



# TP de Especificación

Esperando el Bondi

30 de Marzo de 2022

Algoritmos y Estructuras de Datos I

## Grupo 1

Integrante	LU	Correo electrónico
Polonuer, Joaquin	1612/21	jtpolonuer@gmail.com
González, Facundo	1440/21	facundo2gonzalez2@gmail.com
Jaime, Brian David	411/18	brian.d.jaime97@gmail.com
Guberman, Diego	469/17	diego98g@hotmail.com



**Facultad de Ciencias Exactas y Naturales**  
Universidad de Buenos Aires

Ciudad Universitaria - (Pabellón I/Planta Baja)

Intendente Güiraldes 2610 - C1428EGA

Ciudad Autónoma de Buenos Aires - Rep. Argentina

Tel/Fax: (++54 +11) 4576-3300

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

## 1. Definición de Tipos

```
type Tiempo =  $\mathbb{R}$   
type Dist =  $\mathbb{R}$   
type GPS =  $\mathbb{R} \times \mathbb{R}$   
type Recorrido =  $seq\langle GPS \rangle$   
type Viaje =  $seq\langle Tiempo \times GPS \rangle$   
type Nombre =  $\mathbb{Z} \times \mathbb{Z}$   
type Grilla =  $seq\langle GPS \times GPS \times Nombre \rangle$ 
```

## 2. Constantes

```
aux MIN :  $\mathbb{Z}$  = 1;  
aux MAX :  $\mathbb{Z}$  = 10;
```

## 3. Problemas

### 3.1. Ejercicio 1

```
proc viajeValido (in v: Viaje, out res: Bool) {  
  Pre {true}  
  Post {res = true  $\leftrightarrow$  esViajeValido(v)}  
  pred esViajeValido (v: Viaje) {  
     $(\forall i : \mathbb{Z})(0 \leq i < |v| \longrightarrow_L (esTiempoValido(v[i]_0) \wedge sonCoordenadasValidas(v[i]_1)))$   
  }  
  pred esTiempoValido (t: Tiempo) {  
     $t \geq 0$   
  }  
  pred sonCoordenadasValidas (c: GPS) {  
     $-90,0 \leq c_0 \leq 90,0 \wedge -180,0 \leq c_1 \leq 180,0$   
  }  
}
```

### 3.2. Ejercicio 2

```
proc recorridoValido (in v: Recorrido, out res: Bool) {  
  Pre {true}  
  Post {res = true  $\leftrightarrow$  esRecorridoValido(v)}  
  pred esRecorridoValido (v: Recorrido) {  
     $(\forall i : \mathbb{Z})(0 \leq i < |v| \longrightarrow_L sonCoordenadasValidas(v[i]))$   
  }  
}
```

### 3.3. Ejercicio 3

```

proc enTerritorio (in v: Viaje, in r: Dist, out res: Bool) {
  Pre {esViajeValido(v)}
  Post {res = true  $\leftrightarrow$  estaEnTerritorio(v, r)}
  pred estaEnTerritorio (v: Viaje, r: Dist) {
    ( $\exists c : GPS$ )(sonCoordenadasValidas(c)  $\wedge_L$  ( $\forall i : \mathbb{Z}$ )( $0 \leq i < |v| \longrightarrow_L dist(c, v[i]_1) \leq r$ ))
  }
}

```

### 3.4. Ejercicio 4

```

proc tiempoTotal (in v: Viaje, out t: Tiempo) {
  Pre {esViajeValido(v)}
  Post {maximaDiferenciaTiempo(v, t)}
  pred esMaximaDiferenciaTiempo (t: Tiempo, v: Viaje) {
    (( $\exists i, j : \mathbb{Z}$ )( $0 \leq i, j < |v| \wedge_L |v[i]_0 - v[j]_0| = t$ ))  $\wedge$  ( $\neg(\exists i, j : \mathbb{Z})(0 \leq i, j < |v| \wedge_L |v[i]_0 - v[j]_0| > t)$ )
  }
  pred maximaDiferenciaTiempo (v: Viaje, t: Tiempo) {
    ( $\exists ti, tf : Tiempo$ )( $(esTiempoValido(ti) \wedge esTiempoValido(tf)) \wedge_L (esMinimoTiempo(v, ti) \wedge esMaximoTiempo(v, tf)) \wedge t = tf - ti$ )
  }
  pred esMinimoTiempo (v: Viaje, t: Tiempo) {
    ( $\exists i : \mathbb{Z}$ )( $0 \leq i < |v| \wedge_L (\forall j : \mathbb{Z})(0 \leq j < |v| \longrightarrow_L v[i]_0 \leq v[j]_0) \wedge t = v[i]_0$ )
  }
  pred esMaximoTiempo (v: Viaje, t: Tiempo) {
    ( $\exists i : \mathbb{Z}$ )( $0 \leq i < |v| \wedge_L (\forall j : \mathbb{Z})(0 \leq j < |v| \longrightarrow_L v[i]_0 \geq v[j]_0) \wedge t = v[i]_0$ )
  }
}

```

### 3.5. Ejercicio 5

```

proc distanciaTotal (in v: Viaje, out d: Dist) {
  Pre {esViajeValido(v)}
  Post {distanciaViajeOrdenado(v, d)}
  pred distanciaViajeOrdenado (v: Viaje, d: Dist) {
    ( $\exists vo : Viaje$ )( $esViajeValido(vo) \wedge_L esElViajeOrdenado(v, vo) \wedge d = sumaDistanciasSucesivas(vo)$ )
  }
  pred esElViajeOrdenado (v, vo: Viaje) {
    esPermutacion(v, vo)  $\wedge$  ( $\forall i : \mathbb{Z}$ )( $0 \leq i < |vo| - 1 \longrightarrow_L vo[i]_0 < vo[i + 1]_0$ )
  }
  pred esPermutacion (v1, v2: Viaje) {
    ( $\forall e : Tiempo \times GPS$ )( $\#apariciones(v1, e) = \#apariciones(v2, e)$ )
  }
}

```

```

aux #apariciones (v: Viaje, e: Tiempo  $\times$  GPS) :  $\mathbb{Z}$  =  $\sum_{i=0}^{|v|-1}$  if  $v[i] = e$  then 1 else 0 fi ;
aux sumaDistanciasSucesivas (v: Viaje) : Dist =  $\sum_{i=0}^{|v|-2} dist(v[i]_1, v[i+1]_1)$  ;
}

```

### 3.6. Ejercicio 6

```

proc () {
  Pre {}
  Post {}
}

```