

## Análisis de la programación visual

Conceptos de la programación orientada a objetos es un paradigma en el cual los programas se modelan en torno a objetos que unen toda funcionalidad relacionada con ellos. De este modo en lugar de crear una serie de funciones sin conexión alguna entre ellas, en programación orientada a objetos se crean clases, que representan entidades que se pueden manejar en el programa. La programación orientada a objetos gira mas en torno a los datos que en torno a la lógica, que se delega a un segundo plano ya que forma parte de dichos datos. Los principales conceptos de la Programación orientada a objetos: Clases, objetos e instancias

Una clase es una plantilla. Define de manera genérica como van a ser los objetos de determinado tipo.

Ejemplo de clase: En un juego una clase para representar a personas puede llamarse Persona y tener una serie de atributos como *Nombre*, *Apellidos* o *Edad* y una serie de comportamientos que pueden tener como *Hablar()*, *Caminar()* o *Comer()* y que se implementan como métodos de la clase o también llamadas funciones.

Una clase por sí sola no sirve de nada, para poder utilizar una clase lo primero que hay que hacer es instanciarla es decir crear un objeto en la misma. Un objeto es una entidad concreta que se crea a partir de la plantilla que es la clase. Un ejemplo de objeto en nuestro programa podría ser una persona que se llama Cristina López de 37 años y que en el programa podría hablar, caminar o comer que son las funciones que están definidas en la clase.

## Características y aplicaciones de eventos

Los eventos se envían para notificar al código de cosas interesantes que han ocurrido. Cada evento está representado por un objeto que se basa en la interfaz `Event`, y puede tener campos y/o funciones personalizadas adicionales para obtener más información acerca de lo sucedido. Los eventos pueden representar cualquier cosa desde las interacciones básicas del usuario para notificaciones automatizadas de las cosas que suceden en el modelo de representación.

Los eventos son el medio como interactúa una clase con otras o con el propio usuario, se encarga de avisar que algo ha ocurrido y de manejarlo de una forma o de otra. Cada vez que escribimos con nuestro teclado, que hacemos click en un botón o link, que cambiamos el tamaño de un objeto, estamos generando eventos. Es por ello por lo que cuando programamos debemos tener en cuenta la posibilidad tanto de manejar eventos que sólo implican a nuestra clase como de generar nuestros propios eventos, de modo que los usuarios de nuestras clases puedan decidir cómo reaccionará su código ante ellos.

## Características de componentes y métodos visuales y no visuales.

Los componentes son unos elementos genéricos con una funcionalidad muy concreta, cuya única finalidad es la reutilización. Cada uno de ellos está destinado a realizar una tarea típica en una aplicación.

Los métodos son funciones asociadas al componente que pueden invocarse para que el componente realice distintas acciones.

Los componentes visuales y no visuales se pueden establecer muchas clasificaciones para ellos. Una de ellas es la de visuales o controles. Frente a no visuales.

Un componente es visual cuando tiene una representación gráfica en tiempo de diseño y ejecución (botones, barras de scroll, cuadros de edición, etc.), y se dice no visual en caso contrario (temporizadores, cuadros de diálogo -no visibles en la fase de diseño-, etc).

Los componentes no visuales se pueden colocar en los formularios de la misma manera que los controles, aunque en este caso su posición es irrelevante.

Procesos de desarrollo visual en proyectos distribuidos y de escritorio.

Se puede ejecutar el módulo o programa para chequear la parte visual debido al alto manejo de estilos, controles asp y HTML que se hace en los sitios web desarrollados. También cambia el checklist a seguir en las revisiones e inspecciones, puesto que son más orientados a la interfaz de usuario. Las pruebas unitarias para seguir son las mismas planeadas para la codificación lógica, pero teniendo en cuenta como se altera la interfaz.

Requerimientos visuales de proyectos distribuidos y de escritorio.

Wireframes y maquetas de interfaz

Maqueta de interfaz de usuario: Una versión más gráfica y detallada del wireframe es la maqueta de la interfaz de usuario. No sólo le ayudará a hacerse una idea de cómo funciona su aplicación, sino también de su aspecto.

## Herramientas y lenguajes de programación visual

- Scratch
- Blockly
- Nepo
- Grape
- App inventor
- Ardublock
- Pure data
- Lego Mindstorms

## Lista de referencias

### Referencias

<https://www.campusmvp.es/recursos/post/los-conceptos-fundamentales-sobre-programacion-orientada-objetos-explicados-de-manera-simple.aspx>

<https://developer.mozilla.org/es/docs/Web/Events>

<https://elvex.ugr.es/decsai/builder/intro/3.html>

[https://www.ctr.unican.es/asignaturas/MC\\_OO/Doc/OO\\_08\\_I2\\_Proceso.pdf](https://www.ctr.unican.es/asignaturas/MC_OO/Doc/OO_08_I2_Proceso.pdf)

<https://creately.com/blog/es/diagramas/herramientas-recopilar-requisitos-software/>

<https://www.ionos.mx/digitalguide/paginas-web/desarrollo-web/programacion-visual/>