

# **Practica 4:**

## **Construcción de árboles de sintaxis abstracta**

**Grupo 11:**

**Youssef El Faqir El Rhazoui**

**Enrique Ávila Rodríguez**

# 1. Conjunto de funciones constructoras

**Prog:**  $\text{Sec\_Dec} \times \text{Sec\_Ins} \rightarrow \text{Prog}$

**Sec\_Dec:**  $\text{LDs} \rightarrow \text{Prog}$

**Sec\_Ins:**  $\text{LIs} \rightarrow \text{Prog}$

**LD\_simp:**  $\text{String} \times \text{String} \rightarrow \text{LDs}$

**LD\_comp:**  $\text{String} \times \text{String} \times \text{LDs} \rightarrow \text{LDs}$

**LI\_simp:**  $\text{String} \times \text{Exp} \rightarrow \text{LIs}$

**LI\_comp:**  $\text{String} \times \text{Exp} \times \text{LIs} \rightarrow \text{LIs}$

**Mas:**  $\text{Exp} \times \text{Exp} \rightarrow \text{Exp}$

**Menos:**  $\text{Exp} \times \text{Exp} \rightarrow \text{Exp}$

**And:**  $\text{Exp} \times \text{Exp} \rightarrow \text{Exp}$

**Or:**  $\text{Exp} \times \text{Exp} \rightarrow \text{Exp}$

**Distinto:**  $\text{Exp} \times \text{Exp} \rightarrow \text{Exp}$

**Igual:**  $\text{Exp} \times \text{Exp} \rightarrow \text{Exp}$

**Menor\_que:**  $\text{Exp} \times \text{Exp} \rightarrow \text{Exp}$

**Menor\_igual\_que:**  $\text{Exp} \times \text{Exp} \rightarrow \text{Exp}$

**Mayor\_que:**  $\text{Exp} \times \text{Exp} \rightarrow \text{Exp}$

**Mayor\_igual\_que:**  $\text{Exp} \times \text{Exp} \rightarrow \text{Exp}$

**Por:**  $\text{Exp} \times \text{Exp} \rightarrow \text{Exp}$

**Div:**  $\text{Exp} \times \text{Exp} \rightarrow \text{Exp}$

**Not:**  $\text{Exp} \rightarrow \text{Exp}$

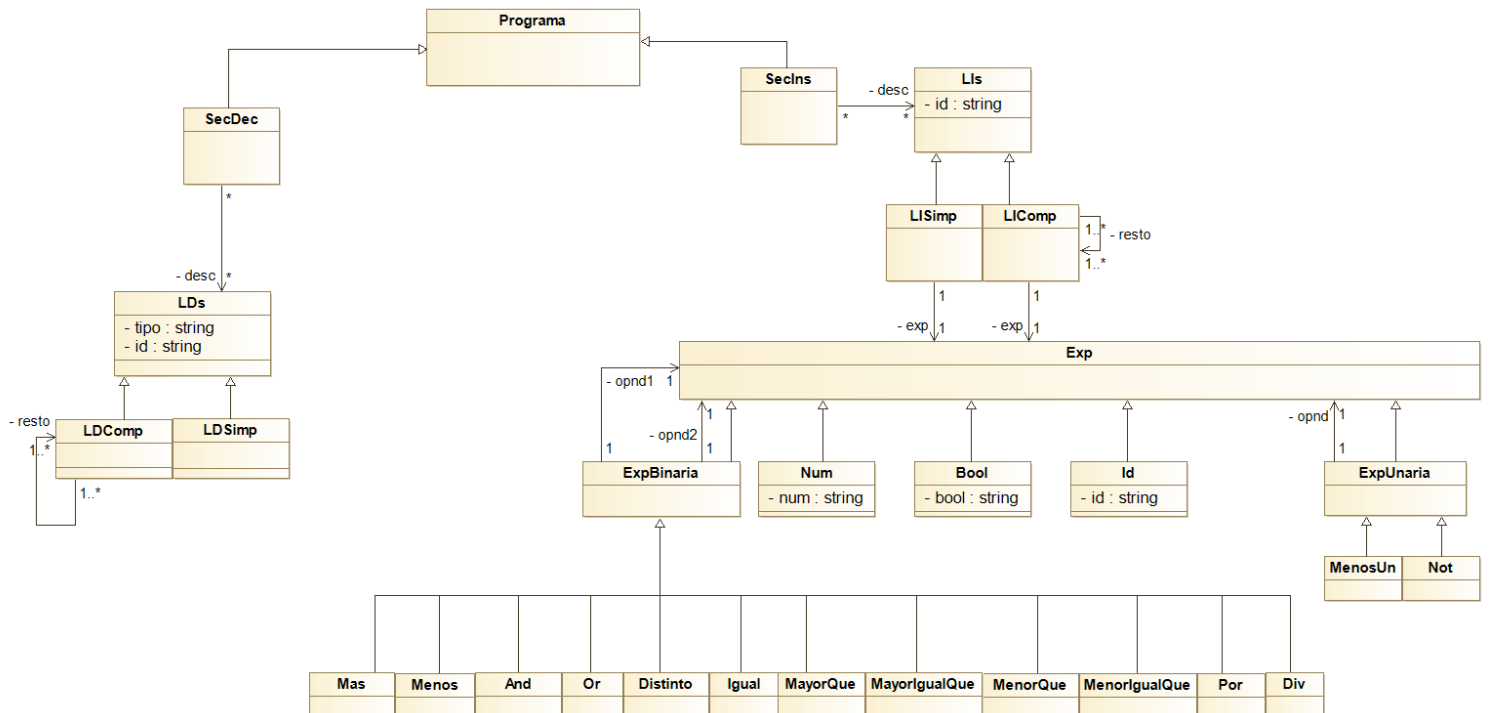
**Menos\_unario:**  $\text{Exp} \rightarrow \text{Exp}$

**Num:**  $\text{String} \rightarrow \text{Exp}$

**Bool:**  $\text{String} \rightarrow \text{Exp}$

**Id:**  $\text{String} \rightarrow \text{Exp}$

## 2. Diagrama de clases



## 3. Gramática de atributos

Prog  $\rightarrow$  Sec\_Dec Sec\_Ins

Sec\_Dec  $\rightarrow$  LDs; D

Sec\_Dec.a = ldCompuesta(LDs.a, D.tipo, D.id)

Sec\_Dec  $\rightarrow$  D

Sec\_Dec.a = ldSimple(D.tipo, D.id)

Sec\_Ins  $\rightarrow$  LIs; I

Sec\_Ins.a = liCompuesta(LIs.a, I.id, I.exp)

Sec\_Ins  $\rightarrow$  I

Sec\_Ins.a = liSimple(I.id, I.exp)

D  $\rightarrow$