

Reporte de la Actividad 2

Hedwin Aaron Encinas Acosta

10 de Febrero de 2015

1. Introduccion

En el siguiente reporte se muestran ejemplos de lenguajes interpretadores y compiladores utilizados en programación, poniendo como ejemplo un mismo programa, pero en 6 diferentes lenguajes. Unos de ellos son lenguajes interpretadores y otros compiladores. Pero de que hablamos al decir lenguajes compiladores e interpretadores :

un lenguaje compilador es un lenguaje de programación cuyas implementaciones son normalmente compiladas (traductores que generan código de máquina a partir de código fuente), es decir, una vez escrito el programa, éste se traduce a partir de su código fuente por medio de un compilador en un archivo ejecutable para una determinada plataforma, y no los intérpretes (ejecutores paso a paso de código fuente, donde no se lleva a cabo una traducción de pre-ejecución.)

Mientras que un interpretador es el lenguaje cuyo código no necesita ser preprocesado mediante un compilador, eso significa que el ordenador es capaz de ejecutar la sucesión de instrucciones dadas por el programador sin necesidad de leer y traducir exhaustivamente todo el código.

Los compiladores e interpretadores que utilizaremos son:

- ANCI C
- C++
- Fortran 90
- Java

- Python
- Ruby

2. Tabla Comparativa

Nombre	Paradigma	Creadores	Año	Extensiones	Comando para Compilación/Interpretación
ANSI C	Imperativo, estructurado	Denis M. Ritchie	1972	.h .c	<code>g++ -o main *.cpp</code>
C++	Multiparadigma, orientado a objetos, imperativo	Bjarne Stroustrup	1983	.h .hh .hpp .hxx .h++ .cc .cpp .cxx .c++	<code>g++ -o main *.cpp</code>
Fortran 95	Estructurado, imperativo, genérico	John Backus	1957	.f .for .f90 .f95	<code>gfortran -std=f95 *.f -o main</code>
Java	Orientado a objetos, imperativo	Sun Microsystems	1995	.java .class .jar .jad	<code>javac java.jar</code>
Python	Orientado a objetos, imperativo, funcional, reflexivo	Guido van Rossum	1991	.py .pyc .pyd .pyo .pyw	<code>python main.py</code>
Ruby	Orientado a objetos, reflexivo	Yukihiro Matsumoto	1995	.rb .rbw	<code>ruby main.rb</code>

3. Programa “adivina la mente” en diferentes lenguajes de programación

3.1. Fortran 90

```

program juegofortran.f90
  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 :3'
end program juegofortran.f90

```

Python

```

import time
print "Hola! Tratare de adivinar un numero.
      Piensa un numero entre 1 y 10."
import time
time.sleep(5)
print "Ahora multiplicalo por 9."
import time
time.sleep(5)
print "Si el numero tiene 2 digitos, sumalos entre si: Ej. 36 -> 3+6=9.
Si tu numero tiene un solo digito, sumale 0."
import time
time.sleep(5)
print "Al numero resultante sumale 4."
import time
time.sleep(10)
print "Muy bien. El resultado es 13 :) "

```

3.2. Ruby

```

puts "Hola! Tratare de adivinar un numero.
      Piensa 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: Ej. 36 -> 3+6=9.
Si tu numero tiene un solo digito, sumale 0."
sleep(5)
puts "Al numero resultante sumale 4."

```

```
sleep(10)
puts "Muy bien. El resultado es 13 :)"
```

3.3. C

```
#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.Hello World!");
    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 :3");
    return(EXIT_SUCCESS);
}
```

3.4. 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 :3\n";
return(0);
}

```

3.5. Java

```

class juego {
    static public void main( String args[] ) {
        System.out.println( "Hola! Trataré de adivinar un número.
        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(10000);  
} catch (InterruptedException ex) {  
    Thread.currentThread().interrupt();  
}  
System.out.println( "Muy bien. El resultado es 13 :3 " );  
}
```

```
} \end{flushleft}
```