

# Programación visual

## Programación orientada a objetos:

La programación orientada a objetos se basa en el concepto de crear un modelo del problema de destino en sus programas. La programación orientada a objetos disminuye los errores y promueve la reutilización del código. Python es un lenguaje orientado a objetos. Los objetos definidos en Python tienen las características siguientes:

**Identidad.** Cada objeto debe ser distinguido y ello debe poder demostrarse mediante pruebas. Las pruebas `is` y `is not` existen para este fin.

**Estado** Cada objeto debe ser capaz de almacenar el estado. Para este fin, existen atributos, tales como variables de instancias y campos.

**Comportamiento.** Cada objeto debe ser capaz de manipular su estado. Para este fin existen métodos.

Python incluye las características siguientes para dar soporte a la programación orientada a objetos:

**Creación de objetos basada en clases.** Las clases son plantillas para la creación de objetos. Los objetos son estructuras de datos con el comportamiento asociado.

**Herencia con polimorfismo.** Python da soporte a la herencia individual y múltiple. Todos los métodos de instancias de Python son polimórficos y se pueden alterar temporalmente mediante subclasses.

**Encapsulación con ocultación de datos.** Python permite ocultar los atributos. Cuando se ocultan los atributos, se puede acceder a los mismos desde fuera de la clase únicamente mediante los métodos de la clase. Las clases implementan métodos para modificar los datos

## Programación orientada a eventos

La programación orientada a eventos se refiere a un modelo de la programación de computadoras, donde se utilizan los eventos que suceden para la determinación del flujo de control de un programa.

No es un tipo de tecnología o lenguaje de programación, sino un enfoque que se implementa durante la etapa de desarrollo del producto. Básicamente, separa la lógica de procesamiento de eventos del resto del código de un programa.

Este tipo de programación está diseñado para descubrir eventos mientras van sucediendo, usando un procedimiento apropiado de manejo de eventos para tratarlos, normalmente mediante una llamada a una función o método.

Teóricamente, el estilo de esta programación es compatible con todos los lenguajes de programación, aunque puede ser diferente en la forma de implementarse.

### Dependencia de eventos

El flujo del programa viene dado por eventos que pueden ser acciones del usuario, mensajes de otros programas, etc., separando la lógica de procesamiento de los eventos del resto del código de un programa, contrastando así con el procesamiento por lotes.

Los eventos en sí pueden ser desde aceptar o rechazar una solicitud de préstamo, denominado evento de alto nivel, hasta que un usuario presione una tecla, que es un evento de bajo nivel.

### Orientada al servicio

Se utiliza para escribir programas diseñados para el servicio sin ralentizar la computadora, ya que la orientación al servicio solo consume poco poder de procesamiento. Además, los servicios se ejecutan por lo general en el trasfondo del sistema operativo.

### Eventos

Es una condición que surge durante la ejecución de un programa y que requiere alguna acción por parte del sistema. Cada evento es diferente por naturaleza, algunos requieren que el programa recobre y muestre cierta información, y otros que se inicien algunos cálculos y cambios de estado.

## Procesos de desarrollo visual en proyectos distribuidos y de escritorio

Caracterización de los Sistemas Distribuidos. Un sistema distribuido es aquel en el cual los componentes, localizados en equipos en red, se comunican y coordinan sus acciones mediante el envío de mensajes. Esta definición presenta las siguientes características significativas de los sistemas distribuidos: concurrencia de los componentes falta de un reloj global fallos independientes de los componentes. Los sistemas distribuidos presentan una serie de retos: Heterogeneidad, si hablar de heterogeneidad nos referimos a la variedad y diferencia que podemos encontrar en los elementos que componen una red de computadoras sobre la que se ejecuta un sistema distribuido. Dicha heterogeneidad no sólo se aplica a las redes y al hardware de las computadoras, sino también a los sistemas operativos, los lenguajes de programación y las implementaciones en las que trabajan los diferentes desarrolladores.

## Herramientas y lenguajes de programación visual

Algunos ejemplos de estas herramientas son: Scratch, adecuada a partir de los 8 años, ScratchJr, que es la versión de Scratch adaptada para niñas y niños de entre 5 y 7 años, Lightbot, etc. Incluso la empresa Google ha creado un precioso doodle llamado Coding for Carrots que ejemplifica este tipo de herramientas.

Estas herramientas se usan con éxito en todos los niveles educativos formales (Infantil, Primaria, Secundaria, Bachillerato y Universidad), y también en escenarios educativos no formales. Se pueden usar en una gran diversidad de materias como lengua española, lengua extranjera, matemáticas, ciencias, historia, artes, etc

## Scratch

El lenguaje de programación visual más popular y utilizado es Scratch. Fue lanzado por primera vez en 2007 y tiene como propósito enseñar a los niños a programar de manera sencilla. Su sucesor para niños de entre 5 y 7 años, ScratchJr, está disponible como una aplicación gratuita. También hay otros lenguajes de programación gráfica basados en Scratch, como BYOB 4.0/Snap!.

## Blockly

Blockly es una llamada biblioteca. Proporciona un editor de programación visual al que se añaden aplicaciones Android, iOS y web. Blockly también utiliza bloques gráficos que encajan entre ellos. Los algoritmos programados se pueden exportar como código JavaScript o Python, por ejemplo.

## NEPO

NEPO es gratuito, está basado en Scratch y utiliza la biblioteca Blockly. Esta biblioteca se ha ampliado con funcionalidades propias. Una ventaja de NEPO son sus interfaces abiertas, que permiten controlar otros sistemas de hardware o robots.

## Grape

Grape es un entorno de desarrollo gráfico. Permite incluso a los principiantes en programación programar con microcontroladores en pasos simples.

## App Inventor

App Inventor proviene originalmente de Google. Esta interfaz gráfica permite programar aplicaciones para teléfonos móviles Android con bloques gráficos.

## Ardublock

Este lenguaje de programación gráfica está especialmente diseñado para programar el microcontrolador Arduino sin introducir texto.

### **Pure Data**

Este lenguaje de programación visual está orientado tanto a flujos de datos como a los entornos de desarrollo. Pure Data permite producir software multimedia interactivo, por ejemplo, para sintetizadores.

### **Lego Mindstorms**

Una serie de productos del fabricante de juguetes Lego, cuyo núcleo es la pieza de Lego programable: los motores eléctricos, sensores y piezas de tecnología propios de Lego permiten construir y programar robots y otros sistemas interactivos.