

Lenguajes de programación

Hugo de Jesús Valenzuela Chaparro

19 de febrero de 2015

1. Compiladores e interpretadores

La función de los **interpretadores** es de interpretar, literalmente, es decir, ejecutan un programa directamente del código sin necesidad de hacer un archivo ejecutable. Por otro lado, los **compiladores** lo que hacen es traducir el código a el lenguaje máquina, creando un ejecutable el cual la máquina ejecutará. Por esa razón cuando se modifica el código de un programa y no se compila, el ejecutable sigue con la misma función, por su parte, cuando se modifica el de código de un interpretador, cambia la función a lo que se modificó.

2. Lenguajes de programación

Nombre	Paradigma	Creadores	Año de aparición	Ext. de archivo
ANSI C	Imperativo	Dennis Ritchie	1972	.c
C++	Orientado a objetos	Bjarne Stroustrup	1983	.cpp
Fortran 90	Orientado a objetos	John Backus	1990	.f90
Java	Orientado a objetos	James Gosling y Sun Microsystems	1995	.java
Python	Orientado a objetos	Guido van Rossum	1991	.py
Ruby	Orientado a objetos	Yukihiro Matsumoto	1995	.rb

3. Ejemplos de códigos

3.1. ANSI C

```
#include <stdio.h>

int main()
{
printf("Hola! Tratare de adivinar un numero.
Piensa en un numero entre 1 y 10\n");
sleep(5);
printf("Ahora multiplicallo por 9\n");
sleep(5);
printf("Si el numero tiene 2 digitos, sumalos entre si.
```

```

Si tu numero tiene un solo digito, sumale 0\n");
sleep(5);
printf("Al numero restante sumale 4\n");
sleep(10);
printf("Muy bien. El resultado es 13 ;D.
Siguele rockeando con todo, animo.\n");
return 0;
}

```

3.2. C++

```

#include <iostream>
#include <unistd.h>
using namespace std;

int main()
{
cout << "Hola! Tratare de adivinar un numero.
Piensa en un numero entre 1 y 10" << endl;
sleep(5);
cout << "Ahora multiplicallo por 9" << endl;
sleep(5);
cout << "Si el numero tiene 2 digitos, sumalos entre si.
Si tu numero tiene un solo digito, sumale 0" << endl;
sleep(5);
cout << "Al numero restante sumale 4" << endl;
sleep(10);
cout << "Muy bien. El resultado es 13 ;D.
Siguele rockeando con todo, animo." << endl;
return 0;
}

```

3.3. Fortran 90

```

PROGRAM AdivinaLaMente

PRINT*, ";Hola! Tratare de adivinar un numero. &
&Piensa en un numero entre 1 y 10"

```

```

CALL Sleep (5)
PRINT*, "Ahora multiplicalo por 9"
CALL Sleep (5)
PRINT*, "Si el numero tiene 2 digitos, sumalos entre si. Si tu numero tiene&
& un solo digito, sumale 0"
CALL Sleep (5)
PRINT*, "Al numero restante sumale 4"
CALL Sleep (10)
PRINT*, "Muy bien. El resultado es 13 ;D. Siguele rockeando con todo, animo."

ENDPROGRAM AdivinaLaMente

```

3.4. Java

```

public class AdivinaLaMente{

    public static void main(String []args){
    try {
System.out.println("Hola! Tratare de adivinar un numero.
Piensa en un numero entre 1 y 10");
Thread.sleep(5000);
System.out.println("Ahora multiplicalo por 9");
Thread.sleep(5000);
System.out.println("Si el numero tiene 2 digitos, sumalos entre si.
Si tu numero tiene un solo digito, sumale 0");
Thread.sleep(5000);
System.out.println("Al numero restante sumale 4")
Thread.sleep(10000)
System.out.println("Muy bien. El resultado es 13 ;D.
Siguele rockeando con todo, animo.");

    }
    catch(Exception e) {
    }
}

```

```
}
```

3.5. Python

```
#!/usr/bin/python
import time

print "Hola! Tratare de adivinar un numero. Piensa en un numero entre 1 y 10"
time.sleep( 5 )
print "Ahora multiplicalo por 9"
time.sleep( 5 )
print "Si el numero tiene 2 digitos, sumalos entre si.
Si tu numero tiene un solo digito, sumale 0"
time.sleep( 5 )
print "Al numero restante sumale 4"
time.sleep( 10 )
print "Muy bien. El resultado es 13 ;D. Siguele rockeando con todo, animo."
```

3.6. Ruby

```
#Adivina la mente en ruby

puts "Hola! Tratare de adivinar un numero. Piensa en un numero entre 1 y 10!"
sleep(5)
puts "Ahora multiplicalo por 9"
sleep(5)
puts "Si el numero tiene 2 digitos, sumalos entre si.
Si tu numero tiene un solo digito, sumale 0"
sleep(5)
puts "Al numero restante sumale 4"
sleep(10)
puts "Muy bien. El resultado es 13 ;D. Siguele rockeando con todo, animo."
```