

# Apuntes de Programación Ensamblador

Leonardo H. Añez Vladimirovna<sup>1</sup>

*Universidad Autónoma Gabriel René Moreno,  
Facultad de Ingeniería en Ciencias de la Computación y Telecomunicaciones,  
Santa Cruz de la Sierra, Bolivia*

5 de agosto de 2018

<sup>1</sup>Correo Electrónico: [toborochi98@outlook.com](mailto:toborochi98@outlook.com)

## Notas del Autor

Estos apuntes fueron realizados durante mis clases en la materia INF221 (Programación Ensamblador), acompañados de referencias de libros, fuentes y código que use a lo largo del curso, en el período II-2018 en la Facultad de Ingeniería en Ciencias de la Computación y Telecomunicaciones.

Para cualquier cambio, observación y/o sugerencia pueden enviarme un mensaje al siguiente correo:

`toborochi98@outlook.com`

# Índice general

<b>1. Introducción</b>	<b>5</b>
1.1. Entendiendo los Niveles de Lenguajes de Programación . . . . .	5



# Capítulo 1

## Introducción

Primeramente hay que entender que es el Lenguaje de Programación *Ensamblador*. Para ello hay que entender los niveles de los lenguajes del computador.

### 1.1. Entendiendo los Niveles de Lenguajes de Programación

Solo hay un lenguaje de programación que cualquier computadora puede entender y ejecutar: su propio código de máquina, *binario nativo*. Este es el nivel de lenguaje más bajo posible en el cual se puede escribir un programa de computadora. Se dice que todos los otros idiomas son de nivel alto o bajo de acuerdo con cuán cerca se puede decir que se asemejan al código de la máquina.

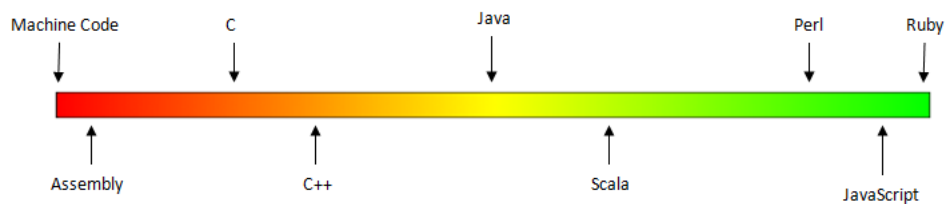


Figura 1.1: Niveles de Programación en el Computador.

Fuente: <http://www.codecommit.com/blog/java/defining-high-mid-and-low-level-languages>

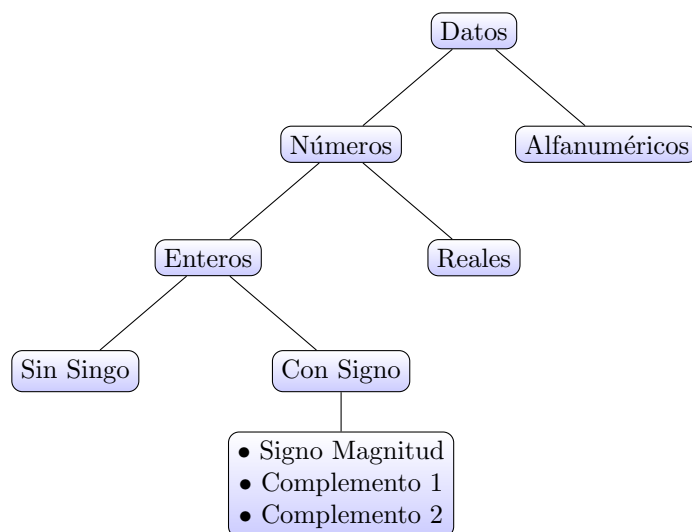


Figura 1.2: Datos en un Computador

Regular ASCII Chart (character codes 0 - 127)															
000	(nul)	016 ▶ (dle)	032 sp	048 0	064 Ø	080 P	096 `	112 p							
001 Ⓒ (soh)	017 ◀ (dc1)	033 !	049 1	065 A	081 Q	097 a	113 q								
002 Ⓓ (stx)	018 ↑ (dc2)	034 "	050 2	066 B	082 R	098 b	114 r								
003 ♥ (etx)	019 !! (dc3)	035 #	051 3	067 C	083 S	099 c	115 s								
004 ⦶ (eot)	020 ‡ (dc4)	036 \$	052 4	068 D	084 T	100 d	116 t								
005 Ⓔ (enq)	021 § (nak)	037 %	053 5	069 E	085 U	101 e	117 u								
006 ♠ (ack)	022 — (syn)	038 &	054 6	070 F	086 V	102 f	118 v								
007 • (bel)	023 ‡ (etb)	039 '	055 7	071 G	087 W	103 g	119 w								
008 ▣ (bs)	024 † (can)	040 ()	056 8	072 H	088 X	104 h	120 x								
009 (tab)	025 ↓ (em)	041 )	057 9	073 I	089 Y	105 i	121 y								
010 (lf)	026 (eof)	042 *	058 :	074 J	090 Z	106 j	122 z								
011 ♂ (vt)	027 — (esc)	043 +	059 ;	075 K	091 [	107 k	123 {								
012 * (np)	028 L (fs)	044 ,	060 <	076 L	092 \	108 l	124								
013 (cr)	029 ↔ (gs)	045 -	061 =	077 M	093 ]	109 m	125 }								
014 ♂ (so)	030 ▲ (rs)	046 .	062 >	078 N	094 ^	110 n	126 ~								
015 ♀ (si)	031 ▼ (us)	047 /	063 ?	079 O	095 _	111 o	127								

# Bibliografía

- [1] Dennis G. Zill, *Lenguaje Ensamblador para Computadoras Basadas en Intel, Quinta Edición*, Thomson Learning, 2005.
- [2] Ken Bigelow, [http://www.play-hookey.com/computers/language\\_levels.html](http://www.play-hookey.com/computers/language_levels.html)