

# CURSO DE FUNDAMENTOS DE PROGRAMACIÓN DESDE CERO



# ¿QUÉ ES UN PARADIGMA DE PROGRAMACIÓN?





# QUÉ ES UN PARADIGMA DE PROGRAMACIÓN

Un paradigma de programación es una forma de pensar y organizar la lógica de un programa.





# QUÉ ES UN PARADIGMA DE PROGRAMACIÓN

Es un modelo o enfoque que define cómo se debe estructurar el código, cómo se resuelven los problemas y qué tipo de pasos sigue el programador para lograrlo.





# QUÉ ES UN PARADIGMA DE PROGRAMACIÓN

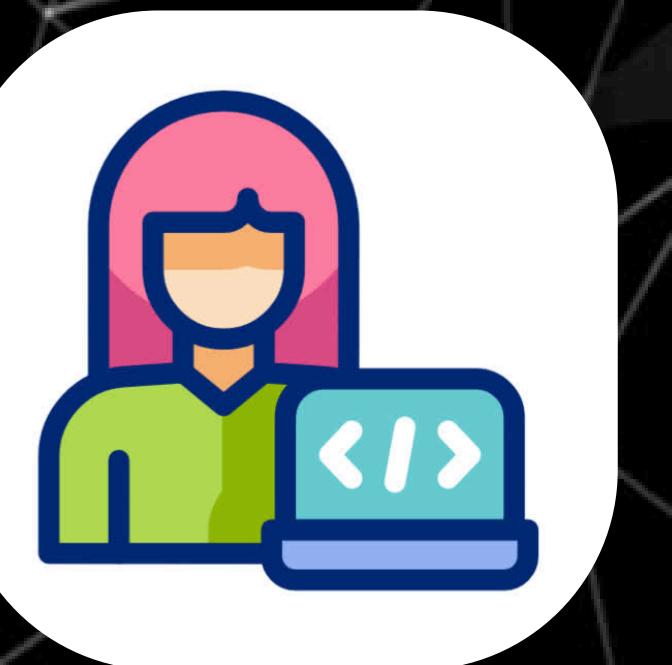
Un paradigma es como un “estilo de pensamiento” que guía cómo se diseña un programa.





# QUÉ ES UN PARADIGMA DE PROGRAMACIÓN

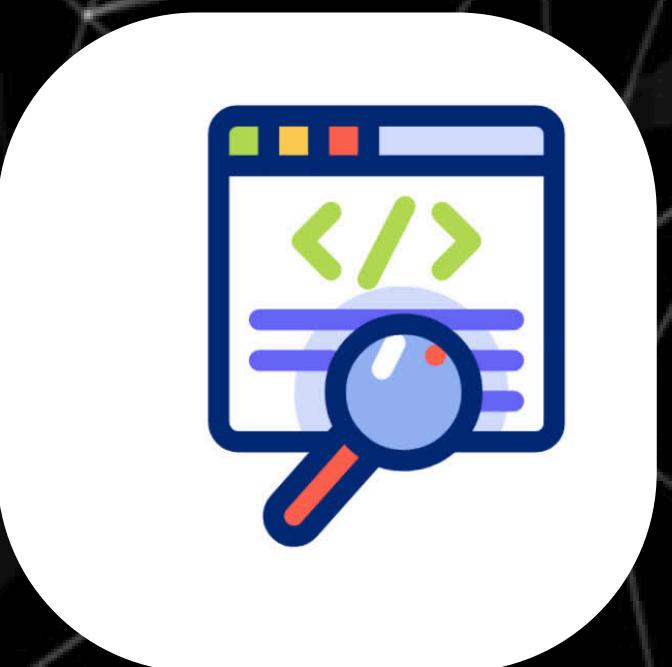
Ejemplo: Como un arquitecto puede elegir distintos estilos para construir una casa, un programador elige un paradigma para construir su software.





# 1. POR QUÉ EXISTEN LOS PARADIGMAS

En programación no hay una sola manera de resolver un problema. A lo largo del tiempo se han desarrollado distintos enfoques o métodos para trabajar.





# 1. POR QUÉ EXISTEN LOS PARADIGMAS

En base a:

El tipo de problema a resolver,

La forma de pensar del programador, y

Las características del lenguaje de programación.



# 1. POR QUÉ EXISTEN LOS PARADIGMAS

Cada paradigma ofrece una forma diferente de organizar el pensamiento lógico y estructurar las instrucciones.





## 2. QUÉ REPRESENTA UN PARADIGMA

Un paradigma define:

- Cómo se comunican las partes del programa.
- Cómo se manejan los datos y las acciones.



## 2. QUÉ REPRESENTA UN PARADIGMA

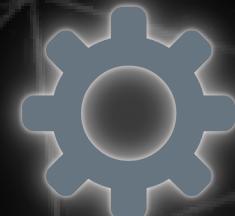
- Cómo se divide un problema (en pasos, en funciones, en objetos, etc.).
- Qué tipo de estructuras lógicas se usan (secuencias, funciones, objetos, etc.).



## 2. QUÉ REPRESENTA UN PARADIGMA

En resumen: El paradigma no es el lenguaje, sino la forma de pensar al usarlo.

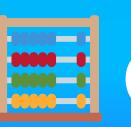




# 3. LOS PARADIGMAS DE PROGRAMACIÓN MÁS COMUNES

A continuación, los principales tipos de paradigmas que existen, explicados de forma teórica y comparativa

# 3. LOS PARADIGMAS DE PROGRAMACIÓN MÁS COMUNES



a) Paradigma imperativo



b) Paradigma estructurado



c) Paradigma orientado a objetos

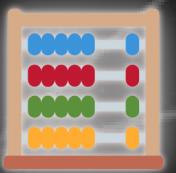


d) Paradigma funcional



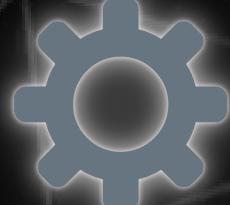
f) Paradigma declarativo

# 3. LOS PARADIGMAS DE PROGRAMACIÓN MÁS COMUNES



## a) Paradigma imperativo

- Es el más antiguo y directo.
- Se basa en dar instrucciones paso a paso para modificar el estado del programa.
- El programador controla explícitamente qué hacer y en qué orden.



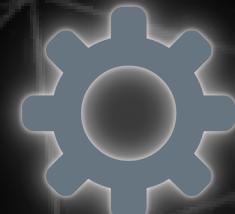
# 3. LOS PARADIGMAS DE PROGRAMACIÓN MÁS COMUNES



Ejemplo mental:

“Primero haz esto, luego aquello,  
y después lo otro.”

Es como seguir una receta, donde cada paso debe cumplirse  
en orden.



# 3. LOS PARADIGMAS DE PROGRAMACIÓN MÁS COMUNES

Información:

Enfoque: Secuencial y detallado.

Ventaja: Fácil de entender para principiantes.

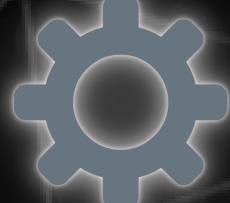
Ejemplo de lenguajes: C, Pascal, BASIC.

# 3. LOS PARADIGMAS DE PROGRAMACIÓN MÁS COMUNES



## b) Paradigma estructurado

- Es una evolución del imperativo.
- Propone dividir el programa en bloques lógicos o estructuras (decisiones, repeticiones, funciones).
- Promueve el uso de una lógica más ordenada y clara.

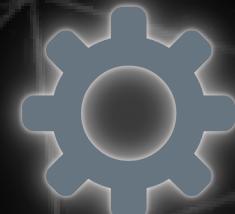


# 3. LOS PARADIGMAS DE PROGRAMACIÓN MÁS COMUNES



Ejemplo mental:

Resolver un problema dividiéndolo en partes pequeñas y controladas:  
inicio → decisión → acción → fin.



# 3. LOS PARADIGMAS DE PROGRAMACIÓN MÁS COMUNES

Información:

Enfoque: Modular y organizado.

Ventaja: Mejora la claridad y reduce errores.

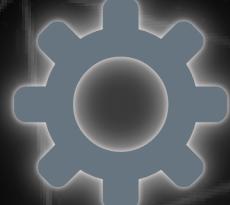
Ejemplo de lenguajes: C, Ada, Fortran moderno.

# 3. LOS PARADIGMAS DE PROGRAMACIÓN MÁS COMUNES



## c) Paradigma orientado a objetos

- Favorece la reutilización y organización del código.
- Cada objeto tiene atributos (datos) y comportamientos (acciones).
- Se centra en modelar el problema a través de “objetos” que representan cosas reales o abstractas.

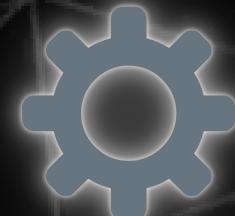


# 3. LOS PARADIGMAS DE PROGRAMACIÓN MÁS COMUNES



Ejemplo mental:

“Un auto tiene características (color, marca) y comportamientos (acelerar, frenar).”  
Cada auto es un objeto dentro del programa.



# 3. LOS PARADIGMAS DE PROGRAMACIÓN MÁS COMUNES

Información:

Enfoque: Basado en entidades que interactúan.

Ventaja: Permite crear programas más complejos.

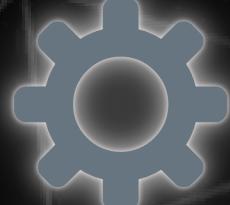
Ejemplo de lenguajes: Java, Python, C++, C#, Ruby.

# 3. LOS PARADIGMAS DE PROGRAMACIÓN MÁS COMUNES



## d) Paradigma funcional

- No se enfoca en instrucciones, sino en transformar datos.
- Se basa en el uso de funciones puras y en evitar cambiar datos.
- El programador describe qué quiere obtener, no cómo hacerlo paso a paso.

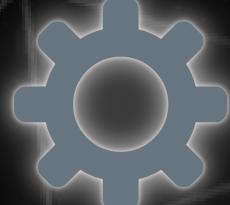


# 3. LOS PARADIGMAS DE PROGRAMACIÓN MÁS COMUNES



Ejemplo mental:

“Quiero una lista de números duplicados.”  
(Sin especificar cómo recorrer la lista, el programa lo hace de forma interna.)



# 3. LOS PARADIGMAS DE PROGRAMACIÓN MÁS COMUNES

Información:

Enfoque: Declarativo y matemático.

Ventaja: Reduce errores y facilita el trabajo de datos.

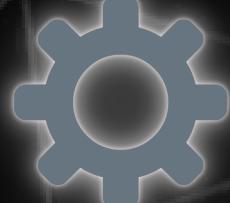
Ejemplo de lenguajes: Haskell, Lisp, Scala y Python.

# 3. LOS PARADIGMAS DE PROGRAMACIÓN MÁS COMUNES



## e) Paradigma lógico

- Se basa en reglas y relaciones lógicas entre hechos.
- El programador describe las condiciones que deben cumplirse, y el programa deduce la respuesta.

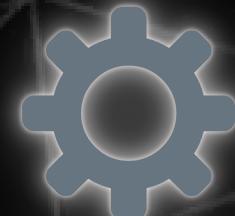


# 3. LOS PARADIGMAS DE PROGRAMACIÓN MÁS COMUNES



Ejemplo mental:

“Si A es padre de B y B es padre de C, entonces A es abuelo de C.”



# 3. LOS PARADIGMAS DE PROGRAMACIÓN MÁS COMUNES

Información:

Enfoque: Basado en la inferencia y la deducción lógica.

Ventaja: Util para sistemas de inteligencia artificial y razonamiento automático.

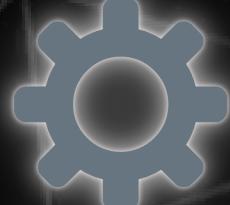
Ejemplo de lenguajes: Prolog.

# 3. LOS PARADIGMAS DE PROGRAMACIÓN MÁS COMUNES



## f) Paradigma declarativo

- El programador indica qué quiere lograr, pero no los pasos exactos para hacerlo.
- El sistema decide cómo obtener el resultado.

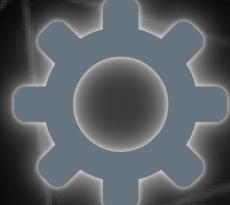


# 3. LOS PARADIGMAS DE PROGRAMACIÓN MÁS COMUNES



Ejemplo mental:

En lugar de decir “haz esto paso a paso”, se dice “quiero este resultado final”.



# 3. LOS PARADIGMAS DE PROGRAMACIÓN MÁS COMUNES

Información:

Enfoque: Orientado a resultados, no a procedimientos.

Ejemplo de lenguajes: SQL (para bases de datos), HTML (para estructura web).



## 4. POR QUÉ SON IMPORTANTES LOS PARADIGMAS

Ayudan a organizar el pensamiento lógico.

Facilitan la comprensión y mantenimiento de los programas.

Permiten elegir el enfoque según el tipo de problema.

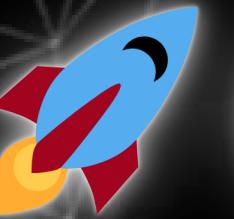
Fomentan buenas prácticas en la resolución de tareas.



## 4. POR QUÉ SON IMPORTANTES LOS PARADIGMAS

Un buen programador no se limita a uno solo: aprende a combinar varios paradigmas según lo que necesite.

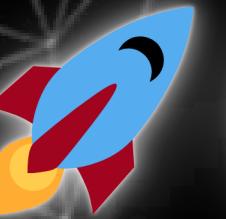




# CONCLUSIÓN

Un paradigma de programación es una forma de pensar y estructurar soluciones dentro del mundo de la programación.

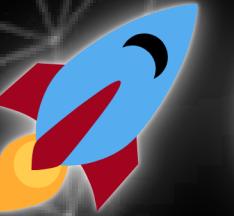




# CONCLUSIÓN

Cada paradigma ofrece una visión distinta de cómo resolver un problema, organizar las instrucciones y manejar los datos.





# CONCLUSIÓN

Información:

Paradigma = estilo de pensamiento lógico para resolver problemas.

No todos los lenguajes usan el mismo paradigma.

Un programador aprende a elegir y combinar el paradigma más adecuado.

**NOS VEMOS EN UN PRÓXIMO  
VIDEO DE ESTE CURSO,  
SALUDOS**

