



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*

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

requiere {

$aulaCuadrada(a) \wedge_L$

$examenValido(solucion) \wedge_L$

$cantValidaEstudiantes(a, cantEstudiantes)$

}

asegura {

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

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

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

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

$(res.solucion = s)$

}

}

pred aulaCuadrada(a : *Aula*) {

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

}

pred examenValido(s : *Examen*) {

$(True)$

}

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

$ifthenelse(esPar(|a|^2), \frac{|a|^2}{2}, \frac{|a|^2+1}{2})$

}

pred noHayAlumnosJuntos(fila : seq<*Examen*>) {

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

}

pred cuantosEstudiantes(a : *Aula*) {

$(True)$

}

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

requiere { $True$ }

asegura {

$(res = True) \leftrightarrow$

$(edr1.aula = edr2.aula) \wedge_L$

$(edr1.solucion = edr2.solucion)$

}

}

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 }
}
}

```