



UAI

**Universidad
Abierta
Interamericana**

“Paradigmas de programación”

Trabajo Práctico N° 11

Alumno: Gianluca Carlini

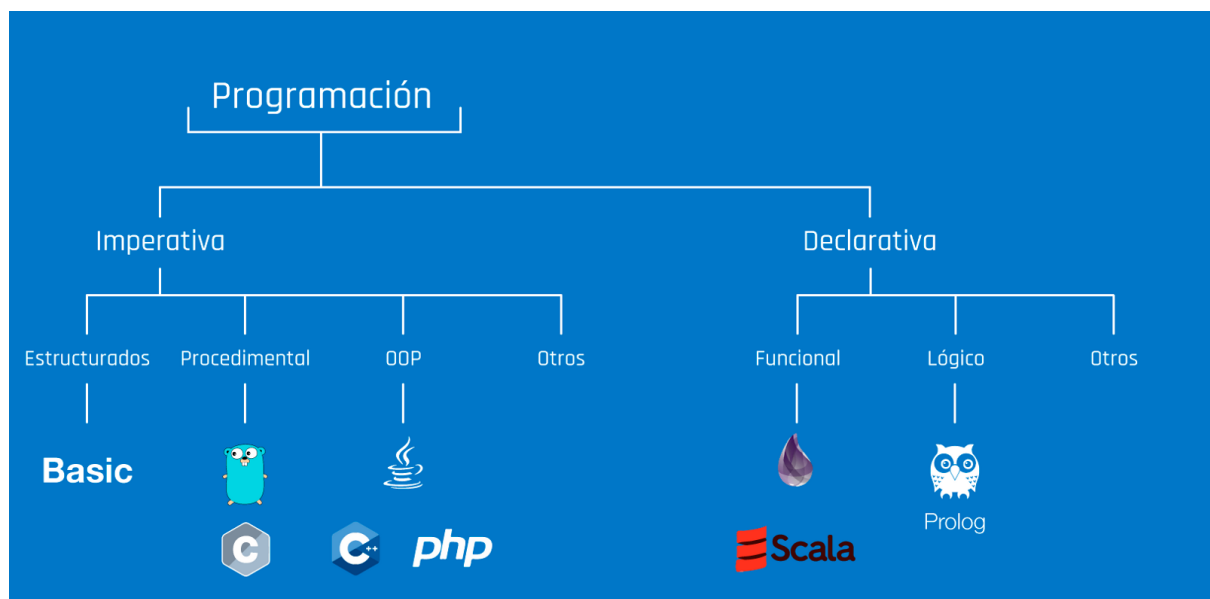
Año: 2022

Para comprender los distintos paradigmas de la programación, primero debemos saber qué significan y/o a qué hacen referencia. Entonces, **¿qué son los paradigmas de programación?** En pocas palabras, son las diferentes formas en las que se puede desarrollar un software. Además, hacen referencia a los diversos enfoques sistemáticos que pueden ser aplicados en todos los niveles del diseño de programas.

En el área de tecnología, un paradigma de programación determina el método de un desarrollador en la construcción de un sistema. Éste elegirá el paradigma que más conveniente sea para realizar ese sistema.

Si bien existen diferentes paradigmas que resultan en diferentes estilos de programación y en diferentes formas de pensar una solución, todos tienen en común los siguientes métodos:

1. **Modularidad:** Es la descomposición lógica de un sistema en entidades más pequeñas. Al descomponer al sistema en unidades más simples, es más manejable. Éstos módulos poseen un grado de cohesión y acoplamiento entre sí.
2. **Recursividad:** Significa aplicar una función como parte de la definición de esa misma función. Está altamente ligado al concepto de *repetición*. Un ejemplo de recursividad es el 'cálculo del factorial'.



Los paradigmas de programación se clasifican en dos:

- **Imperativa:** el programador instruye a la máquina cómo cambiar su estado.

Ventajas y desventajas de los lenguajes de programación imperativa:

Ventajas	Desventajas
Fácilmente legible.	El código se convierte rápidamente en demasiado amplio y difícil de abarcar.
Fácil de aprender en lo relativo a comportamientos.	Mayor riesgo durante la edición.
Modelo fácilmente comprensible para los principiantes.	El mantenimiento bloquea el desarrollo de la aplicación, ya que la programación funciona estrechamente con el sistema.
Se pueden tener en cuenta características de casos especiales de la aplicación	La optimización y la ampliación son más difíciles.

- **Declarativa:** el programador le indicará a la máquina lo que quiere hacer y el resultado que desea, más no el cómo hacerlo.

Ventajas y desventajas de los lenguajes de programación declarativa:

Ventajas	Desventajas
Código más corto y eficiente.	Difícil de comprender para personas ajenas.
Realizable con métodos no conocidos en el momento de la programación.	Basado en una forma de pensar no habitual en las personas.
Optimización sencilla.	Las características de casos de aplicación individuales se pueden considerar en la programación, pero únicamente de forma compleja.
Mantenimiento independiente del desarrollo de la aplicación.	

Paradigma Lógico

Es un paradigma de programación basado en la *Lógica*. Los programas construidos en lenguaje lógico están únicamente contruidos por expresiones lógicas.

Éste paradigma es el que comúnmente se utiliza para la *Inteligencia Artificial*. Éste, se basa en sentencias sobre las que debe demostrarse su veracidad mediante la búsqueda en una enorme base de datos.

Sus características son:

- La sintaxis de los lenguajes de programación lógicos es notablemente diferente de los lenguajes de programación imperativos.
- Unificación de términos.
- Mecanismos de inferencia automática.
- Recursión como estructura de control básica.
- Visión lógica de la computación.
- No tiene un algoritmo que indique los pasos que detallen la manera de llegar a un resultado.
- Las salidas son funcionalmente dependientes de las entradas.

Ya para finalizar, el lenguaje más conocido de este paradigma es el **Prolog**.

Paradigma Funcional

La programación funcional se centra en las **funciones**. En un programa funcional, todos los elementos pueden entenderse como funciones y el código puede ejecutarse mediante

llamadas de función secuenciales. Por el contrario, no se asignan valores de forma independiente.

Lo que hace que la estrategia funcional sea tan importante para la informática y a la vez tan **versátil** es el hecho de que las funciones dentro de este paradigma de programación pueden adoptar diferentes “formas”: estas pueden **enlazarse entre sí** como los datos y **utilizarse como parámetro y como resultado de la función**.

Entre los lenguajes de programación funcional más importantes encontramos:

1. Haskell
2. Scala
3. LISP
4. Clojure

Paradigma Estructurado

Este paradigma está orientado a mejorar la claridad, calidad y tiempo de desarrollo utilizando únicamente subrutinas y/o funciones. Está basada en el teorema del programa estructurado propuesto por Böhm y Jacopini.

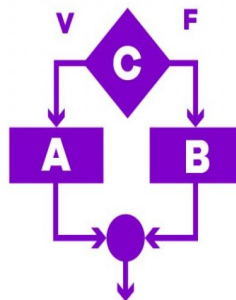
Además, utiliza 3 estructuras básicas:

1. **Secuencia:** La estructura secuencial es la que se da de forma natural en el lenguaje, porque las sentencias se ejecutan en el orden en el que aparecen en el programa, es decir, una detrás de la otra.
2. **Condicional:** La estructura condicional se basa en que una sentencia se ejecuta según el valor que se le atribuye a una variable booleana (aquella que tiene dos valores posibles). Por lo tanto, esta estructura se puede ejecutar de dos formas distintas, dependiendo del valor que tenga su variable. Para las estructuras condicionales, se dispone de la **sentencia if**, que puede combinarse con **elif** y/o **else**.
3. **Iteración:** La estructura de repetición ejecuta una o un conjunto de sentencias siempre que una variable booleana sea verdadera. Para los bucles o iteraciones, los lenguajes de programación usan las estructuras **while** y **for**.

Secuencia



Selección o condicional



Iteración (ciclo o bucle)



Entre las ventajas que encontramos en este paradigma es que nos permite reutilizar código programado y tiene una mejor comprensión de la programación.

El lenguaje de programación más utilizado es **Python**.

Paradigma OOP

La programación orientada a objetos es un paradigma de programación donde utiliza objetos y sus interacciones, para diseñar aplicaciones y programas. Este tipo de programación se utiliza para estructurar un programa de software en piezas simples y reutilizables de planos de código (clases) para crear instancias individuales de objetos.

Está basado en 4 pilares fundamentales:

1. **Herencia:** Define las relaciones jerárquicas entre clases, de forma que atributos y métodos comunes puedan ser reutilizados.
2. **Abstracción:** Cuando el usuario interactúa solo con los atributos y métodos seleccionados de un objeto.
3. **Polimorfismo:** Consiste en diseñar objetos para compartir comportamientos.
4. **Encapsulamiento:** Contiene toda la información importante de un objeto dentro del mismo.

En este paradigma, lo que se busca es dejar de pensar en la lógica pura de los programas, para centrarnos en los objetos. Esto es de suma importancia ya que en grandes sistemas dejaríamos de pensar en funciones para pensar en las interacciones de los diferentes componentes del sistema.

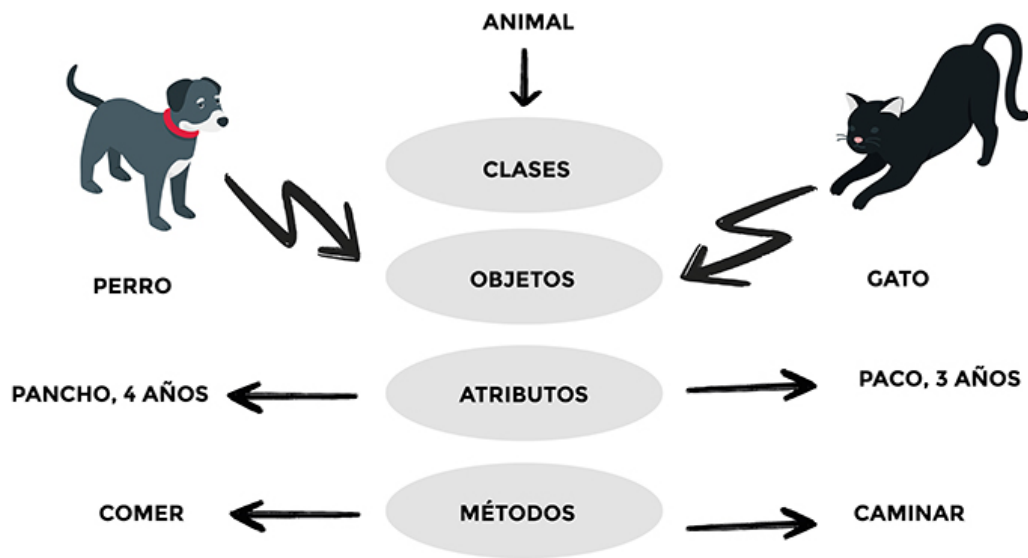
Sus principales ventajas son la reutilización de código, facilidad para pensar soluciones a determinados problemas, conversión de cosas complejas en estructuras simples reproducibles, evita la duplicación de código, protege la información por medio del encapsulamiento, etc.

¿Cómo se crean los programas orientados a objetos?

El primer y más importante concepto de la POO es la **distinción entre clase y objeto**.

Una **clase** es una construcción estática que describe comportamientos comunes y atributos, incluyendo en ella, datos y métodos o funciones.

Un **objeto** es cualquier cosa real o abstracta que posee una estructura que lo define y acciones que lo controlan.



A esta **instancia** de una clase se la llama objeto, quien posee identidad, comportamiento y estado fijo o variable específicas a ese objeto.

Instanciación es la acción por la cual se crean instancias de una clase. Los objetos creados corresponden al tipo de la clase que lo origina. En el ejemplo mostrado, tenemos 2 instancias distintas definidas por la clase 'animal'. Llamar a sus métodos puede tener resultados diferentes. Los **métodos** por defecto son el constructor que inicializa valores y el destructor que libera recursos al finalizar la vida útil de una instancia de una clase creada en memoria.

O sea que un

Objeto

es la

Instancia

de una

Clase

Entre los lenguajes más utilizados en la programación POO se encuentran:

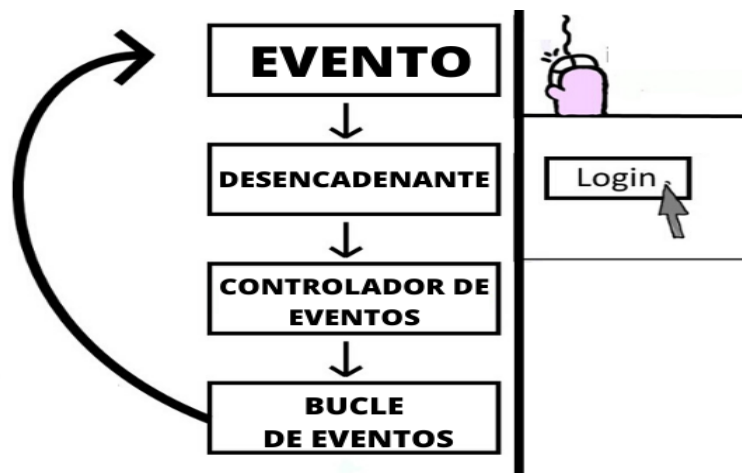
- Java
- C++
- PHP
- JavaScript

Paradigma orientado a Eventos

La programación por eventos es un paradigma de programación en el que tanto la estructura como la ejecución de los programas van determinados por los sucesos que ocurran en el sistema, definidos por el usuario o que ellos mismos provoquen. El programador determinará los eventos a los cuales el programa reacciona y las acciones que seguirá al presentarse cada uno. Esto se conoce como '**manejador de eventos**'.

Este paradigma se basa en 3 pilares:

1. **Evento**
2. **Propiedades**
3. **Métodos**



Pero, ¿qué es un Evento? Un evento es un suceso que ocurre en un sistema. Eventos pueden ser un click, doble click, mover el mouse, mover una tecla especial, etc.

Además, pueden ser divididos en 2 tipos diferentes:

1. Automáticos como al abrir una ventana.
2. Iniciados por el usuario como hacer click en un botón, tocar o arrastrar en la pantalla, etc.

Y, ¿qué es una Propiedad? Es una asignación que describe algo sobre un componente, como por ejemplo: un formulario, un botón de comando, una caja de texto, una etiqueta, una imagen, etc.

Éstas se usan para cambiar la forma de los componentes o controles de la aplicación, como por ejemplo el tamaño de la letra de un control.

Y por último tenemos a los **métodos**, que es una función que es llamada desde el programa, que a diferencia de los procedimientos, estos no son programados por el usuario, sino que ya vienen pre-programados en el lenguaje de programación. Cada tipo de componente tiene sus propios métodos, donde éstos solo pueden ser ejecutados en tiempo de ejecución. Un ejemplo de un método es el método MOVE (en una ventana) que mueve un formulario en un espacio de dos dimensiones en la pantalla.

Vbasic 6 es un buen ejemplo de programación orientada a eventos porque justamente, no tiene herencia (es decir, no se produce la herencia de los atributos ni los métodos de ningún objeto superior).

“Inteligencia Artificial”



La **Inteligencia Artificial (IA)** es la combinación de algoritmos planteados con el propósito de crear máquinas que presenten las mismas capacidades que el ser humano.

Según varios expertos existen varios tipos de IA entre los cuales destacamos:



Sistemas que piensan como humanos

Automatizan actividades como la toma de decisiones, la resolución de problemas y el aprendizaje. Un ejemplo: *redes neuronales artificiales*.



Sistemas que actúan como humanos

Se trata de computadoras que **realizan tareas de forma similar a como lo hacen las personas**. Un ejemplo: *robots*.



Sistemas que piensan racionalmente

Intentan **emular el pensamiento lógico racional de los humanos**.



Sistemas que actúan racionalmente

Son aquellos que tratan de **imitar de manera racional el comportamiento humano**.

PRINCIPALES APLICACIONES DE LA IA



Además, podemos encontrar a la IA en la detección facial de los móviles, en los asistentes virtuales de voz como Siri (Apple), Alexa (Amazon), etc; y está integrada en nuestros dispositivos cotidianos a través de bots o aplicaciones para móviles.

Ventajas IA

- **Automatización de procesos.** Los robots son capaces de reproducir tareas repetitivas y pensar más rápido que los humanos.
- **Minimiza el error humano.** Se reducen los fallos que podemos tener los humanos debido a nuestras limitaciones. Es más, la IA ha sido y es empleada para poder detectar errores que pueden ser indetectables para nuestros ojos. Las máquinas son más precisas.
- **Mantenimiento predictivo.** A través de la IA podemos realizar un mantenimiento predictivo de nuestro equipo industrial, pudiendo incrementar su rendimiento y ciclo de vida.

- **Minimiza el tiempo de análisis de datos.** Permite que este pueda hacerse en tiempo real.
- **Mejora en la toma de decisiones.** Pueden tomar mejores decisiones al disponer de una información mejor estructurada.
- **Aumento de productividad y calidad.** Incrementa tanto la productividad de la maquinaria como la de los propios trabajadores, así como el trabajo realizado.
- **Control y optimización.** Esto se debe a que los procesos que se siguen son más eficientes, con un mayor control y sin apenas errores.
- **Mayor precisión.** Las máquinas pueden llegar a tomar decisiones que anteriormente se tomaban de manera manual o monitorizada, de una forma más precisa.
- **Compatibilidad de sectores.** La IA es aplicable desde el sector sanitario hasta la aviación. Es decir, es aplicable a una gran multitud de sectores.
- **Aplicaciones en tareas complejas.** Es aplicable en labores aplicadas a la minería, exploración de fondos oceánicos u otras que son demasiado complejas para la limitación humana.

Desventajas IA

- **Coste y tiempo de implementación.** La inversión en Inteligencia Artificial es muy elevada al tratarse de máquinas complejas con un alto coste en mantenimiento y reparación.
- **Falta de cualificación.** Para poder implantar proyectos de IA debe contar con profesionales que estén familiarizados con la materia.
- **Ausencia de moral.** Es cierto que la eficacia es mayor, pero no son humanos, carecen de sentimientos, lo que a veces puede ser peligroso ya que no conocen ninguna barrera moral.
- **No existe la creatividad.** Las máquinas no piensan, trabajan bajo parámetros, por lo que la capacidad creativa permanece ausente.

EMPRESAS QUE IMPLEMENTAN LA INTELIGENCIA ARTIFICIAL

APPLE

Apple cuenta con uno de los productos de inteligencia artificial más populares del mercado. Se trata de **Siri**, un **asistente de voz** incluido en los iPhone y en algunos iPad. Las principales funciones de Siri son responder preguntas de los usuarios, realizar acciones a su petición y hacer recomendaciones. Lo consigue utilizando herramientas de **reconocimiento de voz** y de **procesamiento del lenguaje natural**.

FACEBOOK

La red social con más usuarios únicos del mundo dispone de su propio laboratorio de investigación en inteligencia artificial. Su objetivo es servirse de la IA para **ofrecer al usuario una experiencia mucho más personalizada**. Algunas de sus acciones de inteligencia artificial más importantes son el **reconocimiento facial** en el etiquetado de imágenes, los **mapas interactivos** sobre densidad de población y acceso a Internet en el mundo o la **app de Facebook** que emplea redes neuronales e IA para describir el contenido de las imágenes a los usuarios ciegos.

TESLA

La empresa automovilística está trabajando en una **flota de vehículos completamente autónomos**. La introducción de la conducción autónoma es una cuestión que aún genera mucho debate, por lo que corporaciones como Tesla están invirtiendo mucho en investigación para garantizar condiciones de seguridad óptimas.

GOOGLE

Una de las últimas apuestas del gigante Google por la inteligencia artificial se ha materializado en la inversión en una **agencia de noticias escritas por robots**. Sin embargo, la idea de esta agencia no es que los robots actúen como sustitutos o competidores de los periodistas, sino como un recurso más para ellos.