



ESCUELA DE
INGENIERÍA EN CIENCIAS Y SISTEMAS
FACULTAD DE INGENIERÍA
UNIVERSIDAD DE SAN CARLOS DE GUATEMALA



Día, Fecha:	Sábado
Hora de inicio:	12/08/2023 07:10

Lenguajes Formales y de Programación B-

Elder Anibal Pum Rojas

CLASE 3

LABORATORIO LFP "B-"

Aux. Elder Pum



Agenda

- Avisos
- Clase
 - Lenguajes
 - Evolución de los Lenguajes Formales
 - Paradigmas
 - Jerarquía de Chomsky
 - Gramáticas Regulares
 - Python
- Parte Práctica

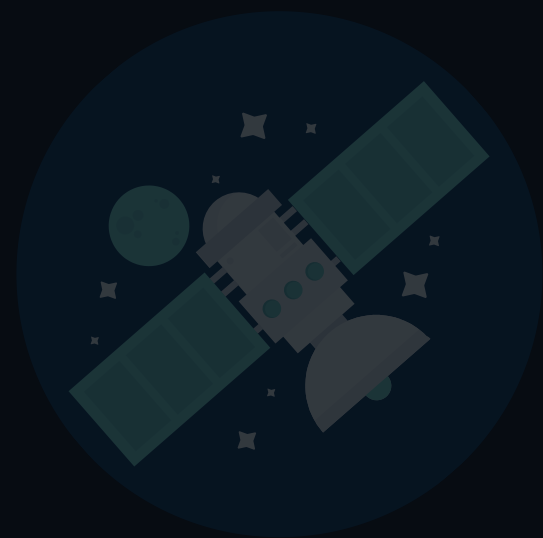




Avisos

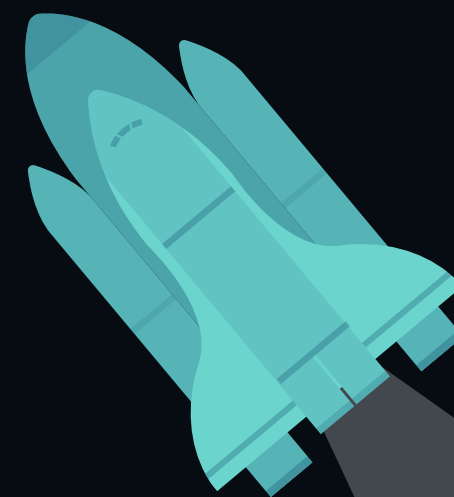
Avisos

- **Práctica a entregarse Pronto**
- **Notas (General)**



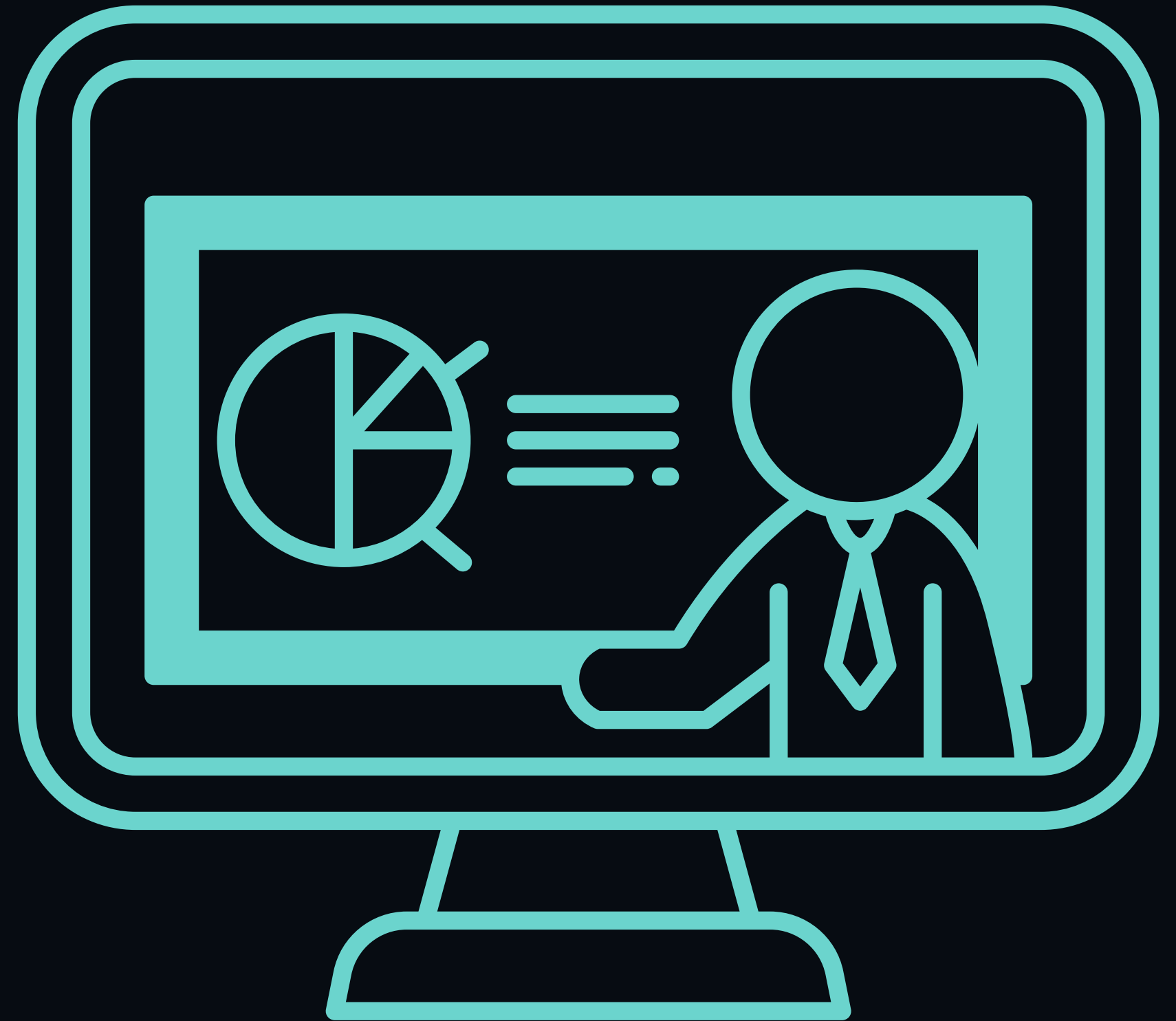
¿DUDAS?

¿DUDAS?





CLASE



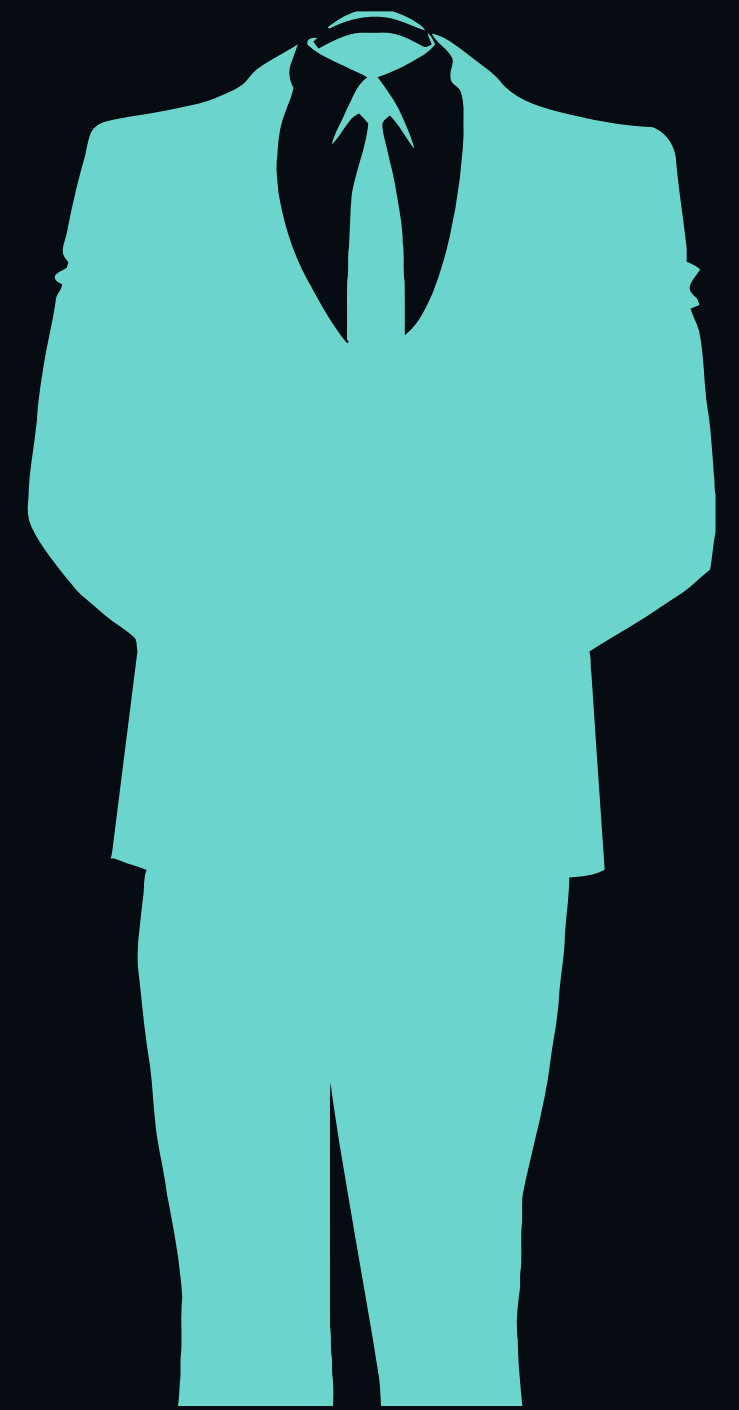
Lenguaje Natural

Es el medio que utilizamos de manera cotidiana para establecer la comunicación entre seres humanos



Lenguaje Formal

Es un lenguaje que ha sido desarrollado para expresar situaciones que se dan en específico en cada área del conocimiento científico.

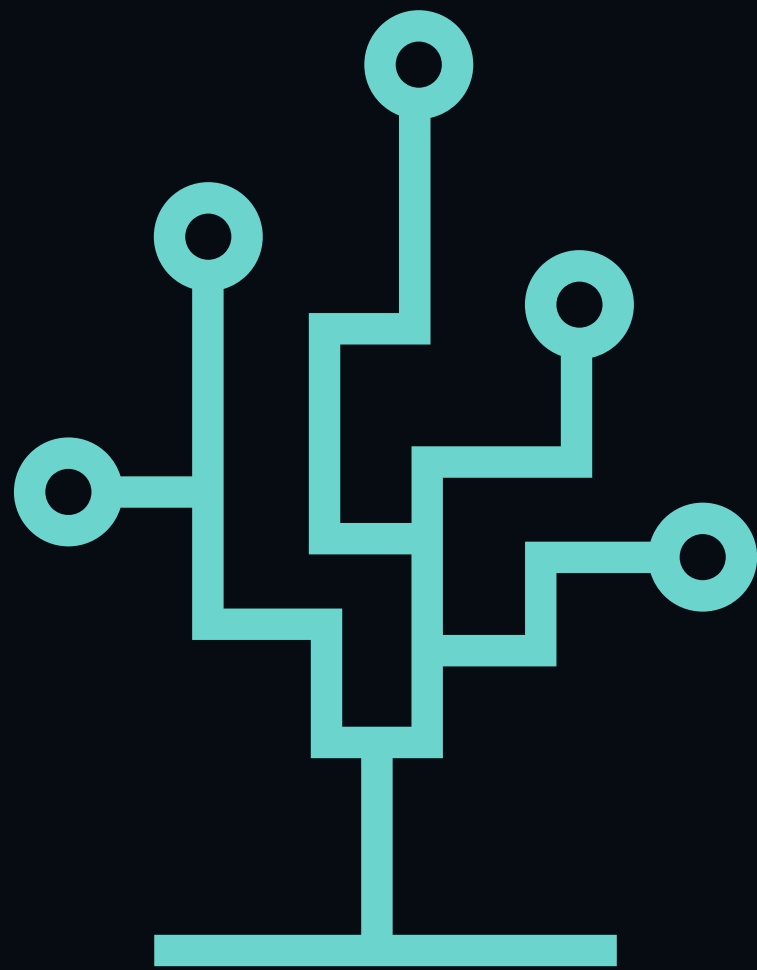


Lenguaje de Programación

Es un lenguaje informático especialmente diseñado para describir el conjunto de acciones consecutivas o instrucciones que un equipo informático debe ejecutar



Evolución de los Lenguajes



Primera Generación



Segunda Generación



Tercera Generación



Cuarta Generación



Quinta Generación

Primera Generación

Sistema Binario

Consistía enteramente de una secuencia de 0's y 1's que los controles de la computadora interpretan como instrucciones, eléctricamente.

Representan los primeros lenguajes de programación que la computadora podía entender, son conocidos como lenguaje máquina.



Segunda Generación

Lenguaje Máquina

Los lenguajes simbólicos, asimismo propios de la máquina simplifican la escritura de las instrucciones y las hacen más legibles. Se refiere al lenguaje ensamblador. Es el lenguaje de máquina combinado con una serie de poderosas macros que permiten declarar estructuras de datos y de control complejas.

```
MOVI R1, 0 ; Contador
MOVI R2, N ; Límite
MOVI R3, 0 ; Suma parcial
$MOVEI R4, V ; @ inicial de V
bucle:
CMPLT R5, R1, R2
BZ R5, fibucle ; Acaba en 5
ADD R5, R1, R1 ; Cada dato
; ocupa 2 bytes
ADD R5, R4, R5
LD R6, 0(R5) ; Leemos un dato
ADD R3, R3, R6 ; Acumulamos
ADDI R1, R1, 1 ; Actualizamos
```

Tercera Generación

Lenguaje Alto Nivel

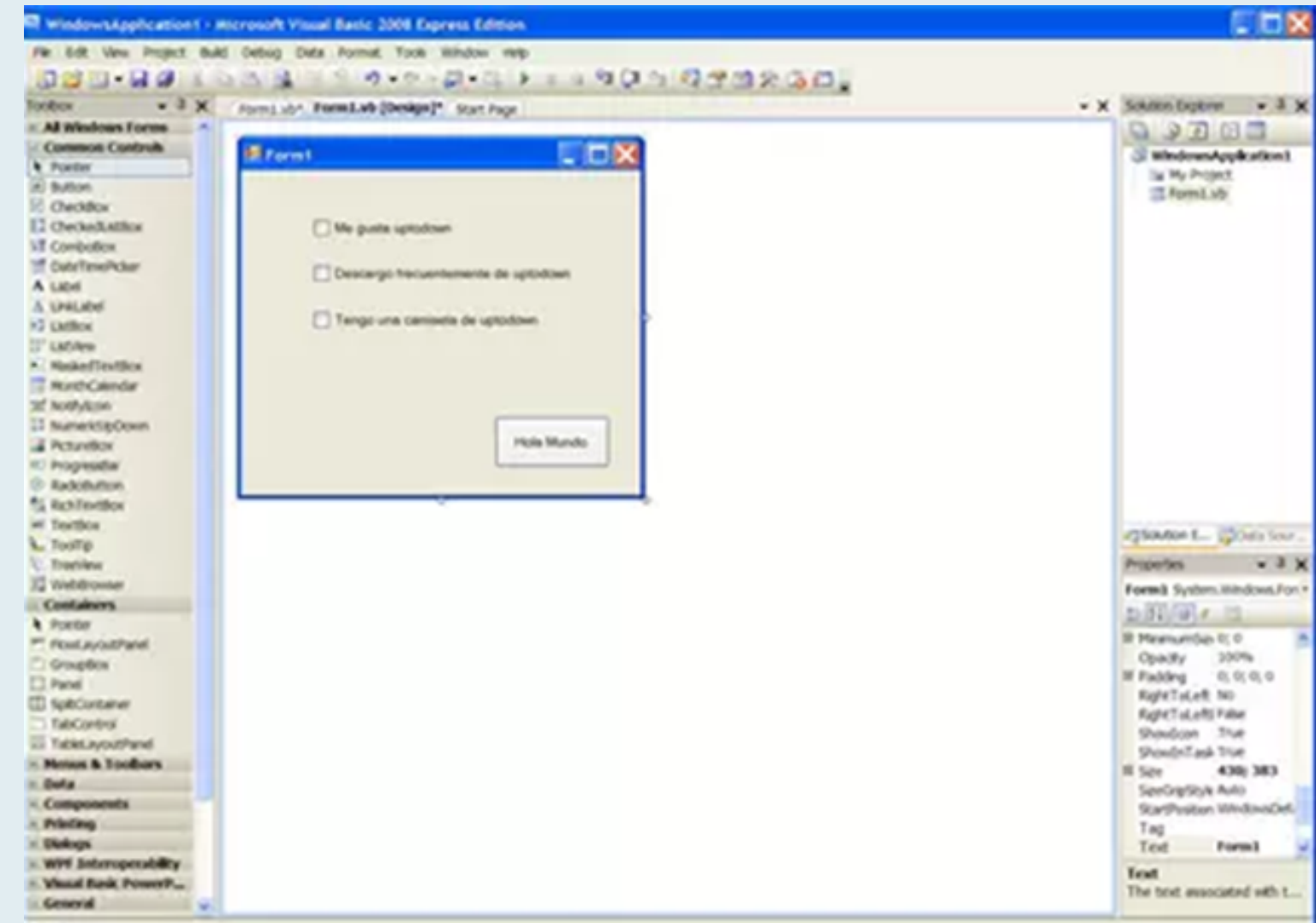
Se conoce como lenguajes de alto nivel. Un lenguaje de alto nivel tiene una gramática y sintaxis similar a las palabras en una oración. Un compilador se encarga de traducir el lenguaje de alto nivel a lenguaje ensamblador o código máquina. Un ejemplo puede ser C.

```
int main(void)
{
    printf("hello, world\n");
    return 0;
}
```

Cuarta Generación

Entornos de desarrollo

Son entornos de desarrollo de aplicaciones constituidos por un conjunto de herramientas integradas entre las que se encuentran editores, compiladores, sistemas para el acceso a bases de datos, generadores de informes, generadores de pantallas. C++, SQL, lenguajes orientados a objetos.

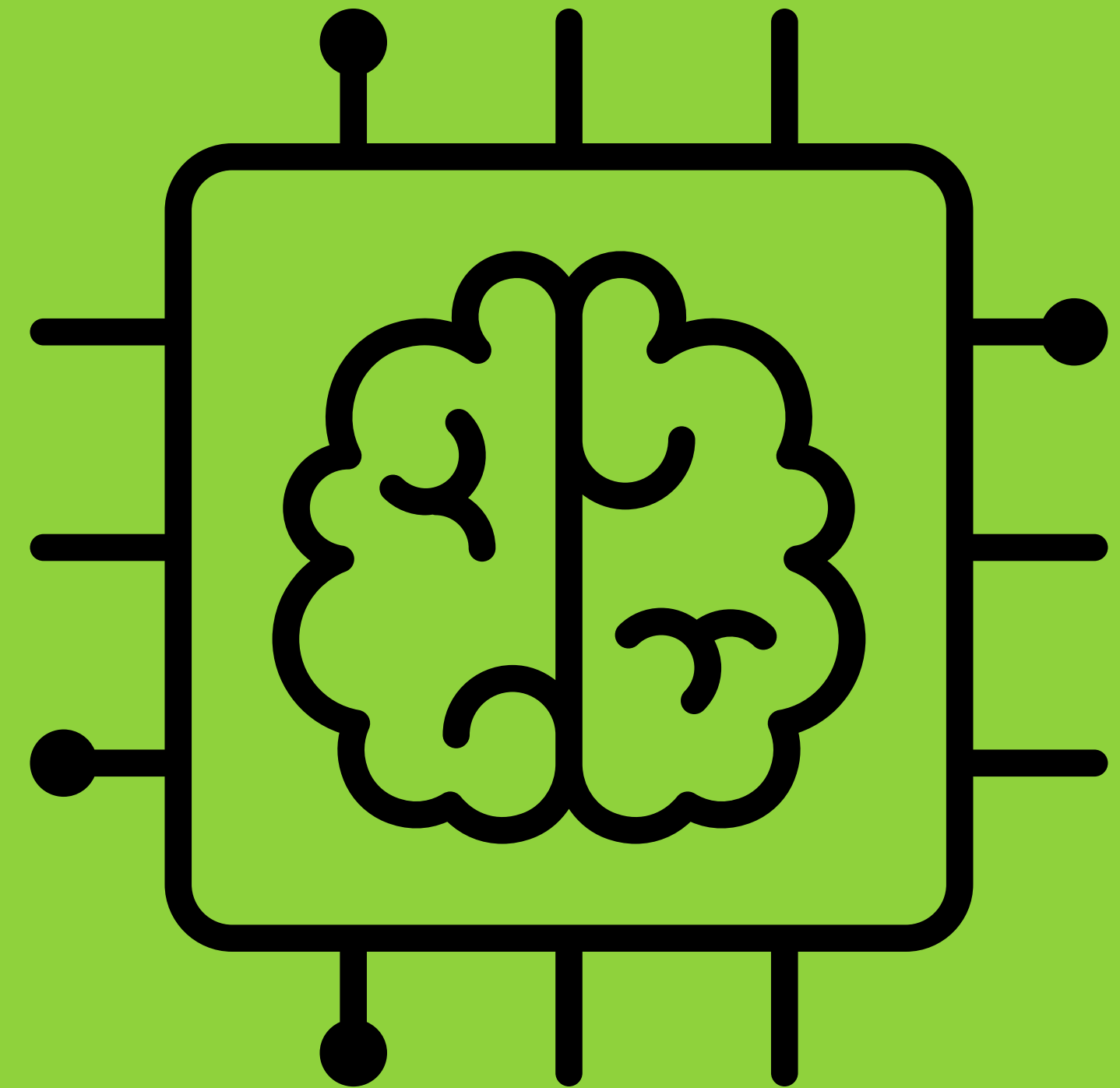


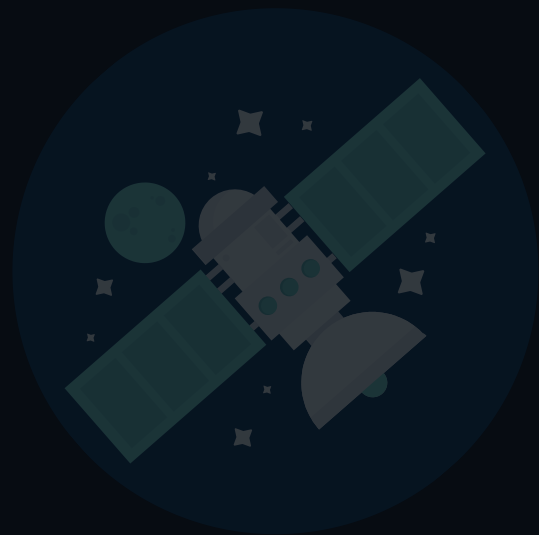
Quinta Generación

Se utilizan principalmente en la investigación de la inteligencia artificial, prolog, ops5 y mercury son ejemplos de lenguajes de quinta generación.

Linea del tiempo:

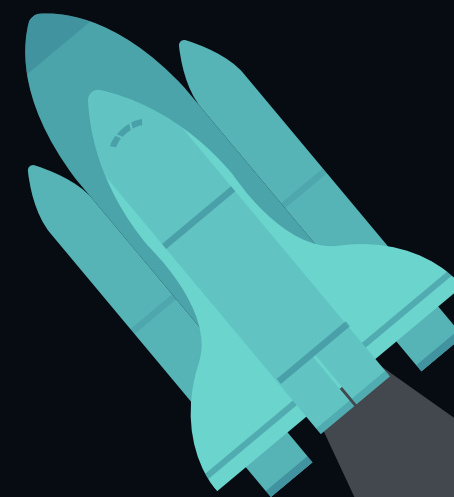
<https://www.timetoast.com/timelines/generaciones-de-los-lenguajes-de-programacion-78279214-40c7-4337-be29-0bcd0e376bd2>



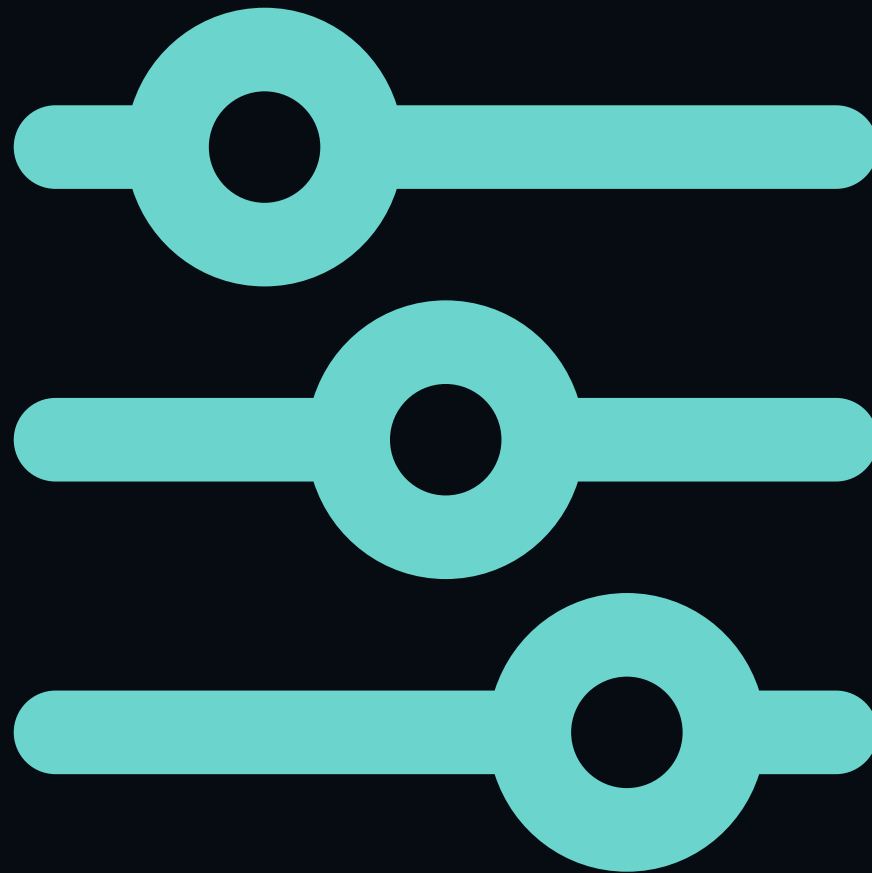


¿DUDAS?

¿DUDAS?

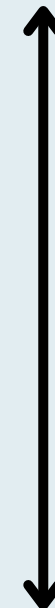


Clasificación por Niveles



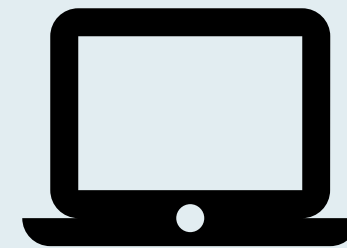
ALTO NIVEL

Los que son más entendibles para el ser humano, como C, C++, Java



BAJO NIVEL

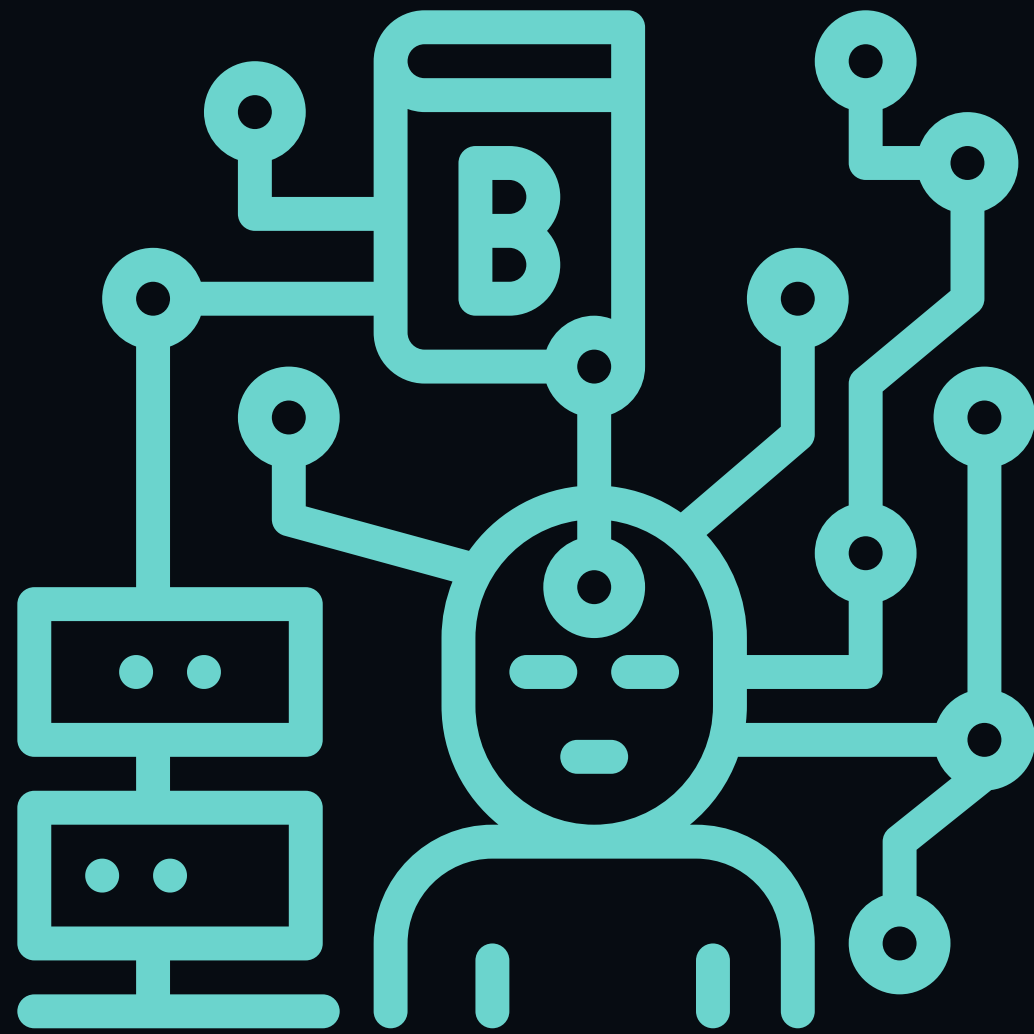
Ensamblador



LENGUAJE MÁQUINA

1 y 0

Paradigmas de los Lenguajes



PARADIGMA

"Teoría o conjunto de teorías cuyo núcleo central se acepta sin cuestionar y que suministra la base de model para resolver problemas y avanzar en el conocimiento".

PARADIGMA IMPERATIVO

PARADIGMA DECLARATIVO

Paradigma Imperativo

Forma más usada y más antigua

Se basa en dar instrucciones al ordenador de como hacer las cosas en forma de algoritmos, en lugar de describir el problema o solución.

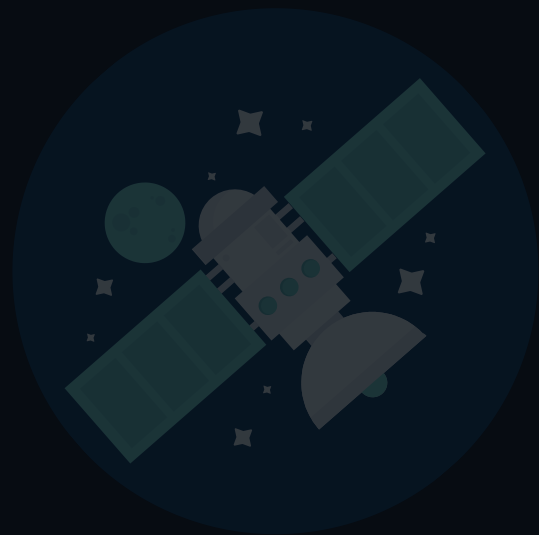


Paradigma Declarativo

No ¿cómo?, sino ¿Qué?

Esta basado en describir el problema declarando propiedades y reglas que deben cumplirse, en lugar de instrucciones.



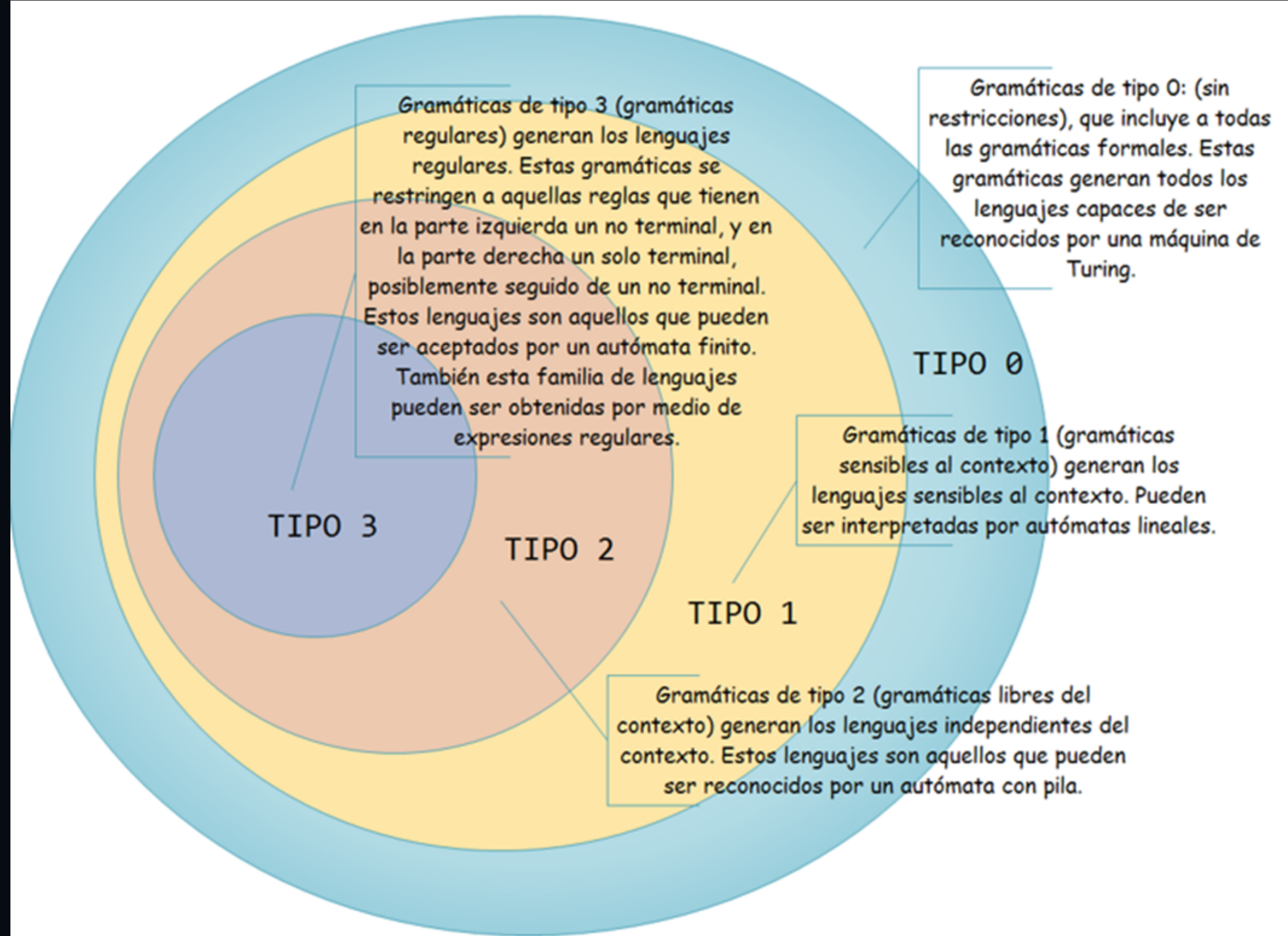


¿DUDAS?

¿DUDAS?



Jerarquía de Chomsky



Lenguajes Regulares

Contienen "regularidades" o repeticiones de los mismos componentes, como por ejemplo en el lenguaje siguiente:

$L1 = \{ AB, ABAB, ABABCCC, ABABABAB \}$

¿Cuál es la regularidad aquí?

Definición

Un lenguaje L es regular si y sólo si se cumple al menos una de las condiciones siguientes:



" L " es finito.

" L " es la unión o la concatenación de otros lenguajes regulares.

" L " es la cerradura de Kleene de algún lenguaje regular, $L = R^*$.

Los lenguajes
regulares son
generados por
una gramática
regular



Un conjunto de símbolos
terminales (variables)



Un conjunto de símbolos
no terminales
(constantes)



Un conjunto de **reglas**

Un símbolo inicial

Los lenguajes
regulares son
generados por
una gramática
regular

Las reglas se restringen a un único "no terminal" en la parte izquierda y una parte derecha compuesta por un único "terminal" que puede estar seguido o no de un único "no terminal".

Es decir, normas del tipo:

- $A \rightarrow aB$
- $A \rightarrow a$

DERIVACIONES



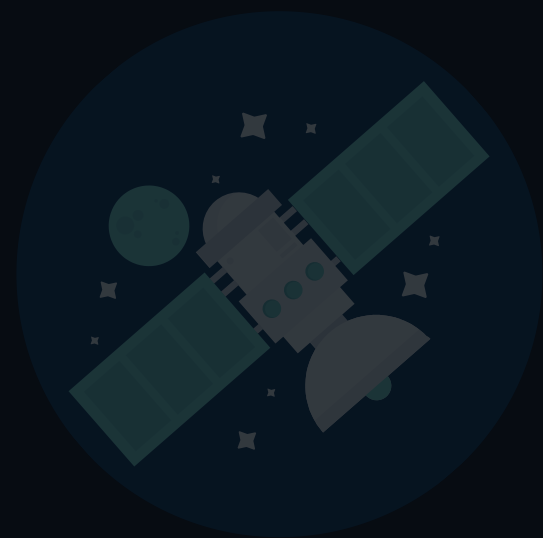
Izquierda

- $A \rightarrow a$
- $A \rightarrow aB$
- $B \rightarrow b$



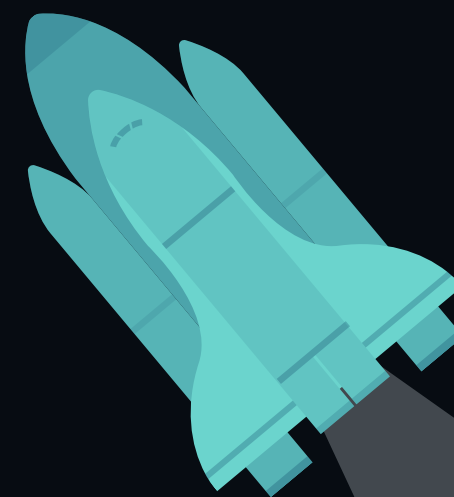
Derecha

$A \rightarrow a$
 $A \rightarrow Ba$
 $B \rightarrow b$



¿DUDAS?

¿DUDAS?



Python

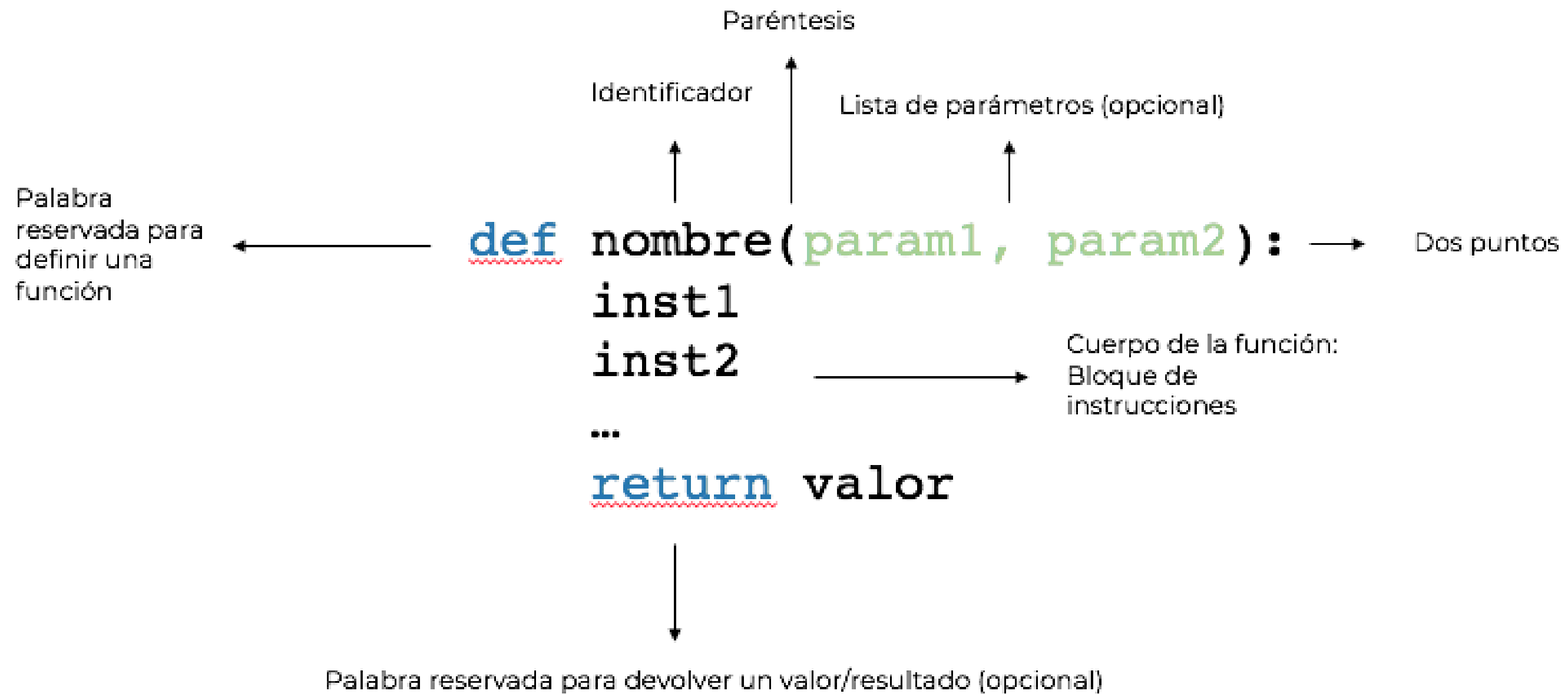


Funciones

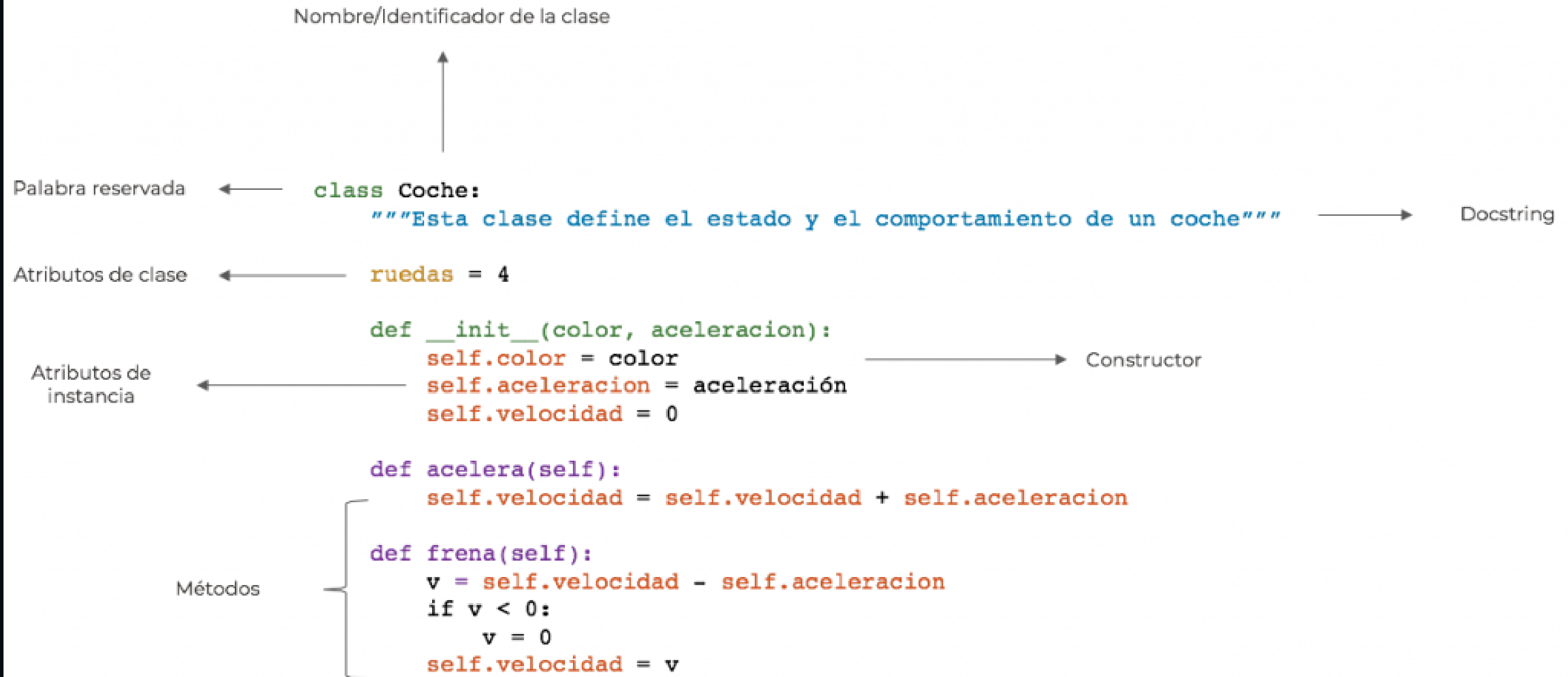


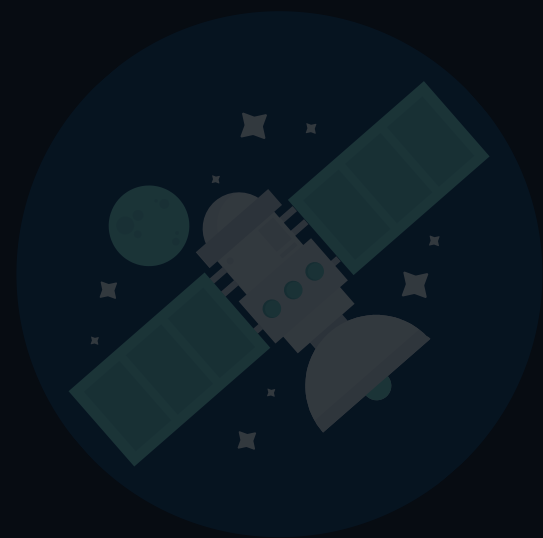
Clases

Funciones



Clases





¿DUDAS?

¿DUDAS?



Gracias por asistir

¿Dudas o preguntas?



LABORATORIO LENGUAJES FORMALES Y DE
PROGRAMACION