

GRUPO 3

Prolog 1y2

Teoría de Lenguajes de Programación

Sara Acevedo Maya

saacevedom@unal.edu.co

Universidad Nacional de Colombia



UNIVERSIDAD
NACIONAL
DE COLOMBIA

Contenido

- 01/ Repaso Teórico
- 02/ Derivaciones y Árboles
- 03/ Hechos y reglas



UNIVERSIDAD
NACIONAL
DE COLOMBIA

01- Repaso Teórico



UNIVERSIDAD
NACIONAL
DE COLOMBIA



Estático vs dinámico

El tipado **estático** realiza una verificación del tipo de las variables es compatible con una operacion dada, esta se realiza antes de la ejecución, ejemplos son C, Java o Scala

El tipado **dinámico** realiza esta misma verificacion pero durante la ejecución del programa ejemplos como python y ruby



UNIVERSIDAD
NACIONAL
DE COLOMBIA



	Tipado Estático	Tipado Dinámico
Comprobación de Tipos	En tiempo de compilación	En tiempo de ejecución
Cambio de Tipos	No cambian durante la ejecución del programa	Pueden cambiar durante la ejecución del programa
Ejemplos de lenguajes	C, C++, Java, TypeScript	Python, JavaScript, Ruby, PHP



Débil vs Fuerte

El **tipado débil** es donde no indicamos el tipo de dato al declarar una variable y permite realizar operaciones entre tipos u objetos.

Ejemplos: Javascript, PHP

```
> let resultado = "x" + 5;  
< undefined  
> resultado  
< "x5"
```

El **tipado fuerte** es donde indicamos el tipo de dato al declarar y no se pueden realizar operaciones entre distintos tipos.

Ejemplos: Java, C, Ruby, Python

```
>>> resultado = "x" + 5  
Traceback (most recent call last):  
  File "<stdin>", line 1, in <module>  
TypeError: must be str, not int  
>>>
```



UNIVERSIDAD
NACIONAL
DE COLOMBIA

Explícito vs Implícito

	Tipado Explícito	Tipado Implícito
Declaración de Tipos	Los tipos se especifican explícitamente por el/la programador/a al declarar	Los tipos se deducen automáticamente según el contexto de uso
Ejemplo (Java)	<code>int edad = 25;</code>	<code>var edad = 25;</code>
Ejemplo (Python)	<code>edad: int = 25</code>	<code>edad = 25</code>



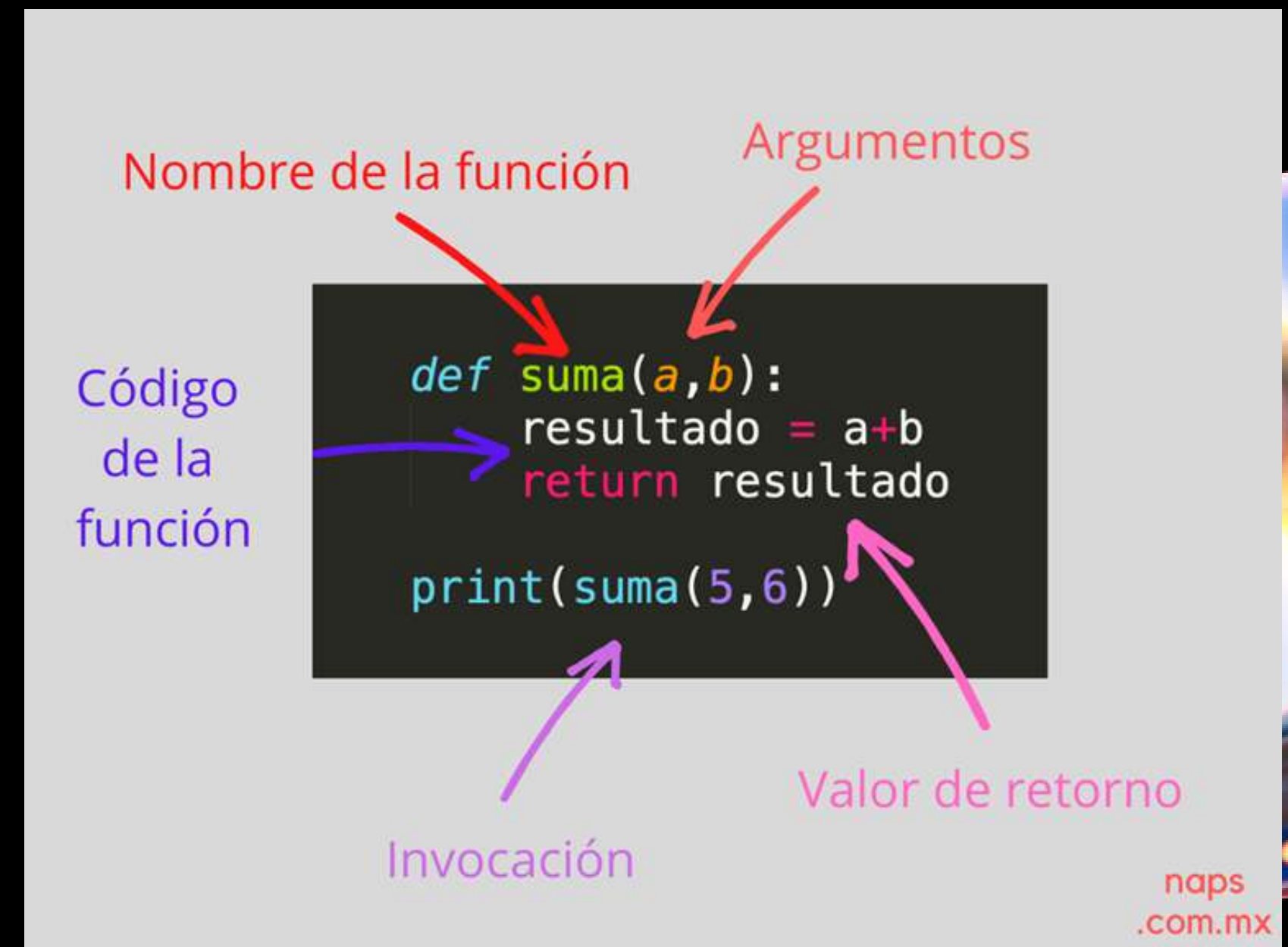
Explicito vs Implicito

```
1  /**
2   * @author Pablo Ruiz Soria
3   */
4  public class Lanzador {
5
6      static float hazMedia(float dato1, float dato2){
7          return (dato1+dato2)/2;
8      }
9
10     public static void main(String[] args) {
11         float numero1 = 4;
12         float numero2 = 7;
13         float medial = hazMedia(numero1,numero2);
14         System.out.println("medial = " + medial);
15         float numero3 = 9;
16         float numero4 = 6;
17         float media2 = hazMedia(numero3, numero4);
18         System.out.println("media2 = " + media2);
19     }
20 }
```

Lanzador > main >

Output - Modulo2Funciones (run) x

run:
medial = 5.5
media2 = 7.5



Nominal vs estructural

El tipado **nominal** es donde dos variables son compatibles si su nombre de tipo es igual

Ejemplo: Java, Scala

El tipado **estructural** es aquel en el cual dos tipos son compatibles por sus características

Ejemplos: Scala, Smalltalk



UNIVERSIDAD
NACIONAL
DE COLOMBIA

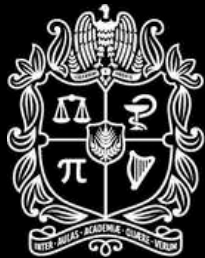
Nominal vs estructural

	Tipado Nominal	Tipado Estructural
Criterio de Comparación	La compatibilidad se basa en los nombres de los tipos.	La compatibilidad se basa en la estructura interna de los tipos.
Ejemplo	(Java) Dos clases con nombres diferentes no son consideradas iguales, incluso si tienen la misma estructura interna.	(TypeScript) Dos objetos con la misma estructura (mismas propiedades) se consideran del mismo tipo, independientemente de sus nombres.



Análisis Léxico vs Sintáctico vs Semántico

	Análisis Léxico	Análisis Sintáctico	Análisis Semántico
Función	Verifica que las palabras estén bien escritas y tengan sentido en el contexto.	Verifica que las palabras estén organizadas correctamente según las reglas gramaticales del lenguaje de programación.	Verifica que las palabras y estructuras tengan un significado lógico y coherente.
Ejemplo	Lee cada palabra para asegurarse de que esté correctamente escrita, como "SALTA" o "CORRE".	Revisa cómo están organizadas las palabras para formar estructuras gramaticales adecuadas, como "SALTA RÁPIDO" o "CORRE LENTO".	Analiza el significado de las palabras y las estructuras para garantizar que tengan sentido, como "SALTA ALTO" o "CORRE RÁPIDO".
Enfoque	Se enfoca en la corrección léxica de las palabras individuales.	Se enfoca en la corrección sintáctica de la estructura del lenguaje.	Se enfoca en la corrección semántica del significado de las palabras y las estructuras.
Resultado	Lista de tokens (palabras válidas) y posibles errores léxicos.	Árbol de análisis (sintaxis gramatical) y posibles errores sintácticos.	Árbol de análisis (estructura lógica) y posibles errores semánticos.



Derivaciones

Imagina que estás construyendo una torre con bloques de juguete. Cada bloque de juguete tiene una letra escrita. Ahora, quieres construir una torre siguiendo algunas reglas.

La **derivación** es como seguir las reglas para apilar los bloques de juguete y construir la torre.

- Cuando haces una derivación por la derecha, siempre agregas un nuevo bloque al final de tu torre.
- Cuando haces una derivación por la izquierda, siempre agregas un nuevo bloque al principio de tu torre.



UNIVERSIDAD
NACIONAL
DE COLOMBIA

Derivaciones

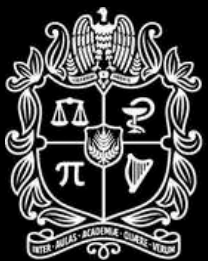
Una derivación es una secuencia de sentencias que comienza con el símbolo de inicio de la gramática y termina con la sentencia que está tratando de derivar.



Derivación izquierda (de izquierda a derecha): siempre reemplace la derivación más a la izquierda cuando se pasa de una expresión gramatical a la siguiente en una derivación.



Derivación derecha (de derecha a izquierda): reemplace siempre el no terminal más a la derecha al pasar de una forma de expresión gramatical a la siguiente en una derivación.



02 - Derivaciones y Árboles

Notaciones

(Prejia, infija y postfija)

	$(2+(3*4))$	$((2+3)*4)$
Prefija	$+ 2 * 3 4$	$* + 2 3 4$
Infija	$2 + 3 * 4$	$(2 + 3) * 4$
Postfija	$2 3 4 * +$	$2 3 + 4 *$

Nota: la notación **prefija**, conocida también como **notación Polaca**, nos indica que el operador va antes de los operandos:

- No requiere paréntesis para indicar el orden de precedencia de operadores.
- Se evalúa de izquierda a derecha hasta que encontremos el primer operador seguido inmediatamente de un par de operandos.
- Se evalúa la expresión binaria y el resultado se cambia como un nuevo operando. Se repite esto mismo que nos quede un solo resultado.

Gramáticas con notación infija

Pueden especificarse como:

$G = (N, T, P, E)$

N – No terminal, T – Terminal

P – Producciones, E – Símbolo de inicio

$N = \{E, T, F\}$

$T = \{\text{identificador, número, +, -, *, /, (,)}\}$

P se define por el conjunto de producciones (o reglas de producción)

$E \rightarrow E + T \mid E - T \mid T$

$T \rightarrow T * F \mid T / F \mid F$

$F \rightarrow (E) \mid \text{identificador} \mid \text{número}$

No terminal: algo que puedo seguir derivando

Terminal: Algo que ya no tiene otra representación

P: Las reglas de producción que son las que me dicen si puedo ir reemplazando (o derivando)

E: Símbolo de inicio o de arranque

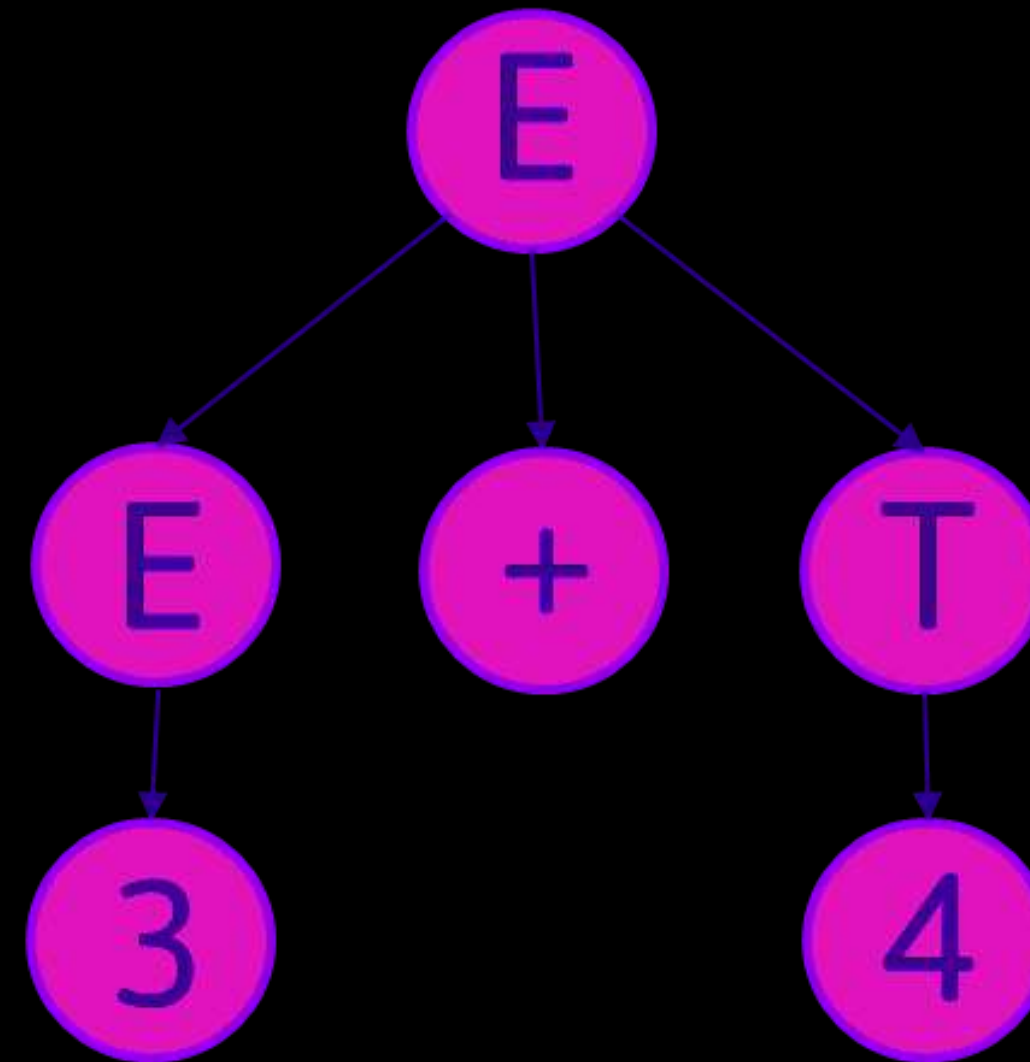


UNIVERSIDAD
NACIONAL
DE COLOMBIA

ÁRBOLES

A partir de una **gramática G** se puede construir un **árbol** que **represente** una **expresión** del lenguaje de la gramática **L(G)**.

3+4



Gramáticas con notación infija

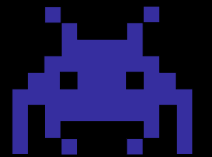
¿La expresión: $(3 / x) * (y+1)$ hace parte de esta gramática ?

$T = \{\text{identificador, número, +, -, *, /, (,)}\}$

$E \rightarrow E + T \mid E - T \mid T$

$T \rightarrow T * F \mid T / F \mid F$

$F \rightarrow (E) \mid \text{identificador} \mid \text{número}$



Derivación izquierda:

$E \Rightarrow ???$



UNIVERSIDAD
NACIONAL
DE COLOMBIA

Gramáticas con notación infija

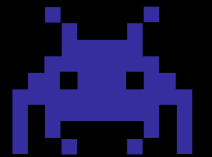
¿La expresión: $(3 / x) * (y+1)$ hace parte de esta gramática ?

$T = \{\text{identificador, número, +, -, *, /, (,)}\}$

$E \rightarrow E + T \mid E - T \mid T$

$T \rightarrow T * F \mid T / F \mid F$

$F \rightarrow (E) \mid \text{identificador} \mid \text{número}$



Derivación izquierda:

$E \Rightarrow T \Rightarrow T * F \Rightarrow F * F \Rightarrow (E) * F \Rightarrow (T) * F \Rightarrow (T / F) * F \Rightarrow (F / F) * F \Rightarrow (3 / F) * F \Rightarrow$
 $(3 / x) * F \Rightarrow (3 / x) * (E) \Rightarrow (3 / x) * (E + T) \Rightarrow (3 / x) * (T + T) \Rightarrow (3 / x) * (F + T) \Rightarrow$
 $(3 / x) * (y + T) \Rightarrow (3 / x) * (y + F) \Rightarrow (3 / x) * (y + 1)$



ÁRBOLES

Árbol de análisis sintáctico para la expresión infija

$$(3 / x) * (y+1)$$

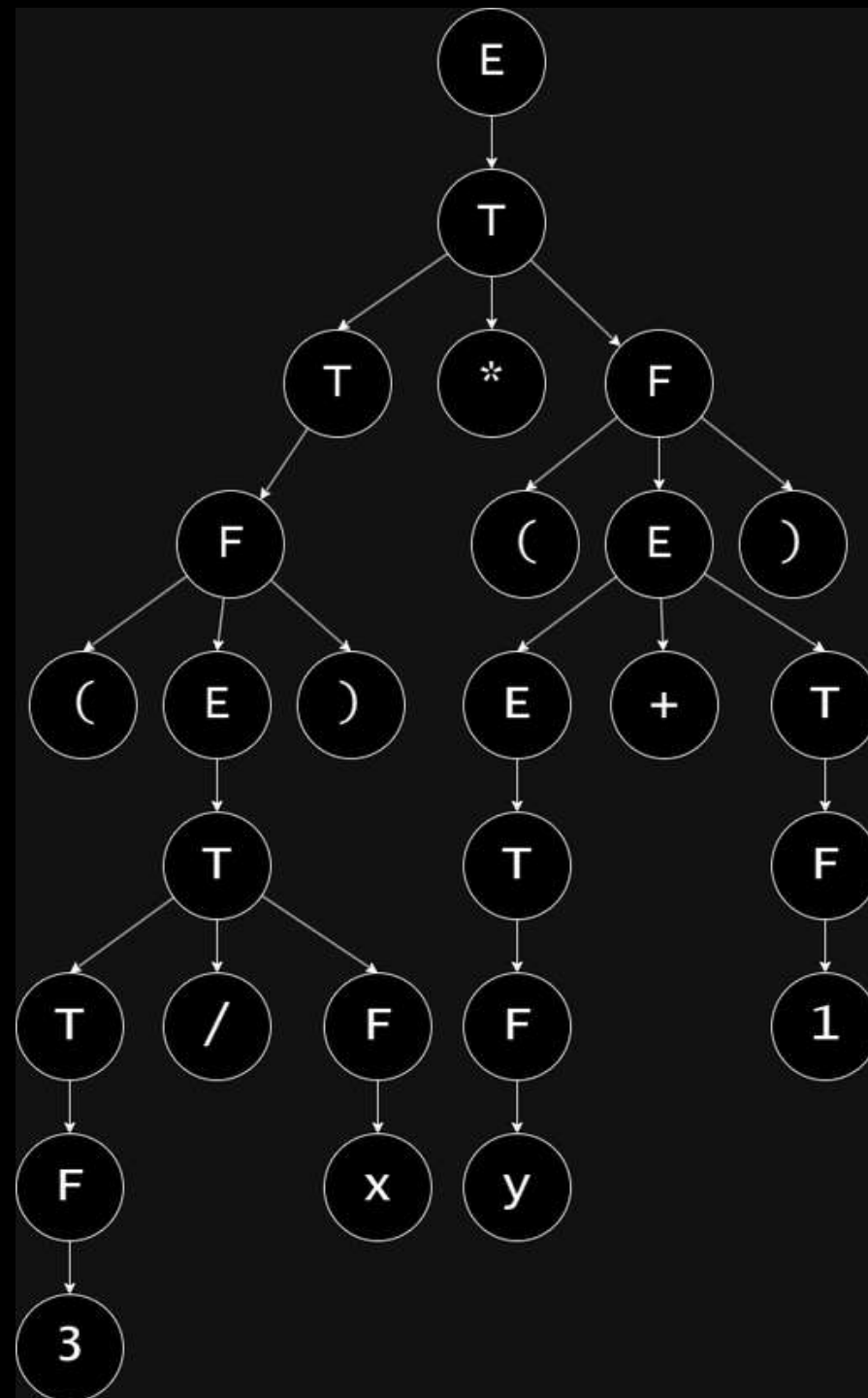
???



UNIVERSIDAD
NACIONAL
DE COLOMBIA

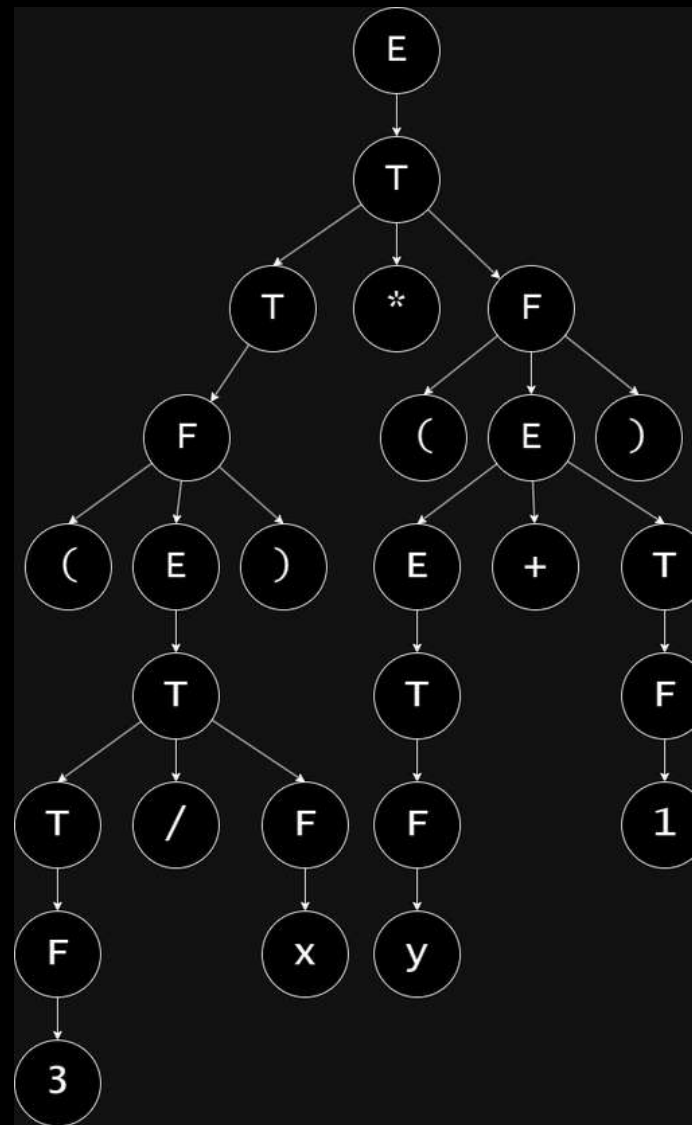
ÁRBOLES

Árbol de análisis sintáctico para la expresión infija
 $(3 / x) * (y+1)$



ÁRBOLES

Árbol de análisis sintáctico **ABSTRACTO** para la expresión infija $(3 / x) * (y+1)$



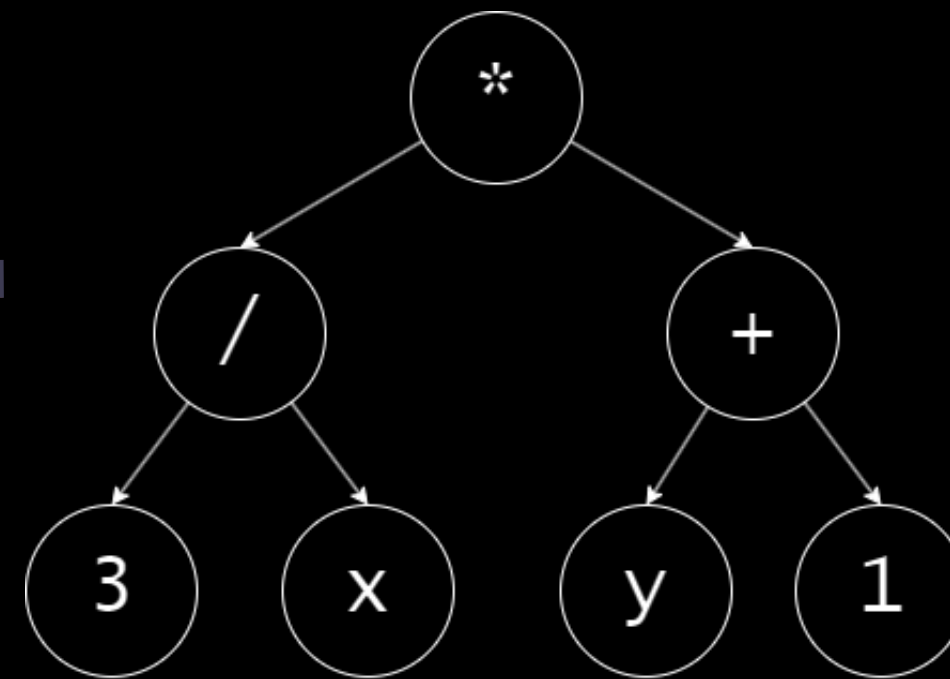
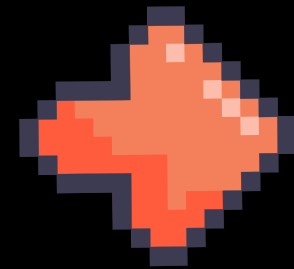
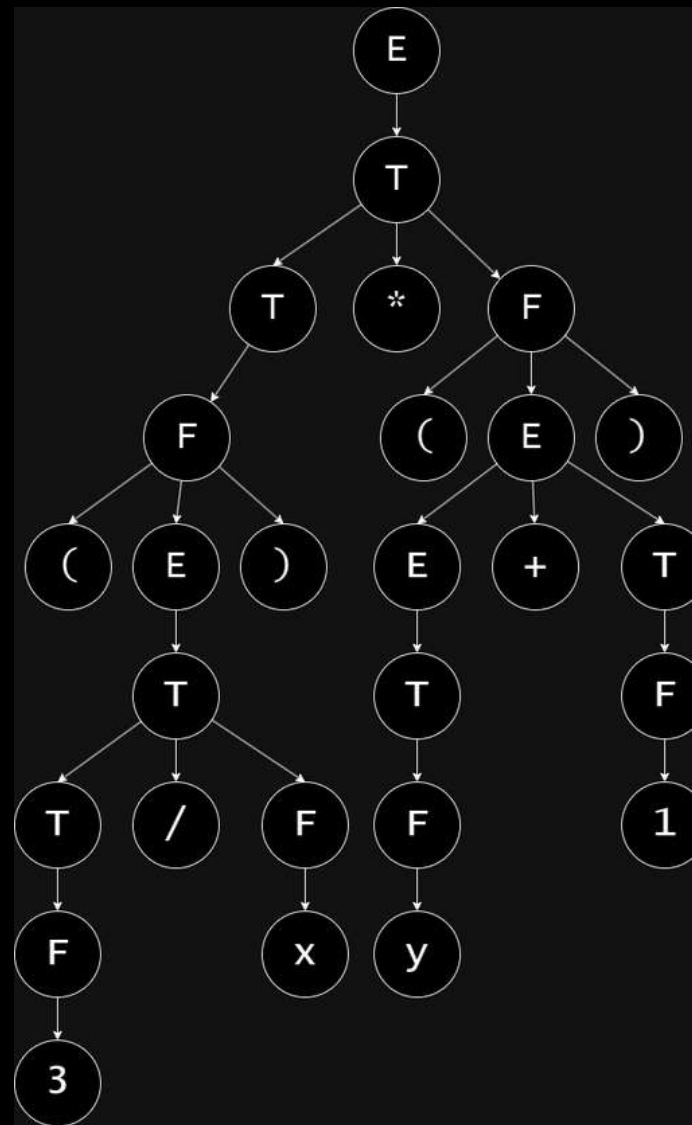
???



UNIVERSIDAD
NACIONAL
DE COLOMBIA

ÁRBOLES

Árbol de análisis sintáctico **ABSTRACTO** para la expresión infija **(3 / x) * (y+1)**



03 - Hechos y reglas



UNIVERSIDAD
NACIONAL
DE COLOMBIA



**¿Que es un
hecho?**



UNIVERSIDAD
NACIONAL
DE COLOMBIA



¿Que es un hecho?

Es algo que ya se sabe que es verdadero.
Para representar un hecho en prolog se
debe iniciar por minuscula y puede o no
tener argumentos.

`es_mamifero(perro).`
`tiene_cuatro_patas(perro).`



UNIVERSIDAD
NACIONAL
DE COLOMBIA



**¿Que es una
regla?**



UNIVERSIDAD
NACIONAL
DE COLOMBIA

¿Que es una regla?

Son elementos que nos permiten realizar inferencias a partir de los hechos que tenemos.

En prolog esta conformada de una cabeza y un cuerpo (podemos decir que la cabeza es la informacion que queremos obtener y el cuerpo contiene la logica para llegar al resultado.

`es_gato(X) :- felino(X), domestico(X).`



UNIVERSIDAD
NACIONAL
DE COLOMBIA



UNIVERSIDAD
NACIONAL
DE COLOMBIA

Sintaxis basica de Prolog

Hechos

H

Inician siempre con minúscula, pueden tener o no argumentos y terminan con un punto, además permite valores alfanuméricos y _

Variables

Var

Son cadenas de letras, digitos y _ que inician por mayusculas

Reglas

R

Se componen de cabeza y cuerpo separados por :- además al lado derecho se encontraran las reglas de inferencia separadas por ,

Comentarios

/**/

Cualquier línea de texto que se encuentre entre los símbolos /**/



UNIVERSIDAD
NACIONAL
DE COLOMBIA

*/*Tipos de hechos*/*

llueve. */*hecho sin argumentos*/*

lee(juan). lee(camila). */*hechos con 1 argumento*/*

caminan(felipe, alejandro). */*hecho con 2 argumento*/*

gusta(juan, rock). gusta(juan, futbol).

gusta(rogelio, futbol). gusta(rogelio, pintar).

gusta(camila, caminar).

gusta(laura, cantar). gusta(laura, pintar).

gusta(andres, nadar). gusta(andres, pintar).

gusta(sebastian, baloncesto).


amigos(X, Y) :- gusta(X, Z), gusta(Y, Z), X \== Y. */*regla para amistades*/*

enemigos(X, Y) :- gusta(X, futbol), gusta(Y, baloncesto). */*regla para enemistades*/*


enemigos(X, Y) :- amigos(X,W), gusta(W, futbol), gusta(Y, baloncesto). */*el enemigo de mi amigo tambien es mi enemigo*/*










UNIVERSIDAD
NACIONAL
DE COLOMBIA



 **SWISH**


File Edit Examples Help

New tab 

 Save ...
 Info & history ...
 Reload
 Open recent
 Backends ...

 Follow ...
 Start TogetherJS ...

 Download
 Print ...

 **No matching files**

If you are a new user:



- Use the Examples tab
- Use the Program or Notebook button above

[help on search](#)

Notebook here

ent CLP s(CASP) profile

user:"me"

 Filter Type 



TALLER

<https://forms.gle/v5ob6uHzky4F474n7>



UNIVERSIDAD
NACIONAL
DE COLOMBIA

Teniendo en cuenta la serie de hechos y reglas entregadas en la siguiente diapositiva, realice las correspondientes reglas para tío, sobrino, primos y bisabuela, recordar discriminar por sexo en el caso de tío y bisabuela (recordar hacer la entrega del archivo apellido_nombre_grupoX.pl **antes del domingo 10 Marzo a las 10:00 am**)

- hijo(sebas,estefa). hijo(sebas,mario).
- hijo(juanita,estefa). hijo(juanita,mario).
- hijo(estefa,carlos). hijo(estefa,claudia).
- hijo(lola, mario). hijo(rodrigo, consuelo).
- hijo(rodrigo,sebas). hijo(marta, juanita).
- hijo(freddy, juanita). hijo(susana, juanita).
- hijo(marta, luis). hijo(freddy, luis).
- hijo(susana, luis).
- hombre(sebas). hombre(mario). hombre(carlos). hombre(rodrigo). hombre(freddy).
- hombre(luis).
- mujer(estefa). mujer(juanita). mujer(claudia). mujer(marta). mujer(susana).
- mujer(lola). mujer(consuelo).
- padre(P,X) :- hijo(X,P), hombre(P).
- madre(M,X) :- hijo(X,M), mujer(M).
- sexo_opuesto(X,Y) :- hombre(X), mujer(Y).
- sexo_opuesto(Y,X) :- hombre(X), mujer(Y).
- abuelo(X,Z) :- padre(X,Y), hijo(Z,Y).
- abuela(Y,Z) :- madre(Y,R), hijo(Z,R).
- hermanos(X, Y) :- hijo(X, Z), hijo(Y, Z), X\=Y.