

Reporte de Actividad 2

Jorge Benz Olguin Aguilar
Departamento de Física
Universidad de Sonora

14 de septiembre de 2015

1. Introducción

Un lenguaje de programación es un lenguaje formal diseñado para expresar procesos que pueden ser llevados a cabo por máquinas como las computadoras.

Pueden usarse para crear programas que controlen el comportamiento físico y lógico de una máquina, para expresar algoritmos con precisión, o como modo de comunicación humana.¹

Está formado por un conjunto de símbolos y reglas sintácticas y semánticas que definen su estructura y el significado de sus elementos y expresiones. Al proceso por el cual se escribe, se prueba, se depura, se compila (de ser necesario) y se mantiene el código fuente de un programa informático se le llama programación.

La implementación de un lenguaje es la que provee una manera de que se ejecute un programa para una determinada combinación de software y hardware. Existen básicamente dos maneras de implementar un lenguaje: compilación e interpretación.

Compilación: es el proceso que traduce un programa escrito en un lenguaje de programación a otro lenguaje de programación, generando un programa equivalente que la máquina será capaz interpretar. Los programas traductores que pueden realizar esta operación se llaman compiladores. Éstos, como los programas ensambladores avanzados, pueden generar muchas líneas de código de máquina por cada proposición del programa fuente.

Interpretación: es una asignación de significados a las fórmulas bien formadas de un lenguaje formal. Como los lenguajes formales pueden definirse en términos puramente sintácticos, sus fórmulas bien formadas pueden no ser más que cadenas de símbolos sin ningún significado. Una interpretación otorga significado a esas fórmulas.

2. Tabla comparativa

	Fortran	C	C++	Java
Paradigma	Lenguaje de programación de alto nivel de propósito general, procedimental e imperativo	Imperativo, procedural y estructurado	Multiparadigma: orientado a objetos, imperativo, programación generica	Orientado a objetos, imperativo
Creadores	John W. Backus	Dennis Ritchie	Bjarne Stroustrup	James Gosling y Sun Microsystems
Año Aparición	Octubre de 1956	1972	1983	1995
Extensión	.f .f90 .f95 .for	.c .h	.h .hh .hpp .hxx .h++ .cc .cpp .cxx .c++	.java .class .jar
Ejemplo de Compiladores e Interpretadores	GNU /usr/bin/gfortran	GNU /usr/bin/gcc	GNU /usr/bin/g++	/usr/bin/javac

	Python	Ruby
Paradigma	Multiparadigma: orientado a objetos, imperativo y funcional	Imperativo, procedural y estructurado
Creadores	Guido van Rossum	Yukihiro Matsumoto
Año Aparición	1991	1995
Extensión	.py, .pyc, .pyd, .pyo, pyw, .pyz	.rb, .rbw
Ejemplo de compiladores e interpretadores	/usr/bin/python	/usr/bin/ruby

2.1. Lenguaje de programación científica

2.1.1. C

El código empleado es:

```
#include <stdio.h>

int main(void)
```

```
{
    printf("Hello, world!\n");
}
```

2.1.2. c++

El código empleado es:

```
#include <iostream>

int main()
{
    std::cout << "Hello, world!\n";
}
```

2.1.3. Fortran

El código empleado es:

```
program hello
    write(*,*) 'Hello, world!'
end program hello
```

2.1.4. Java

El código empleado es:

```
public class HelloWorld {
    public static void main(String[] args) {
        System.out.println("Hello, world!");
    }
}
```

2.1.5. Python

El código empleado es:

```
print "Hello, world!"
```

2.1.6. Ruby

El código empleado es:

```
puts "Hello, world!"
```

3. Bibliografia

- 1.- https://en.wikipedia.org/wiki/C_%28programming_language%29
- 2.- <https://en.wikipedia.org/wiki/C%2B%2B>
- 3.- <https://en.wikipedia.org/wiki/Fortran>
- 4.- https://en.wikipedia.org/wiki/Java_%28programming_language%29
- 5.- https://en.wikipedia.org/wiki/Python_%28programming_language%29
- 6.- https://en.wikipedia.org/wiki/Ruby_%28programming_language%29