

**Práctica Nro. 3**

**Semántica**

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

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

**Ejercicio 2:**

- a. ¿Qué significa compilar un programa?
- b. Describa brevemente cada uno de los pasos necesarios para compilar un programa.
- c. ¿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

var 5: integer;

var a:char;

Begin

    for i:=5 to 10 do begin

        write(a);

        a=a+1;

    end;

End.

*Ayuda: Sintáctico 2, Semántico 3*

**b) Java:**

```
public String tabla(int numero, arrayList<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;
```

```
}
```

*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);
}
```

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

    r = input("Consultar otro numero? 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
  días = 'meses' *30
  hs= 'días * 24'
  puts 'Eso es: meses + ' meses o ' + días + ' 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
```

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)