

# Reporte de Actividad 2

Francisco Javier Real Santoscoy  
Departamento de Física  
Universidad de Sonora

2 de Septiembre del 2015

## 1. Introducción

Los lenguajes de computadora se utilizan para resolver problemas, y ha habido miles de años de trabajo en matemáticas para este fin. Los lenguajes de programación están especificados por reglas para formar instrucciones correctas, someterlas hacia un compilador, el cual traduce el código en un lenguaje comprensible para una máquina en particular, y finalmente ejecutar el programa, es decir, someter la entrada hacia la computadora, la cual transforma en una salida de acuerdo con las instrucciones en el programa.

También la palabra programación se define como el proceso de creación de un programa de computadora, mediante la aplicación de procedimientos lógicos, a través de los siguientes pasos:

El desarrollo lógico del programa para resolver un problema en particular. Escritura de la lógica del programa empleando un lenguaje de programación específico (codificación del programa). Ensamblaje o compilación del programa hasta convertirlo en lenguaje de máquina. Prueba y depuración del programa. Desarrollo de la documentación. Existe un error común que trata por sinónimos los términos 'lenguaje de programación' y 'lenguaje informático'. Los lenguajes informáticos engloban a los lenguajes de programación y a otros más, como por ejemplo HTML (lenguaje para el marcado de páginas web que no es propiamente un lenguaje de programación, sino un conjunto de instrucciones que permiten estructurar el contenido de los documentos).

Permite especificar de manera precisa sobre qué datos debe operar una computadora, cómo deben ser almacenados o transmitidos y qué acciones debe tomar bajo una variada gama de circunstancias. Todo esto, a través de un lenguaje que intenta estar relativamente próximo al lenguaje humano o natural. Una característica relevante de los lenguajes de programación es precisamente que más de un programador pueda usar un conjunto común de instrucciones que sean comprendidas entre ellos para realizar la construcción de un programa de forma colaborativa.

## 2. Antecedentes

### 2.1. Lenguajes de Programación Científica

Para que la computadora entienda nuestras instrucciones debe usarse un lenguaje específico conocido como código máquina, el cual la máquina comprende fácilmente, pero que lo hace excesivamente complicado para las personas. De hecho sólo consiste en cadenas extensas de números 0 y 1.

Para facilitar el trabajo, los primeros operadores de computadoras decidieron hacer un traductor para reemplazar los 0 y 1 por palabras o abstracción de palabras y letras provenientes del inglés; éste se conoce como lenguaje ensamblador. Por ejemplo, para sumar se usa la letra A de la palabra inglesa add (sumar). El lenguaje ensamblador sigue la misma estructura del lenguaje máquina, pero las letras y palabras son más fáciles de recordar y entender que los números.

La necesidad de recordar secuencias de programación para las acciones usuales llevó a denominarlas con nombres fáciles de memorizar y asociar: ADD (sumar), SUB (restar), MUL (multiplicar), CALL (ejecutar subrutina), etc. A esta secuencia de posiciones se le denominó "instrucciones", y a este conjunto de instrucciones se le llamó lenguaje ensamblador. Posteriormente aparecieron diferentes lenguajes de programación, los cuales reciben su denominación porque tienen una estructura sintáctica semejante a la de los lenguajes escritos por los humanos, denominados también lenguajes de alto nivel.

El primer programador de computadora que se haya conocido fue una mujer: Ada Lovelace, hija de Anabella Milbanke Byron y Lord Byron. Anabella inició en las matemáticas a Ada quien, después de conocer a Charles Babbage, tradujo y amplió una descripción de su máquina analítica. Incluso aunque Babbage nunca completó la construcción de cualquiera de sus máquinas, el trabajo que Ada realizó con éstas le hizo ganarse el título de primera programadora de computadoras del mundo. El nombre del lenguaje de programación Ada fue escogido como homenaje a esta programadora.

A finales de 1953, John Backus sometió una propuesta a sus superiores en IBM para desarrollar una alternativa más práctica al lenguaje ensamblador para programar la computadora central IBM 704. El histórico equipo Fortran de Backus consistió en los programadores Richard Goldberg, Sheldon F. Best, Harlan Herrick, Peter Sheridan, Roy Nutt, Robert Nelson, Irving Ziller, Lois Haibt y David Sayre.<sup>2</sup>

El primer manual para el lenguaje Fortran apareció en octubre de 1956, con el primer compilador Fortran entregado en abril de 1957. Esto era un compilador optimizado, porque los clientes eran reacios a usar un lenguaje de alto nivel a menos que su compilador pudiera generar código cuyo desempeño fuera comparable al de un código hecho a mano en lenguaje ensamblador.

En 1960, se creó COBOL, uno de los lenguajes usados aún en la actualidad, en informática de gestión.

A medida que la complejidad de las tareas que realizaban las computadoras aumentaba, se hizo necesario disponer de un método más eficiente para programarlas. Entonces, se crearon los lenguajes de alto nivel, como lo fue BASIC en

las versiones introducidas en los microordenadores de la década de 1980. Mientras que una tarea tan sencilla como sumar dos números puede necesitar varias instrucciones en lenguaje ensamblador, en un lenguaje de alto nivel bastará una sola sentencia.

### 2.1.1. C

Uno de los objetivos de diseño del lenguaje C es que sólo sean necesarias unas pocas instrucciones en lenguaje máquina para traducir cada elemento del lenguaje, sin que haga falta un soporte intenso en tiempo de ejecución. Es muy posible escribir C a bajo nivel de abstracción; de hecho, C se usó como intermediario entre diferentes lenguajes.

En parte a causa de ser de relativamente bajo nivel y de tener un modesto conjunto de características, se pueden desarrollar compiladores de C fácilmente. En consecuencia, el lenguaje C está disponible en un amplio abanico de plataformas (más que cualquier otro lenguaje). Además, a pesar de su naturaleza de bajo nivel, el lenguaje se desarrolló para incentivar la programación independiente de la máquina. Un programa escrito cumpliendo los estándares e intentando que sea portátil puede compilarse en muchos computadores.

C se desarrolló originalmente (conjuntamente con el sistema operativo Unix, con el que ha estado asociado mucho tiempo) por programadores para programadores. Sin embargo, ha alcanzado una popularidad enorme, y se ha usado en contextos muy alejados de la programación de software de sistema, para la que se diseñó originalmente.

### 2.1.2. C++

La intención de su creación fue el extender al lenguaje de programación C mecanismos que permiten la manipulación de objetos. En ese sentido, desde el punto de vista de los lenguajes orientados a objetos, el C++ es un lenguaje híbrido.

Posteriormente se añadieron facilidades de programación genérica, que se sumaron a los paradigmas de programación estructurada y programación orientada a objetos. Por esto se suele decir que el C++ es un lenguaje de programación multiparadigma.

Actualmente existe un estándar, denominado ISO C++, al que se han adherido la mayoría de los fabricantes de compiladores más modernos. Existen también algunos intérpretes, tales como ROOT.

Una particularidad del C++ es la posibilidad de redefinir los operadores, y de poder crear nuevos tipos que se comporten como tipos fundamentales.

El nombre C++ fue propuesto por Rick Mascitti en el año 1983, cuando el lenguaje fue utilizado por primera vez fuera de un laboratorio científico. Antes se había usado el nombre C con clases”. En C++, la expresión C++ significa incremento de C, se refiere a que C++ es una extensión de C.

### 2.1.3. Fortran

A finales de 1953, John W. Backus sometió una propuesta a sus superiores en IBM para desarrollar una alternativa más práctica al lenguaje ensamblador para programar el computador central IBM 704. El histórico equipo FORTRAN de Backus consistió en los programadores Richard Goldberg, Sheldon F. Best, Harlan Herrick, Peter Sheridan, Roy Nutt, Robert Nelson, Irving Ziller, Lois Haibt y David Sayre.<sup>5</sup>

A mediados de 1954 fue terminada una especificación del borrador para el IBM Mathematical Formula Translating System. El primer manual de FORTRAN apareció en octubre de 1956, porque los clientes eran reacios a usar un lenguaje de programación de alto nivel a menos que su compilador pudiera generar código cuyo desempeño fuera comparable al de un código hecho a mano en lenguaje ensamblador.

Mientras que la comunidad era escéptica en que este nuevo a mano, éste redujo por un factor de 20 al número de sentencias de programación necesarias para operar una máquina, y rápidamente ganó aceptación. Durante una entrevista en 1979 con Think, la revista de los empleados de IBM, el creador, John Backus, dijo, "Mucho de mi trabajo ha venido de ser perezoso. No me gustaba escribir programas, y por eso, cuando estaba trabajando en el IBM 701 escribiendo programas para computar trayectorias de misiles, comencé el trabajo sobre un sistema de programación para hacer más fácil escribir programas".<sup>6</sup>

El lenguaje fue ampliamente adoptado por los científicos para escribir programas numéricamente intensivos, que incentivó a los escritores de compiladores a producir compiladores que pudieran generar un código más rápido y más eficiente. La inclusión en el lenguaje de un tipo de datos y de la aritmética de números complejos amplió la gama de aplicaciones para las cuales el lenguaje se adaptaba especialmente e hizo al FORTRAN especialmente adecuado para aplicaciones técnicas tales como la ingeniería eléctrica.

Hacia 1960, las versiones de FORTRAN estaban disponibles para los computadores IBM 709, 650, 1620, y 7090. Significativamente, la cada vez mayor popularidad del FORTRAN estimuló a fabricantes de computadores de la competencia a proporcionar compiladores FORTRAN para sus máquinas, así que por 1963 existían más de 40 compiladores FORTRAN. Por estas razones, el FORTRAN es considerado ser el primer lenguaje de programación ampliamente usado soportado a través de una variedad de arquitecturas de computador.

El desarrollo del FORTRAN fue paralelo a la temprana evolución de la tecnología del compilador. De hecho, muchos avances en la teoría y el diseño de compiladores fueron motivados específicamente por la necesidad de generar código eficiente para los programas en FORTRAN.

## 3. Resultados: Hola Mundo

### 3.1. c

El código empleado es:

```
#include <stdio.h>

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

### 3.2. c++

El código empleado es:

```
program hello
    write(*,*) 'Hello, world! in Fortran 90'
end program hello
```

### 3.3. Fortran

El código empleado es:

```
program hello
    write(*,*) 'Hello, world! in Fortran 90'
end program hello
```

### 3.4. Java

El código empleado es:

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

### 3.5. Python

El código empleado es:

```
print "Hello, world! in Python"
```

### 3.6. Ruby

El código empleado es:

```
puts "Hello, world!in Ruby"
```

## 4. Table Comparativa de los Lenguajes

### 4.1. Lenguajes Compiladores

	Fortran	C	C++	Java
Paradigma	Lenguaje de programación de alto nivel	Procedimental e Imperativo	Imperativo, Procedural y Estructurado	Imperativo, Programación Generica
Creadores	John W. Backus	Dennis Ritchie	Bjarne Stroustrup	James Gosling y Sun Microsystems
Año de aparición	1956	1972	1983	1995
Extensiones de archivo	.f .f90 .f95 .for	.c .h	.h .hh .hpp .hxx .h++ .cc .cpp .cxx .c++	.java .class .jar
Ejemplo de compilación/interpretación	GNU /usr/bin/gfortran	GNU /usr/bin/gcc	GNU /usr/bin/g++	/usr/bin/javac

### 4.2. Lenguajes Interpretadores

	Python	Ruby
Paradigma	Imperativo y Funcional	Procedural y Estructurado
Creadores	Guido van Rossum	Yukihiro Matsumoto
Año de Aparición	1991	1995
Extensiones de Archivo	.py, .pyc, .pyd, .pyo, pyw, .pyz	.rb, .rbw
Ejemplo de compilación/interpretación	/usr/bin/python	/usr/bin/ruby

## 5. Conclusión

Los lenguajes y las Técnicas de Programación son importantes para la utilización de la computadora como una herramienta para resolver problemas.

En computación, un Problema consiste en la necesidad de transformar un grupo de datos iniciales en un grupo diferente de datos finales (resultados).

De este modo, una computadora podrá resolver un problema si alguien

desarrolla un programa que contenga las instrucciones adecuadas que permitan transformar los datos.

Los lenguajes deben ser confiables, comprensibles, eficientes en términos de tiempo de ejecución y consumo de espacio, y deben satisfacer las necesidades de una comunidad, ya sean científicos, hombres de negocios o usuarios no técnicos. Cada uno de estos grupos está acostumbrado a un vocabulario particular y una manera de ver las cosas; de este modo, existe una gran variedad de lenguajes y muy probablemente esto continuará siendo así.

## 6. Bibliografía

[https://es.wikipedia.org/wiki/Lenguaje de programación](https://es.wikipedia.org/wiki/Lenguaje_de_programaci3n)  
[https://es.wikipedia.org/wiki/C \(lenguaje de programación\)](https://es.wikipedia.org/wiki/C_(lenguaje_de_programaci3n))  
<https://es.wikipedia.org/wiki/C>  
<https://es.wikipedia.org/wiki/Fortran>