

# Algoritmos y Estructura de Datos 2

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

## Exorcismo Extremo

TP1

Integrante	LU	Correo electrónico
Rosinov, Gaston Einan	37/18	<a href="mailto:grosinov@gmail.com">grosinov@gmail.com</a>
Schuster, Martin Ariel	208/18	<a href="mailto:m.a.schuster98@gmail.com">m.a.schuster98@gmail.com</a>
Panichelli, Manuel	72/18	<a href="mailto:panicmanu@gmail.com">panicmanu@gmail.com</a>

Reservado para la cátedra

Instancia	Docente	Nota
Primera entrega		
Segunda entrega		

## Índice

<b>1. TAD JUEGO</b>	<b>3</b>
<b>2. TAD HABITACION</b>	<b>6</b>
<b>3. TAD ACCION</b>	<b>7</b>
<b>4. TAD DIRECCION</b>	<b>9</b>
<b>5. EXTENSIONES Y RENOMBRES</b>	<b>10</b>

# 1. TAD JUEGO

## TAD JUEGO

<b>géneros</b>	juego
<b>exporta</b>	observadores, generadores, puntaje, ronda
<b>usa</b>	SECUENCIA, NAT, CONJUNTO, HABITACION, UBICACION, PJ, ACCION, DIRECCION, FANTASMA, PJ, DICCIONARIO, BOOL

### igualdad observacional

$$(\forall j, j' : \text{juego}) \left( j =_{\text{obs}} j' \iff \left( \begin{array}{l} (\text{accionesPJs}(j) =_{\text{obs}} \text{accionesPJs}(j')) \wedge \\ (\text{accionesFan}(j) =_{\text{obs}} \text{accionesFan}(j')) \wedge \\ (\text{localizarJugadores}(j) =_{\text{obs}} \text{localizarJugadores}(j')) \wedge \\ (\text{hab}(j) =_{\text{obs}} \text{hab}(j')) \wedge \\ ((\forall p : \text{pj}) (\text{vivePJ?}(j, p) =_{\text{obs}} \text{vivePJ?}(j', p))) \wedge \\ ((\forall f : \text{fantasma}) ((\text{viveFan?}(j, p) =_{\text{obs}} \text{viveFan?}(j', p)) \wedge \\ (\text{ubicacionInicialFan}(j, f) =_{\text{obs}} \text{ubicacionInicialFan}(j', f)))) \end{array} \right) \right)$$

### observadores básicos

accionesPJs	: juego	→	dicc(pj, secu(accion))
accionesFan	: juego	→	dicc(pj, secu(accion))
hab	: juego	→	hab
vivePJ?	: juego $j \times \text{pj } p$	→	bool $\{p \in \text{jugadores}(j)\}$
viveFan?	: juego $j \times \text{fantasma } f$	→	bool $\{f \in \text{fantasmas}(j)\}$
ubicacionInicialFan	: juego $j \times \text{fantasma } f$	→	ubicacion $\{f \in \text{fantasmas}(f)\}$
localizarJugadores	: juego	→	dicc(pj, ubicacion)

### generadores

iniciar	: conj(pj) $pjs \times \text{secu}(accion) as \times \text{ubicacion } u$	→	juego $\{ \text{esConexa?}(h) \wedge \neg \emptyset?(as) \wedge \neg \emptyset?(pjs) \wedge \text{esValida?}(h, \text{pos}(u)) \}$
proxPaso	: juego $j \times \text{pj } p \times \text{accion } a$	→	juego $\{ p \in \text{jugadores}(j) \wedge \text{vivePJ?}(j, p) \wedge \neg \text{termino?}(j) \wedge \neg \text{esMirar}(a) \}$

### otras operaciones

jugadores	: juego	→	conj(pj)
fantasmas	: juego	→	conj(fantasma)
nombreSiguienteFan	: juego	→	fantasma
puntaje	: juego	→	nat
ronda	: juego	→	nat
paso	: juego	→	nat
cantAcciones	: juego $\times \text{conj}(pj)$	→	nat
terminaRonda	: juego $j \times \text{pj } p \times \text{accion}$	→	bool $\{p \in \text{jugadores}(j)\}$
fantasmaEspecial	: juego	→	fantasma
termino?	: juego	→	bool
estanVivos	: juego $\times \text{conj}(pj) pjs$	→	bool $\{pjs \subseteq \text{jugadores}(j)\}$

ubicacionInicialPJ	: juego $j \times$ pj $p$	$\longrightarrow$ ubicacion	$\{p \in \text{jugadores}(j)\}$
ubicacionPJ	: juego $j \times$ pj $p$	$\longrightarrow$ ubicacion	$\{p \in \text{jugadores}(j)\}$
ubicacionFan	: juego $j \times$ fantamsa $f$	$\longrightarrow$ ubicacion	$\{f \in \text{fantasmas}(j)\}$
deducirUbicacion	: juego $j \times$ ubicacion $u \times$ acciones	$\longrightarrow$ ubicacion	$\{esValida?(hab(j), pos(u))\}$
moriraFantasma	: juego $j \times$ pj $p \times$ accion $\times$ fantasma $f$	$\longrightarrow$ bool	$\{p \in \text{jugadores}(j) \wedge$ $f \in \text{fantasmas}(j) \wedge_L$ $viveFan?(j, f) \wedge$ $vivePJ?(j, p)\}$
moriraPJ	: juego $j \times$ conj(fantasma) $fs \times$ pj $p \times$ accion	$\longrightarrow$ bool	$\{p \in \text{jugadores}(j) \wedge$ $fs \subseteq \text{fantasmas}(j) \wedge_L$ $vivePJ?(j, p)\}$
moriraPJPorFan	: juego $j \times$ fantasma $f \times$ pj $p \times$ accion	$\longrightarrow$ bool	$\{p \in \text{jugadores}(j) \wedge$ $f \in \text{fantasmas}(j) \wedge_L$ $vivePJ?(j, p) \wedge$ $viveFan?(j, f)\}$
accionFan	: juego $j \times$ fantasma $f$	$\longrightarrow$ accion	$\{f \in \text{fantasmas}(j) \wedge_L$ $viveFan?(j, f)\}$
inicializarAcciones	: conj(pj)	$\longrightarrow$ dicc(pj, secu(accion))	
agregarFantasma	: juego $j \times$ ubicacion $u \times$ dicc(fantasma $\times$ secu(accion)) $\times$ fantasma $\times$ secu(accion)	$\longrightarrow$ dicc(fantasma, secu(accion))	$\{esValida?(hab(j), pos(u))\}$
generarAccionesFan	: juego $j \times$ ubicacion $u \times$ secu(accion)	$\longrightarrow$ secu(accion)	$\{esValida?(hab(j), pos(u))\}$
<b>axiomas</b>	$\forall p: \text{pj},$ $\forall pjs: \text{conj}(\text{pj}),$ $\forall f: \text{fantasma},$ $\forall fs: \text{conj}(\text{fantasma}),$ $\forall j: \text{juego},$ $\forall h: \text{hab},$ $\forall u, uInicialPJ: \text{ubicacion},$ $\forall a: \text{accion},$ $\forall as: \text{secu}(\text{accion})$		
accionesPJs(iniciar(pjs, as, u, h))	$\equiv$ inicializarAcciones(pjs)		
accionesPJs(proxPaso(j, p, a))	$\equiv$ <b>if</b> $\neg$ terminaRonda(j, p, a) <b>then</b> definir(p, obtener(p, accionesPJs(j)) $\circ$ a, accionesPJs(j)) <b>else</b> inicializarAcciones(jugadores(j)) <b>fi</b>		
accionesFan(iniciar(pjs, as, u, h))	$\equiv$ definir(nombreSiguienteFan(j), as, vacio)		
accionesFan(proxPaso(j, p, a))	$\equiv$ <b>if</b> $\neg$ terminaRonda(j, p, a) <b>then</b> accionesFan(j) <b>else</b> agregarFantasma(j, ubicacionInicialPJ(j, p), accionesFan(j), nombreSiguienteFan(j), obtener(p, accionesPJs(j)) $\circ$ a ) <b>fi</b>		
hab(iniciar(pjs, as, u, h))	$\equiv$ h		
hab(proxPaso(j, p, a))	$\equiv$ hab(j)		
vivePJ?(iniciar(pjs, as, u, h), p')	$\equiv$ true		

vivePJ?(proxPaso(j, p, a), p')	≡ terminaRonda?(j, p, a) ∨ <b>if</b> p = p' <b>then</b> ¬ moriraPJ(j, fantasmas(j), p, a) <b>else</b> vivePJ?(j, p') ∧ <sub>L</sub> ¬ moriraPJ(j, fantasmas(j), p, a) <b>fi</b>
viveFan?(iniciar(pjs, as, u, h), f)	≡ true
viveFan?(proxPaso(j, p, a), f)	≡ terminaRonda?(j, p, a) ∨ (viveFan?(j, f) ∧ <sub>L</sub> ¬ moriraFantasma(j, p, a, f))
ubicacionInicialFan(iniciar(pjs, as, u, h))	≡ u
ubicacionInicialFan(proxPaso(j, p, a))	≡ <b>if</b> f ∈ fantasmas(j) <b>then</b> ubicacionInicialFan(j, f) <b>else</b> ubicacionInicialPJ(j, p) <b>fi</b>
jugadores(j)	≡ claves(accionesPJs(j))
fantasmas(j)	≡ claves(accionesFan(j))
nombreSiguienteFan(j)	≡ #(claves(accionesFan(j))) + 1
puntaje(j)	≡ ronda(j) - 1
ronda(j)	≡ #(fantasmas(j))
paso(j)	≡ cantAcciones(j, jugadores(j))
cantAcciones(j, pjs)	≡ <b>if</b> ∅?(pjs) <b>then</b> 0 <b>else</b> long(obtener(dameUno(pjs), accionesPJs(j))) + cantAcciones(j, sinUno(pjs)) <b>fi</b>
termino?(j)	≡ ¬ estanVivos(j, jugadores(j))
estanVivos(j, pjs)	≡ <b>if</b> ∅?(pjs) <b>then</b> true <b>else</b> vivePJ?(j, dameUno(pjs)) ∧ estanVivos(j, sinUno(pjs)) <b>fi</b>
fantasmaEspecial(j)	≡ #(claves(accionesFan(j)))
ubicacionInicialPJ(j, p)	≡ obtener(p, localizarJugadores(j))
ubicacionPJ(j, p)	≡ deducirUbicacion(j, ubicacionInicialPJ(j, p), obtener(p, accionesPJs(j)))
ubicacionFan(j, f)	≡ deducirUbicacion(j, ubicacionInicialFan(j, f), obtener(f, accionesFan(j)))
deducirUbicacion(j, u, as)	≡ <b>if</b> vacia?(as) <b>then</b> u <b>else</b> deducirUbicacion(j, ubicacionLuegoDe(prim(as), hab(j), u), fin(as)) <b>fi</b>
agregarFantasma(j, uInicialPJ, accionesFantasmas, f, as)	≡ definir(f, generarAccionesFantasma(j, uInicialPJ, as), accionesFantasmas)
generarAccionesFan(j, uInicialPJ, as)	≡ as & (nada • nada • nada • nada • nada) & invertir(hab(j), uInicialPJ, as)
inicializarAcciones(pjs)	≡ <b>if</b> ∅?(pjs) <b>then</b> vacio <b>else</b> definir(dameUno(pjs), <>, inicializarAcciones(sinUno(pjs))) <b>fi</b>

terminaRonda(j, p, a)	$\equiv$ moriraFantasma(j, p, a, fantasmaEspecial(j))
moriraFantasma(j, p, a, f)	$\equiv$ pos(ubicacionFan(j, f)) $\in$ posicionesAfectadasPor(a, hab(j), ubicacionPJ(j, p))
moriraPJ(j, fs, p, a)	$\equiv$ <b>if</b> $\emptyset?(fs)$ <b>then</b> false <b>else</b> (viveFan?(j, dameUno(fs)) $\wedge_L$ moriraPJPorFan(j, dameUno(fs), p, a)) $\vee_L$ moriraPJ(j, sinUno(fs), p, a) <b>fi</b>
moriraPJPorFan(j, f, p, a)	$\equiv$ $\neg$ moriraFantasma(j, p, a, f) $\wedge$ (pos(ubicacionLuegoDe(a, hab(j), ubicacionPJ(j, p))) $\in$ posicionesAfectadasPor(accionFan(j, f), hab(j), ubicacionFan(j, f)))
accionFan(j, f)	$\equiv$ obtener(accionesFan(j), f)[paso(j) % obtener(accionesFan(j), f)]

Fin TAD

## 2. TAD HABITACION

TAD HABITACION

**géneros** hab**exporta** hab, observadores, generadores, esConexa?**usa** POSICION, BOOL, NAT**igualdad observacional**

$$(\forall h, h' : \text{hab}) \left( h =_{\text{obs}} h' \iff \left( (\forall p : \text{posicion}) (\text{esValida?}(p, h) =_{\text{obs}} \text{esValida?}(p, h') \wedge_L) \right) \right)$$

$$\left( \text{esValida?}(p, h) \Rightarrow_L \text{estaOcupada?}(p, h) =_{\text{obs}} \text{estaOcupada?}(p, h') \right) \right)$$

**observadores básicos**

esValida?	: hab $\times$ posicion	$\longrightarrow$ bool	
estaOcupada?	: hab $h \times$ posicion $p$	$\longrightarrow$ bool	{esValida?(h, p)}

**generadores**

nueva	: nat $n$	$\longrightarrow$ hab	{n>1}
ocupar	: hab $h \times$ posicion $p$	$\longrightarrow$ hab	{esValida?(h, p) $\wedge_L$ $\neg$ estaOcupada?(h, p)}

**otras operaciones**

esConexa?	: hab	$\longrightarrow$ bool	
tamano	: hab	$\longrightarrow$ nat	
posiciones	: hab	$\longrightarrow$ conj(posicion)	
posicionesLibres	: hab $h \times$ conj(posicion) $ps$	$\longrightarrow$ conj(posicion)	{ps $\subseteq$ posiciones(h)}
verificarAlcance	: hab $h \times$ conj(posicion) $ps$	$\longrightarrow$ bool	{ps $\subseteq$ posiciones(h)}
verificarAlcancePos	: hab $h \times$ conj(posicion) $ps \times$ posicion $p$	$\longrightarrow$ bool	{ps $\subseteq$ posiciones(h) $\wedge$ p $\in$ posiciones(h)}

**axiomas**  $\forall h: \text{hab} \forall ps: \text{conj}(\text{posicion}) \forall p: \text{posicion} \forall n, k, tam: \text{nat}$ 

$$\text{esValida?}(nueva(n), p) \equiv 0 \leq \Pi_1(p) < n \wedge 0 \leq \Pi_2(p) < n$$

```

esValida?(ocupar(h,p'),p)      ≡ p = p' ∨L esValida?(h, p)
estaOcupada?(nueva(n),p)      ≡ false
estaOcupada?(ocupar(h,p'),p)  ≡ p = p' ∨ estaOcupada?(h, p)
tamano(nueva(n))              ≡ n
tamano(ocupar(h, p))          ≡ tamano(h)
esConexa?(h)                  ≡ verificarAlcance(h, posicionesLibres(posiciones(h)))
posicionesLibres(h, ps)       ≡ if  $\emptyset?(ps)$ 
                               then  $\emptyset$ 
                               else
                               (if estaOcupada?(h, dameUno(ps)) then  $\emptyset$  else {dameUno(ps)} fi)
                               ∪ posicionesLibres(h, sinUno(ps))
                               fi
verificarAlcance(h, ps)       ≡ if  $\emptyset?(ps)$ 
                               then true
                               else
                               verificarAlcancePos(h, ps, dameUno(ps)) ∧ verificarAlcance(h, p)
                               fi
verificarAlcancePos(h, ps, p) ≡ if  $\emptyset?(ps)$ 
                               then true
                               else
                               esAlcanzable(h, p, dameUno(ps)) ∧ verificarAlcancePos(h, p, sinUno(ps))
                               fi
posiciones(h)                  ≡ darPosiciones(h, tamano(h) - 1, tamano(h) - 1, tamano(h) - 1)
darPosiciones(h, n, k, tam)    ≡ if  $n = 0? \wedge k = 0?$ 
                               then  $\emptyset$ 
                               else if  $k = 0?$ 
                               then Ag((n,k), darPosiciones(h, n - 1, tam, tam))
                               else Ag((n,k), darPosiciones(h, n, k - 1, tam))
                               fi
                               fi

```

**Fin TAD**

### 3. TAD ACCION

**TAD ACCION**

**géneros**      accion

**exporta**      observadores, generadores, genero, otras operaciones

**usa**          DIRECCION, POSICION, UBICACION, BOOL, CONJUNTO, HABITACION, SECUENCIA

**igualdad observacional**

$$(\forall a, a' : \text{accion}) \left( a =_{\text{obs}} a' \iff \left( \begin{array}{l} \text{esNada}(a) =_{\text{obs}} \text{esNada}(a') \wedge \\ \text{esDisparar}(a) =_{\text{obs}} \text{esDisparar}(a') \wedge \\ \text{esMover}(a) =_{\text{obs}} \text{esMover}(a') \wedge \\ \text{esMirar}(a) =_{\text{obs}} \text{esMirar}(a') \wedge \\ ((\text{esMover}(a) \vee \text{esMirar}(a)) \Rightarrow_L \\ \text{direccion}(a) =_{\text{obs}} \text{direccion}(a')) \end{array} \right) \right)$$

**observadores básicos**

esMover                      : accion                       $\longrightarrow$  bool

esMirar                      : accion                       $\longrightarrow$  bool

esDisparar	: accion	→ bool	
esNada	: accion	→ bool	
direccion	: accion $a$	→ direccion	{esMirar( $a$ ) $\vee$ esMover( $a$ )}

**generadores**

mover	: direccion	→ accion
mirar	: direccion	→ accion
disparar	:	→ accion
nada	:	→ accion

**otras operaciones**

ubicacionLuegoDe	: accion $\times$ hab $h \times$ ubicacion $u$	→ conj(posicion)	{esValida?( $h$ , pos( $u$ ))}
posicionesAfectadasPor	: accion $\times$ hab $h \times$ ubicacion $u$	→ conj(posicion)	{esValida?( $h$ , pos( $u$ ))}
$\neg \bullet$	: accion	→ accion	
invertir	: hab $h \times$ ubicacion $u \times$ secu(accion)	→ secu(accion)	{esValida?( $h$ , pos( $u$ ))}

**axiomas**  $\forall d$ : direccion  $\forall u$ : ubicacion,  $\forall a$ : habitacion,  $\forall as$ : secu(accion)

posicionesAfectadasPor(mover( $d$ ), $h$ , $u$ )	$\equiv \emptyset$
posicionesAfectadasPor(mirar( $d$ ), $h$ , $u$ )	$\equiv \emptyset$
posicionesAfectadasPor(nada, $h$ , $u$ )	$\equiv \emptyset$
posicionesAfectadasPor(disparar, $h$ , $u$ )	$\equiv$ <b>if</b> esValida?( $h$ , proxPosEnDir(dir( $u$ ), pos( $u$ )) $\wedge_L$ $\neg$ estaOcupada?( $h$ , proxPosEnDir(dir( $u$ ), pos( $u$ ))) <b>then</b> Ag(proxPosEnDir(dir( $u$ ), pos( $u$ )), posicionesAfectadasPor(disparar, $h$ , $\langle$ proxPosEnDir(dir( $u$ ), pos( $u$ )), dir( $u$ ) $\rangle$ ) <b>else</b> $\emptyset$ <b>fi</b>
invertir( $h$ , $u$ , $as$ )	$\equiv$ <b>if</b> vacia?( $as$ ) <b>then</b> $\langle \rangle$ <b>else</b> invertir( $h$ , ubicacionLuegoDe(prim( $as$ ), $h$ , $u$ ), fin( $as$ )) $\circ$ $\neg$ (prim( $as$ ), $h$ , $u$ ) <b>fi</b>
$\neg$ (mover( $d$ ), $h$ , $u$ )	$\equiv$ <b>if</b> pos(ubicacionLuegoDe(mover( $d$ ), $h$ , $u$ )) = pos( $u$ ) <b>then</b> mirar(opuesta( $d$ )) <b>else</b> mover(opuesta( $d$ )) <b>fi</b>
$\neg$ (mirar( $d$ ), $h$ , $u$ )	$\equiv$ mirar(opuesta( $d$ ))
$\neg$ (disparar, $h$ , $u$ )	$\equiv$ disparar
$\neg$ (nada, $h$ , $u$ )	$\equiv$ nada
ubicacionLuegoDe(nada, $h$ , $u$ )	$\equiv u$
ubicacionLuegoDe(disparar, $h$ , $u$ )	$\equiv u$
ubicacionLuegoDe(mirar( $d$ ), $h$ , $u$ )	$\equiv \langle$ pos( $u$ ), $d$ $\rangle$
ubicacionLuegoDe(mover( $d$ ), $h$ , $u$ )	$\equiv \langle$ ( <b>if</b> esValida?( $h$ , proxPosEnDir( $d$ , pos( $u$ ))) $\wedge_L$ $\neg$ estaOcupada?( $h$ , proxPosEnDir( $d$ , pos( $u$ ))) <b>then</b> proxPosEnDir( $d$ , pos( $u$ )) <b>else</b> pos( $u$ ) <b>fi</b> ), $d$ $\rangle$
esMirar(mirar( $d$ ))	$\equiv$ <b>true</b>
esMirar(mover( $d$ ))	$\equiv$ <b>false</b>



esMirar(disparar)	$\equiv$ false
esMirar(nada)	$\equiv$ false
esMover(mirar(d))	$\equiv$ false
esMover(mover(d))	$\equiv$ <b>true</b>
esMover(disparar)	$\equiv$ false
esMover(nada)	$\equiv$ false
esDisparar(mirar(d))	$\equiv$ false
esDisparar(mover(d))	$\equiv$ false
esDisparar(disparar)	$\equiv$ <b>true</b>
esDisparar(nada)	$\equiv$ false
esNada(mirar(d))	$\equiv$ false
esNada(mover(d))	$\equiv$ false
esNada(disparar)	$\equiv$ false
esNada(nada)	$\equiv$ <b>true</b>
direccion(mirar(d))	$\equiv$ d
direccion(mover(d))	$\equiv$ d

**Fin TAD**

## 4. TAD DIRECCION

**TAD DIRECCION**

<b>géneros</b>	direccion
<b>exporta</b>	observadores, generadores, otras operaciones
<b>usa</b>	BOOL, POSICION, NAT

**igualdad observacional**

$$(\forall d, d' : \text{direccion}) \left( d =_{\text{obs}} d' \iff \left( \begin{array}{l} \text{esArriba}(d) =_{\text{obs}} \text{esArriba}(d') \wedge \\ \text{esAbajo}(d) =_{\text{obs}} \text{esAbajo}(d') \wedge \\ \text{esIzquierda}(d) =_{\text{obs}} \text{esIzquierda}(d') \wedge \\ \text{esDerecha}(d) =_{\text{obs}} \text{esDerecha}(d') \end{array} \right) \right)$$

**observadores básicos**

esArriba	: direccion	$\longrightarrow$ bool
esAbajo	: direccion	$\longrightarrow$ bool
esIzquierda	: direccion	$\longrightarrow$ bool
esDerecha	: direccion	$\longrightarrow$ bool

**generadores**

arriba	:	$\longrightarrow$ direccion
abajo	:	$\longrightarrow$ direccion
izquierda	:	$\longrightarrow$ direccion
derecha	:	$\longrightarrow$ direccion

**otras operaciones**

$\text{opuesta} : \text{direccion} \longrightarrow \text{direccion}$   
 $\text{proxPosEnDir} : \text{direccion} \times \text{posicion} \longrightarrow \text{posicion}$

**axiomas**

$\text{opuesta}(\text{arriba}) \equiv \text{abajo}$   
 $\text{opuesta}(\text{abajo}) \equiv \text{arriba}$   
 $\text{opuesta}(\text{izquierda}) \equiv \text{derecha}$   
 $\text{opuesta}(\text{derecha}) \equiv \text{izquierda}$   
 $\text{proxPosEnDir}(\text{arriba}, p) \equiv \langle \Pi_1(p), \Pi_2(p) + 1 \rangle$   
 $\text{proxPosEnDir}(\text{abajo}, p) \equiv \langle \Pi_1(p), \Pi_2(p) - 1 \rangle$   
 $\text{proxPosEnDir}(\text{izquierda}, p) \equiv \langle \Pi_1(p) - 1, \Pi_2(p) \rangle$   
 $\text{proxPosEnDir}(\text{derecha}, p) \equiv \langle \Pi_1(p) + 1, \Pi_2(p) \rangle$   
 $\text{esArriba}(\text{arriba}) \equiv \text{true}$   
 $\text{esArriba}(\text{abajo}) \equiv \text{false}$   
 $\text{esArriba}(\text{izquierda}) \equiv \text{false}$   
 $\text{esArriba}(\text{derecha}) \equiv \text{false}$   
 $\text{esAbajo}(\text{arriba}) \equiv \text{false}$   
 $\text{esAbajo}(\text{abajo}) \equiv \text{true}$   
 $\text{esAbajo}(\text{izquierda}) \equiv \text{false}$   
 $\text{esAbajo}(\text{derecha}) \equiv \text{false}$   
 $\text{esIzquierda}(\text{arriba}) \equiv \text{false}$   
 $\text{esIzquierda}(\text{abajo}) \equiv \text{false}$   
 $\text{esIzquierda}(\text{izquierda}) \equiv \text{true}$   
 $\text{esIzquierda}(\text{derecha}) \equiv \text{false}$   
 $\text{esDerecha}(\text{arriba}) \equiv \text{false}$   
 $\text{esDerecha}(\text{abajo}) \equiv \text{false}$   
 $\text{esDerecha}(\text{izquierda}) \equiv \text{false}$   
 $\text{esDerecha}(\text{derecha}) \equiv \text{true}$

**Fin TAD****5. EXTENSIONES Y RENOMBRES****TAD FANTASMA ES NAT****TAD PJ ES NAT****TAD POSICION ES TUPLA(NAT, NAT)****TAD NAT extiende NAT****otras operaciones**

$\bullet \% \bullet : \text{nat} \times \text{nat} \longrightarrow \text{nat}$

**axiomas**  $\forall n, m: \text{nat}$

$$n \% m \equiv \text{if } n < m \text{ then } n \text{ else } (n - m) \% m \text{ fi}$$
**Fin TAD****TAD UBICACION extiende TUPLA(POSICION, DIRECCION)****otras operaciones**

direccion

pos : ubicacion  $\longrightarrow$  posiciondir : ubicacion  $\longrightarrow$  direccion**axiomas**  $\forall u$ : ubicacionpos(u)  $\equiv \Pi_1(u)$ dir(u)  $\equiv \Pi_2(u)$ **Fin TAD****TAD SECUENCIA extiende SECUENCIA****otras operaciones** $\bullet[\bullet] : \text{secu}(\alpha) \times \text{nat } i \longrightarrow \alpha$   $\{i < \text{long}(s)\}$ **axiomas**  $\forall s$ : secu( $\alpha$ ),  $\forall i$ : nat $s[i] \equiv \text{if } i = 0? \text{ then } \text{prim}(s) \text{ else } \text{fin}(s)[i - 1] \text{ fi}$ **Fin TAD**