



# TP de Especificación

## Sudoku

21 de Abril de 2017

Algoritmos y Estructuras de Datos I

### Grupo 17

Integrante	LU	Correo electrónico
Maqueda, Ignacio	279/14	ignaciomaqueda95@gmail.com
Parral, Guillermo	280/16	guillermoeparral@gmail.com
Quintela, Gonzalo	089/16	gquintela@dc.uba.ar
Sirio, Tomás	440/16	tomassirio@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

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

# 1. Problemas

```

proc sudoku_esTableroValido (in t: seq⟨seq⟨ℤ⟩⟩, out result: Bool) {
  Pre {True}
  Post {result = esTableroValido(t)}
}

proc sudoku_esCeldaVacía (in t: seq⟨seq⟨ℤ⟩⟩, in f: ℤ, in c: ℤ, out result: Bool) {
  Pre {esTableroValido(t) ∧L 0 ≤ f, c < |t|}
  Post {result = (t[f][c] = 0)}
}

proc sudoku_nroDeCeldasVacías (in t: seq⟨seq⟨ℤ⟩⟩, out result: ℤ) {
  Pre {esTableroValido(t)}
  Post {∑i=0|t|-1 ( ∑j=0|t|-1 if t[i][j] = 0 then 1 else 0 fi)}
}

proc sudoku_primeraCeldaVacíaFila (in t: seq⟨seq⟨ℤ⟩⟩, out result: ℤ) {
  Pre {esTableroValido(t)}
  Post {if (∃i : ℤ)(0 ≤ i < |t| ∧L filaTieneCeldaVacía(t[i]) ∧L (∀j : ℤ)(0 ≤ j < i →L ¬filaTieneCeldaVacía(t[j])))
    then result = i
    else result = -1 fi}
}

proc sudoku_primeraCeldaVacíaColumna (in t: seq⟨seq⟨ℤ⟩⟩, out result: ℤ) {
  Pre {esTableroValido(t)}
  Post {if (∃i : ℤ)(0 ≤ i < |t| ∧L filaTieneCeldaVacía(t[i]) ∧L (∀j : ℤ)(0 ≤ j < i →L ¬filaTieneCeldaVacía(t[j])))
    then result = indicePrimeraCeldaVacíaEnFila(t[i])
    else result = -1 fi}
}

proc sudoku_valorEnCelda (in t: seq⟨seq⟨ℤ⟩⟩, in f: ℤ, in c: ℤ, out result: ℤ) {
  Pre {esTableroValido(t) ∧L 0 ≤ f, c ≤ 8 ∧L t[f][c] ≠ 0}
  Post {result = t[f][c]}
}

proc sudoku_vaciarCelda (inout t: seq⟨seq⟨ℤ⟩⟩, in f: ℤ, in c: ℤ) {
  Pre {esTableroValido(t) ∧L 0 ≤ f, c ≤ 8 ∧L t[f][c] ≠ 0 ∧L t = t0}
  Post {t[f][c] = 0 ∧L (∀i : ℤ)(∀j : ℤ)((0 ≤ i, j < |t| ∧L (i ≠ f ∨ j ≠ c)) →L t[i][j] = t0[i][j])}
}

```

# 2. Predicados y Auxiliares generales

```

pred esMatriz (t: seq⟨seq⟨ℤ⟩⟩) {
  (∀i : ℤ)(∀j : ℤ)(0 ≤ i, j < |t| →L | t[i] | = | t[j] |)
}

fun cantidadFilas (t: seq⟨seq⟨ℤ⟩⟩) : ℤ = |t|;

fun cantidadColumnas (t: seq⟨seq⟨ℤ⟩⟩) : ℤ = if filas(t) > 0 then |t[0]| else 0 fi;

pred esMatrizCuadrada (t: seq⟨seq⟨ℤ⟩⟩) {
  esMatriz(t) ∧ ( cantidadFilas(t) = cantidadColumnas(t) )
}

```

```

pred esTableroValido (t: seq<seq<ℤ>>)) {

  esMatrizCuadrada(t) ∧L |t|=9 ∧L (∀i:ℤ)(∀j:ℤ)(0 ≤ i, j < |t| →L 0 ≤ t[i][j] ≤ 9)}

  pred filaTieneCeldaVacía (f: seq<ℤ>) {
    (∃i:ℤ)(0 ≤ i < |f| ∧L f[i] = 0)
  }

  fun indicePrimeraCeldaVacíaEnFila (s: seq<ℤ>) : ℤ = if ((∃i:ℤ)(0 ≤ i < |s| ∧L s[i] = 0 ∧L
    (∀j:ℤ)(0 ≤ j < i →L s[j] ≠ 0))) then i else -1 fi;

```

### 3. Decisiones tomadas