

# Algoritmos y Estructuras de Datos II

Segundo Cuatrimestre de 2016

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

## Trabajo Practico 1

Especificacion

### Grupo 1

Integrante	LU	Correo electrónico
Integrante 1	Nro/YY	mail@dc.uba.ar
Integrante 2	Nro/YY	mail@dc.uba.ar
Integrante 3	Nro/YY	mail@dc.uba.ar
Integrante 4	Nro/YY	mail@dc.uba.ar

### Reservado para la cátedra

Instancia	Docente	Nota
Primera entrega		
Segunda entrega		

# Índice

1. Especificacion	3
2. Renombres de TADs	3
3. TAD Juego	4
4. TAD Mapa	6
5. TAD Jugador	7
6. TAD Coordenada	8

## 1. Especificacion

Esta es una especificacion del Trabajo Practico 1 del 2<sup>do</sup> cuatrimestre del 2016 presentada por la catedra para la realizacion del Trabajo Practico 2. Ver enunciado:

<http://www.dc.uba.ar/materias/aed2/2016/2c/descargas/tps/tp1/view>

## 2. Renombres de TADs

TAD TIPO es STRING

TAD POKEMONES es DICCIONARIO(NAT, TIPO)

TAD POKEMON es TUPLA(NAT, TIPO)

### 3. TAD Juego

#### TAD JUEGO

**géneros**      juego

##### observadores básicos

mapa	: Juego	→ Mapa	
jugadores	: Juego	→ conj(Jugador)	
posicionJugador	: Jugador $j \times$ Juego $pGo$	→ Coordenada	$\{j \in \text{jugadores}(pGo)\}$
pokemones	: Juego	→ Pokemones	
posicionPokemon	: Pokemon $p \times$ Juego $j$	→ Coordenada	$\{\text{def?}(\Pi_1(p), \text{pokemones}(j))\}$

##### generadores

nuevoJuego	: Mapa	→ Juego	
agJugador	: Jugador $j \times$ Coordenada $c \times$ Juego $pGo$	→ Juego	$\{\text{esPosicionValidaJugador}(c, pGo) \wedge_L \text{id}(j) \notin \text{jugadores}(pGo)\}$
agPokemon	: Nat $n \times$ Tipo $t \times$ Coordenada $c \times$ Juego $pGo$	→ Juego	$\{n \notin \text{pokemones}(j) \wedge_L \text{esPosicionValidaPokemon}(c, pGo)\}$
informarPosicion	: Jugador $j \times$ Coordenada $c \times$ Juego $pGo$	→ Juego	$\{\text{puedeMoverse}(j, pGo) \wedge_L \text{esPosicionValidaEnJuego}(c, pGo)\}$
sumarEspera	: conj(jugador) $cj \times$ Coordenada $c \times$ Juego $pGo$	→ Juego	

##### otras operaciones

atraparPokemon	: Pokemon $p \times$ Jugador $j \times$ Juego $pGo$	→ Juego
esPosicionValidaJugador	: Coordenada $c \times$ Juego $j$	→ bool
esPosicionValidaPokemon	: Coordenada $c \times$ Juego $j$	→ bool
esPosicionValidaEnJuego	: Coordenada $c \times$ Juego $j$	→ bool
puedeMoverse	: Jugador $j \times$ Juego $pGo$	→ bool

**axiomas**       $\forall m: \text{Mapa } \forall j, j1, j2: \text{Jugador } \forall c: \text{Coordenada } \forall pGo: \text{Juego } \forall p: \text{Pokemon } \forall n: \text{Nat } \forall t: \text{Tipo}$

```

mapa(nuevoJuego(m)) ≡ m
jugadores(nuevoJuego(m)) ≡ ∅
pokemones(nuevoJuego(m)) ≡ ∅
mapa(agJugador(j, c, pGo)) ≡ mapa(pGo)
jugadores(agJugador(j, c, pGo)) ≡ Ag(j, jugadores(pGo))
posicionJugador(j1, agJugador(j2, c, pGo)) ≡ if j1 = j2 then c else posicionJugador(j1, pGo) fi
pokemones(agJugador(j, c, pGo)) ≡ pokemones(pGo)
posicionPokemon(p, agJugador(j, c, pGo)) ≡ posicionPokemon(p, pGo)
mapa(agPokemon(n, t, c, pGo)) ≡ mapa(pGo)
jugadores(agPokemon(n, t, c, pGo)) ≡ jugadores(pGo)
posicionJugador(j, agPokemon(n, t, c, pGo)) ≡ posicionJugador(j, pGo)
pokemones(agPokemon(n, t, c, pGo)) ≡ definir(n, t, pokemones(pGo))
posicionPokemon(p, agPokemon(n, t, c, pGo)) ≡ if  $\Pi_1(p) = n$  then c else posicionPokemon(p, pGo) fi
mapa(informarPosicion(j, c, pGo)) ≡ mapa(pGo)
jugadores(informarPosicion(j, c, pGo)) ≡ if nula?(c) then
    jugadores(pGo)
else
    actualizarContadores(if posicionInvalida(posicionJugador(j,
    pGo), c) then
        if cantidadSanciones(j) = 4 then
            jugadores(pGo) - j
        else
            Ag(sancionar(j), jugadores(pGo) - j)
        fi
    else
        jugadores(pGo)
    fi, c, pGo)
fi

```

```

posicionJugador(j1, informarPosicion(j2, c, pGo)) ≡ if j1 ≠ j2 then
    posicionJugador(j1, pGo)
else
    if nula?(c) then
        fueraDelMapa
    else
        if nula?(posicionJugador(j2, pGo)) then
            c
        else
            if posicionInvalida(c) then
                if cantidadDeSanciones(j1) = 4 then
                    fueraDelMapa
                else
                    c
            fi
        else
            c
        fi
    fi
fi
fi
fi
fi

pokemones(informarPosicion(j, c, pGo)) ≡
posicionPokemon(informarPosicion(j, c, pGo)) ≡

Fin TAD

```

## 4. TAD Mapa

**TAD MAPA**

**géneros**      mapa

**igualdad observacional**

$$(\forall m, m' : \text{Mapa}) \left( m =_{\text{obs}} m' \iff \left( \begin{array}{l} (\text{posiciones}(m) =_{\text{obs}} \text{posiciones}(m')) \wedge \\ (\forall c1, c2 : \text{Coordenada}) \\ (\text{existeCamino}(c1, c2, m) \leftrightarrow \text{existeCamino}(c1, c2, m')) \end{array} \right) \right)$$

**exporta**      Mapa, generadores, observadores

**usa**          BOOL, COORDENADA, CONJ()

**observadores básicos**

posiciones      : Mapa  $m$        $\longrightarrow$  Conj(Coordenada)

existeCamino    : Coordenada  $c1 \times$  Coordenada  $c2 \times$  Mapa  $m$        $\longrightarrow$  bool

**generadores**

crear            :       $\longrightarrow$  Mapa

agCoordenada    : Coordenada  $c \times$  Conj(Coordenada)  $cs \times$  Mapa  $m$        $\longrightarrow$  Mapa  
 $\{cs \subseteq \text{posiciones}(m) \wedge c \notin \text{posiciones}(m)\}$

**otras operaciones**

existenCaminos : Coordenada  $c1 \times$  Conj(Coordenada)  $cs \times$  Mapa  $m$        $\longrightarrow$  bool  
 $\{Ag(c1, cs) \in \text{posiciones}(m)\}$

**axiomas**       $\forall c, c1, c2 : \text{Coordenada} \forall cs : \text{conj}(\text{Coordenada}) \forall m : \text{Mapa}$

posiciones(crear())  $\equiv \emptyset$

posiciones(agCoordenada( $c, cs, m$ ))  $\equiv Ag(c, \text{posiciones}(m))$

existeCamino( $c1, c2, \text{crear}()$ )  $\equiv \text{false}$

existeCamino( $c1, c2, \text{agCoordenada}(c, cs, m)$ )  $\equiv c1 == c \wedge c2 \in cs$

existenCaminos( $c, cs, m$ )  $\equiv$  **if** vacio?( $cs$ ) **then**  
false

**else**

existeCamino( $c, \text{dameUno}(cs), m$ )  $\vee$  existenCaminos( $c, \text{sinUno}(cs), m$ )

**fi**

**Fin TAD**

## 5. TAD Jugador

### TAD JUGADOR

#### igualdad observacional

$$(\forall j, j' : \text{Jugador}) \left( j =_{\text{obs}} j' \iff \left( \begin{array}{l} (\text{estado}(j) =_{\text{obs}} \text{estado}(j')) \wedge \\ (\text{cantidadSanciones}(j) =_{\text{obs}} \text{cantidadSanciones}(j')) \wedge \\ (\text{pokedex}(j) =_{\text{obs}} \text{pokedex}(j')) \wedge \\ (\text{estado}(j) =_{\text{obs}} \text{estado}(j')) \wedge \\ (\text{cantidadMovimientos}(j) =_{\text{obs}} \text{cantidadMovimientos}(j')) \end{array} \right) \right)$$

**géneros**      jugador

**exporta**      Jugador, generadores, observadores

**usa**            BOOL, NAT, POKEMONES

#### observadores básicos

estado	: Jugador	→ Bool
cantidadSanciones	: Jugador	→ Nat
pokedex	: Jugador	→ Pokemones
id	: Jugador	→ Nat
cantidadMovimientos	: Jugador	→ Nat

#### generadores

crearJugador	: Nat <i>id</i>	→ Jugador
atraparPokemon	: Pokemon <i>p</i> × Jugador <i>j</i>	→ Jugador
sancionar	: Jugador <i>j</i>	→ Jugador

#### axiomas

estado(crearJugador(id)) ≡ true  
 cantidadSanciones(crearJugador(id)) ≡ 0  
 pokedex(crearJugador(id)) ≡ vacio  
 id(crearJugador(id)) ≡ id  
 cantidadMovimientos(crearJugador(id)) ≡ 0

**Fin TAD**

## 6. TAD Coordenada

### TAD COORDENADA

**géneros**      coordenada

**observadores básicos**

$x$       : Coordenada  $c$        $\longrightarrow$  Nat  
 $y$       : Coordenada  $c$        $\longrightarrow$  Nat  
 $nula?$  : Coordenada  $c$        $\longrightarrow$  Bool

$\{\neg nula?(c)\}$   
 $\{\neg nula?(c)\}$

**generadores**

$fueraDelMapa$  :       $\longrightarrow$  Coordenada  
 $nueva$       : Nat  $x \times$  Nat  $y$        $\longrightarrow$  Coordenada

**axiomas**       $\forall x1, y1: \text{Nat}$

$nula?(fueraDelMapa()) \equiv \text{true}$   
 $nula?(nueva(x1, y1)) \equiv \text{false}$   
 $x(nueva(x1, y1)) \equiv x1$   
 $y(nueva(x1, y1)) \equiv y1$

**Fin TAD**