

Trabajo Práctico I

PokemonGo, Especificación

Algoritmos y Estructuras de Datos II
Segundo Cuatrimestre de 2016

Grupo: Wololo

Integrante	LU	Correo electrónico
Alejandro Dario Echeverri	939/05	<code>ale_echeverri@yahoo.com.ar</code>
Lucas Monzon	785/14	<code>lucasmonzon94@gmail.com</code>
Luis Arroyo	913/13	<code>Luis.arroyo.90@gmail.com</code>
Manuel Cafferata	592/14	<code>manucaferacing@gmail.com</code>

Facultad de Ciencias Exactas y Naturales
Universidad de Buenos Aires

Ciudad Universitaria - (Pabellón I/Planta Baja)

Intendente Güiraldes 2160 - C1428EGA

Ciudad Autónoma de Buenos Aires - Rep. Argentina

Tel/Fax: (54 11) 4576-3359

<http://www.fcen.uba.ar>

Índice

1. TAD MAPA	3
2. TAD VIDEOJUEGO	5
3. Extensión TAD NAT	12

1. TAD MAPA

Posición es Tupla $\langle \text{Nat}, \text{Nat} \rangle$

TAD MAPA

géneros map

exporta generadores, observadores

usa NAT, BOOL

igualdad observacional

$$(\forall m_1, m_2 : \text{map}) \left(m_1 =_{\text{obs}} m_2 \iff \left(\text{Posiciones}(m_1) = \text{Posiciones}(m_2) \wedge_L \left(\forall p_1, p_2 : \text{posición} \right) (p_1, p_2 \in \text{Posiciones}(m_1) \Rightarrow_L \text{Conectadas?}(m_1, p_1, p_2) = \text{Conectadas?}(m_2, p_1, p_2)) \right) \right)$$

observadores básicos

Posiciones : map \longrightarrow conj(posición)

Conectadas? : map $m \times$ posición $p_1 \times$ posición $p_2 \longrightarrow$ bool $\{p_1 \neq p_2 \wedge \{p_1, p_2\} \subseteq \text{Posiciones}(m)\}$

generadores

NuevoMapa : conj(posición) $ps \longrightarrow$ map $\{\text{CoordenadasPositivas}(ps)\}$

Conectar : map $m \times$ posición $p_1 \times$ posición $p_2 \longrightarrow$ map
 $\{p_1 \neq p_2 \wedge \{p_1, p_2\} \subseteq \text{Posiciones}(m) \wedge_L \neg \text{Conectadas?}(m, \{p_1, p_2\})\}$

otras operaciones

CoordenadasPositivas : conj(posición) \longrightarrow bool

Conectadas : map $m \times$ posición $p \times$ conj(posición) $ps \longrightarrow$ conj(posición) $\{\{p\} \cup ps \subseteq \text{Posiciones}(m)\}$

HayCamino? : map $m \times$ posición $p_1 \times$ posición $p_2 \longrightarrow$ bool $\{p_1 \neq p_2 \wedge \{p_1, p_2\} \subseteq \text{Posiciones}(m)\}$

Caminos : map $m \times$ posición $p \longrightarrow$ conj(posición) $\{p \in \text{Posiciones}(m)\}$

CaminosAux : map $m \times$ posición $p \times$ conj(posición) $ps \longrightarrow$ conj(posición) $\{\{p\} \cup ps \subseteq \text{Posiciones}(m)\}$

CaminosConectados : map $m \times$ conj(posición) $ps \times$ conj(posición) $visitadas \longrightarrow$ conj(posición)
 $\{ps \cup visitadas \subseteq \text{Posiciones}(m)\}$

NormaAlCuadrado : posición \times posición \longrightarrow nat

PosicionesNorma2Menor : map $m \times$ posición $p \times$ nat $n \longrightarrow$ conj(posición) $\{p \in \text{Posiciones}(m)\}$

ObtenerPosicionesNormaMenor : conj(posición) \times posición \times nat \longrightarrow conj(posición)

axiomas $\forall m : \text{map}, \forall p, p_1, p_2, p_3, p_4 : \text{posición}, \forall ps, visitadas : \text{conj}(\text{posición}), \forall n, m : \text{nat}$

CoordenadasPositivas(ps) \equiv **if** $\emptyset ?(ps)$ **then**
 true
else
 if $\pi_1(\text{dameUno}(ps)) > 0 \wedge \pi_2(\text{dameUno}(ps)) > 0$ **then**
 CoordenadasPositivas(sinUno(ps))
 else
 false
 fi
fi

Posiciones(NuevoMapa(ps)) \equiv ps

Posiciones(Conectar(m, p_1 , p_2)) \equiv Posiciones(m)

```

Conectadas?(NuevoMapa(ps), p1, p2) ≡ false
Conectadas?(Conectar(m,p1, p2),p3, p4) ≡ if (p3 = p1 ∧ p4 = p2) ∨ (p4 = p1 ∧ p3 = p2) then
    true
    else
        Conectadas?(m, p3, p4)
    fi

Conectadas(m, p, ps) ≡ if ∅?(ps) then
    ∅
    else
        if Conectadas?(m,dameUno(ps), p) then
            Ag(dameUno(ps), Conectadas(m, p, sinUno(ps)))
        else
            Conectadas(m, p, sinUno(ps))
        fi
    fi

HayCamino?(m, p1, p2) ≡ p2 ∈ Caminos(m, p1)
Caminos(m, p) ≡ CaminosAux(m,p,∅)
CaminosAux(m, p,visitadas) ≡ (Conectadas(m,p) - visitadas) ∪ CaminosConectados (m,Conectadas(m,p) -
    visitadas, Ag(p,visitadas))
CaminosConectados(m, ps,visitadas) ≡ if ∅?(ps) then
    ∅
    else
        CaminosAux(m,dameUno(ps),visitadas)
        ∪ CaminosConectados(m,sinUno(ps),visitadas)
    fi

NormaAlCuadrado(p1,p2) ≡ (ValorAbsResta(π1(p1)), π1(p2))2 + (ValorAbsResta(π2(p1)), π2(p2))2
PosicionesNorma2Menor(m,p,n) ≡ ObtenerPosicionesNormaMenor(Posiciones(m),p,n)
ObtenerPosicionesNormaMenor(ps,p,n) ≡ if ∅?(ps) then
    ∅
    else
        if NormaAlCuadrado(p,dameUno(ps)) < n2 then
            Ag(dameUno(ps),ObtenerPosicionesNormaMenor(sinUno(ps),p,n))
        else
            ObtenerPosicionesNormaMenor(sinUno(ps),p,n)
        fi
    fi

```

Fin TAD

2. TAD VIDEOJUEGO

Pokemon es String

Jugador es Nat

TAD VIDEOJUEGO

géneros vg

exporta generadores, observadores

usa NAT, BOOL, STRING, MAPA

igualdad observacional

$$(\forall v, v' : \text{vg}) \left(v =_{\text{obs}} v' \iff \left(\begin{array}{l} \text{Mapa}(v) = \text{Mapa}(v') \wedge \text{Salvajes}(v) = \text{Salvajes}(v') \wedge \text{Jugadores}(v) = \text{Jugadores}(v') \wedge_L (\forall j : \text{Jugador}) j \in v \rightarrow_L (\text{Capturados}(v, j) = \text{Capturados}(v', j) \wedge \text{Ubicacion}(v, j) = \text{Ubicacion}(v', j) \wedge \text{Sanciones}(v, j) = \text{Sanciones}(v', j)) \wedge \text{JugadoresConectados}(v) = \text{JugadoresConectados}(v') \end{array} \right) \right)$$

observadores básicos

Mapa : vg \rightarrow map

Jugadores : vg \rightarrow conj(jugador)

Salvajes : vg \rightarrow dicc(pokemon, conj(posición))

Capturados : vg $v \times \text{jugador } j \rightarrow \text{multiconj}(\text{pokemon})$ $\{j \in \text{Jugadores}(v)\}$

Ubicación : vg $v \times \text{jugador } j \rightarrow \text{posición}$ $\{j \in \text{Jugadores}(v)\}$

Sanciones : vg $v \times \text{jugador } j \rightarrow \text{nat}$ $\{j \in \text{Jugadores}(v)\}$

JugadoresConectados : vg \rightarrow conj(jugador)

TurnosRestantes : vg $v \times \text{jugador } j \times \text{posición } p \rightarrow \text{nat}$
 $\left\{ \begin{array}{l} j \in \text{JugadoresConectados}(v) \wedge j \in \text{Jugadores}(v) \wedge p \in \text{Posiciones}(\text{Mapa}(v)) \wedge_L \\ \text{Sanciones}(v, j) < 5 \wedge \text{Ubicación}(v, j) \in \text{PosicionesNorma2Menor}(\text{Mapa}(v), p, 2) \wedge \text{HayPoke-} \\ \text{mon?}(v, p) \end{array} \right\}$

generadores

NuevoJuego : map \rightarrow vg

AgregarJugador : vg $v \times \text{jugador } j \rightarrow \text{vg}$ $\{j \notin \text{Jugadores}(v)\}$

AgregarPokemon : vg $v \times \text{pokemon } pkm \times \text{posición } p \rightarrow \text{vg}$
 $\left\{ \begin{array}{l} p \in \text{Posiciones}(\text{Mapa}(v)) \wedge (\forall p' : \text{posición}) (p' \in \text{Posiciones}(\text{Mapa}(v)) \wedge_L p' \in \text{PosicionesNorma-} \\ \text{Menor}(\text{Mapa}(v), p, 5)) \Rightarrow_L \neg \text{HayPokemon}(v, p') \end{array} \right\}$

Conectar : vg $v \times \text{jugador } j \times \text{posición } p \rightarrow \text{vg}$
 $\{j \in \text{Jugadores}(v) \wedge_L \text{Sanciones}(v, j) < 5 \wedge j \notin \text{Conectados}(v) \wedge p \in \text{Posiciones}(\text{Mapa}(v))\}$

Desconectar : vg $v \times \text{jugador } j \rightarrow \text{vg}$ $\{j \in \text{Jugadores}(v) \wedge_L \text{Sanciones}(v, j) < 5 \wedge j \in \text{Conectados}(v)\}$

Mover : vg $v \times \text{jugador } j \times \text{posición } p \rightarrow \text{vg}$
 $\{p \in \text{Posiciones}(\text{Mapa}(v)) \wedge j \in \text{Conectados}(v) \wedge j \in \text{Jugadores}(v) \wedge_L \text{Sanciones}(v, j) < 5\}$

otras operaciones

Rareza : vg $v \times \text{pokemon } pkm \rightarrow \text{nat}$ $\{\text{TotalPokemons}(v) > 0\}$

MovimientoVálido : vg $v \times \text{posición } p_1 \times \text{posición } p_2 \rightarrow \text{bool}$
 $\{p_1 \neq p_2 \wedge \{p_1, p_2\} \subseteq \text{Posiciones}(\text{Mapa}(v))\}$

HayPokemon? : vg $v \times \text{posición } p \rightarrow \text{bool}$ $\{p \in \text{Posiciones}(\text{Mapa}(v))\}$

TienePokemon? : vg $v \times \text{posición } p \times \text{conj}(\text{pokemon}) \text{ } cpkm \rightarrow \text{bool}$

$$\{p \in \text{Posiciones}(\text{Mapa}(v)) \wedge \text{cpkm} \subseteq \text{claves}(\text{ps})\}$$

HayPokemonEnRango? : $\text{vg } v \times \text{posición } p \times \text{conj}(\text{posición}) \text{ps} \rightarrow \text{bool}$
 $\{p \in \text{Posiciones}(\text{Mapa}(v)) \wedge \text{ps} \subseteq \text{Posiciones}(\text{Mapa}(v))\}$

PokemonEnPosición : $\text{vg } v \times \text{posición } p \times \text{conj}(\text{pokemon}) \text{cpkm} \rightarrow \text{pokemon}$
 $\{p \in \text{Posiciones}(\text{Mapa}(v)) \wedge \text{cpkm} \subseteq \text{claves}(\text{Salvajes}(v)) \wedge \text{HayPokemonEnRango?}(v, p, \text{cpkm})\}$

PosiciónPokemon : $\text{vg } v \times \text{posición } p \times \text{conj}(\text{posición}) \text{ps} \rightarrow \text{posición}$
 $\{\text{HayPokemonEnRango?}(v, p, \text{ps})\}$

TotalCapturados : $\text{vg } v \times \text{conj}(\text{jugador}) \text{js} \rightarrow \text{nat}$ $\{\text{js} \subseteq \text{Jugadores}(v)\}$

TotalSalvajes : $\text{vg } v \times \text{conj}(\text{pokemon}) \text{cpkm} \rightarrow \text{nat}$ $\{\text{cpkm} \subseteq \text{claves}(\text{Salvajes}(v))\}$

TotalPokemons : $\text{vg} \rightarrow \text{nat}$

TotalSalvajesTipoP : $\text{vg} \times \text{pokemon} \rightarrow \text{nat}$

TotalCapturadosTipoP : $\text{vg } v \times \text{pokemon } \text{pkm} \times \text{conj}(\text{jugador}) \text{js} \rightarrow \text{nat}$ $\{\text{js} \subseteq \text{Jugadores}(v)\}$

TotalTipoP : $\text{vg} \times \text{pokemon} \rightarrow \text{nat}$

PosiciónEnRango : $\text{vg } v \times \text{jugador } j \times \text{conjunto}(\text{posición}) \text{ps} \rightarrow \text{posición}$
 $\left\{ \begin{array}{l} j \in \text{Jugadores}(v) \wedge j \in \text{JugadoresConectados}(v) \wedge \text{ps} \subseteq \text{Posiciones}(\text{Mapa}(v)) \wedge_L (\exists p:\text{posición}) \\ ((p \in \text{ps} \Rightarrow \text{HayPokemon?}(v,p)) \end{array} \right\}$

PosicionesPokemon : $\text{vg} \rightarrow \text{conj}(\text{posición})$

BuscarPosicionesPokemon : $\text{vg } v \times \text{conj}(\text{pokemon}) \text{cpkm} \rightarrow \text{conj}(\text{posición})$
 $\{\text{cpkm} \subseteq \text{claves}(\text{Salvajes}(v))\}$

JugadoresEnRango : $\text{vg} \rightarrow \text{dicc}(\text{posición}, \text{jugador})$

PosiblesCaptores : $v \times \text{ps} \times \text{js} \rightarrow \text{dicc}(\text{posición}, \text{jugador})$
 $\left\{ \begin{array}{l} \text{ps} \subseteq \text{Posiciones}(\text{Mapa}(v)) \wedge (\forall j:\text{jugador}) (j \in \text{js} \Rightarrow (j \in \text{jugadores}(v) \wedge_L \text{Sanciones}(v,j) < 5 \wedge \\ j \in \text{JugadoresConectados}(v))) \end{array} \right\}$

ConectadosYNoEliminados : $\text{vg} \rightarrow \text{conj}(\text{jugador})$

ConectadosYNoEliminadosAux : $\text{vg } v \times \text{conj}(\text{jugador}) \text{js} \rightarrow \text{conj}(\text{jugador})$ $\{\text{js} \subseteq \text{jugadores}(v)\}$

TienenPokemonCerca : $\text{vg } v \times \text{posición } p \times \text{conj}(\text{jugador}) \text{js} \rightarrow \text{conj}(\text{jugador})$
 $\left\{ \begin{array}{l} p \in \text{Posiciones}(\text{Mapa}(v)) \wedge (\forall j:\text{jugador}) (j \in \text{js} \Rightarrow (j \in \text{jugadores}(v) \wedge \\ j \in \text{JugadoresConectados}(v))) \end{array} \right\}$

MovimientoEnRango : $\text{vgv} \times \text{jugadorj1} \times \text{jugadorj2} \times \text{posiciónp} \rightarrow \text{bool}$
 $\{\{j1,j2\} \subseteq \text{jugadores}(v) \wedge p \in \text{Posiciones}(\text{Mapa}(v)) \wedge_L \{j1,j2\} \subseteq \text{conectados}(v)\}$

MovidaNoEnRangoFaltanTurnos : $\text{vg} \times \text{jugador} \times \text{jugador} \times \text{posición} \rightarrow \text{bool}$
 $\{\{j1,j2\} \subseteq \text{jugadores}(v) \wedge p \in \text{Posiciones}(\text{Mapa}(v)) \wedge_L \{j1,j2\} \subseteq \text{conectados}(v)\}$

TurnosCeroSinPokemon : $\text{vg } v \times \text{jugador } j_1 \times \text{jugador } j_2 \times \text{posición } p \rightarrow \text{bool}$
 $\{\{j1,j2\} \subseteq \text{jugadores}(v) \wedge p \in \text{Posiciones}(\text{Mapa}(v)) \wedge_L \{j1,j2\} \subseteq \text{conectados}(v)\}$

HayPokemonCerca? : $\text{vg } v \times \text{jugador } j \rightarrow \text{bool}$ $\{j \in \text{Jugadores}(v)\}$

PuedenAtrapar : $\text{vg } v \times \text{posición } p \times \text{conj}(\text{jugador}) \text{js} \rightarrow \text{conj}(\text{jugador})$
 $\left\{ \begin{array}{l} p \in \text{Posiciones}(\text{Mapa}(v)) \wedge_L (\forall j:\text{jugador}) (j \in \text{js} \Rightarrow (j \in \text{JugadoresConectados}(v) \wedge j \in \text{Jugado-} \\ \text{res}(v) \wedge_L \\ \text{Sanciones}(v,j) < 5 \wedge \text{Ubicación}(v,j) \in \text{PosicionesNorma2Menor}(\text{Mapa}(v),p,2)) \wedge \text{HayPoke-} \\ \text{mon?}(v,p) \end{array} \right\}$

PosicionesCaptura : $\text{vg } v \times \text{posición } p \rightarrow \text{conj}(\text{posición})$ $\{p \in \text{Posiciones}(\text{Mapa}(v))\}$

LejanasConPokemonAtrapable : $\text{vg } v \times \text{conj}(\text{posición}) \text{ps} \times \text{dicc}(\text{posición} \times \text{conj}(\text{jugador})) dH \rightarrow \text{conj}(\text{posición})$

Capturados(Desconectar(v, j_1), j_2) \equiv Capturados(v, j_2)
 Capturados(Mover(v, j_1, p), j_2) \equiv Capturados(v, j_2) \cup
 (**if** $j_1 = j_2 \vee j_2 \notin \text{JugadoresConectados}(v) \vee \text{MovimientoEnRango}(v, j_1, j_2, p)$
 \vee $\text{MovidaNoEnRangoFaltanTurnos}(v, j_1, j_2, p)$ \vee
 $\text{TurnosCeroSinPokemon}(v, j_1, j_2, p) \vee \neg \text{HayPokemonCerca?}(v, j_2)$ **then**
 \emptyset
 else
 { PokemonEnPos(v , PosiciónEnRango(v, j_2 , PosicionesPokemon(v))) }
 fi)

Ubicación(AgregarJugador(v, j_1), j_2) \equiv Ubicación(v, j_2)
 Ubicación(AgregarPokemon(v, pkm, p), j) \equiv Ubicación(v, j)
 Ubicación(Conectar(v, j_1, p), j_2) \equiv **if** $j_1 = j_2$ **then** p **else** Ubicación(v, j_2) **fi**
 Ubicación(Desconectar(v, j_1), j_2) \equiv Ubicación(v, j_2)
 Ubicación(Mover(v, j_1, p), j_2) \equiv **if** $j_1 = j_2$ **then** p **else** Ubicación(v, j_2) **fi**

Sanciones(AgregarJugador(v, j_1), j_1) \equiv **if** $j_1 = j_2$ **then** 0 **else** Sanciones(v, j_2) **fi**
 Sanciones(AgregarPokemon(v, pkm, p), j) \equiv Sanciones(v, j)
 Sanciones(Conectar(v, j_1, p), j_2) \equiv Sanciones(v, j_2)
 Sanciones(Desconectar(v, j_2), j_2) \equiv Sanciones(v, j_2)
 Sanciones(Mover(v, j_1, p), j_2) \equiv Sanciones(v, j_2) +
 (**if** $j_1 = j_2 \wedge \neg \text{MovimientoVálido}(v, \text{Ubicación}(v, j_2), p)$ **then**
 1
 else
 0
 fi)

JugadoresConectados(NuevoJuego(m)) $\equiv \emptyset$
 JugadoresConectados(AgregarJugador(v, j)) \equiv JugadoresConectados(v)
 JugadoresConectados(AgregarPokemon(v, pkm, p)) \equiv JugadoresConectados(v)
 JugadoresConectados(Conectar(v, j, p)) \equiv Ag(j , JugadoresConectados(v))
 JugadoresConectados(Desconectar(v, j)) \equiv JugadoresConectados(v) - { j }
 JugadoresConectados(Mover(v, j, p)) \equiv JugadoresConectados(v)

TurnosRestantes(AgregarJugador(v, j_1), j_2, p) \equiv TurnosRestantes(j_2, p)
 TurnosRestantes(AgregarPokemon(v, pkm, p_1), j, p_2) \equiv **if** $p_2 \in \text{PosicionesNorma2Menor}(\text{Mapa}(v), p_1, 2)$ **then**
 10
 else
 TurnosRestantes(v, j, p_2)
 fi
 TurnosRestantes(Conectar(v, j_1, p_1), j_2, p_2) \equiv **if** $j_1 = j_2$ **then**
 10
 else
 if $\text{MovimientoEnRango}(v, j_1, j_2, p_1)$ **then**
 10
 else
 TurnosRestantes(v, j_2, p_2)
 fi TurnosRestantes(v, j_2, p_2)
 fi
 TurnosRestantes(Desconectar(v, j_1), j_2, p) \equiv TurnosRestantes(j_2, p)


```

TurnosRestantes(Mover(v,j1,p1),j2,p2) ≡ if j1 = j2 then
    TurnosRestantes(v,j2,p2)
else
    if p1 ∈ PosicionesNorma2Menor(Mapa(v),PosiciónEnRango(v,j2,
    PosicionesNorma2Menor(Mapa(v),Ubicación(v,j2),2)),2) then
        10
    else
        TurnosRestantes(v,j2,p2) - 1
    fi
fi

Rareza(v,pkm) ≡ 100 - ( TotalTipoP(v,pkm) / TotalPokemons(v) ) * 100

MovimientoVálido?(v, p1, p2) ≡ HayCamino?(Mapa(v), p1, p2) ∧ 102 > NormaAlCuadrado(p2,p1)

HayPokemon?(v, p) ≡ TienePokemon?(v,p,claves(Salvajes(v)))

TienePokemon?(v, p,cpkm) ≡ if ∅?(cpkm) then
    false
else
    if p ∈ obtener(dameUno(cpkm),Salvajes(v)) then
        true
    else
        TienePokemon?(v, p,sinUno(cpkm))
    fi
fi

PokemonEnPosición(v,p,cpkm) ≡ if p ∈ Obtener(DameUno(cpkm), Salvajes(v)) then
    DameUno(cpkm)
else
    PokemonEnPosición(v, p, cpkm)
fi

HayPokemonEnRango?(v, p, ps) ≡ if ∅?(ps) then
    HayPokemon?(v, p)
else
    HayPokemon?(v, DameUno(ps)) ∨ HayPokemonEnRango?(v, p, SinUno(ps))
fi

PosiciónPokemon(v, p, ps) ≡ if ∅?(ps) then
    p
else
    if HayPokemon?(v, DameUno(ps)) then
        DameUno(ps)
    else
        PosiciónPokemon(v, p, SinUno(ps))
    fi
fi

```

```

TotalCapturados(v,js) ≡ if ∅?(js) then
    0
else
    TotalCapturados(v,sinUno(js)) +
    (if Sanciones(v,dameUno(js)) < 5 then
        #(Capturados(v,dameUno(js)))
    else
        0
    fi)
fi

TotalSalvajes(v,cpkm) ≡ if ∅?(cpkm) then
    0
else
    TotalSalvajes(v,sinUno(cpkm)) + #(obtener(dameUno(cpkm),Salvajes(v)))
fi

TotalPokemons(v) ≡ TotalCapturados(v,jugadores(v)) + TotalSalvajes(v,claves(Salvajes(v)))

TotalSalvajesTipoP(v,pkm) ≡ if def?(pkm,Salvajes(v)) then #obtener(pkm,Salvajes(v)) else 0 fi

TotalCapturadosTipoP(v,js) ≡ if ∅?(js) then
    0
else
    TotalCapturadosTipoP(v,sinUno(js)) +
    (if Sanciones(v,dameUno(js)) < 5 then
        #(pkm,Capturados(v,dameUno(js)))
    else
        0
    fi)
fi

TotalTipoP(v,pkm) ≡ TotalCapturadosTipoP(v,Jugadores(v)) + TotalSalvajesTipoP(v,pkm)

PosiciónEnRango(v,j,ps) ≡ if NormaAlCuadrado(dameUno(ps),Ubicación(v,j)) < 4 ∧ HayPoke-
    mon?(v,dameUno(ps)) then
        dameUno(ps)
    else
        PosiciónEnRango(v,j,sinUno(ps))
fi

JugadoresEnRango(v) ≡ PosiblesCaptores(v,Posiciones(Mapa(v)),ConectadosYNoEliminados(v))

PosicionesPokemon(v) ≡ BuscarPosicionesPokemon(v,claves(Salvajes(v)))

BuscarPosicionesPokemon(v,cpkm) ≡ if ∅?(cpkm) then
    ∅
else
    obtener(dameUno(cpkm),Salvajes(v))
    ∪ BuscarPosicionesPokemon(v,sinUno(cpkm))
fi

```

```

PosiblesCaptore(v,ps,js)  $\equiv$  if  $\emptyset?(ps)$  then
    vacío
else
    if HayPokemon?(v,dameUno(ps)) then
        definir(dameUno(ps),TienenPokemonCerca(v,dameUno(ps),js),
        PosiblesCaptore(v,sinUno(ps),js)
    else
        PosiblesCaptore(v,sinUno(ps),js)
    fi
fi

```

ConectadosYNoEliminados(v) \equiv ConectadosYNoEliminadosAux(v,jugadores(v))

```

ConectadosYNoEliminadosAux(v,js)  $\equiv$  if  $\emptyset?(js)$  then
     $\emptyset$ 
else
    if dameUno(js)  $\in$  JugadoresConectados(v)  $\wedge$ 
    Sanciones(v,dameUno(js)) < 5 then
        Ag(dameUno(js),ConectadosYNoEliminadosAux(v,sinUno(js)))
    else
        ConectadosYNoEliminadosAux(v,sinUno(js))
    fi
fi

```

```

TienenPokemonCerca(v,p,js)  $\equiv$  if  $\emptyset?(js)$  then
     $\emptyset$ 
else
    if Ubicación(v,dameUno(js))  $\in$  PosicionesNorma2Menor(v,p,2) then
        Ag(dameUno(js),TienenPokemonCerca(v,p,sinUno(js)))
    else
        TienenPokemonCerca(v,p,sinUno(js))
    fi
fi

```

MovimientoEnRango(v,j₁,j₂,p) \equiv j₁ \neq j₂ \wedge p \in claves(JugadoresEnRango(v)) \wedge_L
j₂ \in obtener(p,JugadoresEnRango(v))

MovidaNoEnRangoFaltanTurnos(v,j₁,j₂,p) \equiv j₁ \neq j₂ \wedge j₂ \in JugadoresConectados(v) \wedge_L
NormaAlCuadrado(p,PosiciónEnRango(v,j₂,
PosicionesNorma2Menor(Mapa(v), Ubicación(v,j₂),2))) \geq 4 \wedge
TurnosRestantes(v,j₂,PosiciónEnRango(v,j₂,PosicionesPokemon(v)))
> 1

TurnosCeroSinPokemon(v,j₁,j₂,p) \equiv j₁ \neq j₂ \wedge j₂ \in JugadoresConectados(v) \wedge_L
NormaAlCuadrado(p,PosiciónEnRango(v,j₂,
PosicionesNorma2Menor(Mapa(v), Ubicación(v,j₂),2))) \geq 4 \wedge
TurnosRestantes(v,j₂,PosiciónEnRango(v,j₂, PosicionesPokemon(v)))
= 1 \wedge j₂ \neq dameUno(PuedenAtrapar(v, obtener(PosiciónEnRango(v,j₂,
PosicionesPokemon(v)), JugadoresEnRango(v))))

```

PuedenAtrapar(v,p,js)  $\equiv$  if  $\emptyset?(js)$  then
     $\emptyset$ 
else
    if TurnosRestantes(v,dameUno(js),p) = 1 then
        Ag(dameUno(ps),PuedenAtrapar(v,p,sinUno(js)))
    else
        PuedenAtrapar(v,p,sinUno(js))
    fi
fi

```

```

HayPokemonCerca?(v,j)  $\equiv$   $j \in \text{JugadoresConectados}(v) \wedge_L ( \text{PosicionesNorma2Menor}(m,\text{Ubicación}(v,j),2) \cap \text{PosicionesPokemon}(v) ) \neq \emptyset$ 

PosicionesCaptura(v,p)  $\equiv$   $\text{LejanasConPokemonAtrapable}(v, \text{claves}(\text{JugadoresEnRango}(v,\text{Posiciones}(\text{Mapa}(v)) - \text{PosicionesNorma2Menor}(v,p,2), \text{JugadoresConectadosYNoEliminados}(v) )) , \text{JugadoresEnRango}(v,\text{Posiciones}(\text{Mapa}(v)) - \text{PosicionesNorma2Menor}(v,p,2), \text{JugadoresConectadosYNoEliminados}(v) )$ 

LejanasConPokemonAtrapable(v,ps,dH)  $\equiv$  if  $\emptyset?(ps)$  then
     $\emptyset$ 
else
    if  $\neg \emptyset?((\text{PuedenAtrapar}(v,\text{dameUno}(ps),\text{obtener}(\text{dameUno}(ps),dH)))$ 
    then
         $\text{Ag}(\text{dameUno}(ps),\text{LejanasConPokemonAtrapable}(v,\text{sinUno}(ps),dH))$ 
    else
         $\text{LejanasConPokemonAtrapable}(v,\text{sinUno}(ps),dH)$ 
    fi
fi

Capturar(dpkm,cpkm,ps)  $\equiv$  if  $\emptyset?(cpkm)$  then
    dpkm
else
     $\text{definir}(\text{dameUno}(cpkm), \text{obtener}(\text{dameUno}(cpkm),dpkm) - ps , \text{Capturar}(\text{dpkm},\text{sinUno}(cpkm),ps))$ 
fi

PokemonEnPos(v,p)  $\equiv$   $\text{BuscarPokemon}(\text{Salvajes}(v),\text{claves}(\text{Salvajes}(v)),p)$ 

BuscarPokemon(dpkm,cpkm,p)  $\equiv$  if  $p \in \text{obtener}(\text{dameUno}(cpkm),dpkm)$  then
     $\text{dameUno}(cpkm)$ 
else
     $\text{BuscarPokemon}(dpkm,\text{sinUno}(cpkm),p)$ 
fi

```

Fin TAD

3. Extensión TAD NAT

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

$a / b \equiv \text{div}(a,b,1)$

$\text{div} : \text{nat} \times \text{nat} \times \text{nat} \longrightarrow \text{nat}$

$\text{div}(a,b,n) \equiv$ **if** $a \geq b * n$ **then** $1 + \text{div}(a,b,n+1)$ **else** 0 **fi**

$\text{ValorAbsResta} : \text{nat} \times \text{nat} \longrightarrow \text{nat}$

$\text{ValorAbsResta}(n,m) \equiv \text{máx}(n,m) - \text{mín}(n,m)$