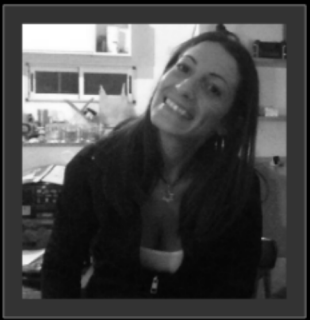
Entendiendo los lenguajes de programación

Publicado originalmente el 29 de Febrero de 2012



Eugenia Bahit

GLAMP Hacker y Programadora eXtrema, avocada a la docencia en el ámbito de la Ingeniería de Software y a la investigación de nuevas técnicas, metodologías y paradigmas relacionadas con la seguridad de aplicaciones. Miembro de la Free Software Foundation y The Linux Foundation.

Los lenguajes de programación, forman parte del grupo de lenguajes informáticos. Ampliamente, puede decirse que un lenguaje informático es un idioma artificial, utilizado por ordenadores, cuyo fin es transmitir información de algo a alguien.

Los **lenguajes informáticos**, pueden clasificarse en:

- lenguajes de programación (Python, PHP, Perl, C, etc.);
- lenguajes de especificación (UML);
- lenguajes de consulta (SQL);
- lenguajes de marcas (HTML, XML);
- lenguajes de transformación (XSLT);
- protocolos de comunicaciones (HTTP, FTP); entre otros.

> “un lenguaje informático es un idioma artificial utilizado por ordenadores

Mientras que algunos lenguajes informáticos como (X)HTML o CSS, han sido

diseñados para diagramar y decidir la forma en la cual la información será presentada al usuario, los lenguajes de programación, tienen como fin, **expresar órdenes e instrucciones precisas, que deben ser llevadas a cabo por una computadora para realizar una o más tareas específicas**. Se utilizan para crear programas que controlan el comportamiento físico o lógico de un ordenador. Están compuestos por una serie de símbolos, reglas sintácticas y semánticas que definen la estructura del lenguaje.

Lenguajes de Programación según su nivel de abstracción

En un primer estado de clasificación, los lenguajes de programación se dividen según su nivel de abstracción, en lenguajes de bajo nivel, lenguajes de medio nivel y lenguajes de alto nivel, dependiendo de su grado de "cercanía al hardware".

Cuanto más cercano al hardware se encuentra el lenguaje, más bajo nivel posee éste. Mientras que cuanto más acercado al usuario se encuentre, más alejado del hardware estará y, en consecuencia, de mayor nivel será el lenguaje.

Lenguajes de Programación de Bajo Nivel

Los **lenguajes de bajo nivel**, son aquellos que dependen intrínsecamente del ordenador. Aquellos programas informáticos, programados con lenguajes de bajo nivel, al ser exclusivamente dependientes del hardware, no pueden migrarse, ya que están justamente diseñados, para un hardware específico.

Existen dos tipos de lenguajes de bajo nivel: el **lenguaje máquina** y el **lenguaje ensamblador**.

El **lenguaje de máquina** (también denominado lenguaje de primera generación) es el sistema de códigos directamente interpretable por un circuito microprogramable, como el microprocesador de una computadora o el microcontrolador de un autómata . Este lenguaje está compuesto por un conjunto de instrucciones que determinan acciones a ser tomadas por la máquina. Un programa consiste en una cadena de estas instrucciones de lenguaje de máquina (más los datos). Estas instrucciones son normalmente ejecutadas en secuencia, con eventuales cambios de flujo causados por el propio programa o eventos externos. El lenguaje de máquina es específico de cada máquina o arquitectura de la máquina, aunque el conjunto de instrucciones disponibles pueda ser similar entre ellas[6].

6 Fuente: http://es.wikipedia.org/wiki/Lenguaje_m%C3%A1quina

```
8B542408 83FA0077 06B80000 0000C383
FA027706 B8010000 00C353BB 01000000
B9010000 008D0419 83FA0376 078BD98B
C84AEBF1 5BC3
```

Función en 32-bits en código de maquina x86, para calcular el enésimo número de la serie de Fibonacci[7]

Un **lenguaje ensamblador**, o **assembler** (assembly language) es un lenguaje de programación de bajo nivel para los ordenadores, microprocesadores, microcontroladores, y otros circuitos integrados programables. Implementa una representación simbólica de los códigos de máquina binarios y otras constantes necesarias para programar una arquitectura dada de CPU y constituye la representación más directa del código máquina específico para cada arquitectura legible por un programador. Esta representación es usualmente definida por el fabricante de hardware, y está basada en los mnemónicos[8] que simbolizan los pasos de procesamiento (las instrucciones), los registros del procesador, las posiciones de memoria, y otras características del lenguaje. Un lenguaje ensamblador es por lo tanto, específico a cierta arquitectura de computador física (o virtual). Esto está en contraste con la mayoría de los lenguajes de programación de alto nivel, que, idealmente son portables[9].

Este lenguaje, también es conocido como **lenguaje de segunda generación**.

```
; HOLA.ASM
; Programa clasico de ejemplo. Despliega una leyenda en pantalla.
STACK     SEGMENT STACK                    ; Segmento de pila
                  DW     64 DUP (?)       ; Define espacio en la pila
STACK     ENDS

DATA      SEGMENT                          ; Segmento de datos
SALUDO      DB    "Hola mundo!!",13,10,"$" ; Cadena
DATA      ENDS

CODE      SEGMENT                          ; Segmento de Codigo
          ASSUME CS:CODE, DS:DATA, SS:STACK

INICIO:                                    ; Punto de entrada al programa
          MOV  AX,DATA                     ; Pone direccion en AX
          MOV  DS,AX                       ; Pone la direccion en los registros
          MOV  DX,OFFSET SALUDO            ; Obtiene direccion del mensaje
          MOV  AH,09H                      ; Funcion: Visualizar cadena
          INT     21H                      ; Servicio: Funciones alto nivel DOS
          MOV  AH,4CH                      ; Funcion: Terminar
```

7 Fuente: http://en.wikipedia.org/wiki/Low-level_programming_language

8 En informática, un mnemónico es una palabra que sustituye a un código de operación (lenguaje de máquina), con lo cual resulta más fácil la programación, es de aquí de donde se aplica el concepto de lenguaje ensamblador. Fuente: http://es.wikipedia.org/wiki/Mnem%C3%B3nico

9 Fuente: http://es.wikipedia.org/wiki/Lenguaje_ensamblador

```
          INT     21H
CODE      ENDS
          END  INICIO                    ; Marca fin y define INICIO
```

Ejemplo desarrollado en lenguaje ensamblador que usa llamadas de MS-DOS (system calls) para imprimir el mensaje Hola mundo!! en pantalla. Extraído de http://homepage.mac.com/eravila/asmix862.html

(para ver la explicación detallada del ejemplo, seguir el enlace anterior)

Para ampliar la información sobre los lenguajes de bajo nivel, puede leerse el siguiente artículo de Karmany.net.

También es recomendable, leer el siguiente artículo sobre Lenguaje Ensamblador en Wikipedia.

Lenguajes de Programación de Medio Nivel

La clasificación de lenguajes de programación, mediante un nivel de abstracción medio, es bastante discutible. Personalmente sostengo sólo por dos niveles de abstracción: bajo nivel y alto nivel. Sin perjuicio de ello, se plasmarán aquí, aquellos argumentos sostenidos, por quienes aceptan este tercer nivel de clasificación.

Quienes sostienen la clasificación de lenguajes de programación medio, argumentan que éstos, **son aquellos lenguajes que se encuentran, justamente, entre los de bajo nivel y los de alto nivel, ya que poseen características que permiten interactuar directamente con el sistema**. Un ejemplo de ello, sería el lenguaje C, el cual puede trabajar (entre otras características) con direcciones de memoria. Sin embargo, dicho acceso, no es efectuado de forma directa (a través de lenguaje máquina o ensamblador), sino que requiere ser “traducido” previamente por su compilador. Por dicha razón, es que asumo a C como lenguaje de alto nivel y descarto la clasificación de lenguajes de medio nivel.

Suele colocarse como ejemplo de lenguaje de programación de medio nivel, anterior a C, a **BCPL**[10], diseñado para escribir Sistemas Operativos y Compiladores.

```
GET "libhdr"

LET start() = VALOF
{ FOR i = 1 TO 5 DO writef("fact(%n) = %i4*n", i, fact(i))
```

10 Ver manual de Referencias de BCPL

```
  RESULTIS 0
}

AND fact(n) = n=0 -> 1, n*fact(n-1)
```

Ejemplo de código BCPL para impresión de factoriales[11]

Lenguajes de Programación de Alto Nivel

Los lenguajes de alto nivel, son aquellos cuya característica principal, consiste en una estructura sintáctica y semántica legible, acorde a las capacidades cognitivas humanas. A diferencia de los lenguajes de bajo nivel, **son independientes de la arquitectura del hardware**, motivo por el cual, asumen mayor portabilidad.

Son ejemplo de lenguajes de alto nivel: Python, Perl, PHP, Ruby, Lisp, Java, Fortran, C++, C#, entre otros.

```
print "Hola Python!"
```

Archivo: hola_mundo.py . Imprime Hola Python! En pantalla

```
<?php
echo "Hola PHP!";
?>
```

Archivo: hola_mundo.php . Imprime Hola PHP! En pantalla

```
print "Hola Perl!";
```

Archivo: hola_mundo.pl . Imprime Hola Perl! En pantalla

Hola Mundo en otros lenguajes (para curiosos): www.holamundo.es

Clasificación de Lenguajes de programación, según su forma de ejecución
Según su forma de ejecución, los lenguajes de programación pueden ser: **compilados** o **interpretados**.

Los **lenguajes de programación compilados**, son lenguajes de alto nivel que requieren que las instrucciones (código fuente del programa), sean traducidas a lenguaje máquina por un compilador, a fin de generar un ejecutable del programa. Ejemplo de lenguajes compilados son Pascal, C, C++, Ada, entre otros.

---

11 Fuente: http://en.wikipedia.org/wiki/BCPL#Examples

```
#include

int main()
{
        printf("Hola mundo");
        return 0;
}
```

Ejemplo en C que imprimirá "Hola mundo" en pantalla tras ser compilado.

```
program Hello;
   begin
      writeln ('Hola mundo')
   end.
```

Mismo ejemplo, pero en Pascal.

Los **lenguajes interpretados**, a diferencia de los compilados, no requieren de un compilador para ser ejecutados sino de un intérprete. Un intérprete, actúa de manera casi idéntica a un compilador, con la salvedad de que ejecuta el programa directamente, sin necesidad de generar previamente un ejecutable. Ejemplo de lenguajes de programación interpretado son Python, PHP, Ruby, Lisp, entre otros.

```
(print "Hola Mundo!")
```

Ejemplo de código Lisp que imprime "Hola Mundo!" en pantalla

Es importante además, hacer notar que la mayoría de los lenguajes de programación, puede ejecutarse tanto de modo compilado como interpretado.