

Especificación “Lollapatuza”

Algoritmos y Estructura de Datos II - FCyEN - UBA

20 de mayo de 2023

1. Lollapatuza

TAD LOLLAPATUZA

généros lolla

observadores básicos

$$\text{puestos} : \text{lolla } l \longrightarrow \text{dicc}(\text{puestoid}, \text{puesto})$$
$$\text{personas} : \text{lolla } l \longrightarrow \text{conj}(\text{persona})$$

generadores

$$\begin{array}{ll} \text{crearLolla} & : \text{dicc}(\text{puetoid}, \text{puesto}) \text{ } ps \times \text{conj}(\text{persona}) \text{ } as \quad \longrightarrow \text{ lolla} \quad \{ \text{vendenAlMismoPrecio}(\text{significados}(ps)) \\ & \wedge \text{NoVendieronAun}(\text{significados}(ps)) \\ & \wedge \neg \emptyset?(as) \\ & \wedge \neg \emptyset?(claves(ps)) \} \end{array}$$
$$\text{vender} : \text{lolla } l \times \text{puestoid } pi \times \text{persona } a \times \text{item } i \times \text{cant } c \longrightarrow \text{lolla } \{a \in \text{personas}(l) \wedge \text{def?}(pi, \text{puestos}(l)) \wedge_L \text{haySuficiente?}(\text{obtener}(pi, \text{puestos}(l)), i, c)\}$$
$$\text{hackear} : \text{lolla } l \times \text{persona } a \times \text{item } i \longrightarrow \text{lolla } \{ \text{ConsumioSinPromoEnAlgunPuesto}(l, a, i) \}$$

otras operaciones

$$\text{masGasto} : \text{lolla } l \longrightarrow \text{persona } \{-\emptyset?(personas(l))\}$$
$$\text{gastoTotal} : \text{lolla } l \times \text{persona } a \rightarrow \text{dinero} \quad \{a \in \text{personas}(l)\}$$
$$\text{menorStock} : \text{lolla } l \times \text{item } i \longrightarrow \text{puestoid}$$

otras operaciones auxiliares

$$\text{gastoSobre} : \text{multiconj}(\text{puesto}) \text{ ps} \times \text{persona } a \longrightarrow \text{dinero}$$
$$\text{masGastoEntre} : \text{lolla } l \times \text{conj}(\text{persona}) \text{ } as \longrightarrow \langle \text{dinero}, \text{persona} \rangle$$

$$\{ \neg \emptyset?(as) \wedge as \subseteq \text{personas}(l) \}$$
$$\text{vendenAlMismoPrecio} : \text{multiconj}(\text{puesto}) \, ps \longrightarrow \text{bool}$$
$$\text{vendenMismoPrecioQue} : \text{multiconj}(\text{puesto}) \, ps \times \text{puesto} \, p \longrightarrow \text{bool}$$
$$\text{consumioSinPromoPuestos} : \text{persona } a \times \text{item } i \times \text{multiconi}(\text{puesto}) \text{ } ps \quad \longrightarrow \text{ bool}$$
$$\text{algunPuestoId} : \text{lolla } l \times \text{persona } a \times \text{item } i \longrightarrow \text{puestoid}$$
$$\{ConsumoSinPromoEnAlgunPuesto(l, a, i)\}$$
$$\text{algunPuestoEntre} : \text{persona } a \times \text{item } i \times \text{dicc}(\text{puestoid} \times \text{puesto}) \text{ } ps \longrightarrow \text{puestoid}$$
$$\{consumioSinPromoPuestos(a, i, significados(ps))\}$$
$$\text{menorStockEntre} : \text{item } i \times \text{dicc}(\text{puestoid} \times \text{puesto}) \text{ } ps \longrightarrow \text{puestoid} \quad \{\neg \emptyset?(claves(ps))\}$$

axiomas

$$(\forall ps: \text{dicc}(\text{puestoid}, \text{puesto}), \forall as: \text{conj}(\text{persona}), \forall l: \text{lolla}, \forall p: \text{puesto}, \forall a: \text{persona}, \forall i: \text{item}, \forall v: \text{multiconj}(\text{item}))$$
$$\text{puestos}(\text{crearLolla}(ps, as)) \equiv ps$$
$$\text{personas}(\text{crearLolla}(ps, as)) \equiv as$$
$$\text{puestos}(\text{vender}(l, pi, a, i, c)) \equiv \text{definir}(pi, \text{vender}(\text{obtener}(pi, \text{puestos}(l)), a, i, c), \text{puestos}(l))$$
$$\text{personas}(\text{vender}(l, pi, a, i, c)) \equiv \text{personas}(l)$$
$$\text{gastoTotal}(l, a) \equiv \text{gastoSobre}(\text{significados}(\text{puestos}(l)), a)$$
$$\text{gastoSobre}(ps, a) \equiv \text{if } \emptyset?(ps) \text{ then } 0 \text{ else } \text{gastosDe}(\text{dameUno}(ps), a) + \text{gastoSobre}(\text{sinUno}(ps), a) \text{ fi}$$
$$\text{masGasto}(l) \equiv \pi_2(\text{masGastoEntre}(l, \text{personas}(l)))$$

```

masGastoEntre( $l, as$ )  $\equiv$  if  $\#(as) = 1 \vee_L \text{gastoTotal}(l, \text{dameUno}(as)) > \pi_1(\text{masGastoEntre}(l, \text{sinUno}(as)))$  then
     $\langle \text{gastoTotal}(l, \text{dameUno}(as)), \text{dameUno}(as) \rangle$ 
else
    if  $\text{gastoTotal}(l, \text{dameUno}(as)) = \pi_1(\text{masGastoEntre}(l, \text{sinUno}(as)))$  then
         $\text{dameUno}(\{ \langle \text{gastoTotal}(l, \text{dameUno}(as)), \text{dameUno}(as) \rangle, \text{masGastoEntre}(l, \text{sinUno}(as)) \})$ 
    else
         $\text{masGastoEntre}(l, \text{sinUno}(as))$ 
    fi
fi

personas( $\text{hackear}(l, a, i)$ )  $\equiv$   $\text{personas}(l)$ 
puestos( $\text{hackear}(l, a, i)$ )  $\equiv$   $\text{definir}(\text{algunPuestoId}(l, a, i),$ 
     $\text{olvidarItem}(\text{obtener}(\text{algunPuestoId}(l, a, i), \text{puestos}(l)), a, v),$ 
     $\text{puestos}(l))$ 
algunPuestoId( $l, a, i$ )  $\equiv$   $\text{algunPuestoEntre}(a, i, \text{puestos}(l))$ 
algunPuestoEntre( $a, i, ps$ )  $\equiv$  if  $i \in \text{menu}(\text{obtenerUno}(ps)) \wedge_L \text{consumioSinPromo}?( \text{obtenerUno}(ps), a, i)$  then
     $\text{dameUno}(\text{claves}(ps))$ 
else
     $\text{algunPuestoEntre}(a, i, \text{sinUnaClave}(ps))$ 
fi
vendenAlMismoPrecio( $ps$ )  $\equiv$  if  $\emptyset?(ps)$  then
    true
else
     $\text{vendenMismoPrecioQue}(ps, \text{dameUno}(ps))$ 
     $\wedge \text{vendenAlMismoPrecio}(\text{sinUno}(ps))$ 
fi
vendenMismoPrecioQue( $ps, p$ )  $\equiv$  if  $\emptyset?(ps)$  then
    true
else
     $\text{vendenMismoPrecio}(\text{obtenerUno}(ps), p)$ 
     $\wedge \text{vendenMismoPrecioQue}(\text{sinUno}(ps), p)$ 
fi
consumioSinPromoPuestos( $a, i, ps$ )  $\equiv$  if  $\emptyset?(\text{claves}(ps))$  then
    false
else
     $(i \in \text{menu}(\text{obtenerUno}(ps)) \wedge_L \text{consumioSinPromo}?( \text{obtenerUno}(ps), a, i))$ 
     $\vee_L \text{consumioSinPromoPuestos}(\text{sinUnaClave}(a, i, ps))$ 
fi
menorStock( $l, i$ )  $\equiv$   $\text{menorStockEntre}(i, \text{puestos}(l))$ 
menorStockEntre( $i, ps$ )  $\equiv$  if  $\#(\text{claves}(ps)) = 1 \vee_L (i \in \text{menu}(\text{obtenerUno}(ps)) \wedge_L$ 
     $\text{stock}(\text{obtenerUno}(ps), i) < \text{stock}(\text{obtener}(\text{menorStockEntre}(i, \text{sinUnaClave}(ps)), ps), i))$ 
then
     $\text{dameUno}(\text{claves}(ps))$ 
else
     $\text{menorStockEntre}(i, \text{sinUnaClave}(ps))$ 
fi

```

Fin TAD

1.1. Predicados Auxiliares

$\text{NoVendieronAun}(ps) \equiv (\forall a : \text{persona})(\forall p : \text{puesto})(p \in \text{significados}(ps) \Rightarrow_L \emptyset?(\text{ventas}(p, a)))$

$\text{ConsumioSinPromoEnAlgunPuesto}(l, a, i) \equiv a \in \text{personas}(l)$
 $\wedge_L (\exists pi : \text{puestoid}) (\text{def}?(pi, \text{puestos}(l)) \wedge_L \text{consumioSinPromo}?(\text{obtener}(pi,$
 $\text{puestos}(l)), a, i))$

2. PuestoDeComida

TAD PUESTODeCOMIDA

géneros puesto

observadores básicos

menu	: puesto p	\longrightarrow conj(item)	
precio	: puesto $p \times$ item i	\longrightarrow dinero	$\{i \in \text{menu}(p)\}$
stock	: puesto $p \times$ item i	\longrightarrow cant	$\{i \in \text{menu}(p)\}$
descuento	: puesto $p \times$ item $i \times$ cant c	\longrightarrow nat	$\{i \in \text{menu}(p)\}$
ventas	: puesto $p \times$ persona a	\longrightarrow multiconj(\langle item, cant \rangle)	

generadores

crearPuesto	: dicc(item, nat) $p \times$ dicc(item, nat) $s \times$ dicc(item, dicc(cant, nat)) d	\longrightarrow puesto	$\{\text{claves}(p) = \text{claves}(s) \wedge \text{claves}(d) \subseteq \text{claves}(p)\}$
vender	: puesto $p \times$ persona $a \times$ item $i \times$ cant c	\longrightarrow puesto	$\{\text{haySuficiente?}(p, i, c)\}$

otras operaciones

haySuficiente?	: puesto $p \times$ item $i \times$ cant c	\longrightarrow bool	$\{i \in \text{menu}(p)\}$
consumioSinPromo?	: puesto $p \times$ persona $a \times$ item i	\longrightarrow bool	$\{i \in \text{menu}(p)\}$
gastosDe	: puesto $p \times$ persona a	\longrightarrow dinero	
olvidarItem	: puesto $p \times$ persona $a \times$ item i	\longrightarrow puesto	$\{i \in \text{menu}(p) \wedge_L \text{consumioSinPromo?}(p, a, i)\}$
vendenMismoPrecio	: puesto \times puesto	\longrightarrow bool	

otras operaciones auxiliares

consumioSinPromoVentas?	: puesto $p \times$ item $i \times$ multiconj(\langle item, cant \rangle) m	\longrightarrow bool	$\{i \in \text{menu}(p) \wedge \text{SonVentasDelMenu}(m, p)\}$
consumioSinPromo1Venta?	: puesto $p \times$ item $i \times \langle$ item, cant \rangle v	\longrightarrow bool	$\{i \in \text{menu}(p) \wedge \pi_1(v) \in \text{menu}(p)\}$
ventasSinPromo	: puesto $p \times$ persona $a \times$ item i	\longrightarrow multiconj(\langle item, cant \rangle)	$\{i \in \text{menu}(p)\}$
ventasSinPromoEntre	: puesto $p \times$ item $i \times$ multiconj(\langle item, cant \rangle) m	\longrightarrow multiconj(\langle item, cant \rangle)	$\{i \in \text{menu}(p) \wedge \text{SonVentasDelMenu}(m, p)\}$
olvidarItemDeVenta	: puesto $p \times$ persona $a \times$ item $i \times$ multiconj(\langle item, cant \rangle) v	\longrightarrow puesto	$\{i \in \text{menu}(p) \wedge \pi_1(v) \in \text{menu}(p) \wedge_L \text{consumioSinPromo1Venta?}(p, a, i, v)\}$
gastosDeVentas	: puesto $p \times$ multiconj(\langle item, cant \rangle) m	\longrightarrow dinero	$\{\text{SonVentasDelMenu}(m, p)\}$
gastosDe1Venta	: puesto $p \times \langle$ item, cant \rangle v	\longrightarrow dinero	$\{\pi_1(v) \in \text{menu}(p)\}$
vendenMismoPrecioMenu	: puesto $p1 \times$ puesto $p2 \times$ conj(item) is	\longrightarrow bool	$\{is \subseteq \text{menu}(p1) \cap \text{menu}(p2)\}$

axiomas

$(\forall p, p1, p2: \text{puesto}, \forall a, a1, a2: \text{persona}, \forall c: \text{cant}, \forall i: \text{item}, \forall v, v1, v2: \langle \text{item}, \text{cant} \rangle, \forall m: \text{multiconj}(\langle \text{item}, \text{cant} \rangle), \forall is: \text{conj}(\text{item}), \forall s: \text{dicc}(\text{item}, \text{nat}), \forall d: \text{dicc}(\text{item}, \text{dicc}(\text{cant}, \text{nat})))$

menu(crearPuesto(p, s, d)) \equiv claves(p)

precio(crearPuesto(p, s, d), i) \equiv obtener(i, p)

stock(crearPuesto(p, s, d), i) \equiv obtener(i, s)

descuento(crearPuesto(p, s, d), i, c) \equiv **if** $c = 0$ **then**

0

else

if $\neg \text{def?}(i, d)$ **then**

0

else

if $\text{def?}(c, \text{obtener}(i, d))$ **then**

obtener($c, \text{obtener}(i, d)$)

else

descuento(crearPuesto(p, s, d), $i, c - 1$)

fi

fi

fi

ventas(crearPuesto(p, s, d), a) $\equiv \emptyset$

```

menu(vender(p, a, i, c)) ≡ menu(p)
precio(vender(p, a, i1, c), i2) ≡ precio(p, i2)
stock(vender(p, a, i1, c), i2) ≡ if i1 = i2 then stock(p, i2) - c else stock(p, i2) fi
descuento(vender(p, a, i1, c1), i2, c2) ≡ descuento(p, i2, c2)
ventas(vender(p, a, i, c), a2) ≡ if a1 = a2 then Ag(⟨i, c⟩, ventas(p, a2)) else ventas(p, a2) fi
haySuficiente?(p, i, c) ≡ stock(p, i) ≥ c
gastosDe(p, a) ≡ gastosDeVentas(p, ventas(p, a))
gastosDeVentas(p, m) ≡ if ∅?(m) then 0 else gastosDeVentas(p, sinUno(m)) + gastosDe1Venta(p, dameUno(m)) fi
gastosDe1Venta(p, v) ≡ aplicarDescuento(π2(v) × precio(p, π1(v)), descuento(p, π1(v), π2(v)))
consumioSinPromo?(p, a, i) ≡ consumioSinPromoVentas?(p, i, ventas(p, a))
consumioSinPromoVentas?(p, i, m) ≡ if ∅?(m) then
    false
  else
    consumoSinPromo1Venta?(p, i, dameUno(m)) ∨
    consumoSinPromoVentas?(p, i, sinUno(m))
  fi
consumioSinPromo1Venta?(p, i, m) ≡ if descuento(p, π1(v), π2(v)) = 0 then true else false fi
ventasSinPromoEntre(p, i, m) ≡ if ∅?(m) then
    ∅
  else
    if consumioSinPromo1Venta?(p, i, dameUno(m)) then
      Ag(dameUno(m), ventasSinPromoEntre(p, i, sinUno(m)))
    else
      ventasSinPromoEntre(p, i, sinUno(m))
    fi
  fi
ventasSinPromo(p, a, i) ≡ ventasSinPromoEntre(p, i, ventas(p, a))
olvidarItemDeVenta(vender(p, a1, i1, c), a2, i2, v2) ≡ if a1 = a2 ∧ ⟨i1, c⟩ = v2 then
    if c = 1 then p else vender(p, a1, c - 1) fi
  else
    vender(olvidarItemDeVenta(p, a2, i, v2), a1, i1, c)
  fi
olvidarItem(p, a, i) ≡ olvidarItemDeVenta(p, a, i, dameUno(ventasSinPromo(p, a, i)))
vendenMismoPrecio(p1, p2) ≡ vendenMismoPrecioMenu(p1, p2, menu(p1) ∩ menu(p2))
vendenMismoPrecioMenu(p1, p2, is) ≡ if ∅?(is) then
    true
  else
    precio(p1, dameUno(is)) = precio(p2, dameUno(is))
    ∧ vendenMismoPrecioMenu(p1, p2, sinUno(is))
  fi

```

Fin TAD

2.1. Predicados Auxiliares

$\text{SonVentasDelMenu}(m, p) \equiv (\forall v : \langle \text{item}, \text{cant} \rangle)(v \in m \Rightarrow_L (\forall i' : \text{item}) \pi_1(v) \in \text{menu}(p))$

3. Dinero

TAD DINERO

géneros dinero

extiende NAT

otras operaciones

div : dinero $n \times \text{nat } k \rightarrow \text{dinero}$

aplicarDescuento : dinero $p \times \text{nat } d \rightarrow \text{dinero}$

$\{0 < k\}$
 $\{d < 100\}$

axiomas

$(\forall n: \text{dinero}, \forall k: \text{nat})$

$\text{div}(n, k) \equiv \text{if } n < k \text{ then } 0 \text{ else } 1 + \text{div}(n - k, k) \text{ fi}$
 $\text{aplicarDescuento}(p, d) \equiv \text{div}(p \times (100 - d), 100)$

Fin TAD

4. Otros TADs

El TAD ITEM es renombre de NAT con género *item*.

El TAD PERSONA es renombre de NAT con género *persona*. Representa el DNI de una persona.

El TAD CANTIDAD es renombre de NAT con género *cant*.

El TAD PUESTOID es renombre de NAT con género *puetoid*.

Extendemos TAD DICCIONARIO con las siguientes operaciones:

$\text{sinUnaClave} : \text{dicc}(\alpha, \beta) \ d \longrightarrow \text{dicc}(\alpha, \beta)$

$\{\neg \emptyset?(claves(d))\}$

$\text{obtenerUno} : \text{dicc}(\alpha, \beta) \ d \longrightarrow \beta$

$\{\neg \emptyset?(claves(d))\}$

$\text{significados} : \text{dicc}(\alpha, \beta) \longrightarrow \text{multiconj}(\beta)$

axiomas

$\text{sinUnaClave}(d) \equiv \text{borrar}(\text{dameUno}(claves(d)), d)$

$\text{obtenerUno}(d) \equiv \text{obtener}(\text{dameUno}(claves(d)), d)$

$\text{significados}(d) \equiv \text{if } \emptyset?(claves(d)) \text{ then } \emptyset \text{ else } \text{Ag}(\text{obtenerUno}(d), \text{significados}(\text{sinUnaClave}(d))) \text{ fi}$