# Índice

# 1. Renombres de TADs

TAD Coordenada es Tupla(Nat,Nat)
TAD Jugador es String
TAD Pokemon es String

# 2. TAD Mapa

**TAD** Mapa

**igualdad observacional**

$$(\forall m, m' : \text{mapa}) \left( m =_{\text{obs}} m' \iff \begin{pmatrix} \text{posiciones(m)} = \text{posiciones(m')} \wedge_{\text{L}} (\forall c\text{: posiciones(m)}) \\ \text{opciones(m,c)} = \text{opciones(m,c')} \end{pmatrix} \right)$$

**géneros** mapa

**exporta** mapa, generadores, observadores, otras operaciones

**usa** Coordenada, Nat

**observadores básicos**

| | | | | |
|---|---|---|---|---|
| posiciones | : mapa | $\longrightarrow$ | conj(Coordenada) | |
| opciones | : Mapa $m$ × Coordenada $c$ | $\longrightarrow$ | conj(Coordenada) | {c ∈ posiciones(m) } |

**generadores**

| | | | | |
|---|---|---|---|---|
| InitMapa | : | $\longrightarrow$ | Mapa | |
| AgCoordenada | : Mapa $m$ × Coordenada $c$ | $\longrightarrow$ | Mapa | |
| Relacionar | : Mapa $m$ × Coordenada $c$ × Coordenada $c'$ | $\longrightarrow$ | Mapa | {c ≠ c' ∧ c,c' ∈ posiciones(m)} |

**otras operaciones**

| | | | | |
|---|---|---|---|---|
| enRango | : Mapa $m$ × Coordenada $c$ × Coordenada $c'$ × Nat $n$ | $\longrightarrow$ | Bool | |
| opcionesTotales | : Mapa $m$ × Coordenada $c$ | $\longrightarrow$ | conj(Coordenada) | {c ∈ posiciones(m)} |
| opcionesTotalesAux | : Mapa $m$ × conj(Coordenada) $C$ × conj(Coordenada) $C'$ | $\longrightarrow$ | conj(Coordenada) | {C, C' ⊆ posiciones(m)} |

**axiomas** $\forall m$: mapa $\forall c, c', c''$: Coordenada $\forall C, C'$: conj(Coordenada) $\forall n$: Nat

| | | |
|---|---|---|
| posiciones(InitMapa) | ≡ | ∅ |
| posiciones(AgCoordenada(m,c)) | ≡ | Ag(c,posiciones(m)) |
| posiciones(Relacionar(m,c,c') | ≡ | Posiciones(m) |
| opciones(InitMapa,c) | ≡ | ∅ |
| opciones(AgCoordenada(m,c),c') | ≡ | **if** $c = c'$ **then** ∅ **else** opciones(m,C') **fi** |
| opciones(Relacionear(m,c,c'),c") | ≡ | **if** c"≠ c ∧ c" ≠ c' **then** opciones(m,c") **else** **if** c"= c **then** Ag(c',opciones(m,c)) **else** Ag(c,opciones(m,c')) **fi** **fi** |
| enRango(m,c,c',n) | ≡ | ((max($\Pi_1$(c),$\Pi_1$(c')) - min($\Pi_1$(c),$\Pi_1$(c'))) *((max($\Pi_1$(c),$\Pi_1$(c')) - min($\Pi_1$(c),$\Pi_1$(c'))) + ((max($\Pi_2$(c),$\Pi_2$(c')) - min($\Pi_2$(c),$\Pi_2$(c'))) *((max($\Pi_2$(c),$\Pi_2$(c')) - min($\Pi_2$(c),$\Pi_2$(c'))) |
| opcionesTotales(m,c) | ≡ | opciones(c) ∪ opcionesTotalesAux(m,opciones(c),Ag(c,∅)) |
| opcionesTotalesAux(m,C,C') | ≡ | opciones(dameUno(C)) ∪ opcionesTotalesAux(m,Opciones(dameUno(c)) ∪ C - (Ag(dameUno(C),C')) , Ag(dameUno(C),C') ) |

**Fin TAD**

# 3. TAD Partida

**TAD** PARTIDA

**igualdad observacional**

$$(\forall p, p' : \text{partida}) \quad \left( p =_{\text{obs}} p' \iff \left( \begin{array}{l} \text{mapa(p) = mapa(p')} \land \text{jugadores(p) = jugadores(p')} \land \\ \text{pokemonesSalvajes(p) = pokemonesSalvajes(p')} \land_{\text{L}} \\ (\forall \text{ a} \in \text{pokemonesSalvajes(p))} \text{ posicionesPokemon(p, a) =} \\ \text{posicionesPokemons(p', a)} \land (\forall \text{ j} \in \text{jugadores(p))} \\ (\text{sanciones(p, j) = sanciones(p', j))} \land \neg\text{eliminado(p, j)} \Rightarrow_{\text{L}} \\ (\text{pokemonesDeJugador(p, j) = pokemonesDeJugador(p', j)} \\ \land \text{ conectado?(p, j) = conectado?(p', j)} \land \text{conectado?(p,} \\ \text{j)} \Rightarrow_{\text{L}} \text{posicionJugador(p, j) = posicionJugador(p', j))} \land \\ (\forall \text{ c} \in \text{coordenadasTotalesPoke(pokemonesSalvajes(p), p))} \\ \text{movimientosParaCapturar(p, c) = movimientosParaCap-} \\ \text{turar(p', c)} \end{array} \right) \right)$$

**géneros** partida

**exporta** partida, generadores, observadores, otras operaciones

**usa** BOOL, NAT, COORDENADA, JUGADOR, POKEMON, CONJUNTO(COORDENADA), CONJUNTO(JUGADOR), CONJUNTO(POKEMON)

**observadores básicos**

| | | | | |
|---|---|---|---|---|
| mapa | : partida | $\longrightarrow$ | mapa | |
| jugadores | : partida | $\longrightarrow$ | conj(jugador) | |
| pokemonsSalvajes | : partida | $\longrightarrow$ | conj(pokemon) | |
| posicionesPokemon | : partida $p$ $\times$ pokemon $a$ | $\longrightarrow$ | conj(coordenada) | {a $\in$ pokemonsSalvajes(p)} |
| pokemonsDeJugador | : partida $p$ $\times$ jugador $j$ | $\longrightarrow$ | multiconj(pokemon) | {j $\in$ jugadores(p) $\land_{\text{L}}$ $\neg$ eliminado(p, j)} |
| posicionJugador | : partida $p$ $\times$ jugador $j$ | $\longrightarrow$ | coordenada | {j $\in$ jugadores(p) $\land_{\text{L}}$ conectado?(p, j) $\land$ $\neg$ eliminado(p, j)} |
| conectado? | : partida $p$ $\times$ jugador $j$ | $\longrightarrow$ | bool | {j $\in$ jugadores(p) $\land_{\text{L}}$ $\neg$ eliminado(p, j)} |
| sanciones | : partida $p$ $\times$ jugador $j$ | $\longrightarrow$ | nat | {j $\in$ jugadores(p)} |
| movimientosParaCapturar | : partida $p$ $\times$ coordenada $c$ | $\longrightarrow$ | nat | $\left\{ \begin{array}{l} \text{c} \in \text{posiciones(mapa(p))} \land_{\text{L}} (\exists \text{ a} \in \text{pokemonsSalvajes(p))} ((\exists \text{ c'} \in \text{posicionesPokemon(p, a))} \text{ c} = \\ \text{c')} \end{array} \right\}$ |

**generadores**

| | | | | |
|---|---|---|---|---|
| nuevaPartida | : mapa | $\longrightarrow$ | partida | |
| AgPokemon | : partida $p$ $\times$ pokemon $\times$ coordenada $c$ | $\longrightarrow$ | partida | $\left\{ \begin{array}{l} \text{c} \in \text{posiciones(mapa(p)} \land_{\text{L}} (\forall \text{ a'} \in \text{pokemonsSalvajes(p))} ((\forall \text{ c'} \in \text{posicionesPokemon(p))}) \\ \neg\text{enRango(mapa(p), c, c', 5))} \end{array} \right\}$ |
| AgJugador | : partida $p$ $\times$ jugador $j$ | $\longrightarrow$ | partida | {j $\notin$ jugadores(p)} |
| conectar | : partida $p$ $\times$ jugador $j$ $\times$ coordenada | $\longrightarrow$ | partida | {j $\in$ jugadores(p) $\land$ $\neg$ eliminado(p, j) $\land$ $\neg$ conectado?(p, j)} |
| desconectar | : partida $p$ $\times$ jugador $j$ | $\longrightarrow$ | partida | {j $\in$ jugadores(p) $\land$ $\neg$ eliminado(p, j) $\land$ conectado?(p, j)} |
| mover | : partida $p$ $\times$ jugador $j$ $\times$ coordenada $c$ | $\longrightarrow$ | partida | {j $\in$ jugadores(p) $\land$ c $\in$ posiciones(mapa(p) $\land_{\text{L}}$ $\neg$ eliminado(p, j)} |

**otras operaciones**

| | | | | |
|---|---|---|---|---|
| capturados | : jugador $j$ $\times$ partida $p$ $\times$ coordenada | $\longrightarrow$ | conj(pokemon) | {j $\in$ noEliminados(p)} |
| capturadosTotales | : partida $p$ $\times$ conj(coordenada) $C$ $\times$ conj(pokemon) $C'$ | $\longrightarrow$ | conj(pokemon) | {$(\forall c \in C) c \in$ coordenadas(mapa(p)) $\land$ ($\forall$ a $\in$ C') a $\in$ pokemonesSalvajes(p)} |
| pokemonesCapturables | : conj(coordenada) $C$ $\times$ jugador $j$ $\times$ coordenada $c$ $\times$ partida $p$ | $\longrightarrow$ | conj(coordenada) | {$(\forall c' \in C) c' \in$ coordenadas(mapa(p)) $\land$ j $\in$ noEliminados(p) $\land c \in$ coordenadas(mapa(p))} |

| | | | | |
|---|---|---|---|---|
| coordenadasTotalesPoke | : conj(pokemon) $C$ × partida $p$ | ⟶ | conj(coordenada) | {(∀ a ∈ C) a ∈ pokemonesSalvajes(p)} |
| hayPokeEnRango | : jugador $j$ × conj(coordenada) $C$ × partida $p$ | ⟶ | bool | {(∀$c \in C$)$c$ ∈ coordenadas(mapa(p)) ∧ j ∈ noEliminados(p)} |
| movFueraDelPoke | : jugador $j$ × jugador $j'$ × partida $p$ × coordenada $c$ | ⟶ | bool | {j,j' ∈ noEliminados(p) ∧$c$ ∈ coordenadas(mapa(p))} |
| coordPoke | : jugador $j$ × conj(coordenada) $C$ | ⟶ | coordenada | {(∀$c \in C$)$c$ ∈ coordenadas(mapa(p)) ∧ j ∈ noEliminados(p)} |
| esElElegido | : jugador $j$ × partida $p$ | ⟶ | bool | {j ∈ noEliminados(p)} |
| jugadoresEnRangoPoke | : coordenada $c$ × conj(jugador) $C$ × partida $p$ | ⟶ | conj(jugador) | {$c$ ∈ coordenadas(mapa(p)) ∧ (∀ j ∈ C) noEliminados(p)} |
| esePokemon | : jugador $j$ × partida $p$ | ⟶ | pokemon | {j ∈ noEliminados(p)} |
| cualEsEsePokemon? | : coordenada $c$ × conj(pokemon) $C$ × partida $p$ | ⟶ | pokemon | {$c$ ∈ coordenadas(mapa(p)) ∧ (∀ j ∈ C) j ∈ noEliminados(p)} |
| pokeParaCapturar | : jugador $j$ × partida $p$ | ⟶ | bool | {j ∈ noEliminados(p)} |
| NoEliminados | : partida | ⟶ | conj(jugador) | |
| FiltrarEliminados | : conj(jugador) $C$ × partida | ⟶ | conj(jugador) | {(∀ j ∈ C)j ∈ jugadores(p)} |
| pokesCapturados | : conj(jugador) $C$ × pokemon $a$ × partida $p$ | ⟶ | nat | {(∀ j ∈ C) j ∈ noEliminados(p)} |
| pokesTotalesDeTipo | : pokemon × partida | ⟶ | nat | |
| pokesTotalesJugadores | : conj(jugador) $C$ × partida $p$ | ⟶ | nat | {(∀ j ∈ C) j ∈ noEliminados(p)} |
| pokesTotales | : partida | ⟶ | nat | |
| rareza | : partida × pokemon | ⟶ | nat | {pokesTotales(p) > 0} |
| eliminado | : partida × jugador $j$ | ⟶ | bool | {(j ∈ jugadores(p)} |
| div | : nat × nat $q$ | ⟶ | nat | {q ≠ 0} |
| mod | : nat × nat $q$ | ⟶ | nat | {q ≠ 0} |

**axiomas** ∀$m$: mapa ∀$p$: partida ∀$a$: pokemon ∀$j, j'$: jugador ∀$c$: coordenada

| | | |
|---|---|---|
| mapa(nuevaPartida(m)) | ≡ | m |
| mapa(AgPokemon(p, a, c)) | ≡ | mapa(p) |
| mapa(AgJugador(p, j)) | ≡ | mapa(p) |
| mapa(conectar(p, j, c)) | ≡ | mapa(p) |
| mapa(desconectar(p, j)) | ≡ | mapa(p) |
| mapa(mover(p, j, c)) | ≡ | mapa(p) |
| jugadores(nuevaPartida(m)) | ≡ | ∅ |
| jugadores(AgPokemon(p, a, c)) | ≡ | jugadores(p) |
| jugadores(AgJugador(p, j)) | ≡ | Ag(j,jugadores(p)) |
| jugadores(conectar(p, j, c)) | ≡ | jugadores(p) |
| jugadores(desconectar(p, j)) | ≡ | jugadores(p) |
| jugadores(mover(p, j, c)) | ≡ | jugadores(p) |
| pokemonesSalvajes(NuevaPartida(m)) | ≡ | ∅ |
| pokemonesSalvajes(AgPokemon(p, a, c)) | ≡ | Ag(a , pokemonesSalvajes(p)) |
| pokemonesSalvajes(AgJugador(p, j)) | ≡ | pokemonesSalvajes(p) |
| pokemonesSalvajes(conectar(p, j, c)) | ≡ | pokemonesSalvajes(p) |
| pokemonesSalvajes(desconectar(p, j)) | ≡ | pokemonesSalvajes(p) |
| pokemonesSalvajes(mover(P, j, c)) | ≡ | pokemonesSalvajes(P) - capturados(j, P, c) |
| posicionesPokemon(NuevaPartida(m), a) | ≡ | ∅ |
| posicionesPokemon(AgPokemon(p, a, c), a) | ≡ | Ag(c, posicionesPokemon(P, a)) |
| posicionesPokemon(AgJugador(p, j), a) | ≡ | posicionesPokemon(p, a) |
| posicionesPokemon(conectar(p, j, c), a) | ≡ | posicionesPokemon(p, a) |
| posicionesPokemon(desconectar(p, j), a) | ≡ | posicionesPokemon(p, a) |

posicionesPokemon(mover(p, j, c), a) ≡ posicionesPokemon(p, a) - pokemonesCapturables(posicionesPokemon(p, a), j, c, p))

pokemonesDeJugador(NuevaPartida(m), j) ≡ ∅
pokemonesDeJugador(AgPokemon(p, a, c), j) ≡ pokemonesDeJugador(P,j)
pokemonesDeJugador(AgJugador(p, j), j) ≡ pokemonesDeJugador(p,j)
pokemonesDeJugador(conectar(p, j, c), j) ≡ pokemonesDeJugador(p,j)
pokemonesDeJugador(desconectar(p, j), j) ≡ pokemonesDeJugador(p,j)
pokemonesDeJugador(mover(p, j', c), j) ≡ **if** j ≠ j' ∧ hayPokeEnRango(j, coordenadasTotalesPoke(pokemonesSalvajes(p), p)) ∧ₗ movFueraDelPoke(j, j', p, c) ∧ₗ pokeParaCapturar(j, p) ∧ₗ esElElegido(j, p) **then**
Ag(esePokemon(j, p), pokemonesDeJugador)
**else**
pokemonesDeJugador(j, p)
**fi**

posicionJugador(AgPokemon(p, a, c), j') ≡ posicionJugador(p,j')
posicionJugador(AgJugador(p, j), j') ≡ posicionJugador(p,j')
posicionJugador(conectar(p, j, c),j') ≡ **if** j = j' **then** c **else** posicionJugador(p, j') **fi**
posicionJugador(desconectar(p, j),j') ≡ posicionJugador(p,j')
posicionJugador(mover(p, j, c), j') ≡ **if** j = j' **then** c **else** posicionJugador(p, j') **fi**
conectado?(AgPokemon(p, a, c),j') ≡ conectado(p, j')
conectado?(AgJugador(p, j), j') ≡ **if** j = j' **then** False **else** conectado(p, j') **fi**
conectado?(conectar(p, j, c), j') ≡ **if** j = j' **then** True **else** conectado(p,j') **fi**
conectado?(desconectar(p, j), j') ≡ **if** j = j' **then** False **else** conectado(p,j') **fi**
conectado?(mover(p, j, c), j') ≡ conectado(p, j')
sanciones(AgPokemon(p, a, c), j') ≡ sanciones(p, j')
sanciones(AgJugador(p, j), j') ≡ **if** j = j' **then** 0 **else** sanciones(p, j') **fi**
sanciones(conectar(p, j, c), j') ≡ sanciones(p, j')
sanciones(desconectar(p, j), j') ≡ sanciones(p, j')
sanciones(mover(p, j, c), j') ≡ **if** j = j' **then**
**if** c ∈ opcionesTotales(mapa(p), posicionJugador(j, p)) ∧ enRango(c, posicionJugador(j, p), 10) **then**
sanciones(j, p)
**else**
snaciones(j, p) + 1
**fi**
**else**
sanciones(j, p)
**fi**

movimientosParaCapturar(AgPokemon(p, a, c'),c) ≡ movimientosParaCapturar(p,c)
movimientosParaCapturar(AgJugador(p, j), c) ≡ movimientosParaCapturar(p,c)
movimientosParaCapturar(conectar(p, j, c'), c) ≡ **if** enRango(c, c', 2) **then**
0
**else**
moverParaCapturar(p,c)
**fi**
movimientosParaCapturar(desconectar(p, j), c) ≡ **if** enRango(c, posicionJugador(p,j), 2) **then**
0
**else**
moverParaCapturar(p,c)
**fi**

movimientosParaCapturar(mover(p, j, c)), c') ≡ **if** enRango(c, c', 2) ∧
enRango(c, posicionJugador(j, p), 2) **then**
moverParaCapturar(c, p)
**else**
**if** ¬ enRango(c, c', 2) ∧
¬ enRango(c, posicionJugador(j, p), 2) **then**
moverParaCapturar(c, p) + 1
**else**
0
**fi**
**fi**

capturados(j,p,c) ≡ capturadosTotales(p, pokemonesCapturables(
coordenadasTotalesPoke(pokemonesSalvajes(p), p),
j, c, p), pokemonesSalvajes(p))

capturadosTotales(p, C, C') ≡ **if** ∅?(C') **then**
∅
**else**
**if** ∅?(posicionesPokemon(dameUno(C'), p) - C)
**then**
Ag(dameUno(C'),
capturadosTotales(p, C, sinUno(C')))
**else**
capturadosTotales(p, C, sinUno(C'))
**fi**
**fi**

pokemonesCapturables(C, j, c, p) ≡ **if** ∅?(C) **then**
∅
**else**
**if** ¬ enRango(dameUno(C), c, 2) ∧
¬enRango(dameUno(C), posicionJugador(j, p), 2)
∧ movimientosParaCapturar(dameUno(C)) == 9
**then**
Ag(dameUno(C),
pokemonesCapturables(sinUno(C), j, c, p))
**else**
pokemonesCapturables(sinUno(C), j, c, p)
**fi**
**fi**

coordenadasTotalesPoke(C, p) ≡ **if** ∅?(C) **then**
∅
**else**
posicionesPokemon(p, dameUno(C)) ∪
coordenadasTotalesPoke(sinUno(C), p)
**fi**

hayPokeEnRango(j, C, p) ≡ **if** ∅?(C) **then**
False
**else**
**if** enRango(posicionJugador(j), dameUno(C), 2)
**then**
True
**else**
hayPokeEnRango(j,sinUno(C),p)
**fi**
**fi**

movFueraDelPoke(j, j', p, c) ≡ ¬ enRango(posicionJugador(j'), coordPoke(j,
coordenadasTotalesPoke(pokemonesSalvajes(p),
p)), 2) ∧ ¬ enRango(c, coordenadasTotalesPoke(
pokemonesSalvajes(p), p), 2))

coordPoke(j, C) ≡ **if** enRango(j, dameUno(C), 2) **then**
dameUno(C)
**else**
coordPoke(j, sinUno(C))
**fi**

esElElegido(j,p) ≡ dameUno(jugadoresEnRangoPoke(coordPoke(j, coordenadasTotalesPoke(pokemonesSalvajes(p), noEliminados(p), p)))) == j

jugadoresEnRangoPoke(c, C, p) ≡ **if** ∅?(C) **then**
∅
**else**
**if** enRango(posicionJugador(p,dameUno(C)), c, 2) **then**
Ag(dameUno(C), jugadoresEnRangoPoke(c, sinUno(C), p))
**else**
jugadoresEnRangoPoke(c, sinUno(C))
**fi**
**fi**

esePokemon(j,p) ≡ cualEsEsePokemon?(coordPoke(j, coordenadasTotalesPoke(pokemonesSalvajes(p), p)), pokemonesSalvajes(p))

cualEsEsePokemon?(c, C, p) ≡ **if** c ∈ posicionesPokemon(p,dameUno(C)) **then**
dameUno(C)
**else**
cualEsEsePokemon?(c, sinUno(C), p)
**fi**

pokeParaCapturar(j, p) ≡ movimientosParaCapturar(coordPoke(j, coordenadasTotalesPoke(pokemonesSalvajes(p), p)), p) == 9

NoEliminados(p) ≡ FiltrarEliminados(Jugadores(p), p)

FiltrarEliminados(C, p) ≡ **if** ∅?(C) **then**
∅
**else**
**if** ¬ eliminado(p, DameUno(C)) **then**
Ag(DameUno(C), FiltrarEliminados(SinUno(C), p))
**else**
FiltrarEliminados(SinUno(C), p)
**fi**
**fi**

pokeCapturados(C, a, p) ≡ #(a, pokemonsDeJugador(p, DameUno(C))) + pokeCapturados(SinUno(C), a, p)

pokesTotalesDeTipo(a, p) ≡ pokeCapturados(NoEliminados(p), p) + #posicionesPokemon(p, a)

pokesTotalesJugadores(C, p) ≡ **if** ∅?(C) **then**
O
**else**
#pokemonsDeJugador(p, DameUno(C)) + pokesTotalesJugadores(SinUno(C), p)
**fi**

pokesTotales(p) ≡ pokesTotalesJugadores(NoEliminados(p), p) + #coordenadasTotalesPoke(pokemonesSalvajes(p), p)

rareza(p, a) ≡ 100 - div(pokesTotalesDeTipo(a, p) * 100, pokesTotales(p)) - mod(pokesTotalesDeTipo(a, p) * 100, pokesTotales(p))

eliminado(p, j) ≡ sanciones(p, j) ≥ 5

div(p, q) ≡ **if** p < q **then** 0 **else** 1 + div(p - q, q) **fi**

mod(p, q) ≡ **if** p < q **then** p **else** mod(p - q, q) **fi**

**Fin TAD**