



DEPARTAMENTO
DE COMPUTACION

Facultad de Ciencias Exactas y Naturales - UBA

Descripción del tp

Subtítulo del tp

6 de septiembre de 2024

Materia de la carrera

Grupo 42

Integrante	LU	Correo electrónico
Krivonosoff, Thiago	310/24	thiagokribas@gmail.com
Pelli, Agustin	002/01	email2@dominio.com
Miguel, Facundo	003/01	email3@dominio.com
Montenegro, Ulises	477/24	ulinicolasmonte@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. Ejemplo de sección

1.1. Subsección: ambientes comunes de L^AT_EX

Lo principal: las fórmulas. Se puede poner en una linea, como $x_i = x_{i-1} + x_{i-2}$, o ponerse más grande:

$$\sum_{i=0}^n i$$

(1)

Y se pueden citar ecuaciones con `\eqref{nombreDeEq}`: (1)
Ejemplo de itemizado:

- Item 1
- Item 2
- Item 3

Ejemplo de enumerado con menor distancia entre items:

1. Item 1
2. Item 2
3. Item 3

Podemos escribir mucho texto. Mucho texto. Mucho texto. Mucho texto. Mucho texto. Mucho texto. Mucho texto. Mucho texto. Mucho texto. Mucho texto. Mucho texto. Mucho texto. Mucho texto.

Otro párrafo. Otro párrafo. Otro párrafo. Otro párrafo. Otro párrafo. Otro párrafo. Otro párrafo. Otro párrafo. Otro párrafo. Otro párrafo. Otro párrafo. Otro párrafo. Otro párrafo. Otro párrafo. Otro párrafo. Otro párrafo.

Le agregamos una separación entre párrafos. Le agregamos una separación entre párrafos. Le agregamos una separación entre párrafos. Le agregamos una separación entre párrafos. Le agregamos una separación entre párrafos.

La tabla 1 es un ejemplo de cómo se hace una tabla.

Col1	Col2	Col2	Col3
1	6	87837	787
2	7	78	5415
3	545	778	7507
4	545	18744	7560
5	88	788	6344

Tabla 1: Ejemplo de tabla

La figura 2 es un ejemplo de cómo se agrega una imagen.



Figura 1: Ejemplo de figura

```
1 res := 0;
2 i := 0;
3 while (i < s.size()) do
4     res := res + s[i];
5     i := i + 1
6 endwhile
```

Código 1: Ejemplo de código (usando los estilos de la cátedra, ver las macros para más detalles)

Si se pone un label al `lstlisting`, se puede referenciar: Código 1.



(a) Logo de LaTeX



(b) Logo de TeX

Figura 2: Ejemplo para poner dos figuras juntas. Y citarlas por separado a (a) y (b).

1.2. Macros de la cátedra para especificar

```

proc nombre (in paramIn : N, inout paramInout : seq(Z)) : tipoRes
  requiere {expresionBooleana1}
  asegura {expresionBooleana2}
  aux auxiliar1 (parametros) : tipoRes = expresion;
  pred pred1 (parametros) {
    expresion
  }

aux auxiliarSuelto (parametros) : tipoRes = expresion;
pred predSuelto (parametros) {
  (∀variable : tipo) (algo →L expresion)
}
pred predSuelto (parametros) {
  (∃variable : tipo) (algo ∧L expresion)
}

```

A partir de aca empieza el TP.

2. Problema 1

```

proc grandesCiudades (in ciudades : seq(Ciudad)) : seq(Ciudad)
  requiere {noRepetidos(ciudades) ∧ noHabitantesNegativos(ciudades)}
  asegura {|res| ≤ |ciudades|}
  asegura {(∀elem : Ciudad) ((elem.habitantes > 50000 ∧ elem ∈ ciudades) → elem ∈ res)}

pred noRepetidos (in ciudades : seq(Ciudad)) {
  (∀i : Z) (0 ≤ i < |ciudades|) →L (∀j : Z) (0 ≤ j < |ciudades|) →L (i ≠ j → ciudades[i].nombre ≠
ciudades[j].nombre)
}
pred noHabitantesNegativos (in ciudades : seq(Ciudad)) {
  (∀i : Z) (0 ≤ i < |ciudades|) →L ciudades[i].habitantes ≥ 0
}

```

3. Problema 2

```

proc sumaDeHabitantes (in menoresDeCiudades : seq(Ciudad), in mayoresDeCiudades : seq(Ciudad)) : seq(Ciudad)
  requiere {noRepetidos(menoresDeCiudades) ∧ noRepetidos(mayoresDeCiudades)}
  requiere {noHabitantesNegativos(menoresDeCiudades) ∧ noHabitantesNegativos(mayoresDeCiudades)}
  requiere {mismosElementos(menoresDeCiudades, mayoresDeCiudades)}
  requiere {|menoresDeCiudades| = |mayoresDeCiudades|}
  asegura {|res| = |mayoresDeCiudades|}
  asegura {(∀elem : Ciudad) ((elem ∈ res) →
(∀i : Z) (0 ≤ i < |res|) →L (∃j : Z) (0 ≤ j < |res|)) ∧L
mayoresDeCiudades[i].nombre = menoresDeCiudades[j].nombre →
res[i].nombre = mayoresDeCiudades[i].nombre ∧
res[i].habitantes = mayoresDeCiudades[i].habitantes + menoresDeCiudades[j].habitantes)}

```

```

pred mismosElementos (in s1: seq⟨Ciudad⟩, in s2: seq⟨Ciudad⟩) {
  (∀i : ℤ) (0 ≤ i < |s1|) →L (∃j : ℤ) (0 ≤ j < |s1|) ∧L s1[i].nombre = s2[j].nombre
}

```

4. Problema 3

```

proc hayCamino (in distancia: seq⟨seq⟨ℤ⟩⟩, in desde: ℤ, in hasta: ℤ) : Bool
  requiere {esCuadrada(distancia)}
  requiere {0 ≤ desde < |distancia|}
  requiere {0 ≤ hasta < |distancia|}
  requiere {filaIgualColumna(distancia)}
  requiere {matrizTodosPositivos(distancia)}
  asegura {(∃sec : seq⟨ℤ⟩) ((∀i : ℤ) (0 ≤ i < |sec|) →L 0 ≤ sec[i] < |distancia| ∧
    sec[0] = desde ∧ sec[|sec| - 1] = hasta ∧
    todosConexionAnterior(sec, distancia))}

pred todosConexionAnterior (in sec: seq⟨ℤ⟩, in mat: seq⟨seq⟨ℤ⟩⟩) {
  (∀j : ℤ) (1 ≤ i < |sec| →L mat[sec[i]][sec[i - 1]] ≠ 0)
}

pred esCuadrada (in mat: seq⟨seq⟨ℤ⟩⟩) {
  (∀i : ℤ) (0 ≤ i < |mat|) →L |mat| = |mat[i]|
}

pred filaIgualColumna (in mat: seq⟨seq⟨ℤ⟩⟩) {
  (∀i : ℤ) (0 ≤ i < |mat|) →L (∀j : ℤ) (0 ≤ j < |mat|) →L mat[i][j] = mat[j][i]
}

pred matrizTodosPositivos (in mat: seq⟨seq⟨ℤ⟩⟩) {
  (∀i : ℤ) (0 ≤ i < |mat|) →L (∀j : ℤ) (0 ≤ j < |mat|) →L mat[i][j] ≥ 0
}

```