

# Trabajo Practico I, Alta Seguridad nos cuida

Algoritmos y Estructuras de Datos II, DC, UBA.

## Índice

<b>1. TAD UNIVERSIDAD</b>	<b>2</b>
<b>2. TAD AGENTE</b>	<b>11</b>
<b>3. TAD POSICION</b>	<b>12</b>
<b>4. TAD DIRECCION</b>	<b>14</b>
<b>5. TAD SITUACIONESRODEO</b>	<b>14</b>
<b>6. TAD TIPOELEM</b>	<b>15</b>

# 1. TAD UNIVERSIDAD

## TAD UNIVERSIDAD

**géneros** uni

**exporta** uni, Generadores, Observadores Basicos

**usa** NAT, CONJU( $\alpha$ ), BOOL, TUPLA( $\alpha_1, \dots, \alpha_n$ )

### igualdad observacional

$$(\forall u, u' : \text{uni}) \left( u =_{\text{obs}} u' \iff \left( \begin{array}{l} \text{alto?}(u) =_{\text{obs}} \text{alto?}(u') \wedge \text{ancho?}(u) =_{\text{obs}} \text{ancho?}(u') \wedge \text{obstaculos?}(u) \\ =_{\text{obs}} \text{obstaculos?}(u') \wedge \text{agentes?}(u) =_{\text{obs}} \text{agentes?}(u') \wedge \text{estudian-} \\ \text{tes?}(u) =_{\text{obs}} \text{estudiantes?}(u') \wedge \text{hippies?}(u) =_{\text{obs}} \text{hippies?}(u') \end{array} \right) \right)$$

*Los Agentes son manejados como triplas que tienen al Id, al Agente en cuestion, y su posicion. Los estudiantes son triplas con Id, Estudiante y Posicion. Los Hippies son duplas de Id y Posicion.*

*Los Estudiantes son representados como una secuencia de direcciones en las cuales moverse. Si en algun momento su camino esta bloqueado, no se mueven, ya que consideramos que lo interesante se encuentra en el camino que debe recorrer.*

### observadores básicos

alto? : uni  $\rightarrow$  nat

ancho? : uni  $\rightarrow$  nat

obstaculos? : uni  $\rightarrow$  conj(pos)

agentes? : uni  $\rightarrow$  conj( $\langle id, as, pos \rangle$ )

hippies? : uni  $\rightarrow$  conj( $\langle id, pos \rangle$ )

estudiantes? : uni  $\rightarrow$  conj( $\langle id, est, pos \rangle$ )

### generadores

TAD id ES Nat TAD hip ES Id TAD est ES Secu(Dir)

nuevaUni : conj( $\langle as, pos \rangle$ )  $\times$  nat  $\times$  nat  $\times$  conj(pos)  $\rightarrow$  uni

agregarE : uni  $\times$  id  $\times$  est  $\times$  pos  $\rightarrow$  uni

agregarH : uni  $\times$  hip  $\times$  pos  $\rightarrow$  uni

moverAS : uni  $\times$  id  $\rightarrow$  uni

moverH : uni  $\times$  id  $\rightarrow$  uni

moverE : uni  $\times$  id  $\rightarrow$  uni

### otras operaciones

capturar : pos  $\times$  conj( $\langle id, agente, pos \rangle$ )  $\rightarrow$  conj( $\langle agente, pos \rangle$ )

sancionar : pos  $\times$  conj( $\langle id, agente, pos \rangle$ )  $\rightarrow$  conj( $\langle agente, pos \rangle$ )

queTipoHay : pos  $\times$   $\langle cAs, cH, cEst \rangle \rightarrow$  tipo

cuatroVecinosShort : pos  $\times$   $\langle cAs, cH, cEst \rangle \times$  uni  $\rightarrow$  multiconj(tipo)

cuatroVecinos : pos  $\times$   $\langle cAs, cH, cEst \rangle \times$  uni  $\times$  conj(pos)  $\rightarrow$  multiconj(tipo)

queSituacion : multiconj(tipo)  $\rightarrow$  conj(situ)

damePosicionesAs : conj( $\langle id, agente, pos \rangle$ )  $\rightarrow$  conj(pos)

damePosicionesEst : conj( $\langle id, agente, pos \rangle$ )  $\rightarrow$  conj(pos)

damePosicionesH : conj( $\langle id, pos \rangle$ )  $\rightarrow$  conj(pos)

moverEstudianteYChequearSituaciones : id  $is \times$  uni  $u \rightarrow \langle cAs, cH, cEst \rangle$   
 $\{\exists t \in \text{estudiantes?}(u) / \Pi_0(t) = is\}$

dameEstudiante : id  $is \times$  conj( $\langle i, est, pos \rangle$ ) conjEst  $\rightarrow$  est  
 $\{\exists t \in \text{conjEst} / \Pi_0(t) = is\}$

dameEstudiantePos : id  $is \times$  conj( $\langle i, est, pos \rangle$ ) conjEst  $\rightarrow$  pos  
 $\{\exists t \in \text{conjEst} / \Pi_0(t) = is\}$

$\text{sacarEstudianteId} : \text{id } is \times \text{conj}(\langle i, \text{est}, \text{pos} \rangle) \text{ conjEst} \longrightarrow \text{conj}(\langle i, \text{est}, \text{pos} \rangle) \quad \{\exists t \in \text{conjEst} / \Pi_0(t) = is\}$   
 $\text{entradas?} : \text{nat} \times \text{nat} \longrightarrow \text{conj}(\text{pos})$   
 $\text{dirLibres} : \text{pos} \times \text{uni} \longrightarrow \text{conj}(\text{dir})$   
 $\text{dirValidas} : \text{pos} \times \text{nat} \times \text{nat} \longrightarrow \text{conj}(\text{dir})$   
 $\text{dirNoOcupadas} : \text{conj}(\text{dir}) \times \text{pos} \times \text{uni} \longrightarrow \text{conj}(\text{dir})$   
 $\quad \{\neg(\text{col?}(\text{pos}) = 0 \wedge \{o\} \in \text{cDirs}) \wedge \neg(\text{fila?}(\text{pos}) = 0 \text{ y } \{n\} \in \text{cDirs})\}$   
 $\text{posNoOcupadas} : \text{conj}(\text{pos}) \times \text{uni} \longrightarrow \text{conj}(\text{pos})$   
 $\text{moverAgenteYChequearSituacion} : \text{id } is \times \text{uni } u \longrightarrow \langle \text{cAs}, \text{cH}, \text{cEst} \rangle \quad \{\exists t \in \text{agentes?}(u) / \Pi_0(t) = is\}$   
 $\text{posibleMovAs} : \text{As} \times \text{pos} \times \text{uni} \longrightarrow \text{conj}(\text{dir})$   
 $\text{dameAgente} : \text{id } is \times \text{conj}(\langle i, \text{as}, \text{pos} \rangle) \text{ conjAs} \longrightarrow \text{as} \quad \{\exists t \in \text{conjAs} / \Pi_0(t) = is\}$   
 $\text{dameAgentePos} : \text{id } is \times \text{conj}(\langle i, \text{as}, \text{pos} \rangle) \text{ conjAs} \longrightarrow \text{pos} \quad \{\exists t \in \text{conjAs} / \Pi_0(t) = is\}$   
 $\text{sacarAgenteId} : \text{id } is \times \text{conj}(\langle i, \text{as}, \text{pos} \rangle) \text{ conjAs} \longrightarrow \text{conj}(\langle i, \text{as}, \text{pos} \rangle) \quad \{\exists t \in \text{conjAs} / \Pi_0(t) = is\}$   
 $\text{moverHippieYChequearSituacion} : \text{id } is \times \text{uni} \times u \longrightarrow \langle \text{cAs}, \text{cH}, \text{cEst} \rangle \quad \{\exists t \in \text{hippies?}(u) / \Pi_0(t) = is\}$   
 $\text{posibleMovH} : \text{pos} \times \text{uni} \longrightarrow \text{conj}(\text{dir})$   
 $\text{dameHippiePos} : \text{id } is \times \text{conj}(\langle i, \text{pos} \rangle) \text{ conjHip} \longrightarrow \text{pos} \quad \{\exists t \in \text{conjHip} / \Pi_0(t) = is\}$   
 $\text{sacarHippieId} : \text{id } is \times \text{conj}(\langle i, \text{pos} \rangle) \text{ conjHip} \longrightarrow \text{conj}(\langle i, \text{pos} \rangle) \quad \{\exists t \in \text{conjHip} / \Pi_0(t) = is\}$   
 $\text{chequearSituacionShort} : \langle \text{conj}(\langle i, \text{as}, \text{pos} \rangle), \text{conj}(\langle i, \text{pos} \rangle), \text{conj}(\langle i, \text{est}, \text{pos} \rangle) \rangle \times \text{uni} \longrightarrow$   
 $\quad \langle \text{conj}(\langle i, \text{as}, \text{pos} \rangle), \text{conj}(\langle i, \text{pos} \rangle), \text{conj}(\langle i, \text{est}, \text{pos} \rangle) \rangle$   
 $\text{chequearSituacion} : \text{MACRO1} \times \text{MACRO1} \times \text{uni} \longrightarrow \text{MACRO1}$   
 $\quad \text{MACRO1} = \langle \text{conj}(\langle i, \text{as}, \text{pos} \rangle), \text{conj}(\langle i, \text{pos} \rangle), \text{conj}(\langle i, \text{est}, \text{pos} \rangle) \rangle$   
 $\text{agAgenteTripla} : \langle i, \text{as}, \text{pos} \rangle \times \langle \text{conj}(\langle i, \text{as}, \text{pos} \rangle), \text{conj}(\langle i, \text{pos} \rangle), \text{conj}(\langle i, \text{est}, \text{pos} \rangle) \rangle \longrightarrow$   
 $\quad \langle \text{conj}(\langle i, \text{as}, \text{pos} \rangle), \text{conj}(\langle i, \text{pos} \rangle), \text{conj}(\langle i, \text{est}, \text{pos} \rangle) \rangle$   
 $\text{agHippieTripla} : \langle i, \text{pos} \rangle \times \langle \text{conj}(\langle i, \text{as}, \text{pos} \rangle), \text{conj}(\langle i, \text{pos} \rangle), \text{conj}(\langle i, \text{est}, \text{pos} \rangle) \rangle \longrightarrow$   
 $\quad \langle \text{conj}(\langle i, \text{as}, \text{pos} \rangle), \text{conj}(\langle i, \text{pos} \rangle), \text{conj}(\langle i, \text{est}, \text{pos} \rangle) \rangle$   
 $\text{agEstudianteTripla} : \langle i, \text{est}, \text{pos} \rangle \times \langle \text{conj}(\langle i, \text{as}, \text{pos} \rangle), \text{conj}(\langle i, \text{pos} \rangle), \text{conj}(\langle i, \text{est}, \text{pos} \rangle) \rangle \longrightarrow$   
 $\quad \langle \text{conj}(\langle i, \text{as}, \text{pos} \rangle), \text{conj}(\langle i, \text{pos} \rangle), \text{conj}(\langle i, \text{est}, \text{pos} \rangle) \rangle$   
 $\text{estudiantesAdyacentesPos} : \text{pos} \times \text{conj}(\langle i, \text{est}, \text{pos} \rangle) \longrightarrow \text{conj}(\text{est})$   
 $\text{convertirHippieAEst} : \langle \text{id}, \text{pos} \rangle \times \text{conj}(\langle \text{id}, \text{est}, \text{pos} \rangle) \longrightarrow \langle \text{id}, \text{est}, \text{pos} \rangle$   
 $\text{convertirEstAHippie} : \langle \text{id}, \text{est}, \text{pos} \rangle \longrightarrow \langle \text{id}, \text{pos} \rangle$   
 $\text{sacarHippieTripla} : \langle \text{id}, \text{pos} \rangle \times \langle \text{conj}(\langle i, \text{as}, \text{pos} \rangle), \text{conj}(\langle i, \text{pos} \rangle), \text{conj}(\langle i, \text{est}, \text{pos} \rangle) \rangle \longrightarrow$   
 $\quad \langle \text{conj}(\langle i, \text{as}, \text{pos} \rangle), \text{conj}(\langle i, \text{pos} \rangle), \text{conj}(\langle i, \text{est}, \text{pos} \rangle) \rangle$   
 $\text{dondeEstaAgente} : \text{id } is \times \text{uni } u \longrightarrow \text{pos} \quad \{\exists t \in \text{agentes?}(u) / \Pi_0(t) = is\}$   
 $\text{dondeEstaHippie} : \text{id } is \times \text{uni } u \longrightarrow \text{pos} \quad \{\exists t \in \text{hippies?}(u) / \Pi_0(t) = is\}$   
 $\text{dondeEstaEstudiante} : \text{id } is \times \text{uni } u \longrightarrow \text{pos} \quad \{\exists t \in \text{estudiantes?}(u) / \Pi_0(t) = is\}$   
 $\text{masVigilante} : \text{uni} \longrightarrow \langle \text{id}, \text{as}, \text{pos} \rangle$   
 $\text{losMasVigilantes} : \text{conj}(\langle \text{id}, \text{as}, \text{pos} \rangle) \times \text{nat} \longrightarrow \text{conj}(\langle \text{id}, \text{as}, \text{pos} \rangle)$   
 $\text{losMasVigilantesShort} : \text{conj}(\langle \text{id}, \text{as}, \text{pos} \rangle) \longrightarrow \text{conj}(\langle \text{id}, \text{as}, \text{pos} \rangle)$   
 $\text{agenteConMenorPlaca} : \text{conj}(\langle \text{id}, \text{as}, \text{pos} \rangle) \times \text{nat} \longrightarrow \langle \text{id}, \text{as}, \text{pos} \rangle$   
 $\text{agenteConMenorPlacaShort} : \text{conj}(\langle \text{id}, \text{as}, \text{pos} \rangle) \longrightarrow \langle \text{id}, \text{as}, \text{pos} \rangle$   
 $\text{maxCapturas} : \text{conj}(\langle \text{id}, \text{as}, \text{pos} \rangle) \longrightarrow \text{nat}$   
 $\text{minPlaca} : \text{conj}(\langle \text{id}, \text{as}, \text{pos} \rangle) \longrightarrow \text{nat}$   
 $\text{cuantosHippies} : \text{uni} \longrightarrow \text{nat}$

cuantosEstudiantes : uni  $\rightarrow$  nat

**axiomas**  $\forall cAs: \text{conj}(<\text{id}, \text{as}, \text{pos}>) \forall cH: \text{conj}(<\text{id}, \text{pos}>) \forall cEst: \text{conj}(<\text{id}, \text{est}, \text{pos}>) \forall cDirs: \text{conj}(\text{dir})$   
 $\forall al, an, maxC, minN: \text{nat} \forall cObs: \text{conj}(\text{pos}) \forall u: \text{uni} \forall i: \text{id} \forall e: <\text{id}, \text{est}, \text{pos}> \forall agente: <\text{id}, \text{as},$   
 $\text{pos}> \forall h: <\text{id}, \text{pos}> \forall p: \text{pos} \forall p4v: \text{conj}(\text{pos}) \forall tripla, triplaInfo: <\text{conj}(<\text{id}, \text{as}, \text{pos}>), cH \text{ conj}(<\text{id},$   
 $\text{pos}>), \text{conj}(<\text{id}, \text{est}, \text{pos}>)> \forall mcT: \text{multiconj}(\text{tipo})$

Observadores Basicos

alto? (nuevaUni(cAs, al, an, cObs))  $\equiv$   
alto?(u)

alto? (agregarE(u, i, e, p))  $\equiv$   
alto?(u)

alto? (agregarH(u, i, p))  $\equiv$   
alto?(u)

alto? (moverAs(u, i))  $\equiv$   
alto?(u)

alto? (moverH(u, i))  $\equiv$   
alto?(u)

alto? (moverE(u, i))  $\equiv$   
alto?(u)

ancho? (nuevaUni(cAs, al, an, cObs))  $\equiv$   
an

ancho? (agregarE(u, i, e, p))  $\equiv$   
ancho?(u)

ancho? (agregarH(u, i, p))  $\equiv$   
ancho?(u)

ancho? (moverAs(u, i))  $\equiv$   
ancho?(u)

ancho? (moverH(u, i))  $\equiv$   
ancho?(u)

ancho? (moverE(u, i))  $\equiv$   
ancho?(u)

obstaculos? (nuevaUni(cAs, al, an, cObs))  $\equiv$   
cObs

obstaculos? (agregarE(u, i, e, p))  $\equiv$   
obstaculos?(u)

obstaculos? (agregarH(u, i, p))  $\equiv$   
obstaculos?(u)

obstaculos? (moverAs(u, i))  $\equiv$   
obstaculos?(u)

obstaculos? (moverH(u, i))  $\equiv$   
obstaculos?(u)

obstaculos? (moverE(u, i))  $\equiv$   
obstaculos?(u)

agentes? (nuevaUni(cAs, al, an, cObs))  $\equiv$   
cAs

agentes? (agregarE(u, i, e, p))  $\equiv$   
 $\Pi_1$ ( chequearSituacionShort( agregarEstudianteTripla(  $\langle i, e, p \rangle$ ,  
 $\langle$  agentes?(u), hippies?(u), estudiantes?(u)  $\rangle \rangle$  ) ) )

agentes? (agregarH(u, i, pos))  $\equiv$   
 $\Pi_1$ ( chequearSituacionShort( agregarHippieTripla(  $\langle i, p \rangle$ ,  
 $\langle$  agentes?(u), hippies?(u), estudiantes?(u)  $\rangle \rangle$  ) ) )

agentes? (moverAs(u, i))  $\equiv$   
 $\Pi_1$ ( moverAgenteYSancionarYCapturar(i, u) )

agentes? (moverE(u, i))  $\equiv$   
 $\Pi_1$ ( moverEstudiante(i, u) )

agentes? (moverH(u, i))  $\equiv$   
 $\Pi_1$ ( moverHippie(i, u) )

hippies? (nuevaUni(cAs, al, an, cObs))  $\equiv$   
 $\emptyset$

hippies? (agregarE(u, i, e, p))  $\equiv$   
 $\Pi_2$ ( chequearSituacionShort( agregarEstudianteTripla(  $\langle i, e, p \rangle$ ,  
 $\langle$  agentes?(u), hippies?(u), estudiantes?(u)  $\rangle \rangle$  ) ) )

hippies? (agregarH(u, i, p))  $\equiv$   
 $\Pi_2$ ( chequearSituacionShort( agregarHippieTripla(  $\langle i, p \rangle$ ,  
 $\langle$  agentes?(u), hippies?(u), estudiantes?(u)  $\rangle \rangle$  ) ) )

hippies? (moverAs(u, i))  $\equiv$   
 $\Pi_2$ ( moverAgenteYSancionarYCapturar(i, u) )

hippies? (moverE(u, i))  $\equiv$   
 $\Pi_2$ ( moverEstudiante(i, u) )

hippies? (moverH(u, i))  $\equiv$   
 $\Pi_2$ ( moverHippie(i, u) )

estudiantes? (nuevaUni(cAs, al, an, cObs))  $\equiv$   
 $\emptyset$

estudiantes? (agregarE(u, i, e, p))  $\equiv$   
 $\Pi_3$ ( chequearSituacionShort( agregarEstudianteTripla(  $\langle i, e, p \rangle$ ,  
 $\langle$  agentes?(u), hippies?(u), estudiantes?(u)  $\rangle \rangle$  ) ) )

estudiantes? (agregarH(u, i, p))  $\equiv$   
 $\Pi_3$ ( chequearSituacionShort( agregarHippieTripla(  $\langle i, p \rangle$ ,  
 $\langle$  agentes?(u), hippies?(u), estudiantes?(u)  $\rangle \rangle$  ) ) )

estudiantes? (moverAs(u, i))  $\equiv$   
 $\Pi_3$ ( moverAgenteYSancionarYCapturar(i, u) )

estudiantes? (moverE(u, i))  $\equiv$

$\Pi_3(\text{moverEstudiante}(i, u))$

$\text{estudiantes?}(\text{moverH}(u, i)) \equiv$   
 $\Pi_3(\text{moverHippie}(i, u))$

Otras Operaciones

$\text{capturar}(p, cAs) \equiv$   
**if** (  $p \in \text{posiciones4Vecinas}(\Pi_3(\text{dameUno}(cAs)))$  ) **then**  
 $\text{Ag}(\langle \Pi_1(cAs), \text{darCaptura}(\text{seg}(\text{dameUno}(cAs))), \Pi_3(\text{dameUno}(cAs)) \rangle, \text{capturar}(p, \text{sinUno}(cAs)))$   
**else**  
 $\text{Ag}(\text{dameUno}(cAs), \text{capturar}(p, \text{sinUno}(cAs)))$   
**fi**

$\text{sancionar}(p, cAs) \equiv$   
**if** (  $p \in \text{posiciones4Vecinas}(\Pi_3(\text{dameUno}(cAs)))$  ) **then**  
 $\text{Ag}(\langle \Pi_1(cAs), \text{darSancion}(\text{seg}(\text{dameUno}(cAs))), \Pi_3(\text{dameUno}(cAs)) \rangle, \text{capturar}(p, \text{sinUno}(cAs)))$   
**else**  
 $\text{Ag}(\text{dameUno}(cAs), \text{sancionar}(p, \text{sinUno}(cAs)))$   
**fi**

$\text{queTipoHay}(p, \text{tripla}, u) \equiv$   
**if** (  $\text{fila?}(p) = \text{alto?}(u) \vee \text{col?}(p) = \text{ancho?}(u)$  ) **then**  
 $\text{FueraDeRango}$   
**else**  
**if** (  $p \in \text{obstaculos?}(u)$  ) **then**  
 $\text{Obstaculo}$   
**else**  
**if** (  $p \in \text{damePosicionesAs}(\Pi_1(\text{tripla}))$  ) **then**  
 $\text{Agente}$   
**else**  
**if** (  $p \in \text{damePosicionesH}(\Pi_2(\text{tripla}))$  ) **then**  
 $\text{Hippie}$   
**else**  
**if** (  $p \in \text{damePosicionesEst}(\Pi_3(\text{tripla}))$  ) **then**  $\text{Estudiante}$  **else**  $\emptyset$  **fi**  
**fi**  
**fi**  
**fi**

$\text{cuatroVecinosShort}(p, \text{tripla}, u) \equiv$   
 $\text{cuatroVecinos}(p, \text{tripla}, u, \text{posiciones4Vecinas}(p))$

$\text{cuatroVecinos}(p, \text{tripla}, u, p4v) \equiv$   
**if** (  $\text{vacio?}(p4v)$  ) **then**  
 $\text{vacio}$   
**else**  
 $\text{Ag}(\text{queTipoHay}(\text{dameUno}(p4v), \text{tripla}, u), \text{cuatroVecinos}(p, \text{tripla}, u, \text{sinUno}(p4v)))$   
**fi**

$\text{queSituacion}(mcT) \equiv$   
 $\text{movRest?}(mcT) \cup \text{unAgente?}(mcT) \cup \text{dosHippies?}(mcT) \cup \text{cuatroEstudiantes?}(mcT)$

$\text{damePosicionesAs}(cAs) \equiv$   
**if**  $\text{vacio?}(cAs)$  **then**  $\emptyset$  **else**  $\text{Ag}(\Pi_2(\text{dameUno}(cAs)), \text{damePosicionesAs}(\text{sinUno}(cAs)))$  **fi**

$\text{damePosicionesEst}(cEst) \equiv$   
**if**  $\text{vacio?}(cEst)$  **then**  $\emptyset$  **else**  $\text{Ag}(\Pi_3(\text{dameUno}(cEst)), \text{damePosicionesEst}(\text{sinUno}(cEst)))$  **fi**

$\text{damePosicionesH}(cH) \equiv$

```

if vacio?(cH) then  $\emptyset$  else Ag(  $\Pi_2$ ( dameUno(cH) ), damePosicionesH( sinUno(cH) ) ) fi

    moverEstudianteYChequearSituaciones(i, u)  $\equiv$ 
if ( vacio?( dameEstudiante( i, estudiantes?(u) ) ) ) then
    { agentes?(u), hippies?(u), estudiantes?(u) }
else
    if  $\neg$ (  $\Pi_1$ ( dameEstudiante(i, u) )  $\in$  dirLibres( dameEstudiantePos(i, estudiantes?(u) ), u ) ) then
    { agentes?(u), hippies?(u), estudiantes?(u) }
    else
    chequearSituacionShort( agregarEstudianteTripla( { i, fin( dameEstudiante(i, u) ), mover(
    dameEstudiantePos(i, estudiantes?(u) ),  $\Pi_1$ ( dameEstudiante(i, u) ) } ), { agentes?(u), hippies?(u),
    sacarEstudianteId( i, estudiantes?(u) ) } ) )
fi
fi

    dameEstudiante(i, cEst)  $\equiv$ 
if (  $\Pi_1$ ( dameUno(cEst) ) = i ) then  $\Pi_2$ ( dameUno(cEst) ) else dameEstudiante(i, sinUno(cEst) ) fi

    dameEstudiantePos(i, cEst)  $\equiv$ 
if (  $\Pi_1$ ( dameUno(cEst) ) = i ) then  $\Pi_3$ ( dameUno(cEst) ) else dameEstudiantePos( i, sinUno(cEst) ) fi

    sacarEstudianteId(i, cEst)  $\equiv$ 
if ( vacio?(cEst) ) then
     $\emptyset$ 
else
    if (  $\Pi_1$ ( dameUno(cEst) ) = i ) then
    sinUno(cEst)
    else
    Ag( dameUno(cEst), sacarEstudiante( i, sinUno(cEst) ) )
fi
fi

    entradas?(an, al)  $\equiv$ 
if ( an = 0 ) then  $\emptyset$  else Ag( pos(0, an-1), Ag( pos(al-1, an-1), entradas(an-1, al) ) ) fi

    dirLibres(p, u)  $\equiv$ 
    dirNoOcupadas( dirValidas( p, ancho?(u), alto?(u) ), p, u )

    dirValidas(p, an, al)  $\equiv$ 
    {n,s,e,o} - (if col?(p) = 0 then {o} else  $\emptyset$  fi) - (if col?(p) = an-1 then {e} else  $\emptyset$  fi) - (if fila?(p) = 0
then {n} else  $\emptyset$  fi) - (if fila?(p) = al-1 then {s} else  $\emptyset$  fi)

    dirNoOcupadas( cDirs, p, u )  $\equiv$ 
if (vacio(cDirs)) then
     $\emptyset$ 
else
    if ( mover( p, dameUno(cDirs) )  $\in$  damePosicionesH( hippies?(u) )  $\vee$  mover( p, dameUno(cDirs) )
     $\in$  obstaculos?(u)  $\vee$  mover( p, dameUno(cDirs) )  $\in$  damePosicionesAs( agentes?(u) )  $\vee$  mover( p,
    dameUno(cDirs) )  $\in$  damePosicionesEst( estudiantes?(u) ) ) then
    dirNoOcupadas( sinUno(cDirs), p, u )
    else
    Ag( dameUno(cDirs), dirNoOcupadas( sinUno(cDirs), p, u ) )
fi
fi

    posNoOcupadas(cPos, u)  $\equiv$ 

```

```

if ( vacio?(cPos) ) then
     $\emptyset$ 
else
    if  $\neg$  dameUno(cPos)  $\in$  ( damePosicionesAs( agentes?(u) )  $\cup$  damePosicionesEst( estudiantes?(u) )  $\cup$  dame-
    PosicionesH( hippies?(u) )  $\cup$  obstaculos?(u) ) then
        Ag( dameUno(cPos), posNoOcupadas(sinUno(cPos), u) )
    else
        posNoOcupadas(sinUno(cPos), u)
    fi
fi

    moverAgenteYChequearSituacion(i, u)  $\equiv$ 
if ( vacio?( hippies?(u) )  $\wedge$  ( dameAgentePos( i, agentes?(u) )  $\in$  entradas?( alto?(u), ancho?(u) ) ) )  $\vee$  inactivo?(
dameAgente(i, u) ) then
    > agentes?(u), hippies?(u), estudiantes?(u) <
else
    if vacio?( posibleMovAs( dameAgente( i, agentes?(u) ), dameAgentePos( i, agentes?(u) ), u ) ) then
        > agentes?(u), hippies?(u), estudiantes?(u) <
    else
        chequearSituacionShort( agregarAgenteTripla( ) i, dameAgente( i, agentes?(u) ), mover( dameAgentePos(i,
        agentes?(u) ), dameUno( posibleMovAs( dameAgente( i, agentes?(u) ), dameAgentePos( i, agentes?(u) ),
        u ) ) ) <, > sacarAgenteId(i, agentes?(u) ), hippies?(u), estudiantes?(u) < )
    fi
fi

    posibleMovAs (agente, p, u)  $\equiv$ 
if ( inactivo?(agente) ) then
     $\emptyset$ 
else
    if ( vacio?( hippies?(u) ) ) then
        if ( dirLibres(agente, u)  $\cap$  direccionesOptimas( dameUno( cPosMasCercanaShort( p, posNoOcupadas(
        entradas(an, al), u ) ) ) ) =  $\emptyset$  ) then
            dirLibres( p, u )
        else
            dirLibres( p, u )  $\cap$  direccionesOptimas( dameUno( cPosMasCercanaShort( p, posNoOcupadas(
            entradas(an, al), u ) ) ) )
        fi
    else
        if dirLibres(p, u)  $\cap$  direccionesOptimas( dameUno( cPosMasCercanaShort( p, hippies?(u) ) ) ) =  $\emptyset$  then
            dirLibres(p, u)
        else
            dirLibres(p, u)  $\cap$  direccionesOptimas( dameUno( cPosMasCercanaShort( p, hippies?(u) ) ) )
        fi
    fi
fi

    dameAgente(i, cAs)  $\equiv$ 
if (  $\Pi_1$ ( dameUno(cAs) ) = i ) then  $\Pi_2$ ( dameUno(cAs) ) else dameAgente( i, sinUno(cAs) ) fi

    dameAgentePos(i, cAs)  $\equiv$ 
if (  $\Pi_1$ (dameUno(cAs)) = i ) then  $\Pi_3$ ( dameUno(cAs) ) else dameAgente( i, sinUno(cAs) ) fi

    sacarAgenteId(i, cAs)  $\equiv$ 
if ( vacio?(cAs) ) then
     $\emptyset$ 
else
    if (  $\Pi_1$ ( dameUno(cAs) ) = i ) then
        sinUno(cAs)
    else
        Ag( dameUno(cAs), sacarAgente( i, sinUno(cAs) ) )
    fi
fi

```



```

    moverHippieYChequearSituacion(i, u)  $\equiv$ 
if ( vacio?( posibleMovH( dameHippiePos( $i, u$ ),  $u$ ) ) ) then
     $\langle$  agentes?( $u$ ), hippies?( $u$ ), estudiantes?( $u$ )  $\rangle$ 
else
    chequearSituacionShort( agregarHippieTripla(  $\langle$   $i$ , mover( dameHippiePos( $i, u$ ), dameUno( posibleMovH(
    dameHippiePos( $i, u$ ),  $u$ ) ) )  $\rangle$ ,  $\langle$  agentes?( $u$ ), sacarHippieId(  $i$ , hippies?( $u$ ), estudiantes?( $u$ ) ) ) )
fi

    posibleMovH (p, u)  $\equiv$ 
if vacio?( estudiantes?( $u$ ) ) then
     $\emptyset$ 
else
    if dirLibres( $p, u$ )  $\cap$  direccionesOptimas( dameUno( cPosMasCercanaShort(  $p$ , damePosicionesEst(
    estudiantes?( $u$ ) ) ) ) ) =  $\emptyset$  then
    dirLibres( $p, u$ )
    else
    dirLibres( $p, u$ )  $\cap$  direccionesOptimas( dameUno( cPosMasCercanaShort(  $p$ , damePosicionesEst(
    estudiantes?( $u$ ) ) ) ) )
    fi
fi

    dameHippiePos(i, cH)  $\equiv$ 
if prim(dameUno( $cH$ )) =  $i$  then seg(dameUno( $cH$ )) else dameHippiePos( $i$ , sinUno( $cH$ )) fi

    sacarHippieId(i, cH)  $\equiv$ 
if vacio( $cH$ ) then
     $\emptyset$ 
else
    if prim(dameUno( $cH$ )) =  $i$  then sinUno( $cH$ ) else Ag(dameUno( $cH$ ), sacarHippie( $i$ , sinUno( $cH$ ))) fi
fi

    chequearSituacionShort(tripla, u)  $\equiv$ 
    chequearSituacion( tripla, tripla,  $u$ )

    agAgenteTripla(a,  $\langle$  cAs, cH, cEst  $\rangle$ )  $\equiv$ 
     $\langle$  Ag( $a, cAs$ ),  $cH, cEst$   $\rangle$ 

    agHippieTripla(h,  $\langle$  cAs, cH, cEst  $\rangle$ )  $\equiv$ 
     $\langle$   $cAs$ , Ag( $h, cH$ ),  $cEst$   $\rangle$ 

    agEstudianteTripla(e,  $\langle$  cAs, cH, cEst  $\rangle$ )  $\equiv$ 
     $\langle$   $cAs, cH$ , Ag( $e, cEst$ )  $\rangle$ 

    estudiantesAdyacentesPos(p, cEst)  $\equiv$ 
if dameEstudiantePos( dameUno( $cEst$ ) )  $\in$  posiciones4Vecinas( $p$ ) then
    Ag(  $\Pi_2$ ( dameUno( $cEst$ ) ), estudiantesAdyacentesPos( $p$ , sinUno( $cEst$ ) ) )
else
    estudiantesAdyacentesPos( $p$ , sinUno( $cEst$ ))
fi

```

*Cuando chequeamos los cambios de situaciones al moverse un agente, hippie o estudiante, primero vemos que pasa con los Agentes segun sus 4-vecinos, luego con los Hippies, y por ultimo con los Estudiantes. Cada uno de estos conjuntos no siguen ningun orden particular.*

```

chequearSituacion(  $\langle$  cAs, cH, cEst  $\rangle$  , triplaInfo, u)  $\equiv$ 

```

```

if vacio?(cH) then
  if vacio?(cEst) then
     $\langle cAs, cH, cEst \rangle$ 
  else
    if movRest, unAgente  $\subseteq$  queSituacion( cuatroVecinosShort( dameEstudiantePos( dameUno(cEst)), triplaInfo, u ) ) then
      if dosHippies  $\in$  queSituacion( cuatroVecinosShort( dameEstudiantePos( dameUno(cEst)), triplaInfo, u ) ) then
        agregarHippieTripla( convertirEstudianteAHippie(dameUno(cEst)) , chequearSituacion(  $\langle$  sancionar( dameEstudiantePos( dameUno(cEst) ), cAs), cH, sinUno(cEst) ), triplaInfo, u ) )
      else
        agregarEstudianteTripla( dameUno(cEst), chequearSituacion(  $\langle$  sancionar( dameEstudiantePos( dameUno(cEst) ), cAs), cH, sinUno(cEst) ), triplaInfo, u ) )
      fi
    else
      if dosHippies  $\in$  queSituacion( cuatroVecinosShort(dameEstudiantePos(dameUno(cEst)), triplaInfo, u ) ) then
        agregarHippieTripla( convertirEstudianteAHippie(dameUno(cEst)), chequearSituacion(  $\langle cAs, cH, sinUno(cEst) \rangle$ , triplaInfo, u ) )
      else
        agregarEstudianteTripla( dameUno(cEst), chequearSituacion(  $\langle cAs, cH, sinUno(cEst) \rangle$ , triplaInfo, u ) )
      fi
    fi
  fi
else
  if movRest, unAgente  $\subseteq$  queSituacion( cuatroVecinosShort( dameHippiePos( dameUno(cH) ), triplaInfo, u ) ) then
    chequearSituacion(  $\langle$  capturar(dameHippiePos(dameUno(cH)), cAs), sinUno(cH), cEst ), sacarHippieTripla(dameUno(cH), triplaInfo, u )
  else
    if movRest, cuatroEstudiantes  $\subseteq$  queSituacion( cuatroVecinosShort( dameHippiePos( dameUno(cH) ), triplaInfo, u ) ) then
      agregarEstudianteTripla( convertirHippieAEst(dameUno(cH), cEst), chequearSituacion(  $\langle cAs, sinUno(cH), cEst \rangle$ , sacarHippieTripla(dameUno(cH), triplaInfo, u ) )
    else
      agregarHippieTripla( dameUno(cH), chequearSituacion(  $\langle cAs, sinUno(cH), cEst \rangle$ , triplaInfo, u ) )
    fi
  fi
fi

```

$\text{convertirHippieAEst}(h, cEst) \equiv$   
 $\langle \Pi_0(h), \text{dameUno}(\text{estudiantesAdyacentesPos}(\Pi_1(h), cEst)), \Pi_1(h) \rangle$

$\text{convertirEstAHippie}(e) \equiv$   
 $\langle \Pi_0(e), \Pi_2(e) \rangle$

$\text{sacarHippieTripla}(h, \langle cAs, cH, cEst \rangle) \equiv$   
 $\langle cAs, \text{sacarHippieId}(\Pi_0(h), cH), cEst \rangle$

$\text{dondeEstaAgente}(i, u) \equiv$   
 $\text{dameAgentePos}(i, \text{agentes?}(u))$

$\text{dondeEstaHippie}(i, u) \equiv$   
 $\text{dameHippiePos}(i, \text{hippies?}(u))$

$\text{dondeEstaEstudiante}(i, u) \equiv$   
 $\text{dameEstudiantePos}(i, \text{estudiantes?}(u))$

$\text{masVigilante}(u) \equiv$   
 $\text{agenteConMenorPlacaShort}(\text{losMasVigilantesShort}(\text{agentes?}(u)))$

```

    losMasVigilantes(cAs, maxC)  $\equiv$ 
if vacio?(cAs) then
     $\emptyset$ 
else
    if hippiesCapturados( $\Pi_1$ (dameUno(cAs))) = maxC then
        Ag(dameUno(cAs), losMasVigilantes(sinUno(cAs), maxC))
    else
        losMasVigilantes(sinUno(cAs), maxC)
    fi
fi

    losMasVigilantesShort(cAs)  $\equiv$ 
    losMasVigilantes(cAs, maxCapturas(cAs))

    agenteConMenorPlaca(cAs, minN)  $\equiv$ 
if vacio?(cAs) then
     $\emptyset$ 
else
    if numPlaca( $\Pi_1$ (dameUno(cAs))) = minN then
        Ag(dameUno(cAs), agenteConMenorPlaca(sinUno(cAs), minN))
    else
        agenteConMenorPlaca(sinUno(cAs), minN)
    fi
fi

    agenteConMenorPlacaShort(cAs)  $\equiv$ 
    agenteConMenorPlaca(cAs, minPlaca(cAs))

    maxCapturas(cAs)  $\equiv$ 
if vacio?(cAs) then 0 else max(hippiesCapturados( $\Pi_1$ (dameUno(cAs))), maxCapturas(sinUno(cAs))) fi

    minPlaca(cAs)  $\equiv$ 
if vacio?(cAs) then 0 else min(numPlaca( $\Pi_1$ (dameUno(cAs))), minPlaca(sinUno(cAs))) fi

    cuantosHippies(u)  $\equiv$ 
    # hippies?(u)

    cuantosEstudiantes(u)  $\equiv$ 
    # estudiantes?(u)

```

**Fin TAD**

## 2. TAD AGENTE

**TAD AGENTE**

```

géneros      as

exporta      as, Generadores, Observadores Basicos, inactivo?

usa          NAT, BOOL

igualdad observacional
     $(\forall a, a' : \text{as}) \left( a =_{\text{obs}} a' \iff \left( \text{numPlaca}(a) =_{\text{obs}} \text{numPlaca}(a') \wedge \text{hippiesAtrapados}(a) =_{\text{obs}} \text{hippie-} \right. \right.$ 
     $\left. \left. \text{sAtrapados}(a') \wedge \text{numSanciones}(a) =_{\text{obs}} \text{numSanciones}(a') \right) \right)$ 

observadores básicos
    numPlaca : as  $\longrightarrow$  nat
    hippiesCapturados : as  $\longrightarrow$  nat

```

```

    numSanciones : as  → nat

generadores
    nuevoAs : nat  → as
    darCaptura : as a  → as
    darSancion : as  → as

otras operaciones
    inactivo? : as  → bool

axiomas       $\forall n: \text{nat } \forall a: \text{as}$ 
Observadores Basicos
    numPlaca(nuevoAs(n)) ≡
n
    numPlaca(darCaptura(a)) ≡
numPlaca(a)
    numPlaca(darSancion(a)) ≡
numPlaca(a)
    hippiesCapturados(nuevoAs(n)) ≡
0
    hippiesCapturados(darCaptura(a)) ≡
1+ hippiesCapturados(a)
    hippiesCapturados(darSancion(a)) ≡
hippiesCapturados(a)
    numSanciones(nuevoAs(n)) ≡
0
    numSanciones(darCaptura(a)) ≡
numSanciones(a)
    numSanciones(darSancion(a)) ≡
1+ numSanciones(a)

Otras Operaciones
    inactivo?(a) ≡
if numSanciones(a) > 3 then true else false fi

Fin TAD

```

### 3. TAD POSICION

**TAD POSICION**

```

géneros      pos

exporta      pos, Generadores, Observadores Basicos, Otras Operaciones

usa          NAT ,BOOL, CONJUNTO( $\alpha$ ), DIR

igualdad observacional
     $(\forall p, p' : \text{pos}) (p =_{\text{obs}} p' \iff (\text{fila?}(p) =_{\text{obs}} \text{fila}(p') \wedge \text{col?}(p) =_{\text{obs}} \text{col}(p')))$ 

observadores básicos
    fila? : pos  → nat
    col? : pos  → nat

generadores

```

```

nuevaPos : nat  $f \times$  nat  $c \longrightarrow$  pos {(f ≥ 0) ∧ (c ≥ 0)}

otras operaciones

direccionesOptimas : pos  $\times$  pos  $\longrightarrow$  conj(dir)
cPosMasCercana : pos  $\times$  conj(pos)  $\times$  nat  $\longrightarrow$  conj(pos)
cPosMasCercanaShort : pos  $\times$  conj(pos)  $\longrightarrow$  conj(pos)
menorDistancia : pos  $\times$  conj(pos)  $cPos \longrightarrow$  nat {¬ Vacía?(cPos)}
dist : pos  $\times$  pos  $\longrightarrow$  nat
posiciones4Vecinas : pos  $\longrightarrow$  conj(pos)
mover : pos  $p \times$  dir  $\longrightarrow$  pos {¬(col?(p) = 0 ∧ (o) = dir) ∧ ¬(fila?(p) = 0 ∧ (n) = dir)}

axiomas  $\forall p, p': pos \wedge \forall f, c, menorDist: nat \forall d: dir$ 
  fila?(nuevaPos(f,c))  $\equiv$ 
f
  col?(nuevaPos(f,c))  $\equiv$ 
c
  direccionesOptimas(p, p')  $\equiv$ 
{n,s,e,o} - (if fila?(p) <= fila?(p') then {n} else ∅ fi)
- (if fila?(p) >= fila?(p') then {s} else ∅ fi)
- (if col?(p) <= col?(p') then {o} else ∅ fi)
- (if col?(p) >= col?(p') then {e} else ∅ fi)
  PosMasCercanaShort(p, cPos)  $\equiv$ 
cPosMasCercana(p, cPos, menorDistancia(p,cPos))
  cPosMasCercana(p, cPos, menorDist)  $\equiv$ 
if vacío?(cPos) then
  ∅
else
  if dist(p, dameUno(cPos)) = menorDist then
    Ag(p, cPosMasCercana(p, sinUno(cPos), menorDist))
  else
    cPosMasCercana(p, sinUno(cPos), menorDist)
  fi
fi
  menorDistancia(p, cPos)  $\equiv$ 
if ¬ vacía?(cpos) then
  min(dist(p, dameUno(cPos)), menorDistancia(p, sinUno(cpos)))
else
  dist(p, dameUno(cPos))
fi
  dist(p,p')  $\equiv$ 
(max(fila?(p), fila?(p')) - min(fila?(p), fila?(p')) + (max(col?(p), col?(p')) - min(col?(p), col?(p'))
  mover(p, d)  $\equiv$ 
if (d = n) then
  nuevaPos(fila?(p)-1, col?(p))
else
  if (d=s) then
    nuevaPos(fila?(p)+1, col?(p))
  else
    if (dir = o) then nuevaPos(fila?(p), col?(p)-1) else nuevaPos(fila?(p), col?(p)+1) fi
  fi
fi
  posiciones4Vecinas(p)  $\equiv$ 

```

```

if (fila?(p) = 0 )  $\wedge$  (col?(p) = 0) then
  { nuevaPos(1,0), nuevaPos(0,1)}
else
  if fila?(p) = 0 then
    { nuevaPos(0, col?(p)-1), nuevaPos(0, col?(p)+1), pos(1, col?(p))}
  else
    if col?(p) = 0 then
      { nuevaPos(fila?(p)-1, 0), nuevaPos(fila?(p)+1, 0), nuevaPos(fila?(p), 1)}
    else
      { nuevaPos(fila?(p), col?(p)-1), nuevaPos(fila?(p), col?(p)+1),
        nuevaPos(fila?(p)+1, col?(p)), nuevaPos(fila?(p)-1, col?(p)) }
    fi
  fi
fi
Fin TAD

```

## 4. TAD DIRECCION

**TAD DIRECCION**

**géneros**      dir

**exporta**      dir, Generadores

**usa**

**igualdad observacional**

$$\left( \begin{array}{l} n =_{\text{obs}} n \wedge s =_{\text{obs}} s \wedge e =_{\text{obs}} e \wedge o =_{\text{obs}} o \wedge \neg(s =_{\text{obs}} n) \wedge \neg(s =_{\text{obs}} o) \wedge \neg(s =_{\text{obs}} e) \wedge \neg(n =_{\text{obs}} s) \wedge \neg(n \\ =_{\text{obs}} e) \wedge \neg(n =_{\text{obs}} o) \wedge \neg(e =_{\text{obs}} s) \wedge \neg(e =_{\text{obs}} n) \wedge \neg(e =_{\text{obs}} o) \wedge \neg(o =_{\text{obs}} s) \wedge \\ \neg(o =_{\text{obs}} n) \wedge \neg(o =_{\text{obs}} e) \end{array} \right)$$

**observadores básicos**

**generadores**

$n : \longrightarrow \text{dir}$

$s : \longrightarrow \text{dir}$

$e : \longrightarrow \text{dir}$

$o : \longrightarrow \text{dir}$

**otras operaciones**

**Fin TAD**

## 5. TAD SITUACIONESRODEO

**TAD SITUACIONESRODEO**

**géneros**      situ

**exporta**      situ, Generadores

**usa**

**igualdad observacional**

$$\left( \begin{aligned} & \text{movRest} =_{\text{obs}} \text{movRest} \wedge \text{unAgente} =_{\text{obs}} \text{unAgente} \wedge \text{dosHippies} =_{\text{obs}} \text{dosHippies} \wedge \text{cuatroEstudian-} \\ & \text{tes} =_{\text{obs}} \text{cuatroEstudiantes} \wedge \neg(\text{movRest} =_{\text{obs}} \text{unAgente}) \wedge \neg(\text{movRest} =_{\text{obs}} \text{dosHippies}) \wedge \neg(\text{movRest} \\ & =_{\text{obs}} \text{cuatroEstudiantes}) \wedge \neg(\text{unAgente} =_{\text{obs}} \text{movRest}) \wedge \neg(\text{unAgente} =_{\text{obs}} \text{dosHippies}) \wedge \neg(\text{unAgente} \\ & =_{\text{obs}} \text{cuatroEstudiantes}) \wedge \neg(\text{dosHippies} =_{\text{obs}} \text{movRest}) \wedge \neg(\text{dosHippies} =_{\text{obs}} \text{unAgente}) \wedge \neg(\text{dosHippies} \\ & =_{\text{obs}} \text{cuatroEstudiantes}) \wedge \neg(\text{cuatroEstudiantes} =_{\text{obs}} \text{movRest}) \wedge \neg(\text{cuatroEstudiantes} =_{\text{obs}} \text{unAgente}) \wedge \\ & \neg(\text{cuatroEstudiantes} =_{\text{obs}} \text{dosHippies}) \end{aligned} \right)$$

**observadores básicos**

**generadores**

movRest :  $\longrightarrow$  situ

unAgente :  $\longrightarrow$  situ

dosHippies :  $\longrightarrow$  situ

cuatroEstudiantes :  $\longrightarrow$  situ

**otras operaciones**

**Fin TAD**

## 6. TAD TIPOELEM

**TAD TIPOELEM**

**géneros**      tipo

**exporta**      tipo, Generadores

**usa**

**igualdad observacional**

$$\left( \begin{aligned} & \text{Vacio} =_{\text{obs}} \text{Vacio} \wedge \text{Obstaculo} =_{\text{obs}} \text{Obstaculo} \wedge \text{Agente} =_{\text{obs}} \text{Agente} \wedge \text{Hippie} =_{\text{obs}} \text{Hippie} \wedge \text{Estudiante} \\ & =_{\text{obs}} \text{Estudiante} \wedge \text{FueraDeRango} =_{\text{obs}} \text{FueraDeRango} \quad // \text{ y son diferentes de manera cruzada} \end{aligned} \right)$$

**observadores básicos**

**generadores**

Vacio :  $\longrightarrow$  tipo

Obstaculo :  $\longrightarrow$  tipo

Agente :  $\longrightarrow$  tipo

Hippie :  $\longrightarrow$  tipo

Estudiante :  $\longrightarrow$  tipo

FueraDeRango :  $\longrightarrow$  tipo

**otras operaciones**

**Fin TAD**