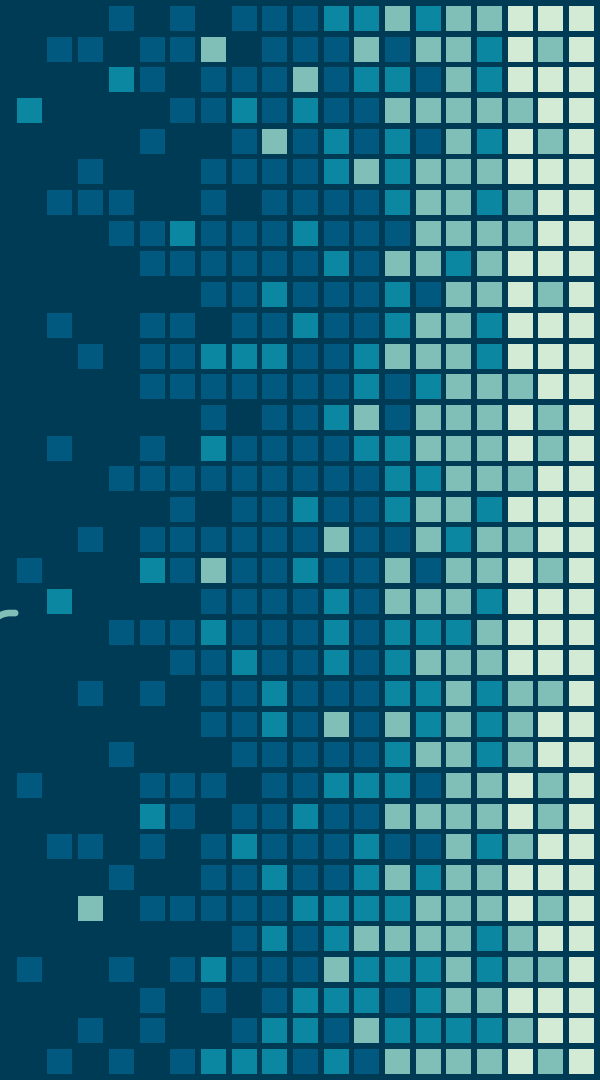


Introducción al Lenguaje Ensamblador

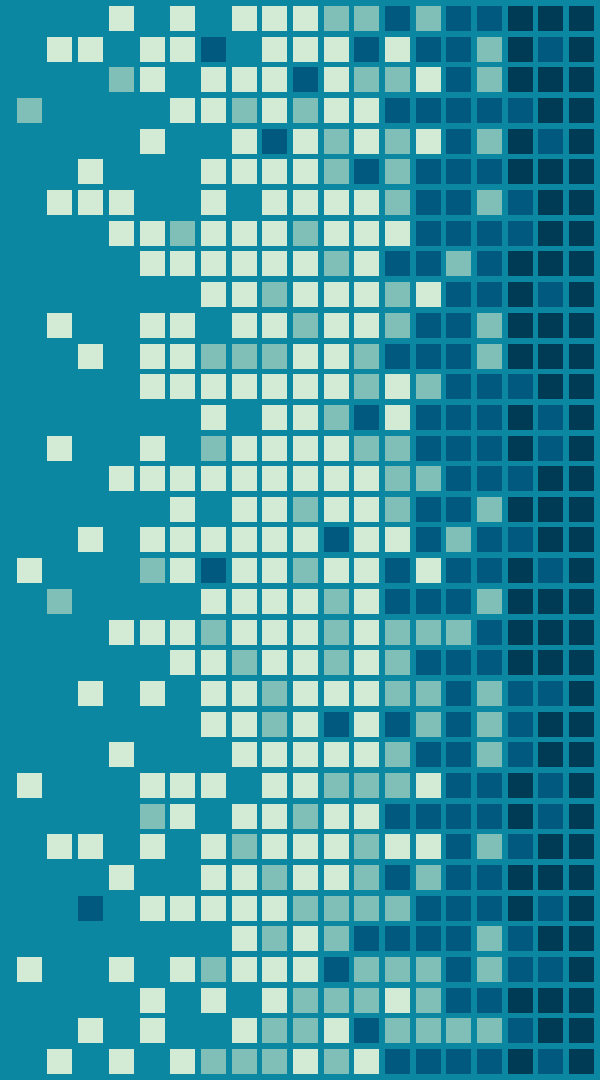


1. Lenguajes de Programación

Definición



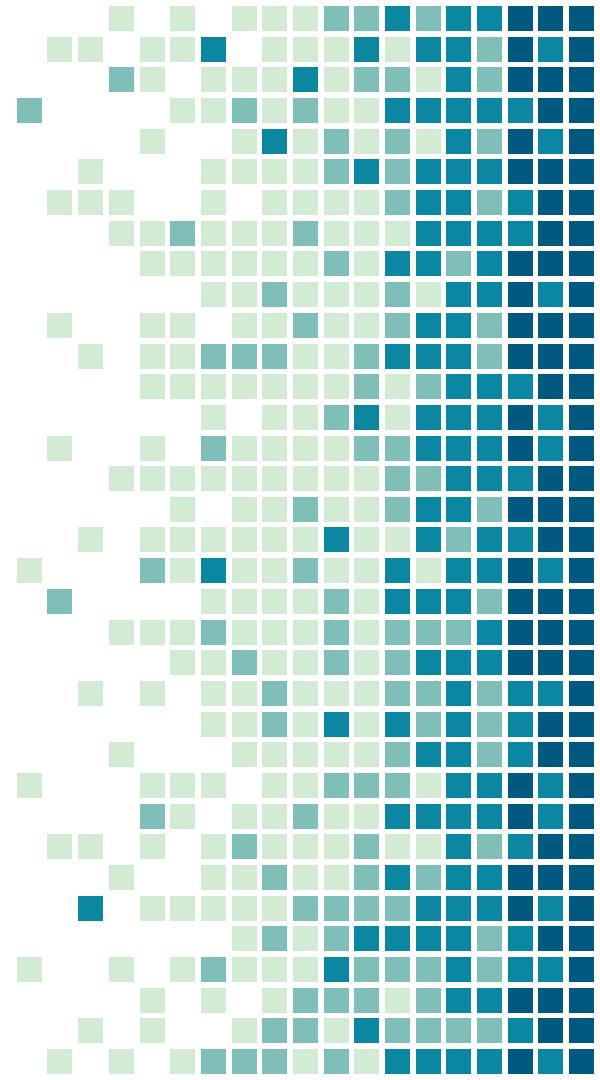
“ Los lenguajes de programación sirven para escribir programas que permitan la comunicación usuario – máquina.
(Joyanes; 2003)



1.1

Lenguajes de Programación

Clasificación



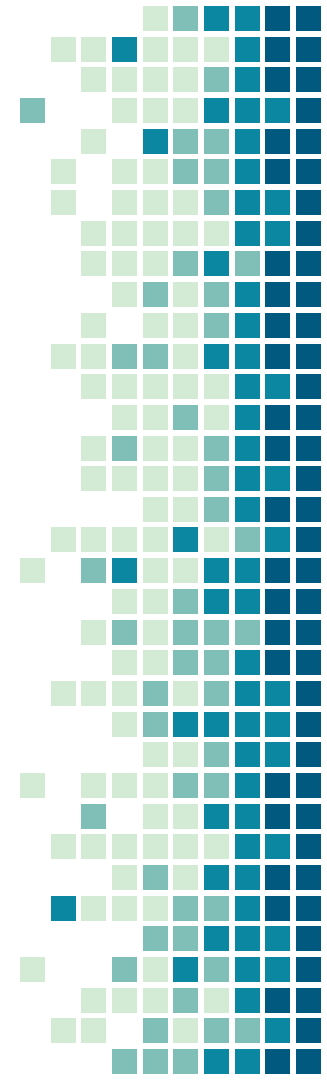
Según Paradigma de Programación:

Imperativos

Se ordena a la computadora cómo realizar una tarea, siguiendo una serie de pasos o instrucciones.

Declarativos

Se le indica a la computadora qué es lo que se desea obtener o qué es lo que se esta buscando (Lógicos y Funcionales).



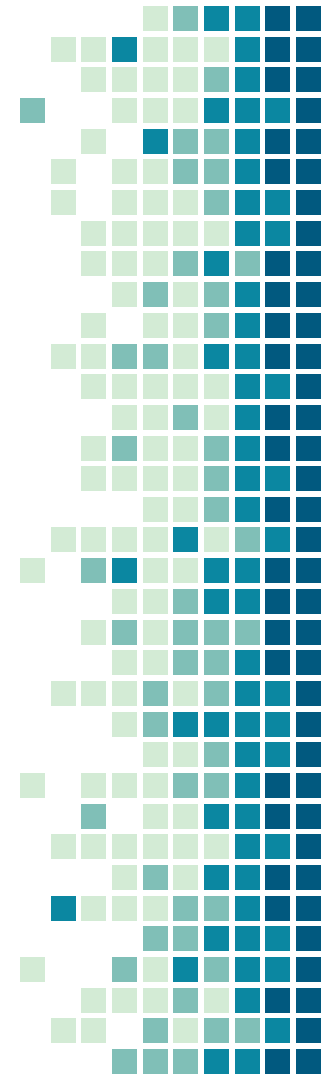
Según forma de Ejecución:

Compilados

Aquellos que se ejecutan como un todo, pasando por uno o varios procesos de traducción hasta ser comprensibles por la computadora.

Interpretados

Aquellos que se ejecutan instrucción por instrucción, haciendo traducción de un determinado lenguaje a un idioma comprensible por la computadora.



Según su Nivel de Abstracción:

Alto Nivel

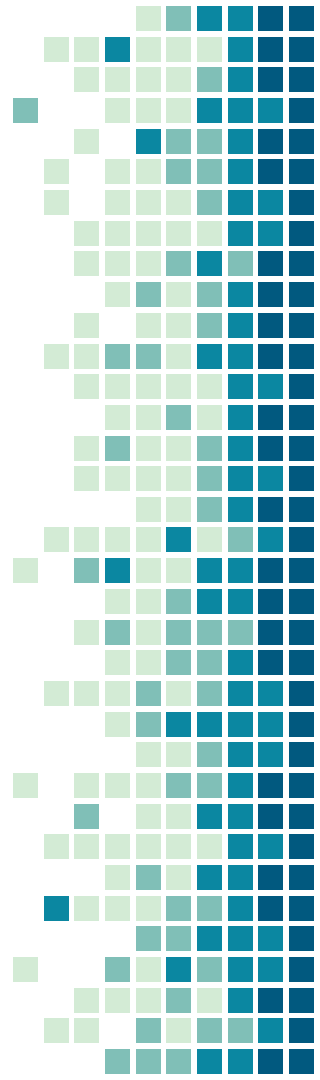
Lenguajes independientes de la arquitectura del computador, por lo que se puede migrar de una máquina a otra tomando en cuenta únicamente los factores de compatibilidad.

Mediano Nivel

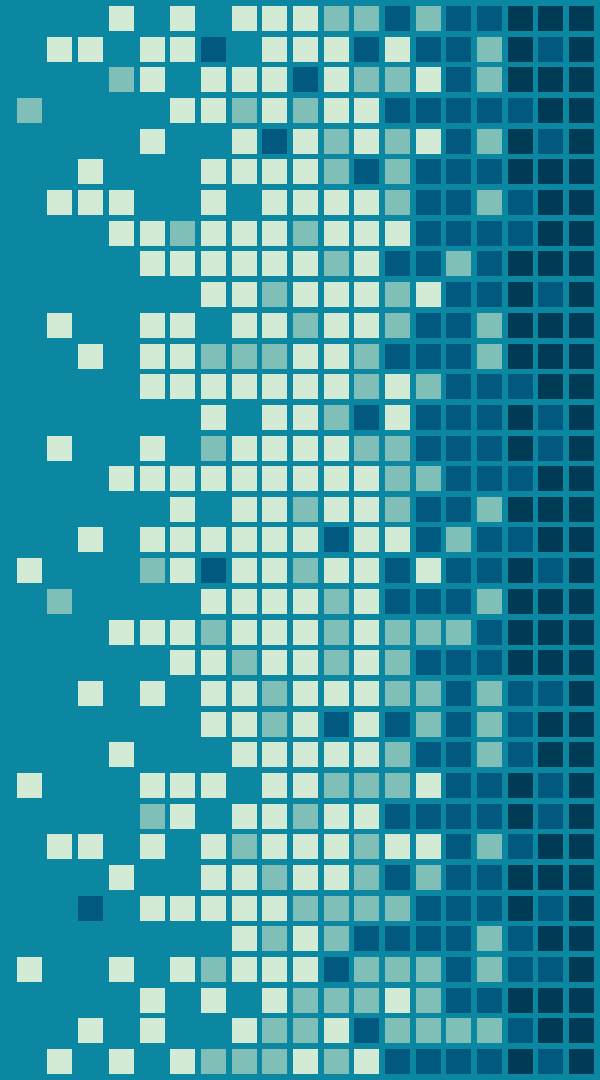
Tienen características que los acercan a los lenguajes de bajo nivel, pero, al mismo tiempo, ciertas cualidades que lo hacen un lenguaje más cercano al humano y, por tanto, de alto nivel.

Bajo Nivel

Son lenguajes dependientes arquitectura del computador, no se puede migrar o utilizar en otras máquinas sin recodificación.



“ Los lenguajes más cercanos a la arquitectura hardware se denominan lenguajes de bajo nivel y los que se encuentran más cercanos a los programadores y usuarios se denominan lenguajes de alto nivel.



Lenguaje de Bajo Nivel:

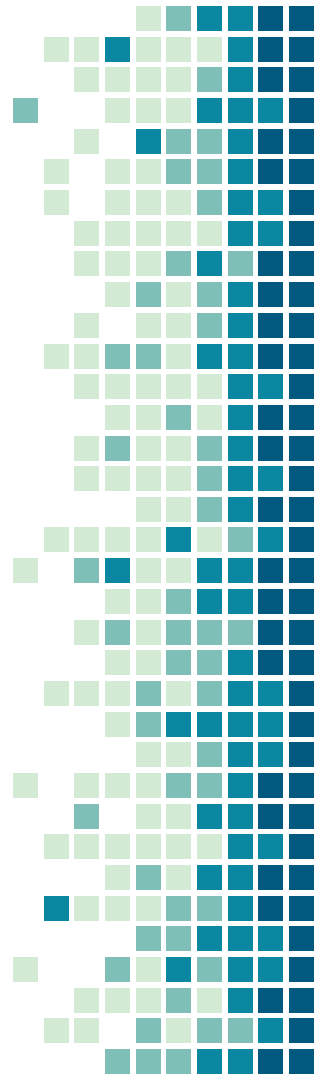
Lenguaje de Máquina

Es el que da órdenes a la máquina, que son las operaciones fundamentales para su funcionamiento, en código binario o código máquina, consistente en ceros y unos.

Lenguaje Ensamblador

Es un derivado del lenguaje máquina y está formado por abreviaturas de letras y números.

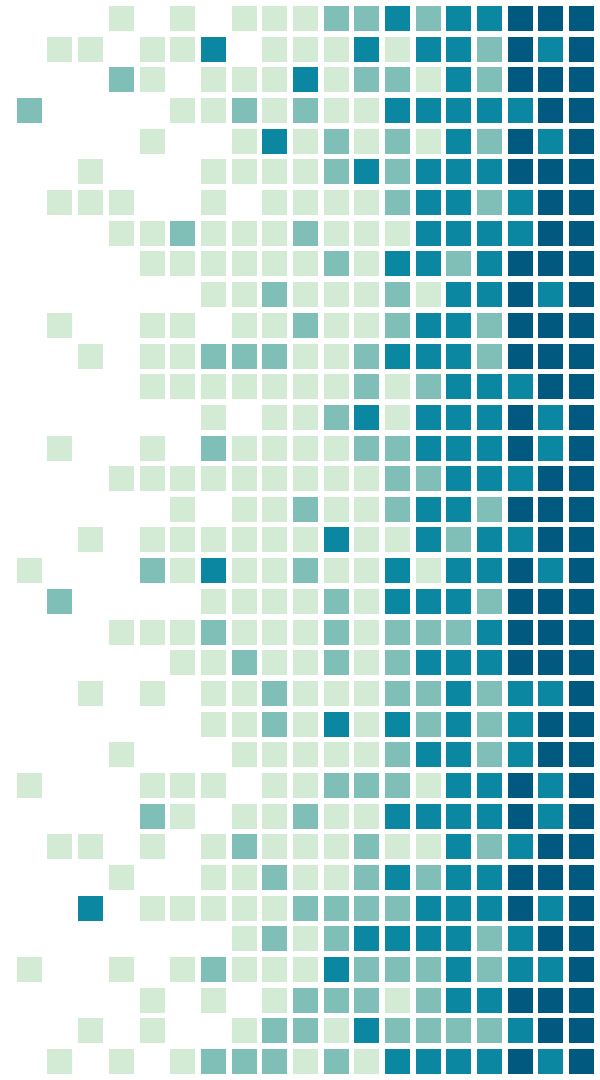
Se trabaja a nivel de instrucciones, es decir, su programación es al más fino detalle.



1.2

Lenguajes de Programación

Evolución

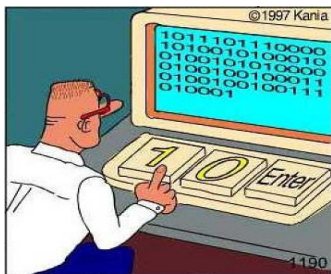


Por generaciones:

Primera
Lenguaje de
Máquina

Segunda
Lenguajes
Ensambladores

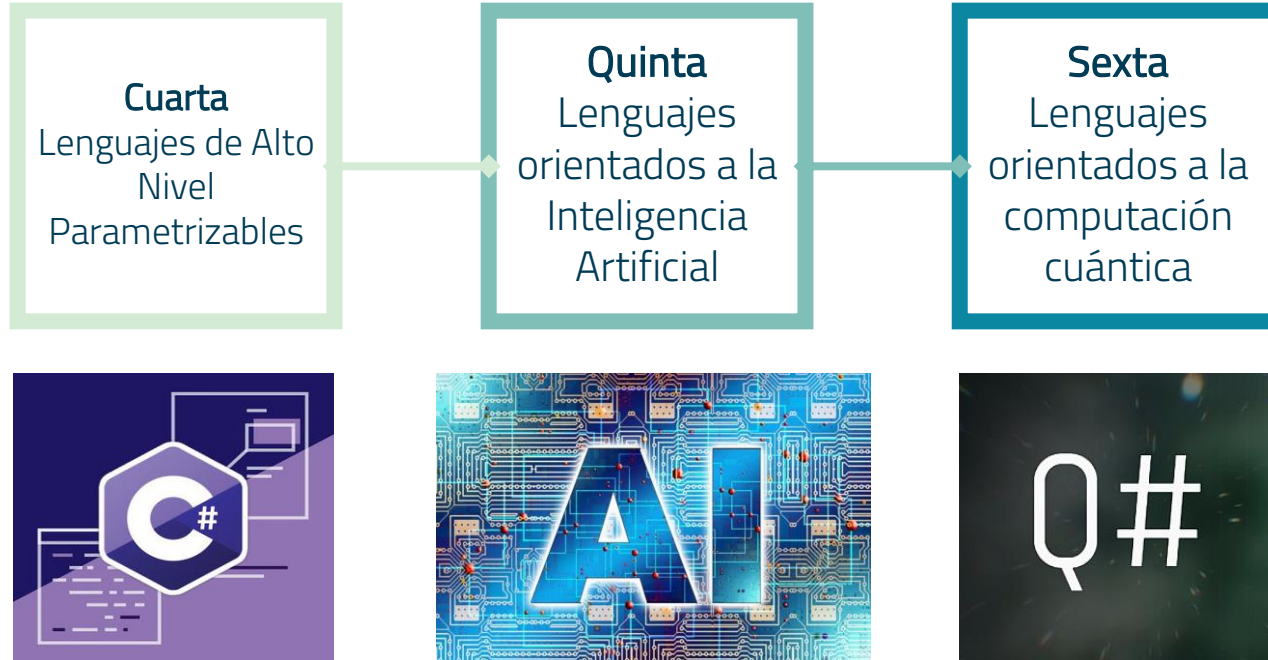
Tercera
Lenguajes de
Alto Nivel
Funcionales



**Lenguaje
Ensamblador**



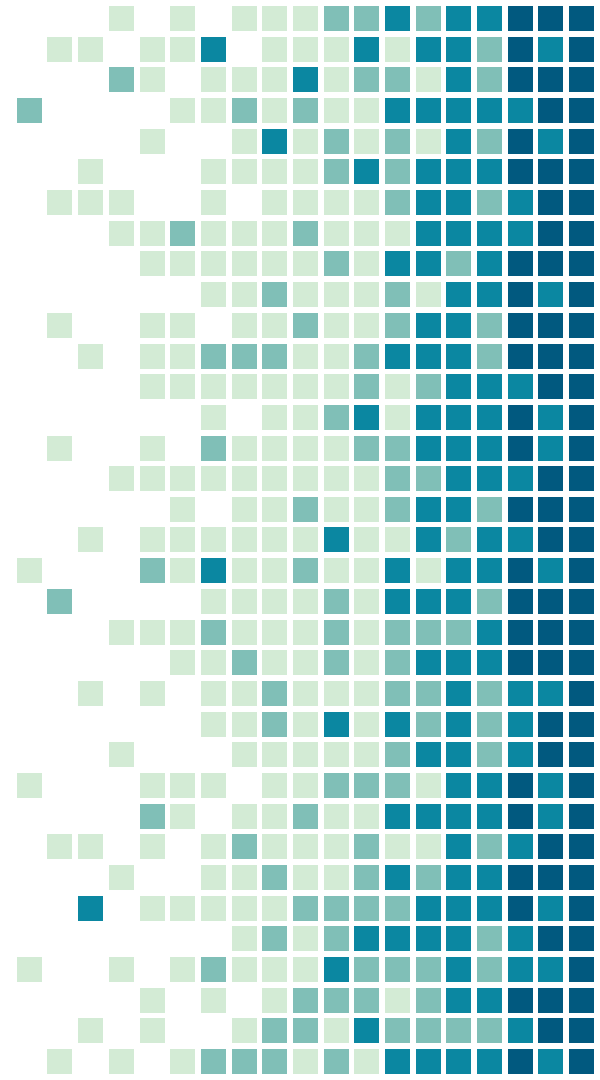
Por generaciones:



2.2

Lenguaje Ensamblador

Aplicaciones típicas

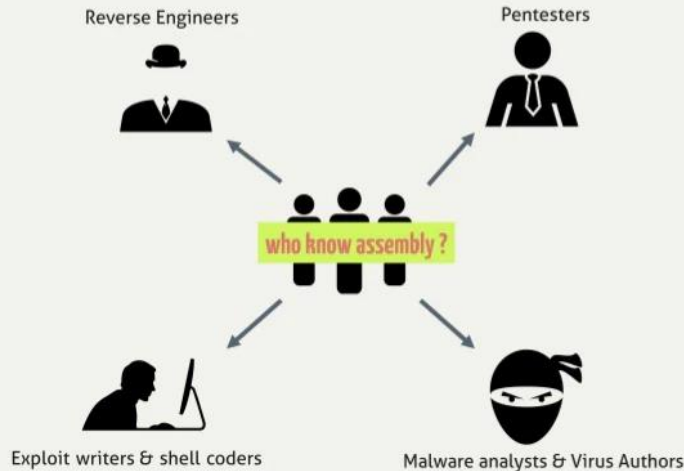




En los inicios...

- Sistemas Operativos IBM PC DOS.
- Lotus 1, 2, 3.
- Atari 800.
- Mega Drive/Genesis.
- Super Nintendo Entertainment System.
- Sega Saturn.

Who should know assembly ?



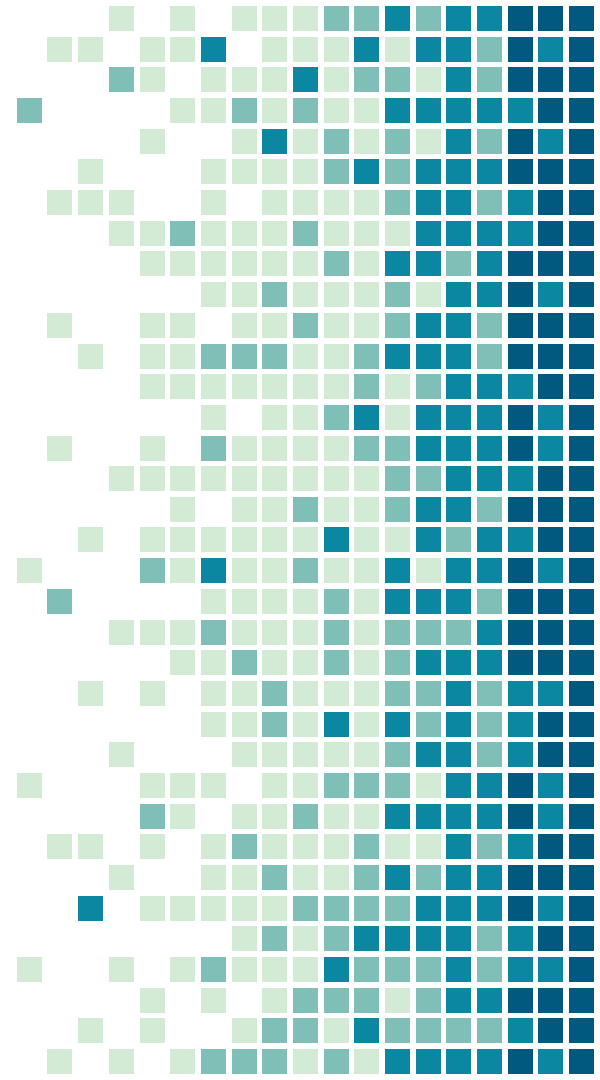
En la actualidad...

- Controladores de dispositivos
- Sistemas embebidos
- Sistemas de tiempo real
- Programación de hardware
- Hacking

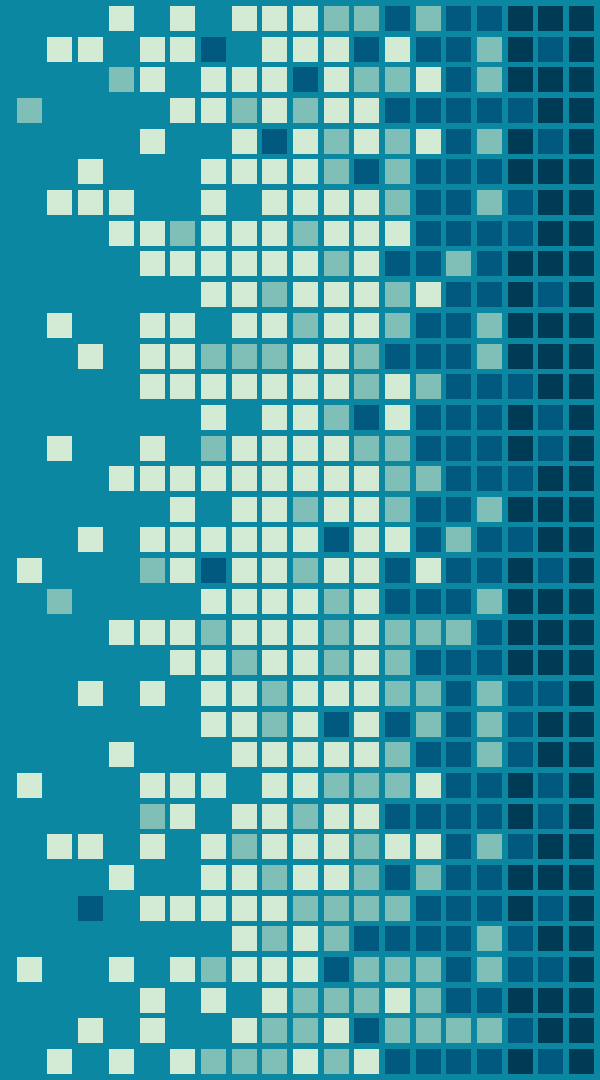
2.3

Lenguaje Ensamblador

Desambiguación



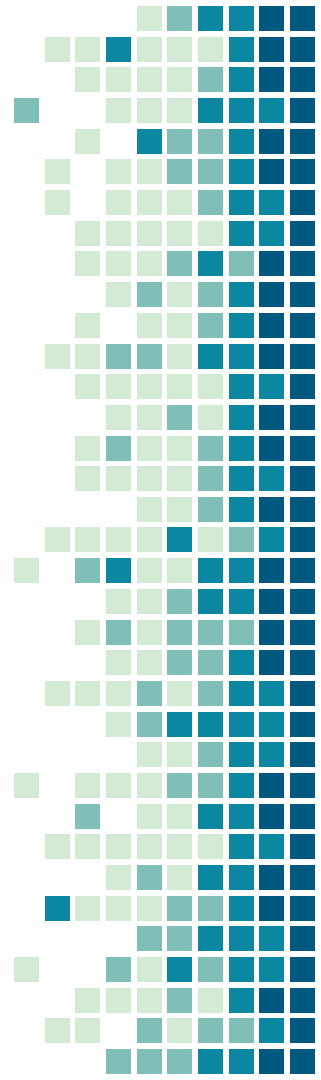
“ Las máquinas
(computadoras), al ser
electrónicas, solo pueden
percibir la presencia o
ausencia de energía, el paso
o la interrupción de corriente
eléctrica.





Lenguaje de Máquina

Basado en el código binario, conjunto de '0' y '1', es el único que puede ser interpretado por las máquinas (computadoras).



Instrucción
de alto nivel

Instrucción en
ensamblador

Instrucción en
ensamblador

Instrucción
binaria

00011010
10010101
01101110
01010110

Instrucción
binaria

01101010
10101110
00010101
10101000

Instrucción
binaria

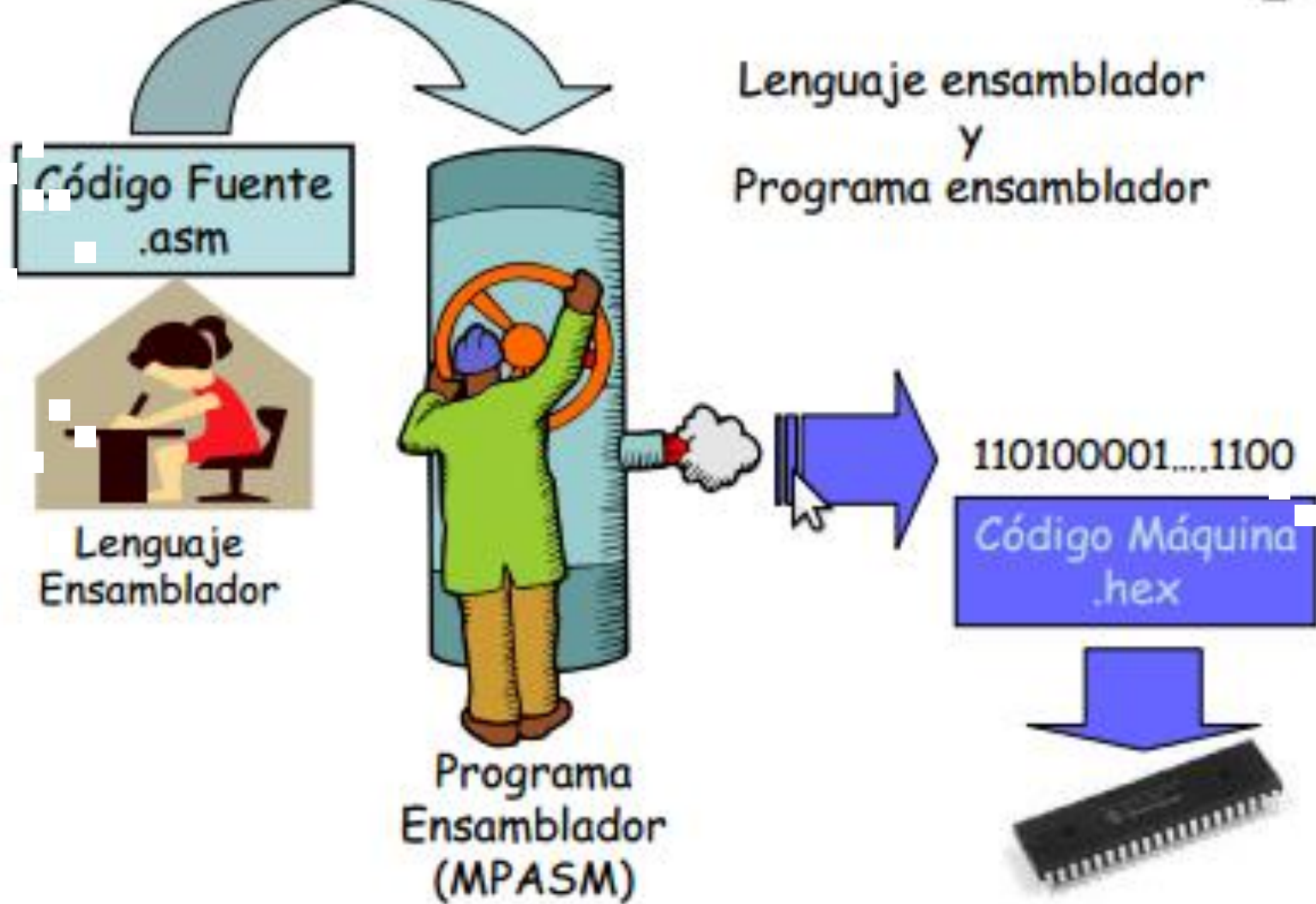
01100101
00101001
11101011
10101111

Instrucción
binaria

11010000
10100011
01011110
00110010

Instrucción
binaria

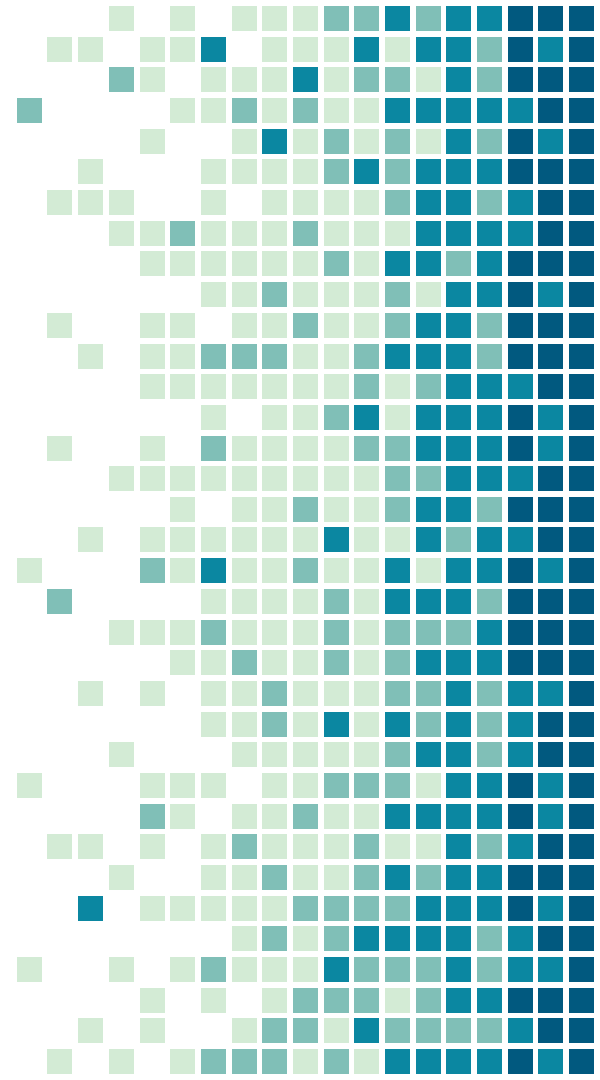
10100011
10001001
00101010
00000001



2.4

Lenguaje Ensamblador

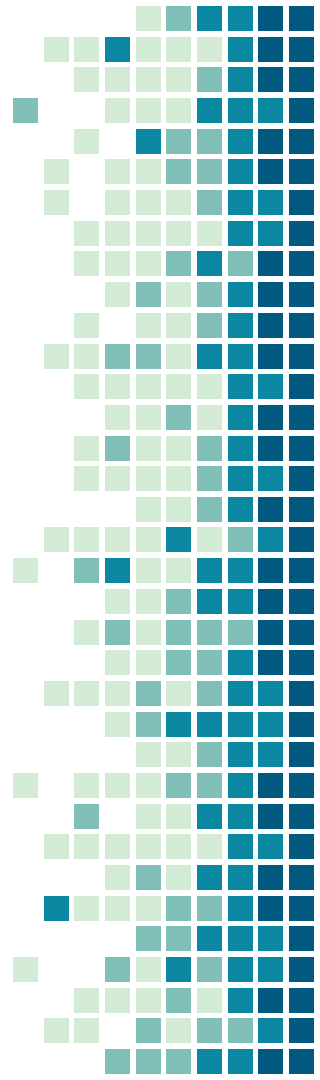
Características





Ventajas

- Mayor velocidad de ejecución.
- Poca demanda de recursos.
- Control preciso de tareas.
- Ejecución directa sobre el Hardware.
- Traducción única.



python

```
print("hello world")
```

java

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

c++

```
#include<iostream>
using namespace std;
int main()
{
    cout<<"hello world "<<endl;
    return 0;
}
```

Assembly:

```
global _main
extern _printf
section .text
_main:
    push    message
    call    _printf
    add     esp, 4
    ret
message:
    db 'Hello, World', 10, 0
```



Desventajas

- Programación compleja (para el usuario).
- Codificación prolongada (tiempo del usuario).
- Programa fuente extenso.
- Portabilidad reducida o nula.
- Errores irreversibles.



¿Dudas?

