



DEPARTAMENTO  
DE COMPUTACION

Facultad de Ciencias Exactas y Naturales - UBA

# Trabajo práctico

## Especificacion de TADs

September 10, 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 : *Paso*  
obs entregas : seq(*Alumno*)

```
proc EdR(in dimensionAula :  $\mathbb{Z}$ , in s : Paso, in cantEstudiantes :  $\mathbb{Z}$ ) : EdR {  
  requiere {  
    (dimensionAula > 0)  $\wedge_L$   
    pasoValido(s)  $\wedge_L$   
    cantValidaEstudiantes(dimensionAula, cantEstudiantes)  
  }  
  asegura {  
    (|res.aula| = dimensionAula)  $\wedge_L$   
    aulaCuadrada(res.aula)  $\wedge_L$   
    noHayAlumnosJuntos(res.aula)  $\wedge_L$   
    (cuantosEstudiantes(res.aula) = cantEstudiantes)  $\wedge_L$   
    examenesSinResponder(res.aula)  $\wedge_L$   
    (res.solucion = s)  $\wedge_L$   
    (res.entregas =  $\langle \rangle$ )  
  }  
}  
  
pred pasoValido(s : Paso) {  
  ( $\forall i : \mathbb{Z}$ ) (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  $\leq \text{ifThenElseFi}(\text{esPar}(|a|), \frac{|a|^2}{2}, \frac{|a|+1}{2} * |a|)$ )  
}  
  
pred esPar(a :  $\mathbb{Z}$ ) { (mod(a, 2) = 0) }  
pred aulaCuadrada(a : Aula) {  
  ( $\forall i : \mathbb{Z}$ ) (0  $\leq i < |a| \rightarrow_L |a[i]| = |a|$ )  
}  
  
pred noHayAlumnosJuntos(a : Aula) {  
  ( $\forall i : \mathbb{Z}$ ) (0  $\leq i < |a| \rightarrow_L$   
  ( $\forall j : \mathbb{Z}$ ) (0  $\leq j < |a[i]| - 1 \rightarrow_L$   
  (a[i][j]  $\neq \langle \rangle \rightarrow a[i][j+1] = \langle \rangle$ )))  
}  
  
aux cuantosEstudiantes(a : Aula) :  $\mathbb{Z} = \sum_{i=0}^{|a|-1} \sum_{j=0}^{|a[i]|-1} \text{ifThenElseFi}(a[i][j] \neq \langle \rangle, 1, 0)$   
pred examenesSinResponder(a : Aula) {  
  ( $\forall i : \mathbb{Z}$ ) (0  $\leq i < |a| \rightarrow_L (\forall j : \mathbb{Z}) (0 \leq j < |a[i]| \rightarrow_L \text{examenSinResponder}(a[i][j].examen))$ )  
}  
  
pred examenSinResponder(e : Examen) {  
  (|e| = 1  $\wedge (\forall i : \mathbb{Z}) (0 \leq i < |e| \rightarrow e[0][i] = \text{" "})$ )  
}
```

```

}
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 : Alumno,inout aula : Aula) : {
  requiere {
    (|aula[alumno.f][alumno.c]| > 0)  $\wedge_L$ 
    ( $\exists i : Coordinada$ )(esAlumnoCercano(i,alumno) $\wedge$ )
  }
  asegura { res }
}
}
proc consultarDarkWeb(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 }
}
}
}

```