}) \cup$ $\text{generarJoin}(c, \text{sinUno}(\text{regs}), \text{regs2})$ fi

```

generarRegistrosDelJoin(c, reg, regs) ≡ if  $\emptyset?(regs)$  then
     $\emptyset$ 
else
    generarRegistrosDelJoin(c, reg, sinUno(regs))  $\cup$ 
    if valor(c, dameUno(regs)) == valor(c, reg) then
        combinarRegistros(campos(dameUno(regs)) – campos(reg),
            reg, dameUno(regs))
    else
         $\emptyset$ 
    fi
fi

combinarRegistros(cc, reg, reg2) ≡ if  $\emptyset?(cc)$  then
    reg
else
    combinarRegistros(sinUno(cc),
        definir(dameUno(cc), valor(dameUno(cc), reg2), reg), reg2)
fi

criterios(nuevaDB()) ≡  $\emptyset$ 
criterios(agregarTabla(tbl, t, db)) ≡ criterios(db)
criterios(insertarEntrada(reg, t, db)) ≡ criterios(db)
criterios(crearIndice(t, c, db)) ≡ criterios(db)
criterios(realizarBusqueda(t, crit, db)) ≡ ag(crit, criterios(db))

criteriosMasUsados(db) ≡ filtrarCriteriosUsadosNVeces(criterios(db),
    mayorUsoDeCriterios(criterios(db), db), db)

mayorUsoDeCriterios(crits, db) ≡ if  $\emptyset?(crits)$  then
    0
else
    if usoCriterio(dameUno(crits), db) >
        mayorUsoDeCriterios(sinUno(crits), db) then
        usoCriterio(dameUno(crits), db)
    else
        mayorUsoDeCriterios(sinUno(crits), db)
    fi
fi

filtrarCriteriosUsadosNVeces(crits, n, db) ≡ if  $\emptyset?(crits)$  then
     $\emptyset$ 
else
    if usoCriterio(dameUno(crits), db) == n then
        ag(dameUno(crits),
            filtrarCriteriosUsadosNVeces(sinUno(crits), n, db)
        )
    else
        filtrarCriteriosUsadosNVeces(sinUno(crits), n, db)
    fi
fi

buscar(crit,t,db) ≡ agregarTodosLosRegistros(
    nuevaTabla(claves(dameTabla(t, db)), estructura(dameTabla(t, db))),
    filtrarRegistrosSegunCriterio(crit, registros(dameTabla(t, db)))
)

```

```
filtrarRegistrosSegunCriterio(crit, regs)    ≡ if  $\emptyset?$ (crits) then  
                                         regs  
                                         else  
                                           filtrarRegistrosSegunRestriccion(dameUno(crit),  
                                           filtrarRegistrosSegunCriterio(sinUno(crit),regs) )  
                                         fi  
  
filtrarRegistrosSegunRestriccion(rest, regs) ≡ if porIgual(rest) then  
                                           filtrarRegistros(definir(campo(rest),dato(rest),vacío),  
                                           campo(rest), regs)  
                                           else  
                                             regs – filtrarRegistros(definir(campo(rest),dato(rest),vacío),  
                                             campo(rest), regs)  
                                           fi  
  
agregarTodosLosRegistros(tbl,regs)        ≡ if  $\emptyset?$ (regs) then  
                                         tbl  
                                         else  
                                           agregarRegistro(dameUno(regs),  
                                           agregarTodosLosRegistros(tbl,sinUno(regs)))  
                                         fi
```

Fin TAD