

# Algoritmos y Estructuras de Datos II

## Trabajo Práctico 1

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

### Pacalgo2

Los inertes

Integrante	LU	Correo electrónico
Bruno Robbio	480/09	brobbio@hotmail.com
Nicolas Andres Kinaschuk	248/15	nicolaskinaschuk@gmail.com
Pedro Joel Burgos	804/18	facultadburgospedrojoel@hotmail.com
Valentina Madelaine Saravia Ruiz	257/18	valentina.saraviaruiz@gmail.com

Reservado para la cátedra

Instancia	Docente	Nota
Primera entrega		
Segunda entrega		

## 1. Introducción

### 1.1. Consideraciones

- No se puede arrancar el mapa ganando o perdiendo
- El mapa puede no tener un camino ganador o perdedor
- El vértice del mapa es la esquina inferior izquierda

## 2. Desarrollo

### 2.1. TP 1

#### TAD CASILLERO

**extiende**    Tupla(int, int)

**usa**        Int

**géneros**    casillero

**exporta**    casillero, +, -, aDistanciaMenosDeN

#### otras operaciones

$\bullet + \bullet$  : casillero  $\times$  casillero  $\longrightarrow$  casillero

$\bullet - \bullet$  : casillero  $\times$  casillero  $\longrightarrow$  casillero

aDistanciaMenosDeN : casillero  $\times$  nat  $\longrightarrow$  conj(casillero)

#### axiomas

$\pi_1(c1 + c2) \equiv \pi_1(c1) + \pi_1(c2)$

$\pi_2(c1 + c2) \equiv \pi_2(c1) + \pi_2(c2)$

$\pi_1(c1 - c2) \equiv \pi_1(c1) - \pi_1(c2)$

$\pi_2(c1 - c2) \equiv \pi_2(c1) - \pi_2(c2)$

aDistanciaMenosDeN( $c, n$ )  $\equiv$  **if**  $n=0?$  **then**  
                                   { $c$ }  
                                   **else**  
                                   (aDistanciaMenosDeN( $c + \langle 1,0 \rangle, n - 1$ )  $\cup$   
                                   aDistanciaMenosDeN( $c - \langle 1,0 \rangle, n - 1$ )  $\cup$   
                                   aDistanciaMenosDeN( $c + \langle 0,1 \rangle, n - 1$ )  $\cup$   
                                   aDistanciaMenosDeN( $c - \langle 0,1 \rangle, n - 1$ )  $\cup$   
                                   aDistanciaMenosDeN( $c, n - 1$ ))  
                                   **fi**

**Fin TAD**

**TAD MAPA**

**usa** Casillero, Conjunto, Bool  
**géneros** mapa  
**exporta** mapa, observadores, generadores, casillerosLibres

**igualdad observacional**

$$(\forall m_1, m_2 : \text{mapa}) \left( m_1 =_{\text{obs}} m_2 \iff \begin{pmatrix} \text{conjFantasmas}(m_1) =_{\text{obs}} \text{conjFantasmas}(m_2) \wedge \\ \text{conjParedes}(m_1) =_{\text{obs}} \text{conjParedes}(m_2) \wedge \\ \text{conjChocolates}(m_1) =_{\text{obs}} \text{conjChocolates}(m_2) \wedge \\ \text{dimensiones}(m_1) =_{\text{obs}} \text{dimensiones}(m_2) \wedge \\ \text{casilleroInicial}(m_1) =_{\text{obs}} \text{casilleroInicial}(m_2) \wedge \\ \text{vértice}(m_1) =_{\text{obs}} \text{vértice}(m_2) \wedge \\ \text{casilleroDeLlegada}(m_1) =_{\text{obs}} \text{casilleroDeLlegada}(m_2) \end{pmatrix} \right)$$

**observadores básicos**

$\text{conjFantasmas} : \text{mapa} \rightarrow \text{conj}(\text{casillero})$   
 $\text{conjParedes} : \text{mapa} \rightarrow \text{conj}(\text{casillero})$   
 $\text{conjChocolates} : \text{mapa} \rightarrow \text{conj}(\text{casillero})$   
 $\text{dimensiones} : \text{mapa} \rightarrow \text{tupla}(\text{nat}, \text{nat})$   
 $\text{vértice} : \text{mapa} \rightarrow \text{casillero}$   
 $\text{casilleroInicial} : \text{mapa} \rightarrow \text{casillero}$   
 $\text{casilleroDeLlegada} : \text{mapa} \rightarrow \text{casillero}$

**generadores**

$\text{nuevoMapa} : \text{tupla}(\text{nat}, \text{nat}) \text{ dimensión} \times \text{casillero vértice} \times \text{casillero inicio} \times \text{casillero fin} \times$   
 $\text{conj}(\text{casillero}) \text{ fantasmas} \times \text{conj}(\text{casillero}) \text{ paredes} \times \text{conj}(\text{casillero}) \text{ chocolates} \rightarrow \text{mapa}$

$$\left\{ \begin{array}{l} \emptyset?(fantasmas \cap paredes) \wedge \emptyset?(chocolates \cap paredes) \wedge \\ \emptyset?(fantasmas \cap chocolates) \wedge (\text{inicio} \neq \text{fin}) \wedge \\ \text{dentroDeLasDimensiones}(\text{dimensión}, \text{vértice}, \text{inicio}) \wedge \\ \text{dentroDeLasDimensiones}(\text{dimensión}, \text{vértice}, \text{fin}) \wedge \\ (\forall f \in \text{fantasmas})(\text{dentroDeLasDimensiones}(\text{dimensión}, \text{vértice}, f) \wedge \\ (f \notin \text{aDistanciaMenosDeN}(\text{inicio}, f, 3) \vee (\text{inicio} \in \text{chocolates}))) \wedge \\ (\forall c \in \text{chocolates})(\text{dentroDeLasDimensiones}(\text{dimensión}, \text{vértice}, c)) \wedge \\ (\forall p \in \text{paredes})(\text{dentroDeLasDimensiones}(\text{dimensión}, \text{vértice}, p)) \end{array} \right\}$$

**otras operaciones**

$\text{casillerosLibres} : \text{mapa} \rightarrow \text{conj}(\text{casillero})$   
 $\text{dentroDeLasDimensiones} : \text{tupla}(\text{int} \times \text{int}) \times \text{casillero} \times \text{casillero} \rightarrow \text{bool}$   
 $\text{casillerosADerecha} : \text{casillero } c \times \text{mapa } m \rightarrow \text{conj}(\text{casillero})$   
 $\{\text{dentroDeLasDimensiones}(\text{dimensión}(m), \text{vértice}(m), c)\}$   
 $\text{casillerosIzquierda} : \text{casillero } c \times \text{mapa } m \rightarrow \text{conj}(\text{casillero})$   
 $\{\text{dentroDeLasDimensiones}(\text{dimensión}(m), \text{vértice}(m), c)\}$   
 $\text{casillerosHorizontales} : \text{casillero } c \times \text{mapa } m \rightarrow \text{conj}(\text{casillero})$   
 $\{\text{dentroDeLasDimensiones}(\text{dimensión}(m), \text{vértice}(m), c)\}$   
 $\text{casillerosArriba} : \text{casillero } c \times \text{mapa } m \rightarrow \text{conj}(\text{casillero})$   
 $\{\text{dentroDeLasDimensiones}(\text{dimensión}(m), \text{vértice}(m), c)\}$

$\text{casillerosAbajo} : \text{casillero } c \times \text{mapa } m \longrightarrow \text{conj}(\text{casillero})$   
 $\{\text{dentroDeLasDimensiones}(\text{dimensión}(m), \text{vértice}(m), c)\}$   
 $\text{casillerosVerticales} : \text{casillero } c \times \text{mapa } m \longrightarrow \text{conj}(\text{casillero})$   
 $\{\text{dentroDeLasDimensiones}(\text{dimensión}(m), \text{vértice}(m), c)\}$   
 $\text{generarCasillerosHorizontales} : \text{mapa } m \times \text{conj}(\text{casillero}) \ C \longrightarrow \text{conj}(\text{casillero})$   
 $\{(\forall c \in C)(\text{dentroDeLasDimensiones}(\text{dimensión}(m), \text{vértice}(m), c))\}$

**axiomas**

$\text{conjFantasmas}(\text{nuevoMapa}(\text{dimensión}, \text{vértice}, \text{inicio}, \text{fin}, \text{fantasmas}, \text{paredes}, \text{chocolates})) \equiv \text{fantasmas}$   
 $\text{conjParedes}(\text{nuevoMapa}(\text{dimensión}, \text{vértice}, \text{inicio}, \text{fin}, \text{fantasmas}, \text{paredes}, \text{chocolates})) \equiv \text{paredes}$   
 $\text{conjChocolates}(\text{nuevoMapa}(\text{dimensión}, \text{vértice}, \text{inicio}, \text{fin}, \text{fantasmas}, \text{paredes}, \text{chocolates})) \equiv \text{chocolates}$   
 $\text{dimensiones}(\text{nuevoMapa}(\text{dimensión}, \text{vértice}, \text{inicio}, \text{fin}, \text{fantasmas}, \text{paredes}, \text{chocolates})) \equiv \text{dimensión}$   
 $\text{casilleroInicial}(\text{nuevoMapa}(\text{dimensión}, \text{vértice}, \text{inicio}, \text{fin}, \text{fantasmas}, \text{paredes}, \text{chocolates})) \equiv \text{inicio}$   
 $\text{casilleroDeLlegada}(\text{nuevoMapa}(\text{dimensión}, \text{vértice}, \text{inicio}, \text{fin}, \text{fantasmas}, \text{paredes}, \text{chocolates})) \equiv \text{fin}$   
 $\text{vértice}(\text{nuevoMapa}(\text{dimensión}, \text{vértice}, \text{inicio}, \text{fin}, \text{fantasmas}, \text{paredes}, \text{chocolates})) \equiv \text{vértice}$   
 $\text{dentroDeLasDimensiones}(\text{dimensión}, \text{vértice}, \text{casilla}) \equiv (0 \leq \pi_1(\text{casilla}) - \pi_1(\text{vértice}) < \pi_1(\text{dimensión})) \wedge$   
 $(0 \leq \pi_2(\text{casilla}) - \pi_2(\text{vértice}) < \pi_2(\text{dimensión}))$   
 $\text{casillerosLibres}(m) \equiv \text{generarCasillerosHorizontales}(m, \text{casillerosVerticales}(\text{vértice}(m), m)) - (\text{conjFantasmas}(m)$   
 $\cup \text{conjParedes}(m))$   
 $\text{generarCasillerosHorizontales}(m, \text{casilleros}) \equiv \text{if } \text{casilleros} = \emptyset \text{ then}$   
 $\emptyset$   
 $\text{else}$   
 $\text{casillerosHorizontales}(\text{dameUno}(\text{casilleros}, m)) \cup$   
 $\text{generarCasillerosHorizontales}(m, \text{sinUno}(\text{casilleros}))$   
 $\text{fi}$   
 $\text{casillerosHorizontales}(c, m) \equiv \text{casillerosADerecha}(c, m) \cup \text{casillerosAIzquierda}(c, m)$   
 $\text{casillerosADerecha}(c, m) \equiv (\text{if } \text{dentroDeLasDimensiones}(\text{dimensión}(m), \text{vértice}(m), c + \langle 1, 0 \rangle) \text{ then}$   
 $\text{casillerosADerecha}(c + \langle 1, 0 \rangle, m)$   
 $\text{else}$   
 $\emptyset$   
 $\text{fi}) \cup \{c\}$   
 $\text{casillerosAIzquierda}(c, m) \equiv (\text{if } \text{dentroDeLasDimensiones}(\text{dimensión}(m), \text{vértice}(m), c - \langle 1, 0 \rangle) \text{ then}$   
 $\text{casillerosAIzquierda}(c - \langle 1, 0 \rangle, m)$   
 $\text{else}$   
 $\emptyset$   
 $\text{fi}) \cup \{c\}$   
 $\text{casillerosVerticales}(c, m) \equiv \text{casillerosArriba}(c, m) \cup \text{casillerosAbajo}(c, m)$   
 $\text{casillerosArriba}(c, m) \equiv (\text{if } \text{dentroDeLasDimensiones}(\text{dimensión}(m), \text{vértice}(m), c + \langle 0, 1 \rangle) \text{ then}$   
 $\text{casillerosArriba}(c + \langle 0, 1 \rangle, m)$   
 $\text{else}$   
 $\emptyset$   
 $\text{fi}) \cup \{c\}$   
 $\text{casillerosAbajo}(c, m) \equiv (\text{if } \text{dentroDeLasDimensiones}(\text{dimensión}(m), \text{vértice}(m), c - \langle 0, 1 \rangle) \text{ then}$   
 $\text{casillerosAbajo}(c - \langle 0, 1 \rangle, m)$   
 $\text{else}$   
 $\emptyset$   
 $\text{fi}) \cup \{c\}$

**Fin TAD**

**TAD PACALGO2****usa** Mapa**géneros** pacalgo2**exporta** pacalgo2, observadores, generadores**igualdad observacional**

$$(\forall p_1, p_2 : \text{pacalgo2}) \left( p_1 =_{\text{obs}} p_2 \iff \begin{pmatrix} \text{verMapa}(p_1) =_{\text{obs}} \text{verMapa}(p_2) \wedge \\ \text{posiciónActual}(p_1) =_{\text{obs}} \text{posiciónActual}(p_2) \wedge \\ \text{nivelDeChocolate}(p_1) =_{\text{obs}} \text{nivelDeChocolate}(p_2) \wedge \\ \text{chocolatesRestantes}(p_1) =_{\text{obs}} \text{chocolatesRestantes}(p_2) \wedge \\ \text{ganó?}(p_1) =_{\text{obs}} \text{ganó?}(p_2) \wedge \\ (\text{ganó?}(p_1) \Rightarrow_L \text{puntaje}(p_1) =_{\text{obs}} \text{puntaje}(p_2)) \end{pmatrix} \right)$$

**observadores básicos**verMapa : pacalgo2  $\longrightarrow$  mapaposiciónActual : pacalgo2  $\longrightarrow$  casilleropuntaje : pacalgo2  $\longrightarrow$  nat {ganó?(p)}nivelDeChocolate : pacalgo2  $\longrightarrow$  natchocolatesRestantes : pacalgo2  $\longrightarrow$  conj(casillero)**generadores**inicializarJuego : mapa  $\longrightarrow$  pacalgo2arriba : pacalgo2  $p \longrightarrow$  pacalgo2  
 $\{( \text{posiciónActual}(p) + \langle 0, 1 \rangle ) \in \text{direccionesPosibles}(p) \wedge \neg \text{ganó?}(p) \wedge \neg \text{perdió?}(p) \}$ abajo : pacalgo2  $p \longrightarrow$  pacalgo2  
 $\{( \text{posiciónActual}(p) - \langle 0, 1 \rangle ) \in \text{direccionesPosibles}(p) \wedge \neg \text{ganó?}(p) \wedge \neg \text{perdió?}(p) \}$ derecha : pacalgo2  $p \longrightarrow$  pacalgo2  
 $\{( \text{posiciónActual}(p) + \langle 1, 0 \rangle ) \in \text{direccionesPosibles}(p) \wedge \neg \text{ganó?}(p) \wedge \neg \text{perdió?}(p) \}$ izquierda : pacalgo2  $p \longrightarrow$  pacalgo2  
 $\{( \text{posiciónActual}(p) - \langle 1, 0 \rangle ) \in \text{direccionesPosibles}(p) \wedge \neg \text{ganó?}(p) \wedge \neg \text{perdió?}(p) \}$ **otras operaciones**direccionesPosibles : pacalgo2  $\longrightarrow$  conj(casillero)perdió? : pacalgo2  $\longrightarrow$  boolganó? : pacalgo2  $\longrightarrow$  boolpasos : pacalgo2  $\longrightarrow$  nat**axiomas**verMapa(inicializarJuego(m))  $\equiv m$ verMapa(arriba(p))  $\equiv \text{verMapa}(p)$ verMapa(abajo(p))  $\equiv \text{verMapa}(p)$ verMapa(izquierda(p))  $\equiv \text{verMapa}(p)$ verMapa(derecha(p))  $\equiv \text{verMapa}(p)$ posiciónActual(inicializarJuego m)  $\equiv \text{casilleroInicial}(m)$ posiciónActual(arriba(p))  $\equiv \text{posiciónActual}(p) + \langle 0, 1 \rangle$ posiciónActual(abajo(p))  $\equiv \text{posiciónActual}(p) - \langle 0, 1 \rangle$

<code>posiciónActual(izquierda(p))</code>	$\equiv$ <code>posiciónActual(p) - <math>\langle 1, 0 \rangle</math></code>
<code>posiciónActual(derecha(p))</code>	$\equiv$ <code>posiciónActual(p) + <math>\langle 1, 0 \rangle</math></code>
<code>pasos(inicializarJuego m)</code>	$\equiv$ 0
<code>pasos(arriba(p))</code>	$\equiv$ 1 + <code>pasos(p)</code>
<code>pasos(abajo(p))</code>	$\equiv$ 1 + <code>pasos(p)</code>
<code>pasos(izquierda(p))</code>	$\equiv$ 1 + <code>pasos(p)</code>
<code>pasos(derecha(p))</code>	$\equiv$ 1 + <code>pasos(p)</code>
<code>puntaje(p)</code>	$\equiv$ <code>pasos(p)</code>
<code>nivelDeChocolate(inicializarJuego(m))</code>	$\equiv$ <b>if</b> <code>casilleroInicial(m) <math>\in</math> conjChocolates(m)</code> <b>then</b> 10 <b>else</b> 0 <b>fi</b>
<code>nivelDeChocolate(arriba(p))</code>	$\equiv$ <b>if</b> <code>(posiciónActual(p) + <math>\langle 0, 1 \rangle</math>) <math>\in</math> chocolatesRestantes(p)</code> <b>then</b> 10 <b>else</b> <b>if</b> <code>0?(nivelDeChocolate(p))</code> <b>then</b> <code>nivelDeChocolate(p)</code> <b>else</b> <code>nivelDeChocolate(p) - 1</code> <b>fi</b> <b>fi</b>
<code>nivelDeChocolate(abajo(p))</code>	$\equiv$ <b>if</b> <code>(posiciónActual(p) - <math>\langle 0, 1 \rangle</math>) <math>\in</math> chocolatesRestantes(p)</code> <b>then</b> 10 <b>else</b> <b>if</b> <code>0?(nivelDeChocolate(p))</code> <b>then</b> <code>nivelDeChocolate(p)</code> <b>else</b> <code>nivelDeChocolate(p) - 1</code> <b>fi</b> <b>fi</b>
<code>nivelDeChocolate(izquierda(p))</code>	$\equiv$ <b>if</b> <code>(posiciónActual(p) - <math>\langle 1, 0 \rangle</math>) <math>\in</math> chocolatesRestantes(p)</code> <b>then</b> 10 <b>else</b> <b>if</b> <code>0?(nivelDeChocolate(p))</code> <b>then</b> <code>nivelDeChocolate(p)</code> <b>else</b> <code>nivelDeChocolate(p) - 1</code> <b>fi</b> <b>fi</b>
<code>nivelDeChocolate(derecha(p))</code>	$\equiv$ <b>if</b> <code>(posiciónActual(p) + <math>\langle 1, 0 \rangle</math>) <math>\in</math> chocolatesRestantes(p)</code> <b>then</b> 10 <b>else</b> <b>if</b> <code>0?(nivelDeChocolate(p))</code> <b>then</b> <code>nivelDeChocolate(p)</code> <b>else</b> <code>nivelDeChocolate(p) - 1</code> <b>fi</b> <b>fi</b>
<code>chocolatesRestantes(inicializarJuego(m))</code>	$\equiv$ <b>if</b> <code>casilleroInicial(m) <math>\in</math> conjChocolates(m)</code> <b>then</b> <code>conjChocolates(m) - { casilleroInicial(m) }</code> <b>else</b> <code>conjChocolates(m)</code> <b>fi</b>

<code>chocolatesRestantes(arriba(p))</code>	$\equiv$ <b>if</b> ( <code>posiciónActual(p) + <math>\langle 0, 1 \rangle</math></code> ) $\in$ <code>chocolatesRestantes(p)</code> <b>then</b> <code>chocolatesRestantes(p) - { (posiciónActual(p) + <math>\langle 0, 1 \rangle</math>) }</code> <b>else</b> <code>chocolatesRestantes(p)</code> <b>fi</b>
<code>chocolatesRestantes(abajo(p))</code>	$\equiv$ <b>if</b> ( <code>posiciónActual(p) - <math>\langle 0, 1 \rangle</math></code> ) $\in$ <code>chocolatesRestantes(p)</code> <b>then</b> <code>chocolatesRestantes(p) - { (posiciónActual(p) - <math>\langle 0, 1 \rangle</math>) }</code> <b>else</b> <code>chocolatesRestantes(p)</code> <b>fi</b>
<code>chocolatesRestantes(izquierda(p))</code>	$\equiv$ <b>if</b> ( <code>posiciónActual(p) - <math>\langle 1, 0 \rangle</math></code> ) $\in$ <code>chocolatesRestantes(p)</code> <b>then</b> <code>chocolatesRestantes(p) - { (posiciónActual(p) - <math>\langle 1, 0 \rangle</math>) }</code> <b>else</b> <code>chocolatesRestantes(p)</code> <b>fi</b>
<code>chocolatesRestantes(derecha(p))</code>	$\equiv$ <b>if</b> ( <code>posiciónActual(p) + <math>\langle 1, 0 \rangle</math></code> ) $\in$ <code>chocolatesRestantes(p)</code> <b>then</b> <code>chocolatesRestantes(p) - { (posiciónActual(p) + <math>\langle 1, 0 \rangle</math>) }</code> <b>else</b> <code>chocolatesRestantes(p)</code> <b>fi</b>
<code>perdió?(p)</code>	$\equiv$ $\neg \emptyset?(\text{conjFantasmas}(\text{verMapa}(p)) \cap$ $\text{aDistanciaMenosDeN}(\text{posiciónActual}(p), 3)) \wedge$ $0?(\text{nivelDeChocolate}(p))$
<code>ganó?(p)</code>	$\equiv$ <code>posiciónActual(p) = casilleroDeLlegada(verMapa(p))</code>
<code>direccionesPosibles(p)</code>	$\equiv$ <b>if</b> $0?(\text{nivelDeChocolate}(p))$ <b>then</b> $(\text{aDistanciaMenosDeN}(\text{posiciónActual}(p), 1) - \text{posiciónActual}(p))$ $\cap$ $\text{casillerosLibres}(\text{verMapa}(p))$ <b>else</b> $(\text{aDistanciaMenosDeN}(\text{posiciónActual}(p), 1) - \text{posiciónActual}(p))$ $\cap$ $(\text{casillerosLibres}(\text{verMapa}(p)) \cup \text{conjFantasmas}(\text{verMapa}(p)))$ <b>fi</b>

**Fin TAD**

## 2.2. TP 2

TAD persona es String

### TAD RANKING

**usa** Int

**géneros** ranking

**exporta** ranking, generadores, contrincante, superioresInmediatos

#### observadores básicos

$\text{ver} : \text{ranking} \longrightarrow \text{dicc}(\text{persona}, \text{nat})$

#### generadores

$\text{iniciar} : \longrightarrow \text{ranking}$

$\text{cargarPuntaje} : \text{ranking} \times \text{persona} \times \text{nat} \longrightarrow \text{ranking}$

#### otras operaciones

$\text{contrincante} : \text{ranking } r \times \text{persona } j \longrightarrow \text{tupla}(\text{persona}, \text{nat})$   
 $\{\text{def?}(j, \text{ver}(r))\}$

$\text{superioresInmediatos} : \text{ranking } r \times \text{persona } j \longrightarrow \text{conj}(\text{persona})$   
 $\{\text{def?}(j, \text{ver}(r))\}$

$\text{elMaximoEntreMenores} : \text{conj}(\text{nat}) \text{ puntajes} \times \text{nat } p \longrightarrow \text{nat}$

$\text{todosLosPuntajes} : \text{ranking } r \times \text{conj}(\text{persona}) \text{ nombres} \longrightarrow \text{conj}(\text{nat})$   
 $\{\text{nombres} \subseteq \text{claves}(\text{ver}(r))\}$

$\text{jugadoresConPuntajeIgual} : \text{ranking } r \times \text{conj}(\text{persona}) \text{ nombres} \times \text{nat} \longrightarrow \text{conj}(\text{persona})$   
 $\{\text{nombres} \subseteq \text{claves}(\text{ver}(r))\}$

$\text{menoresQueP} : \text{conj}(\text{nat}) \times \text{nat} \longrightarrow \text{conj}(\text{nat})$

$\text{maximoConj} : \text{conj}(\text{nat}) \text{ } C \longrightarrow \text{nat} \quad \{\neg \emptyset?(C)\}$

#### axiomas

$\text{ver}(\text{iniciar}()) \equiv \text{vacío}$

$\text{ver}(\text{cargarPuntaje}(\text{ranking}, \text{nombre}, \text{puntaje})) \equiv \text{if } \text{def?}(\text{nombre}, \text{ver}(\text{ranking})) \text{ then}$   
 $\quad \text{if } \text{puntaje} < \text{obtener}(\text{ver}(r), \text{nombre}) \text{ then}$   
 $\quad \quad \text{definir}(\text{ver}(\text{ranking}), (\text{nombre}, \text{puntaje}))$   
 $\quad \text{else}$   
 $\quad \quad \text{ver}(\text{ranking})$   
 $\quad \text{fi}$   
 $\text{else}$   
 $\quad \text{definir}(\text{ver}(\text{ranking}), (\text{nombre}, \text{puntaje}))$   
 $\text{fi}$

$\text{contrincante}(r, j) \equiv \text{dameUno}(\text{superioresInmediatos}(r, j))$

$\text{superioresInmediatos}(r, j) \equiv \text{jugadoresConPuntajeIgual}(r, \text{claves}(\text{ver}(r)),$   
 $\text{elMaximoEntreMenores}(\text{todosLosPuntajes}(r),$   
 $\text{obtener}(\text{ver}(r), j))$

$\text{elMaximoEntreMenores}(\text{puntajes}, p) \equiv \text{if } \emptyset?(\text{menoresQueP}(\text{puntajes}, p)) \text{ then}$   
 $\quad p$   
 $\text{else}$   
 $\quad \text{maximoConj}(\text{menoresQueP}(\text{puntajes}, p))$   
 $\text{fi}$



menoresQueP( <i>puntajes</i> , <i>p</i> )	$\equiv$ <b>if</b> $\emptyset?(puntajes)$ <b>then</b> $\emptyset$ <b>else</b> <b>if</b> $dameUno(puntajes) < p$ <b>then</b> $Ag(dameUno(puntajes),$ $menoresQueP(sinUno(puntajes), p))$ <b>else</b> $menoresQueP(sinUno(puntajes), p)$ <b>fi</b> <b>fi</b>
maximoConj( <i>C</i> )	$\equiv$ <b>if</b> $\emptyset?(sinUno(C))$ <b>then</b> $dameUno(C)$ <b>else</b> $m\acute{a}x(dameUno(C), maximoConj(sinUno(C)))$ <b>fi</b>
todosLosPuntajes( <i>r</i> )	$\equiv$ <b>if</b> $\emptyset?(ver(r))$ <b>then</b> $\emptyset$ <b>else</b> $Ag(obtener(ver(r), dameUno(claves(ver(r)))),$ $todosLosPuntajes(r))$ <b>fi</b>
jugadoresConPuntajeIgual( <i>r</i> , <i>nombres</i> , <i>puntaje</i> )	$\equiv$ <b>if</b> $\emptyset?(nombres) \vee \emptyset?(claves(ver(r)))$ <b>then</b> $\emptyset$ <b>else</b> <b>if</b> $obtener(ver(r), dameUno(nombres)) = puntaje$ <b>then</b> $Ag(dameUno(nombres),$ $jugadorConPuntajeIgual(r, sinUno(nombres),$ $puntaje))$ <b>else</b> $jugadorConPuntajeIgual(r, sinUno(nombres), puntaje)$ <b>fi</b> <b>fi</b>

### Fin TAD

TAD FICHÍN

<b>usa</b>	Int, Ranking, Paclgo2
<b>géneros</b>	fichin
<b>exporta</b>	observadores, generadores, otras operaciones

## observadores básicos

conocerRanking	:	fichin	→	ranking	
verJuegoActual	:	fichin $f$	→	paclgo2	$\{-\text{estáLibre?}(f)\}$
jugadorActual	:	fichin $f$	→	persona	$\{-\text{estáLibre?}(f)\}$

generadores

nuevoFichin	:	$\longrightarrow$	fichin	
nuevaPartida	:	fichin $f \times$ pacalgo2 $p \times$ persona $j$	$\longrightarrow$	fichin $\{est\acute{a}Libre?(f)\}$
palancaArriba	:	fichin $f$	$\longrightarrow$	fichin $\{\neg est\acute{a}Libre?(f)\}$
palancaAbajo	:	fichin $f$	$\longrightarrow$	fichin $\{\neg est\acute{a}Libre?(f)\}$
palancaDerecha	:	fichin $f$	$\longrightarrow$	fichin $\{\neg est\acute{a}Libre?(f)\}$

palancaIzquierda : fichin  $f \rightarrow$  fichin { $\neg$ estáLibre?( $f$ )}

#### otras operaciones

verContrincante : fichin  $f \rightarrow$  persona {jugadorActual( $f$ ) $\in$ claves(ver(conocerRanking( $f$ )))}

estáLibre? : fichin  $\rightarrow$  Bool

#### axiomas

estáLibre?( $f$ )  $\equiv$  ganó?(verJuegoActual( $f$ )) $\vee$  perdió?(verJuegoActual( $f$ ))

conocerRanking(nuevoFichin())  $\equiv$  iniciar()

conocerRanking(nuevaPartida( $f, p, j$ ))  $\equiv$   
conocerRanking( $f$ )

conocerRanking(palancaArriba( $f$ ))  $\equiv$  **if** ganó?(arriba(verJuegoActual( $f$ ))) **then**  
    cargarPuntaje(conocerRanking( $f$ ),  
    puntaje(verJuegoActual(palancaArriba( $f$ ))))  
**else**  
    conocerRanking( $f$ )  
**fi**

conocerRanking(palancaAbajo( $f$ ))  $\equiv$  **if** ganó?(abajo(verJuegoActual( $f$ ))) **then**  
    cargarPuntaje(conocerRanking( $f$ ),  
    puntaje(verJuegoActual(palancaAbajo( $f$ ))))  
**else**  
    conocerRanking( $f$ )  
**fi**

conocerRanking(palancaDerecha( $f$ ))  $\equiv$  **if** ganó?(derecha(verJuegoActual( $f$ ))) **then**  
    cargarPuntaje(conocerRanking( $f$ ),  
    puntaje(verJuegoActual(palancaDerecha( $f$ ))))  
**else**  
    conocerRanking( $f$ )  
**fi**

conocerRanking(palancaIzquierda( $f$ ))  $\equiv$  **if** ganó?(izquierda(verJuegoActual( $f$ ))) **then**  
    cargarPuntaje(conocerRanking( $f$ ),  
    puntaje(verJuegoActual(palancaIzquierda( $f$ ))))  
**else**  
    conocerRanking( $f$ )  
**fi**

verJuegoActual(nuevaPartida( $f, p, j$ ))  $\equiv p$

verJuegoActual(palancaArriba( $f$ ))  $\equiv$  arriba(verJuegoActual( $f$ ))

verJuegoActual(palancaAbajo( $f$ ))  $\equiv$  abajo(verJuegoActual( $f$ ))

verJuegoActual(palancaDerecha( $f$ ))  $\equiv$  derecha(verJuegoActual( $f$ ))

verJuegoActual(palancaIzquierda( $f$ ))  $\equiv$  izquierda(verJuegoActual( $f$ ))

**Fin TAD**