



**DEPARTAMENTO
DE COMPUTACION**

Facultad de Ciencias Exactas y Naturales - UBA

TP de Especificación

14 de septiembre de 2020

Algoritmos y Estructuras de Datos I

Grupo 3

Integrante	LU	Correo electrónico
Fernández Spandau, Luciana	368/20	fernandezspandau@gmail.com
Reyna Maciel, Guillermo	393/20	guille.j.reyna@gmail.com
Casado Farall, Joaquín	072/20	joakinfarall@gmail.com
Companeetz, Ezequiel	415/20	echucompa@gmail.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

<https://exactas.uba.ar>

1. Ejercicio 1

1.1. Predicado principal

```
pred esValido (t : Toroides) {  
  esMatriz(t) ∧ |t[0]| ≥ 3 ∧ |t| ≥ 3  
}
```

1.2. Predicados auxiliares

```
pred esMatriz (m : seq<seq<T>>) {  
  (∀ i : ℤ) (0 ≤ i < |m| → (|m[i]| > 0 ∧ (∀ j : ℤ) (0 ≤ j < |m| → |m[i]| = |m[j]|)))  
}
```

2. Ejercicio 2

```
pred toroideMuerto (t : toroide) {  
  (∀ i : ℤ) ((0 ≤ i < |t|) → (∀ j : ℤ) ((0 ≤ j < |t[i]|) → (t[i][j] = false))))  
}
```

3. Ejercicio 3

```
pred posicionesVivas (t : Toroides, vivas : ℤ × ℤ) {  
  (∀ i : ℤ) ((∀ j : ℤ) (0 ≤ i < |t| ∧ 0 ≤ j < |t[i]|) → ((i, j) ∈ vivas ↔ t[i][j] = true))  
}
```

4. Ejercicio 4

4.1. Función principal

```
aux densidadPoblacion (t : toroide) : ℝ =  
  cantidadVivos(t) / (|t| · |t[0]|);
```

4.2. Funciones auxiliares

```
aux cantidadVivos (t : toroide) : ℝ =  
  ∑i=0|t|-1 ∑j=0|t[i]|-1 if t[i][j] = true then 1 else 0 fi;
```

5. Ejercicio 5

5.1. Función principal

```
aux cantVecinosVivos (t : Toroides, f : ℤ, c : ℤ) : ℤ =  
  ∑i=0|t|-1 ∑j=0|t[i]|-1 if t[i][j] = true ∧ sonVecinos(i, j, f, c, t) then 1 else 0 fi;
```

5.2. Predicados auxiliares

```
pred sonVecinos (x, y : ℤ, v1, v2 : ℤ, t : Toroides) {  
  (v1 ≠ x ∨ v2 ≠ y) ∧ (distancia(x, v1, |t|), distancia(y, v2, |t[0]|) ≤ 1)  
}
```

/*Siendo (x,y) las coordenadas de la posición a evaluar y (v1, v2) las del vecino */

5.3. Funciones auxiliares

```
aux abs (n :  $\mathbb{R}$ ) :  $\mathbb{R}$  = if n < 0 then - n else n fi ;  
aux distancia (x1 :  $\mathbb{Z}$ , x2 :  $\mathbb{Z}$ , rango:  $\mathbb{Z}$ ) :  $\mathbb{R}$  = abs(x1 - x2) mod rango ;  
/*Siendo rango la variable para tener en cuenta los vecinos de los bordes*/
```

6. Ejercicio 6

```
pred evolucionDePosicion (t : toroide, posicion :  $\mathbb{Z} \times \mathbb{Z}$ ) {  
  if t[(posicion)0][(posicion)1] = true  
  then 2 ≤ cantVecinosVivos(t, (posicion)0, (posicion)1) ≤ 3  
  else cantVecinosVivos(t, (posicion)0, (posicion)1) = 3 fi  
}
```

7. Ejercicio 7

```
pred evolucionToroide (t1: Toroide, t2: Toroide) {  
  (|t1| = |t2| ∧ |t1[0]| = |t2[0]|) ∧  
  (∀i :  $\mathbb{Z}$ )(∀j :  $\mathbb{Z}$ )(0 ≤ i < |t1| ∧ 0 ≤ j < |t1[i]|) →  
  evolucionDePosicion(t1, (i, j)) ↔ t2[i][j] = true  
}
```

8. Decisiones tomadas

En cada ejercicio asumimos que el usuario debe cumplir con el contrato pasando un toroide válido, para simplificar la escritura.

Ademas de eso no tomamos ninguna decision muy relevante al momento de resolver los ejercicios, mas que interpretar los enunciados y a partir de eso escribir las especificaciones que pensamos adecuadas.