

# LENGUAJES DE PROGRAMACIÓN.

Olga María Fimbres Morales

## INTRODUCCIÓN.

Existen distintos lenguajes de programación, y aunque todos tienen básicamente la misma función, resulta más conveniente elegir uno entre ellos dependiendo de la tarea que deseemos realizar. Pueden ser muy similares entre ellos, pero cada uno posee características especiales que ningún otro tiene. Es por eso, que a pesar de poder realizar una misma tarea en diversos lenguajes, cada uno de ellos será diferente y la forma en que trabajemos con el lenguaje también debe de cambiar. Por ejemplo, pueden compartir instrucciones, pero unos necesitan mayores especificaciones que otros para llevarlas a cabo.

## TABLA COMPARATIVA.

Nombre	Paradigma	Creadores	Año de aparición	Extensiones de archivo	Ejemplo de compilación/interpretación del archivo fuente
ANSI C	Imperativo, estructurado	Dennis M. Ritchie, Bell Labs	1972	.h .c	gcc archivo.c -o comando
C++	Multiparadigma, imperativo, programación genérica	Bjarne Stroustrup, Bell Labs	1983	.h .hh .hpp .hxx .h++ .cc .cpp .cxx .c++	g++ -o comando.archivo.cpp
Fortran 90	Imperativo, procedimental	John W. Backus	1991	.f90	gfortran archivo.f90 -o comando
Java	Imperativo, orientado a objetos	James Gosling, Sun Microsystems	1995	.java .class .jar .jad	javac archivo.java
Python	Multiparadigma: imperativo, funcional, reflexivo	Python Software Foundation	1991	.py .pyc .pyd .pyo .pyw	python archivo.py
Ruby	Multiparadigma, reflexivo	Comunidad de desarrolladores de Ruby	1995	.rb rbw	ruby archivo.rb

## EJEMPLO DE UN PROGRAMA.

Programa: Adivina el resultado de las operaciones mentales al haber escogido un número previamente y de común acuerdo con el usuario.

### ANSI C

```
#include <stdio.h>
#include <stdlib.h>

int main(void)
{
    puts("¡Hola! Trataré de adivinar un número.");
    puts("Piensa un número entre 1 y 10.");
    sleep (5);
    puts("Ahora multiplícalo por 9.");
    sleep (5);
    puts("Si el número tiene 2 dígitos, súmalos entre si: Ej. 36 -> 3+6=9.");
    puts("Si tu número tiene un solo dígito, súmale 0.");
    sleep(5);
    puts("Al número resultante súmale 4.");
    sleep(10);
    puts("Muy bien. El resultado es 13 :)");
    return EXIT_SUCCESS;
}
```

### C++

```
#include <iostream>
#include <unistd.h>

main()
{
    std::cout << "¡ Hola! Trataré de adivinar un número." << std::endl;
    std::cout << "Piensa un número entre 1 y 10." << std::endl;
    sleep(5);
    std::cout << "Ahora multiplícalo por 9." << std::endl;
    sleep(5);
}
```

```

std::cout << "Si el número tiene 2 dígitos, súmalos entre si:
Ej. 36 -> 3+6=9." << std::endl;
std::cout << "Si tu número tiene un solo dígito, súmale 0." << std::endl;
sleep(5);
std::cout << "Al número resultante súmale 4." << std::endl;
sleep(10);
std::cout << "Muy bien. El resultado es 13 :)" << std::endl;
return 0;
}

```

## Fortran 90

```

PROGRAM Adivina
  WRITE (*,*) "Hola! Trataré de adivinar un número."
  WRITE (*,*) "Piensa un número entre 1 y 10."
  CALL sleep(5)
  WRITE (*,*) "Ahora multiplícalo por 9."
  CALL sleep(5)
  WRITE (*,*) "Si el número tiene 2 dígitos, súmalos entre si:
Ej. 36 -> 3+6=9."
  WRITE (*,*) "Si tu número tiene un solo dígito, súmale 0."
  CALL sleep(5)
  WRITE (*,*) "Al número resultante súmale 4."
  CALL sleep(10)
  WRITE (*,*) "Muy bien. El resultado es 13 :)"
END PROGRAM

```

## Java

```

class Adivina {
  static public void main( String args[] ) {
    System.out.println("!Hola! Trataré de adivinar un número." );
    System.out.println("Piensa un número entre 1 y 10.");
    try {
      Thread.sleep(5000);
    } catch (InterruptedException ex) {
      Thread.currentThread().interrupt();
    }
    System.out.println("Ahora multiplícalo por 9.");
    try {
      Thread.sleep(5000);
    } catch (InterruptedException ex) {
      Thread.currentThread().interrupt();
    }
  }
}

```

```

    System.out.println("Si el número tiene 2 dígitos, súmalos entre si:
    Ej. 36 -> 3+6=9. Si tu número tiene un solo dígito, súmale 0.");
try {
    Thread.sleep(5000);
} catch (InterruptedException ex) {
    Thread.currentThread().interrupt();
}
    System.out.println("Al número resultante súmale 4.");
try {
    Thread.sleep(10000);
} catch (InterruptedException ex) {
    Thread.currentThread().interrupt();
}
    System.out.println("Muy bien. El resultado es 13 :)");
}
}

```

## Python

```

# -*- coding: utf-8 -*-
#!/usr/bin/python
import time

```

```

print "Hola! Trataré de adivinar un número"
print "Piensa un número entre 1 y 10."
time.sleep( 5 )
print "Ahora multiplícalo por 9."
time.sleep( 5 )
print "Si el número tiene 2 dígitos, súmalos entre si: Ej. 36 -> 3+6=9.
Si tu número tiene un solo dígito, súmale 0."
time.sleep( 5 )
print "Al número resultante súmale 4."
time.sleep( 10 )
print "Muy bien. El resultado es 13 :)"

```

## Ruby

```

# -*- coding: utf-8 -*-
puts "Hola! Trataré de adivinar un número."
puts "Piensa un número entre 1 y 10."
sleep(5)

```

```
puts "Ahora multiplícalo por 9."
sleep(5)
puts "Si el número tiene 2 dígitos, súmalos entre si:
Ej. 36 -> 3+6=9. Si tu número tiene un solo dígito, súmale 0."
sleep(5)
puts "Al número resultante súmale 4."
sleep(10)
puts "Muy bien. El resultado es 13 :)"
```