Paradigmas de programación

DigitalHouse>

Certified Tech Developer
The Ultimate Degree

Índice

1 Paradigmas de programación

¿Qué son los paradigmas de programación?

Diferentes estilos de usar la programación para resolver un problema.

Programación Imperativa

Los programas consisten en una sucesión de instrucciones o conjunto de sentencias, paso a paso. Dentro del paradigma imperativo encontramos estos enfoques:

Programación estructurada

La programación estructurada es un tipo de programación imperativa donde el flujo de control se define mediante bucles anidados, condicionales y subrutinas, en lugar de a través de GO TO.

Programación procedimental

Este paradigma de programación consiste en basarse en un número muy bajo de expresiones repetidas, englobadas todas en un procedimiento o función y llamarlo cada vez que tenga que ejecutarse.

Programación modular

Consiste en dividir un programa en módulos o subprogramas con el fin de hacerlo más manejable y legible. Se trata de una evolución de la programación estructurada para resolver problemas de programación más complejos.

Programación Declarativa

Este paradigma no necesita definir algoritmos puesto que describe el problema en lugar de encontrar una solución al mismo. Así, utiliza el principio del razonamiento lógico para responder a las preguntas o cuestiones consultadas.

Se divide en dos:

Programación lógica	Programación funcional
Un programa puede ser descripto definiendo ciertas relaciones entre conjuntos de objetos, a partir de las cuales otras pueden ser calculadas empleando deducción.	Se basa en la evaluación de expresiones, no en la ejecución de instrucciones.

Programación Orientada a Objetos

Se construyen modelos de objetos que representan elementos (objetos) del problema a resolver, que tienen características y funciones. Permite separar los diferentes componentes de un programa. Se acerca de alguna manera a cómo expresaríamos las cosas en la vida real.

Programación Reactiva

Se basa en escuchar lo que emite un evento o cambios en el flujo de datos, en donde los objetos reaccionan a los valores que reciben de dicho cambio.

Los sistemas reactivos deben ser:

- **Responsivos:** aseguran la calidad del servicio cumpliendo unos tiempos de respuesta establecidos.
- **Resilientes:** se mantienen responsivos incluso cuando se enfrentan a situaciones de error.
- **Elásticos:** se mantienen responsivos incluso ante aumentos en la carga de trabajos.
- **Orientados a mensajes**: minimizan el acoplamiento entre componentes al establecer interacciones basadas en el intercambio de mensajes de manera asíncrona.

2 Comunicación y lenguaje

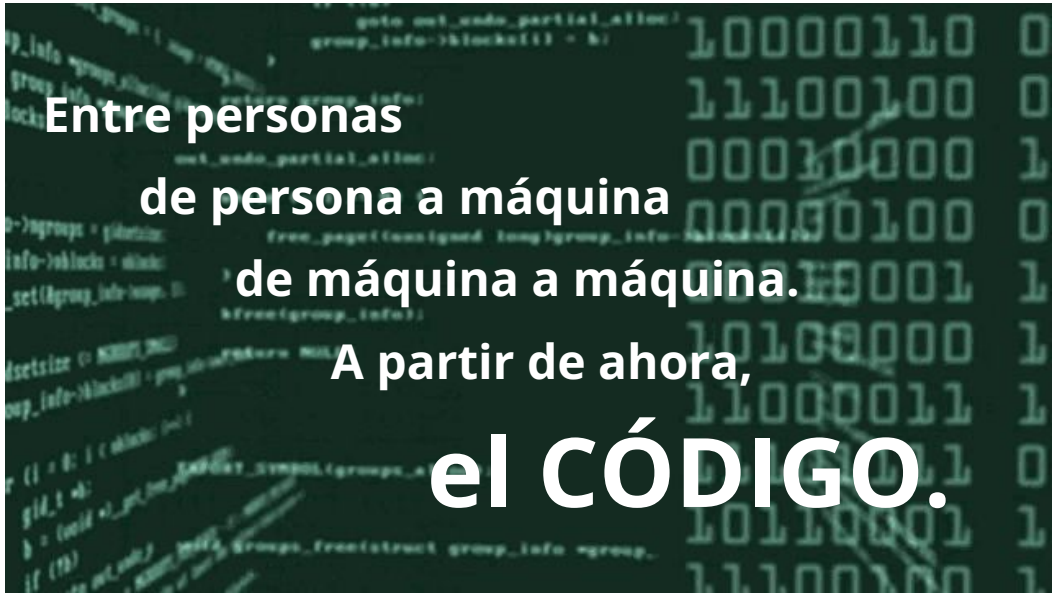
Comunicación

“

Transmisión de señales de un emisor y a un receptor mediante un código común (el lenguaje).

”

El lenguaje



¡Uf! ¿Y eso? ¿No hay más sencillo?

3 Órdenes mediante símbolos (lightbot)

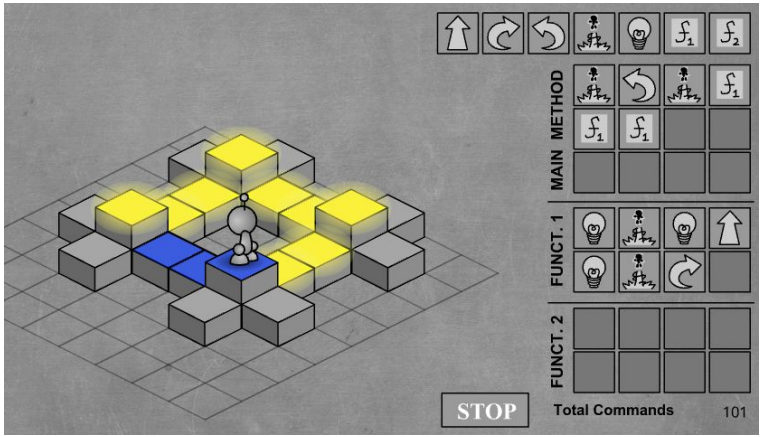
Un símbolo carga un significado.
Una computadora es una máquina que los interpreta, gracias a los significados puestos por el humano.

Contienen un significado diseñado por la persona que programó el juego.

Símbolos primitivos

F1 y F2 (Funciones 1 y 2)

Símbolos primitivos que tendrán el significado diseñado por nosotros, la persona que programa el comportamiento del personaje.



Programa principal

Acá determinamos cuál será el comportamiento del personaje.

Funciones o procedimientos

Acá determinamos cuál será el comportamiento de nuestras funciones y, por lo tanto, del personaje.

Sintaxis: modo estructurado de combinar los símbolos y de asignarles significados.

4 Lenguaje y ambigüedad

El texto es la evolución del símbolo. Una combinación de estos símbolos amplía las posibilidades. Podemos ahora codificar un mensaje en este lenguaje textual.

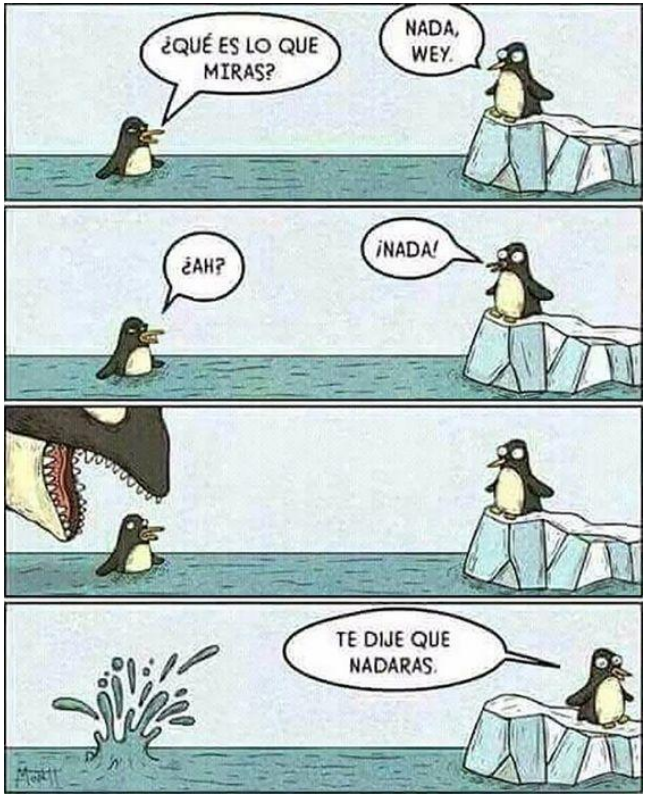
Lo textual en lo cotidiano

Cuando alguien dice *“¡a comer!, siéntense en la mesa”* no quiere que se sienten “**en**” la mesa, sino en las sillas que hay alrededor.

“Tomate un taxi”, “*el nene no me come”*, entre otras frases, invitan a una doble interpretación.

El cerebro humano puede interpretar estas ambigüedades, resignificarlas hasta que tengan sentido dentro del contexto —cena, estar en la calle, charlando con el dr.—.

Las computadoras —por el momento— no pueden hacer eso. Son máquinas muy complejas que necesitan reglas simples. No debe haber ambigüedad.



DigitalHouse>