

Compiladores e Interpretadores

1. Introducción

- **Compilador** que analiza el programa fuente y "lo traduce.^a otro equivalente escrito en otro lenguaje (por ejemplo, en el lenguaje de la máquina). Su acción equivale a la de un traductor humano, que toma un libro y produce otro equivalente escrito en otra lengua.
- **Intérprete** que analiza el programa fuente y "lo ejecuta directamente", sin generar ningún código equivalente. Su acción equivale a la de un intérprete humano, que traduce las frases que oye sobre la marcha, sin producir ningún escrito permanente. Intérpretes y compiladores tienen diversas ventajas e inconvenientes que los hacen complementarios:
- Un intérprete facilita la búsqueda de errores, pues la ejecución de un programa puede interrumpirse en cualquier momento para estudiar el entorno (valores de las variables, etc.). Además, el programa puede modificarse sobre la marcha, sin necesidad de volver a comenzar la ejecución.
- Un compilador suele generar programas más rápidos y eficientes, ya que el análisis del lenguaje fuente se hace una sola vez, durante la generación del programa equivalente. En cambio, un intérprete se ve obligado generalmente a analizar cada instrucción tantas veces como se ejecute (incluso miles o millones de veces).

2. Comparación entre lenguajes de programación

Nombre	paradigma	creadores	año	extensiones de archivo	ejemplo
ANSI C	Imperativo (procedural), estructurado.	Dennis M. Ritchie	1972	.h .c	gcc
C++	Imperativo (procedural), estructurado.	Bjarne Stroustrup	1983	. .h, .hh, .hpp, .hxx, .h++, .cc, .cpp, .cxx, .c++	g++
Fortran90	Multiparadigma orientado a objetos, imperativo, programación genérica.	John Backus	1957	.f, .for, .f90, .f95	gfortran
Java	Multiparadigma estructurado, imperativo (procedural, orientado a objetos), genérico.	Sun Microsystems (Oracle corporation)	1995	.java, .class, .jar, .jad	javac
Python	Multiparadigma Orientado a objetos, imperativo, funcional, reflexivo.	Guido van Rossum	1991	.pyd, .pyo, .pyw, .py, .pyc	python
Ruby	orientado a objetos, imperativo, orientado a objetos, reflexivo.	Yukihiro Matsumoto	1995	.rb, .rbw	ruby

3. Juego: Adivinar el número

- Juego ANSI C

```
/* Juego in C, Ansi-style */
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
int main(void)
{
puts("Hola! Trataré de adivinar un número."
"Piensa un número entre 1 y 10.");
sleep(5);
puts("Ahora multiplícalo por 9.");
sleep(5);
puts("Si el número tiene 2 dígitos, súmalos
entre si:Ej. 36 -> 3+6=9. Si tu número tiene
un solo dígito, súmale 0.");
sleep(5);
puts("Al número resultante súmale 4.");
sleep(10);
puts("Muy bien. El resultado es 13 c:");
return(EXIT_SUCCESS);
}
```

- Juego C++

```
#include <iostream>
#include <unistd.h>
int main()
{
std::cout << "Hola! Trataré de adivinar un número.
Piensa en un número entre 1 y 10\n";
sleep(5);
std::cout << "Ahora multiplícalo por 9.\n";
sleep(5);
std::cout << "Si el número tiene 2 dígitos,
súmalos entre si: Ej. 36 -> 3+6=9. Si tu número
```

```

tiene un solo dígito, súmale 0.\n";
sleep(5);
std::cout << "Al número resultante súmale 4.\n";
sleep(10);
std::cout << "Muy bien. El resultado es 13 c:\n";
return(0);
}

```

■ Juego Fortran

```

program hello
write(*,*) 'Hola! Trataré de adivinar un número.
Piensa un número entre 1 y 10.'
call sleep(5)
write(*,*) 'Ahora multiplícalo por 9.'
call sleep(5)
write(*,*) 'Si el número tiene 2 dígitos,
súmalos entre si: Ej. 36 -> 3+6=9. Si tu
número tiene un solo dígito, súmale 0.'
call sleep(5)
write(*,*) 'Al número resultante súmale 4.'
call sleep(10)
write(*,*) 'Muy bien. El resultado es 13 c:'
end program hello

```

■ Juego Java

```

class Adivina {
static public void main( String args[] ) {
System.out.println("!Hola! Trataré de adivinar un número." );
System.out.println("Piensa un número entre 1 y 10.");
try {
Thread.sleep(5000);
} catch(InterruptedException ex) {
Thread.currentThread().interrupt();
}
System.out.println("Ahora multiplícalo por 9.");
}
}

```

```

try {
Thread.sleep(5000);
} catch(InterruptedException ex) {
Thread.currentThread().interrupt();
}
System.out.println("Si el número tiene 2 dígitos, súmalos
entre si: Ej. 36 -> 3+6=9. Si tu número tiene un solo dígito,
súmale 0.");
try {
Thread.sleep(5000);
} catch(InterruptedException ex) {
Thread.currentThread().interrupt();
}
System.out.println("Al número resultante súmale 4.");
try {
Thread.sleep(5000);
} catch(InterruptedException ex) {
Thread.currentThread().interrupt();
}
System.out.println("Muy bien. El resultado es 13 :)");
try {
Thread.sleep(5000);
} catch(InterruptedException ex) {
Thread.currentThread().interrupt();
}
}
}
}

```

■ Juego Python

```

#!/usr/bin/python
# -*- coding: latin-1 -*-
import time
print "Hola! Trataré de adivinar un número."
print "Piensa un número entre 1 y 10."
time.sleep(5)
print "Ahora multiplicallo por 9."

```

```

time.sleep(5)
print "Si el número tiene 2 dígitos, súmalos entre si:
Ej. 36 -> 3+6=9. Si tu número tiene un solo dígito,
súmalo 0."
time.sleep(5)
print "Al número resultante súmalo 4."
time.sleep(10)
print "Muy bien. El resultado es 13 c:"

```

■ Juego Ruby

```

# -*- coding: utf-8 -*-

#!/usr/bin/ruby
# -*- coding: latin-1 -*-
puts "Hola! Trataré de adivinar un número.
Piensa un número entre 1 y 10."
sleep(5)
puts "Ahora multiplicallo por 9."
sleep(5)
puts "Si el número tiene 2 dígitos, súmalos
entre si: Ej. 36 -> 3+6=9. Si tu número tiene
un solo dígito, sumalo 0."
sleep(5)
puts "Al número resultante súmalo 4."
sleep(10)
puts "Muy bien. El resultado es 13 c: "

```