



Universidad  
**Tecmilenio**®



Te invitamos a que realices el siguiente ejercicio mental, el cual te tomará cinco minutos y te servirá para obtener una mejor claridad en los conceptos que aprenderemos el día de hoy.

Respira.

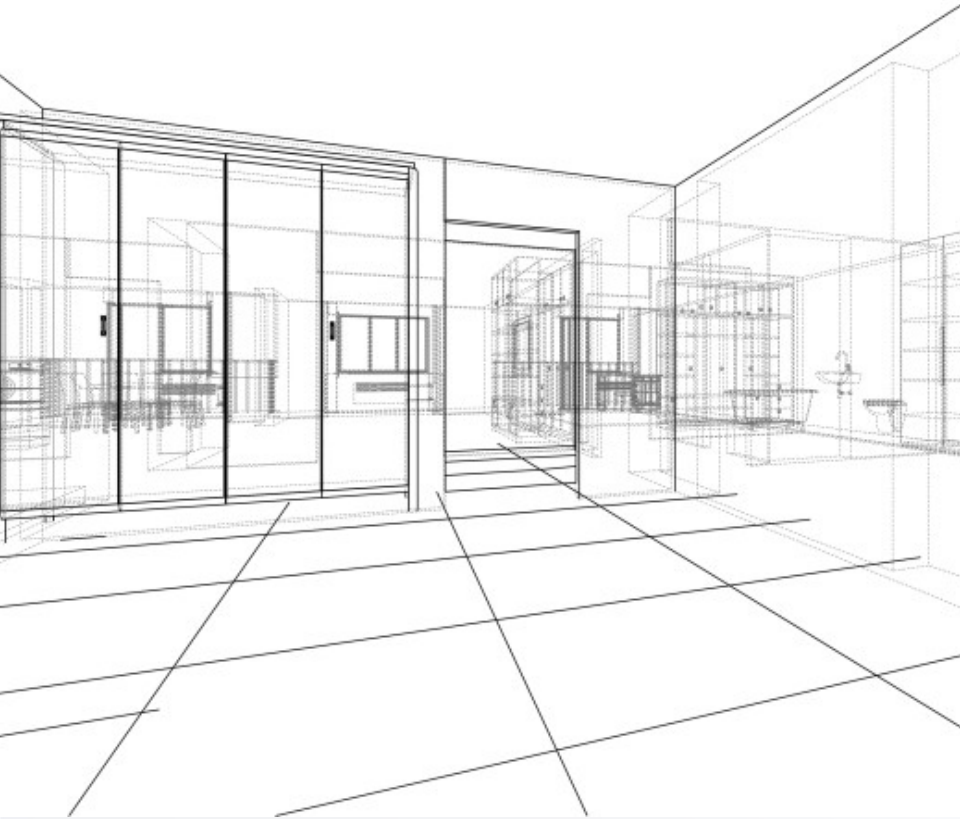
Liga: [https://youtu.be/\\_hSq87FzoIY](https://youtu.be/_hSq87FzoIY)



# Interactividad y diseño multimedia

## 1. Videojuegos casuales





En este tema empezarás a sumergirte a conceptos y técnicas de programación, por lo que te aconsejamos que tengas a la mano información sobre Unity.

También aprenderás el concepto de gameObjects, con el cual te vas a empezar a familiarizar, ya que lo estarás usando todo el tiempo al trabajar en Unity.

Revisarás información importante sobre el lenguaje de programación que estarás usando en la consola de Unity.



## GameObjects

- Son todo lo que vemos en una escena, por lo tanto, cada objeto en tu videojuego es un GameObject.
- Están conformados por componentes o components.



## Scripts

Es todo aquello que ocurre en un videojuego tal y como lo dice un guion y en el momento que se le indica. Estos scripts brindan una funcionalidad diferente dependiendo de lo que se mande a ejecutar y a cuál objeto se le coloque.



## Lenguajes de programación para Unity

### C#

---

Con este lenguaje se quiso mejorar a los dos anteriores de los que deriva el C y el C++. Este lenguaje le brinda un soporte especial a Unity y es el más utilizado por sus usuarios.

### JavaScript

---

Es un lenguaje de programación orientado a objetos y eventos. JavaScript se diseñó con una sintaxis similar al C.

### UnityScript

---

Muy similar en sintaxis a JavaScript, pero con importantes diferencias. Por ejemplo, UnityScript implementa clases, mientras que JavaScript maneja objetos.



- Reflexiona y responde lo siguiente.
  - a) ¿Qué es un GameObject?
  - b) Menciona los tres lenguajes de programación para Unity.







Como aprendiste en el tema, la programación de scripts es la base de todos los videojuegos, sin ellos los videojuegos no tendrían funcionamiento.

Gracias a la buena aceptación de Unity por parte de los usuarios a través de los años, es más sencillo generar nuevos scripts para desarrollar un videojuego desde cero.

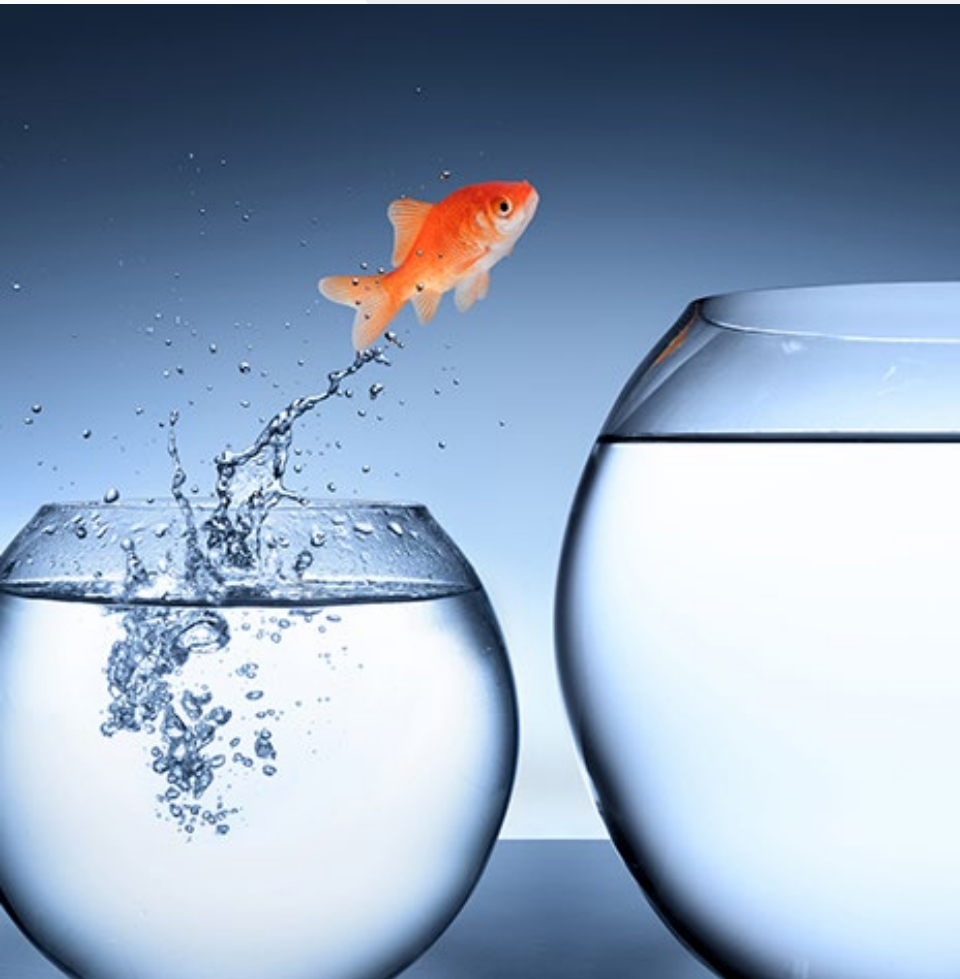
Así que ten paciencia y comienza por lo básico.



# Interactividad y diseño multimedia

## 6. Introducción a la programación en C#





Los scripts en Unity permiten agregar comportamiento a los GameObjects, ¿qué quiere decir esto? Es lo que utilizamos para definir cómo cada parte de tu videojuego funcionará, es el motor principal del videojuego.

Antes de empezar a programar scripts más complejos en Unity, es importante entender más a fondo cómo funciona su principal lenguaje de programación, C#. Como ya se mencionó antes, Unity permite utilizar otros lenguajes de programación, pero C# es el lenguaje más aceptado por el programa y con mayor compatibilidad.



## Operador punto (.)

Se utiliza en medio de dos palabras o líneas de código y se emplea para elegir elementos en un elemento compuesto, es decir, que contiene múltiples propiedades o elementos.

## Punto y coma (;)

Se utiliza para terminar una declaración, por lo que siempre se verá al final.

## Comentarios

Como su nombre lo indica, sirven para dejarte notas, a ti o a otro programador, sobre el código que estás escribiendo, para que sea entendible, fácil de editar y que te permita hacer correcciones.

## Clases

No define ninguna información, pero sí define cómo se utilizará esa información, de qué partes consiste y qué operaciones se pueden realizar con ella.

Sirve como base para saber qué funciones y eventos estarán disponibles para programar.

## MonoBehaviour

Es la clase base para todos los scripts de Unity, aparece por default al crear un nuevo script, y por lo general, escribirás todo el código dentro de ella.

Proporciona una lista de las funciones y eventos que son estándar para la programación de scripts adjuntos a GameObjects.



**Int:**

es una variable que contiene un número entero (es decir, sin decimales), este puede ser positivo o negativo.

**Float:**

similar a las variables int, pues también están hechas para contener números, pero este tipo de variable puede contener decimales.

**Bool:**

el tipo de dato lógico o booleano puede contener valores binarios, es decir, puede tomar dos valores que se excluyen mutuamente, por ejemplo, falso o verdadero.

**String:**

se utiliza principalmente para guardar datos como nombres, códigos, direcciones, etcétera.



● Reflexiona y responde lo siguiente.

- a) ¿Qué es un operador punto?
- b) ¿Qué es una clase?
- c) ¿Qué es una variable?





Programar en Unity puede parecer una tarea muy compleja, el lenguaje C# puede resultar algo intimidante en un inicio, no te desespere, es cuestión de práctica y familiaridad con el lenguaje.

Como te diste cuenta a lo largo del tema, basta seguir algunas reglas y entender la sintaxis del programa para crear programas muy útiles con poco esfuerzo.

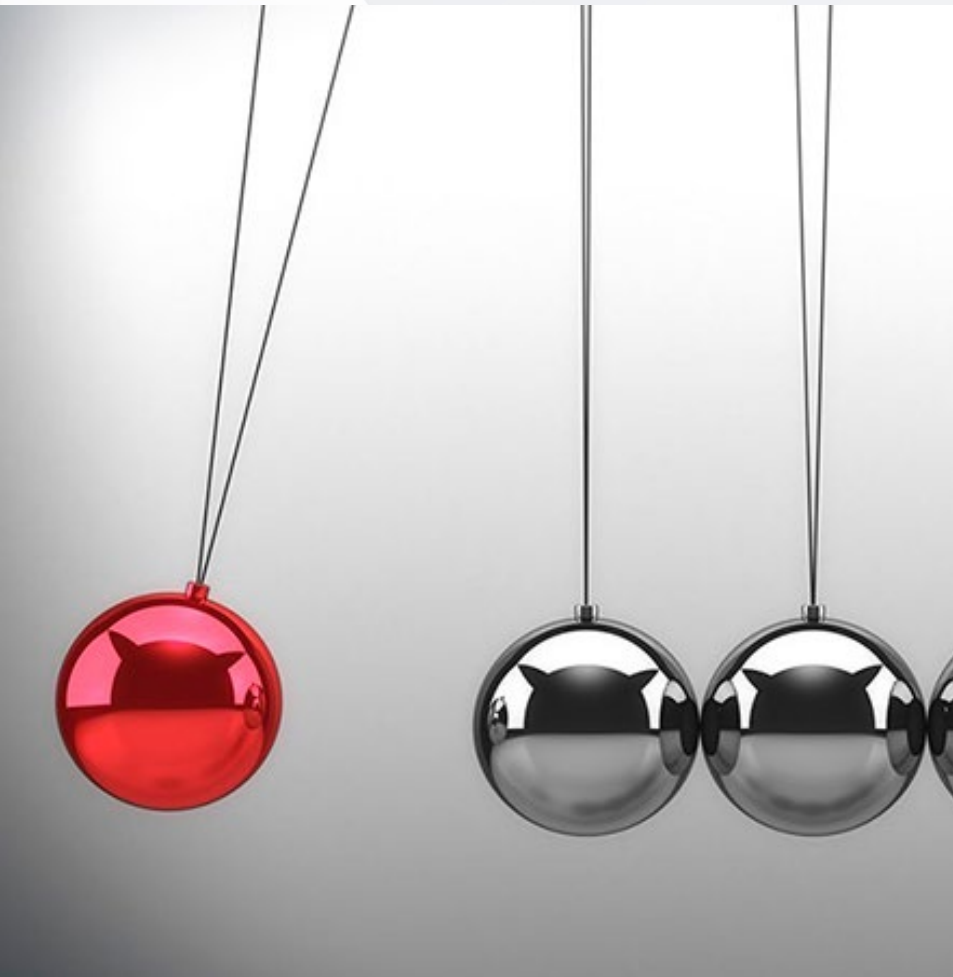




# Interactividad y diseño multimedia

## 7. Eventos y uso de componentes





A lo largo de los temas anteriores te has familiarizado con el lenguaje C#, que es un lenguaje con flexibilidad en su control de versiones y es uno de los lenguajes más potentes en la programación.

En este tema aprenderás a modificar y controlar un `GameObject`, utilizando sus componentes y un concepto nuevo llamado “eventos”.



## Awake

---

Esta función regularmente es utilizada para inicializar un objeto o referenciar otros scripts, es decir, para asignarle propiedades o variables al objeto que utilizarás más adelante o que son parte de otro objeto.

## Start

---

Esta función es llamada antes de la actualización del primer cuadro o antes de la primera actualización de física de un objeto.



## Update

Función que es llamada justo antes de que cada cuadro del juego sea renderizado, y por ello, antes de que se calculen las animaciones.

## FixedUpdate

Es una función muy similar a update, pero es llamada cada vez que se hace una actualización del motor de física de Unity, en lugar del renderizado de la animación.

## LateUpdate

Función que se llama después de que update y fixedupdate han sido llamadas para todos los objetos en la escena y después de que todas las animaciones hayan sido calculadas.



## **OnCollisionEnter**

La función OnCollisionEnter se llamará una vez que se haya hecho un contacto.

## **OnCollisionStay**

OnCollisionStay se llamará cuando un contacto se mantenga por una cantidad delimitada de tiempo.

## **OnCollisionExit**

OnCollisionExit será llamado una vez que un contacto se haya roto.

## **OnTrigger, OnTriggerStay y OnTriggerExit**

Estas funciones pueden ser llamadas varias veces en sucesiones si un contacto o más son detectados durante la actualización de física.ue llega, etcétera.



- Reflexiona y responde lo siguiente.
  - a) ¿Cómo funciona Update?
  - b) ¿Cómo funciona FixedUpdate?
  - c) ¿Para qué sirven las funciones en Unity?





En este tema has aprendido más de la programación C#, por lo que ya sabes que Unity tiene una gran variedad de componentes para generar un videojuego de gran calidad. También conociste nuevos conceptos, como los diferentes tipos de funciones de eventos y su utilización para crear scripts que se ejecuten en el momento correcto y bajo las condiciones exactas.

Es importante que te familiarices con el procedimiento de programación para obtener el resultado deseado en tus scripts. Todo esto llevará a que tu experiencia de uso de Unity sea más sencilla y agradable.



# Interactividad y diseño multimedia

## 8. 2D GameObjects (sprites)







En Unity se trabaja con sprites para proyectos 2D. Los sprites han sido utilizados desde hace años en los videojuegos para representar todo lo que se observa en escena; fondos, jugador, enemigos, barras de estado, entre otros.

Los sprites son mapas de bits en 2D que se dibujan en un destino de representación, se consideran las unidades fundamentales en el desarrollo de videojuegos 2D.

Por lo tanto, un sprite es cualquier objeto que aparece en nuestro juego, en el entorno 2D. Tiene asociado atributos de imagen, posición y velocidad. Los sprites pueden ser estáticos o dinámicos.



## Sprites

- Los sprites son objetos 2D que contienen una imagen, esta tiene posición y orientación. Unity proporciona diversas herramientas con las que puedes crear un sprite, editar y renderizar contenido de sprites en paquetes.
- El modo de sprite tiene dos opciones, single y multiple; se utiliza single cuando es solo un elemento y será utilizado como un sprite individual. Selecciona multiple cuando el asset contiene varios elementos, tales como un atlas, sprite sheet.



## Sprite tools

### Sprite renderer

---

Muestra imágenes como sprites para el entorno 2D y 3D. Despliega imágenes que han sido importadas como sprite type. La propiedad más importante del sprite renderer es precisamente sprite, esto indica que renderizará, por defecto, al sprite.

### Sprite editor

---

Es utilizado para colocar imágenes, conteniendo múltiples elementos, como un atlas o sprite sheets; posteriormente, los separa en sprites individuales.



## Sprite tools

### Sprite packer

---

Permite crear un sprite atlas, solo debes ir al menú principal Window/Sprite Packer.

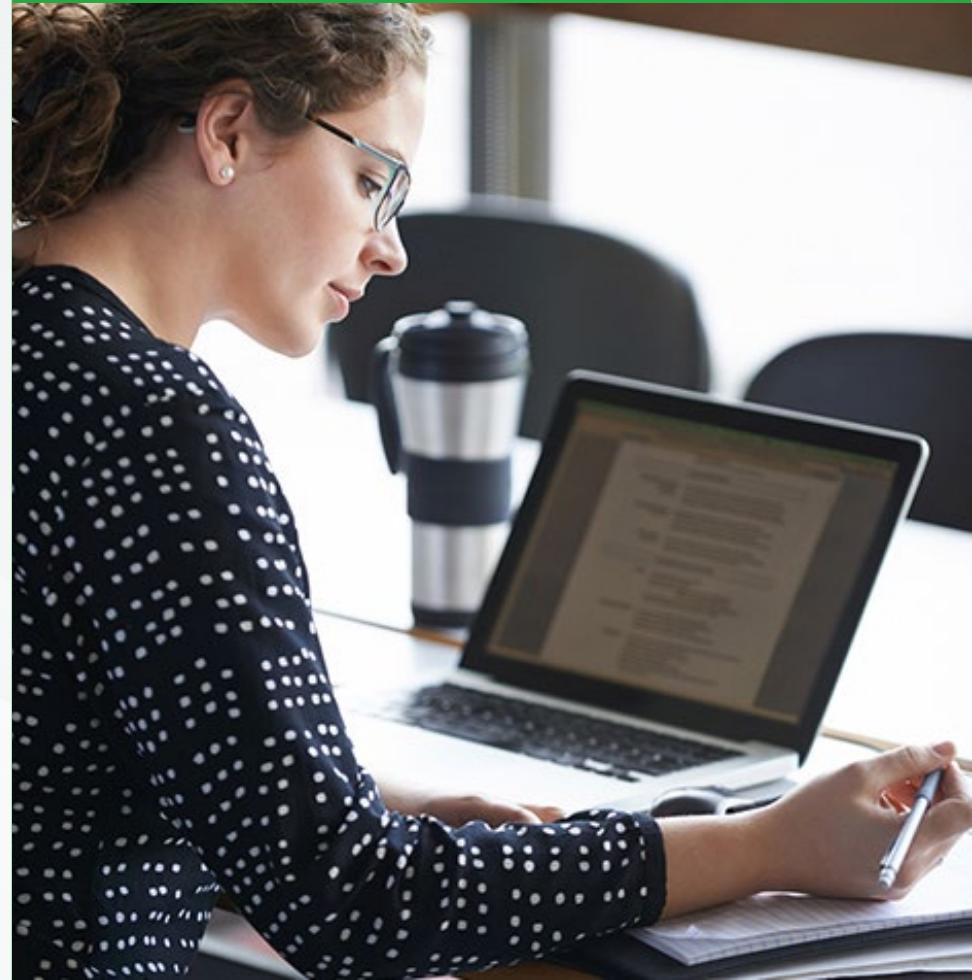
### Sprite creator

---

Esta herramienta te permite crear archivos temporales de sprites. Estos los puedes utilizar al momento de trabajar una versión de prueba, y posteriormente, reemplazarlos con los sprites que utilizarás.



- Reflexiona y responde lo siguiente.
  - a) ¿Qué es un sprite?
  - b) ¿Para qué sirve un sprite sheet?
  - c) Menciona dos herramientas de los sprites en Unity.





Recuerda que al desarrollar un videojuego en 2D se utilizan assets de GameObjects en 2D sprites. Esta técnica se conoce por el surgimiento del desarrollo de videojuegos. Los sprites sirven para la optimización del uso de imágenes.

Pudiste comprobar que Unity cuenta con herramientas para crear tus propios sprites dentro del programa para editar tus sprite sheets o sprite atlas y te brinda diversas opciones para el renderizado de imágenes en el entorno 2D.

- ● Ya puedes crear un proyecto nuevo de Unity que contenga un sprite sheet y la separación de los elementos dentro de él.