



DEPARTAMENTO
DE COMPUTACION

Facultad de Ciencias Exactas y Naturales - UBA

Trabajo práctico

Especificación de TADs

September 8, 2025

Algoritmos y Estructuras de Datos

BobElConstructorPorCopia

Integrante	LU	Correo electrónico
Choque, Leandro	252/25	leandroch2002@gmail.com
Musi, Santino	965/24	santinomusi1@gmail.com
Rojas, Damian	209/25	dam.rojas1@gmail.com
Martell, Juan Bautista	622/25	Juanbamartell@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 Supongo que acá iría una descripción

Breve descripción.

Luego veremos bien el formato, esto de momento es para tener un esqueleto.

2 Especificacion

TAD EdR {

obs aula : *Aula*

obs solucion : *Examen*

obs entregas : *Aula*

proc EdR(in a : *Aula*, in s : *Examen*, in cantEstudiantes : \mathbb{Z}) : EdR {

requiere {

$aulaCuadrada(a) \wedge_L$

$aulaVacía(a) \wedge_L$

$examenValido(solucion) \wedge_L$

$cantValidaEstudiantes(a, cantEstudiantes)$

}

asegura {

$(|res.aula| = |a|) \wedge_L$

$aulaCuadrada(res.aula) \wedge_L$

$((\forall i : \mathbb{Z}) (0 \leq i < |res.aula| \rightarrow_L noHayAlumnosJuntos(res.aula[i]))) \wedge_L$

$(cuantosEstudiantes(res.aula) = cantEstudiantes) \wedge_L$

$(res.solucion = s) \wedge_L$

$aulaVacía(res.entregas)$

$(\forall j : \mathbb{Z}) (0 \leq j < |a| \rightarrow_L (\forall k : \mathbb{Z}) (0 \leq k < |a[j]| \rightarrow_L |a[j][k]| \leq 1))$

}

}

pred aulaCuadrada(a : *Aula*) {

$(\forall i : \mathbb{Z}) (0 \leq i < |a| \rightarrow_L |a[i]| = |a|)$

}

pred aulaVacía(a : *Aula*) {

$(\forall i : \mathbb{Z}) (0 \leq i < |a| \rightarrow_L (\forall j : \mathbb{Z}) (0 \leq j < |a[i]| \rightarrow_L |a[i][j]| = 0))$

}

pred examenValido(s : *Examen*) {

$(\forall i : \mathbb{Z}) (0 \leq i < |s| \rightarrow_L s[i] \in \text{conj}(\text{"0"}, \text{"1"}, \text{"2"}, \text{"3"}, \text{"4"}, \text{"5"}, \text{"6"}, \text{"7"}, \text{"8"}, \text{"9"}, \text{" "}))$

}

pred cantValidaEstudiantes(a : *Aula*, e : \mathbb{Z}) {

$(e \leq \text{ifThenElse}(\text{esPar}(|a|), \frac{|a|^2}{2}, \frac{|a|+1}{2} * |a|))$

}

pred noHayAlumnosJuntos(fila : $\text{seq}\langle \text{Examen} \rangle$) {

$(\forall i : \mathbb{Z}) (0 \leq i < |fila| - 1 \rightarrow_L (|fila[i]| > 0 \rightarrow |fila[i+1]| = 0))$

}

aux cuantosEstudiantes(a : *Aula*) : $\mathbb{Z} = \sum_{i=0}^{|a|-1} \sum_{j=0}^{|a[i]|-1} \text{IfThenElse}(|a[i][j]| > 0, 1, 0)$

proc igualdad(in edr1, edr2 : EdR,) : Bool {

requiere { *True* }

asegura {

```

        (res = True)  $\leftrightarrow$ 
        (edr1.aula = edr2.aula)  $\wedge$ 
        (edr1.solucion = edr2.solucion)  $\wedge$ 
        (edr1.entregas = edr2.entregas)
    }
}
proc copiarse(in alumno : Estudiante) : EdR {
    requiere { True }
    asegura { res }
}
proc publicarResolucion(in completar, Completar) : EdR {
    requiere { True }
    asegura { res }
}
proc resolver(in completar, Completar) : EdR {
    requiere { True }
    asegura { res }
}
proc entregar(in alumno : Estudiante) : EdR {
    requiere { True }
    asegura { res }
}
proc chequearCopias(in completar, Completar) : seq < Estudiante > {
    requiere { True }
    asegura { res }
}
proc corregir(in completar, Completar) : seq << estudiante, nota >> {
    requiere { True }
    asegura { res }
}
}

```