

## Práctica Nro. 3

### Semántica

**Objetivo:** Interpretar el concepto de semántica de los lenguajes de programación.

**Ejercicio 1:** ¿Qué define la semántica?

**Ejercicio 2:**

- ¿Qué significa compilar un programa?
- Describe brevemente cada uno de los pasos necesarios para compilar un programa.
- ¿En qué paso interviene la semántica y cual es su importancia dentro de la compilación?

**Ejercicio 3:** Con respecto al punto anterior ¿es lo mismo compilar un programa que interpretarlo? Justifique su respuesta mostrando las diferencias básicas, ventajas y desventajas de cada uno.

**Ejercicio 4:** Explique claramente la diferencia entre un error sintáctico y uno semántico. Ejemplifique cada caso.

**Ejercicio 5:** Sean los siguientes ejemplos de programas. Analice y diga qué tipo de error se produce (Semántico o Sintáctico) y en qué momento se detectan dichos errores (Compilación o Ejecución).

*Aclaración: Los valores de la ayuda pueden ser mayores.*

**a) Pascal**

Program **P** **Sintáctico** Falta el ; , se detecta en compilación

var **5: integer;** **No se puede asignar a un número como nombre a una variable integer**  
var a:char;  
Begin

for **i:=5** to 10 do begin **No esta declarada la variable i.**  
write(**a**); **No esta inicializada la variable a**  
**a=a+1;** **No se le puede asignar un valor a una variable char.**  
end; **Deberia ser a:= a+1; .Falta el ; . Se detecta en compilación.**

End.

Ayuda: **Sintáctico** 2, **Semántico** 3

**b) Java:**

public String tabla(int numero, **a**rrayList<Boolean> listado)

```
{
    String result = null;
    for(i = 1; i < 11; i--) {
        result += numero + "x" + i + "=" + (i*numero) + "\n";
        listado.get(listado.size()-1)=(BOOLEAN) numero>i;
    }
    return true;
}
```

**Lógico:** La i en el for siempre sera menor a 11. Por lo que se trata de un loop.

**Sintáctico:**

**\_arrayList** deberia ser ArrayList  
**\_BOOLEAN** deberia ser Boolean  
**\_listado.get(listado.size()-1)** deberia ser una variable

**Semántico:**

**\_la variable i** no esta inicializada. Compilación  
**\_numero > i.** No se puede transformar un integer en un boolean. Se detecta en compilacion.  
**\_return true:** ya se habia especificado que se iba a devolver un String. Compilacion.

Ayuda:

**Sintácticos** 4, **Semánticos** 3, **Lógico** 1

### c) C

```
# include <stdio.h>

int suma; /* Esta es una variable global */

int main()
{ int indice;

  encabezado;

  for (indice = 1 ; indice <= 7 ; indice ++ )
    cuadrado (indice);

  final(); Llama a la función final */

  return 0;
}

cuadrado (numero)

int numero;

{ int numero_cuadrado;

  numero_cuadrado == numero * numero;

  suma += numero_cuadrado;

  printf("El cuadrado de %d es %d\n",
    numero, numero_cuadrado);
}
```

Sintáctico:

- \_ encabezado faltan () si es una función declarada.
- \_ falta el /\* en el comentario de la invocación de final() sería /\*Llama a la función final \*/
- \_ las llaves de cuadrado(numero) se encuentran mal colocadas.

Semántico:

- \_ encabezado no esta declarada. Se detecta en compilación
- \_ cuadrado(indice) no esta declarada puesto que tiene mal las llaves. En compilación.
- \_ la función final() no esta declarada. Compilación
- \_ numero\_cuadrado y numero no estan inicializados
- \_ suma no esta inicializada. Compilación.
- \_ cuadrado(numero) no esta en ningun bloque porque se podria tratar de una invocación pero no es el caso.

Ayuda: Sintácticos 2, Semánticos 6

### d) Python

```
#!/usr/bin/python
print "\nDEFINICION DE NUMEROS PRIMOS"
r = 1
while r = True:
    N = input("\nDame el numero a analizar: ")
    i = 3
    fact = 0
    if (N mod 2 == 0) and (N != 2):
        print "\nEl numero %d NO es primo\n" % N
    else:
        while i <= (N^0.5):
            if (N % i) == 0:
                mensaje="\nEl numero ingresado NO es primo\n" % N
                msg = mensaje[4:6]
                print msg
                fact = 1
            i+=2
        if fact == 0:
            print "\nEl numero %d SI es primo\n" % N
```

Sintáctico:

- \_ Faltan los paréntesis en los print.
- \_ la condición del primer while debería ser ==
- \_ el mod en py es %
- \_ en while i<= (N^0.5) la potencia es con \*\*

Semántico:

- \_ while r = true se esta comparando un entero con un boolean. Compilación.
- \_ N = input(...) se debería parsear lo ingresado a int con (int) porque posteriormente N se usa como entero. Se detecta en compilación
- \_ falta el %d en , "\nEl numero ingresado NO es primo\n" % N. Se detecta en ejecución.

```
r = input("Consultar otro número? SI (1) o NO (0)---> ")
```

Ayuda: Sintácticos 2, Semánticos 3

e) Ruby

def ej1

  Puts 'Hola, ¿Cuál es tu nombre?'

  nom = gets.chomp

  puts 'Mi nombre es ', + nom

  puts 'Mi sobrenombre es 'Juan"

  puts 'Tengo 10 años'

  meses = edad\*12

  dias = 'meses' \*30

  hs= 'dias \* 24'

  puts 'Eso es: meses + ' meses o ' + dias + ' días o ' + hs + ' horas'

  puts 'vos cuántos años tenés'

  edad2 = gets.chomp

  edad = edad + edad2.to\_i

  puts 'entre ambos tenemos ' + edad + ' años'

  puts '¿Sabes que hay ' + name.length.to\_s + ' caracteres en tu nombre, ' + name + ' ?'

end

Sintáctico:

\_la mayoría de los puts deberían ser Puts con mayúscula.

\_puts 'Mi sobrenombre es 'Juan" falta cerrar el ".

Semántico:

\_ edad no esta definida . Compilación.

\_ "meses" \* 30 no se puede combinar un string con un int.

\_ puts 'Eso es: meses + ' meses o ' + dias + ' días o ' + hs + ' horas'

No existen las variables meses y dias. Compilación.

\_ la variable name no esta definida. Compilación.

Ayuda: **Semánticos** +4

**Ejercicio 5:** Dado el siguiente código escrito en pascal. Transcriba la misma funcionalidad de acuerdo al lenguaje que haya cursado en años anteriores. Defina brevemente la sintaxis (sin hacer la gramática) y semántica para la utilización de arreglos y estructuras de control del ejemplo.

Procedure ordenar\_arreglo(var arreglo: arreglo\_de\_caracteres; cont: integer);

var

  i: integer; ordenado: boolean;

  aux: char;

begin

  repeat

    ordenado:=true;

    for i:=1 to cont-1 do

      if ord(arreglo[i])>ord(arreglo[i+1])

        then begin

          aux:=arreglo[i];

          arreglo[i]:=arreglo[i+1];

          arreglo[i+1]:=aux; ordenado:=false

        end;

    until ordenado;

end;

*Observación: Aquí sólo se debe definir la instrucción y qué es lo que hace cada una; detallando*

*alguna particularidad del lenguaje respecto de ella. Por ejemplo el for de java necesita definir una variable entera, una condición y un incremento para dicha variable.*

**Ejercicio 6:** Explique cuál es la semántica para las variables predefinidas en lenguaje Ruby **self** y **nil**. ¿Qué valor toman; cómo son usadas por el lenguaje?

**Ejercicio 7:** Determine la semántica de null y undefined para valores en javascript. ¿Qué diferencia hay entre ellos?

**Ejercicio 8:** Determine la semántica de la sentencia break en C, PHP, javascript y Ruby. Cite las características más importantes de esta sentencia para cada lenguaje

**Ejercicio 9:**

Defina el concepto de ligadura y su importancia respecto de la semántica de un programa. ¿Qué diferencias hay entre ligadura estática y dinámica? Cite ejemplos (proponer casos sencillos)