



---

## AGRADECIMIENTOS

Adrián: A Cristina, Yolanda y Manuel por su apoyo constante. Y a todos mis alumnos, por ser mi fuente de motivación, de aprendizaje y los que hacen posible que ejerza mi vocación.

Fernando: Para Rosa y Fernando. A mi amigo y maestro Juan Carlos, que desde su aportación en *LaPasionPorEscribir.blogspot.com.es*, nos ayuda a todos a expresarnos un poco mejor.

Javier: A todos aquellos que no se han rendido, a todos los que han luchado y a todos los que se esfuerzan día tras día. Nunca dejéis de tener hambre por aprender: El futuro es vuestro.





---

## **SOBRE LOS AUTORES**

### **ADRIÁN DOMÍNGUEZ DÍAZ**

Ingeniero informático especializado en desarrollo de apps y videojuegos. Desde 2012 trabaja como formador en estas áreas, impartiendo cursos y másteres en distintas instituciones y empresas como la Universidad de Alcalá o la escuela CICE.

Desde 2014 ha compatibilizado su labor docente con labores de director y desarrollador en Chaotic Kingdoms, estudio del que es cofundador, desarrollando apps y juegos dirigidos al marketing, la salud o la educación. En 2017 finalizó esta aventura empresarial para dedicarse de lleno a su vocación docente.

Actualmente ejerce como director del Máster en Desarrollo de Videojuegos con Unity en CICE, labor que compatibiliza con la investigación y la realización de su tesis doctoral en la Universidad de Alcalá, en las áreas de la gamificación y el uso de nuevas tecnologías en la educación.

Para más información sobre Adrián puedes consultar su web personal <http://adriandominguez.tech> o contactar con él vía Twitter en [@profvideojuegos](https://twitter.com/profvideojuegos).

## **FERNANDO NAVARRO PULIDO**

Especialista en ingeniería informática aplicada a la formación y RRHH, ha desarrollado desde el año 1999 multitud de sistemas de información aplicados a estas áreas, para grandes corporaciones como Deloitte, Endesa, la Confederación Española de Cajas de Ahorros, la Universidad de Deusto, etc., incluyendo el videojuego y la gamificación como elementos innovadores dentro de los procesos formativos.

Desde el año 2007 ha colaborado con distintos estudios en el desarrollo de varios videojuegos para distintas plataformas.

Actualmente ejerce su labor profesional como director técnico del estudio Digital Dream Games & VR Factory, especializado en la integración de videojuegos educativos en sistemas de formación on-line, gamificación de contenidos y desarrollo de aplicaciones de transformación digital, basadas en realidad aumentada y virtual.

## **JAVIER MANUEL CASTRO GONZÁLEZ**

Informático por vocación, Javier creció en una familia de programadores, adquiriendo una pasión por el desarrollo de videojuegos que le ha acompañado desde su niñez. Sus especialidades son la programación de lógica, gráfica, core y scripting para videojuegos.

Desde 2014 se dedica al mundo de la enseñanza en estas áreas, impartiendo clases magistrales y cursos en distintas empresas como ConMasFuturo o CICE. Además, desde 2007 ha trabajado de forma profesional como programador en distintas empresas y colaborado con diversos grupos independientes, desarrollando software educativo, herramientas de imprenta digital, videojuegos, diseño, modelado, texturizado y rigging 3D.

Actualmente ocupa el puesto de formador en el Master en Desarrollo de Videojuegos con Unity en CICE, mientras continua de forma activa en el desarrollo de videojuegos con herramientas como Unreal Engine y Unity3D en diversos grupos con los que se relaciona.

---

# INTRODUCCIÓN

Unity es actualmente uno de los *game engines* más utilizados en la industria del desarrollo de videojuegos. Desde el lanzamiento de Unity 3 en 2010, este motor ha sufrido un gran crecimiento tanto a nivel técnico, ofreciendo cada vez más y mejores funcionalidades; como en cuanto a volumen de usuarios, siendo adoptado por todo tipo de desarrolladores y empresas, desde aficionados hasta colosos de la industria como Nintendo o Square Enix. Este éxito se ha debido sin duda al lema de la compañía, enfocada en “democratizar el desarrollo de videojuegos” con sus herramientas. Además del avanzado y completo motor que nos ofrece Unity, algunas de sus características, como sus versiones gratuitas, la tienda de recursos o las capacidades de compilación multiplataforma, han sido elementos diferenciales clave para el éxito de Unity por encima de otros motores.

El objetivo de este libro es que los lectores adquieran una base sólida de conocimiento sobre Unity, descubriendo y comprendiendo los conceptos, elementos y herramientas esenciales para el desarrollo de videojuegos con este motor. Dada la gran cantidad de funcionalidades que ofrece Unity, el alcance del libro se ha acotado a aquellas partes del motor que resultan esenciales en el desarrollo de cualquier tipo de videojuego. A cambio, cada uno de los temas cubiertos es abordado en profundidad, tratando de que el lector no solamente aprenda a utilizar el motor, sino que, sobre todo, entienda los conceptos subyacentes a cada técnica y herramienta explicada. Los distintos capítulos van acompañados también de consejos de experto, basados en la experiencia de los autores, con los que el lector podrá evitar errores comunes y acelerar su proceso de aprendizaje y posterior trabajo con Unity. Por último, gracias a los contenidos descargables que acompañan al libro, el lector podrá acceder a proyectos con los que ver de forma práctica todo lo aprendido durante la lectura.

El libro se divide en los siguientes capítulos:

- Capítulo 1. Introducción a Unity
- Capítulo 2. Conceptos básicos
- Capítulo 3. *Scripting*
- Capítulo 4. Creación de niveles
- Capítulo 5. Físicas y colisiones
- Capítulo 6. Interfaz de usuario
- Apéndice

# 1

---

## INTRODUCCIÓN A UNITY

En este capítulo explicaremos qué es Unity y cuáles son sus características más importantes. Después procederemos a su descarga e instalación paso a paso, y a la creación y apertura de un nuevo proyecto, descubriendo la estructura de carpetas que siguen los proyectos. A lo largo del capítulo podrás aprender más sobre los siguientes temas:

1. El motor Unity
2. Descarga e instalación
3. Primer uso y activación
4. Creación de un nuevo proyecto
5. Apertura de un proyecto
6. Estructura de carpetas del proyecto
7. Editor de código fuente

### 1.1 EL MOTOR UNITY

---

Unity es un motor de desarrollo de videojuegos, una *suite* de herramientas diseñadas para facilitar la labor de trabajar en los distintos aspectos de un videojuego, tales como los gráficos, las físicas, las animaciones o la inteligencia artificial. Además, Unity tiene determinadas características que han ayudado a su adopción por parte de empresas y profesionales y a su rápido crecimiento, que se enumeran a continuación:

- **Gratuito:** Unity dispone de una licencia gratuita totalmente funcional para desarrolladores *indie* y pequeños estudios, así como un plan de precios escalable según el volumen de negocio de cada estudio. Esto hace de Unity una herramienta asequible que se adapta a las necesidades de cada desarrollador.
- **Multiplataforma:** Unity permite crear juegos para sistemas de sobremesa como Windows o MAC OS, dispositivos móviles como iOS y Android, consolas como Playstation, Xbox o Wii, e incluso para otros dispositivos como televisores inteligentes Samsung o Apple TV, así como los distintos sistemas de realidad virtual que hay en el mercado. La compilación para gran parte de estas plataformas es completamente gratuita, lo que permite ahorrar grandes costes en el desarrollo de juegos multiplataforma.
- **Extensible:** la funcionalidad de Unity puede ser incrementada mediante *plugins*, consiguiendo de esta forma que el motor maneje y gestione elementos para los que, en un principio, no estaba pensado. Esta característica permite a los desarrolladores usar Unity para todo tipo de proyectos y utilizar *plugins* propios o de terceros para acelerar su trabajo.
- **Tienda de recursos:** Unity no solo es extensible, sino que además ofrece un portal para comprar y vender extensiones y otro tipo de recursos, denominado Asset Store. Dentro de este portal es posible encontrar todo tipo de complementos y recursos desarrollados por terceros, tanto gratuitos como de pago. Además de ser útil para los desarrolladores de juegos, también permite a numerosas compañías forjar su negocio alrededor de esta tienda, como creadores de herramientas y contenido.
- **Servicios:** Unity ofrece una serie de servicios adicionales, en su mayor parte gratuitos, que ayudan en distintas fases del negocio de los videojuegos. Por ejemplo, Unity Ads permite añadir publicidad en juegos para dispositivos móviles, Unity Analytics facilita la captura de analíticas sobre interacción de usuarios y Unity Networking ofrece infraestructura para el desarrollo de juegos *online* y *matchmaking*.

## 1.2 DESCARGA E INSTALACIÓN

---

Unity puede descargarse gratuitamente desde la web <https://unity3d.com/es/> accediendo a la página **Obtén Unity ahora** que aparece al entrar y seleccionando el tipo de licencia **Personal** en caso de que queramos utilizar la licencia gratuita.

Al respecto de las distintas licencias de Unity, conviene indicar que, aunque existen cuatro modelos, Personal, Plus, Pro y Enterprise, en todas ellas están incluidas las mismas funcionalidades y herramientas del motor. Las versiones de pago se distinguen únicamente por una serie de servicios externos adicionales que pueden ser de especial interés para desarrolladores profesionales y empresas, pero no afectan de ninguna forma a las funcionalidades del propio motor. En consecuencia, todo lo explicado en este libro podrá llevarse a cabo con cualquier modelo de licencia, ya sea gratuita o de pago.

Una vez descargado el instalador de Unity, procederemos a abrirlo para instalar el motor. Tras aceptar las condiciones de la licencia, durante el proceso de instalación podremos elegir qué paquetes deseamos instalar y para qué plataformas habilitaremos la compilación de proyectos. Para agilizar la instalación te recomendamos activar únicamente la instalación de Unity (primer *checkbox*) y los *Standard Assets* (tercer *checkbox*) que utilizaremos en algunos de los capítulos del libro:

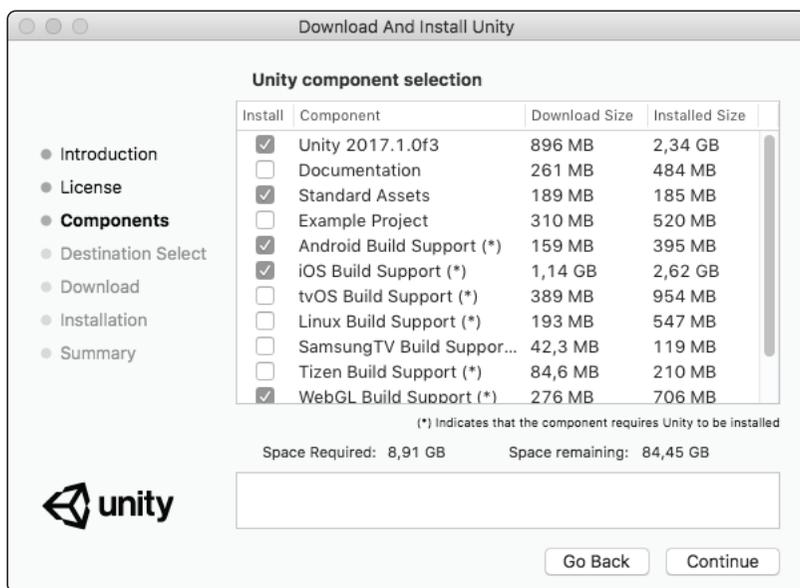


Figura 1.1. Opciones de instalación

Una vez seleccionados los paquetes a instalar y la ruta de instalación, Unity procederá a descargar todos los paquetes necesarios y a instalarlos en nuestro equipo.

**TRUCO**

Pueden descargarse e instalarse paquetes y plataformas adicionales en cualquier momento después de la instalación, por lo que no debes preocuparte si se te ha olvidado seleccionar alguno de los paquetes durante la instalación.

### 1.3 PRIMER USO Y ACTIVACIÓN DE LICENCIA

Cuando abramos el motor por primera vez, este nos solicitará los datos de nuestra cuenta Unity. En caso de no disponer de esta cuenta, necesaria para poder utilizar la herramienta, procederemos a registrarnos pulsando sobre el enlace indicado o a través de la web de Unity. Tras ello introduciremos el email y contraseña con los que nos hayamos registrado, y continuaremos:

unity Hello!

Sign into your Unity Account

If you don't have a Unity Account, please create one to access Unity services and resources.

Email \*

Password \*

[Forgot your password?](#)  
[Can't find your confirmation email?](#)

Remember my choice

Figura 1.2. Ingreso de datos de cuenta de desarrollador

Una vez introducida la cuenta de Unity, se abrirá la ventana inicial. Esta ventana, que se mostrará por defecto cada vez que iniciemos el motor, muestra el contenido de los proyectos recientes con los que hemos trabajado, así como opciones para crear nuevos proyectos o cargar otros que no se encuentran en la lista de recientes.

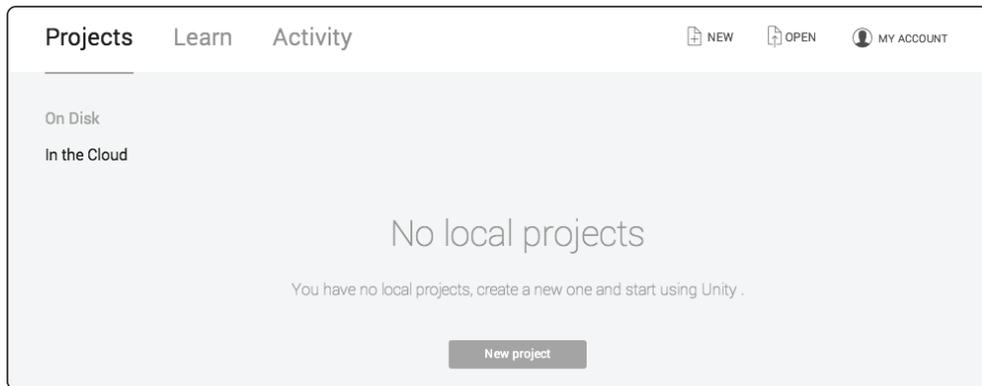


Figura 1.3. Ventana inicial

Desde esta pantalla también podemos acceder a las opciones de nuestra cuenta de Unity, pulsando para ello el botón **My Account**. Aunque una vez validada la licencia podemos cerrar la sesión y trabajar con Unity de forma *offline*, recomendamos mantener la sesión iniciada ya que nos ahorrará introducir los datos de nuestra cuenta al acceder a determinadas secciones, como el Asset Store o los servicios.

## 1.4 CREACIÓN DE UN NUEVO PROYECTO

Para crear un nuevo proyecto podemos seleccionar la opción **New**, con lo que nos aparecerá la siguiente ventana:

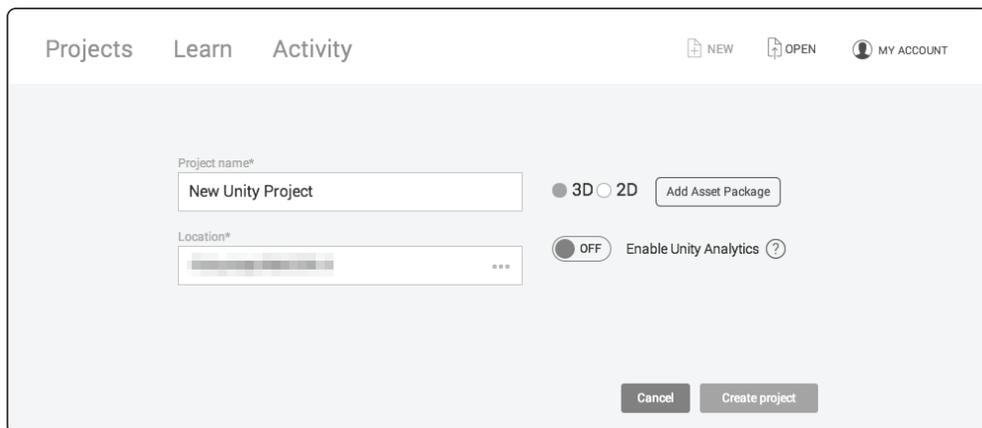


Figura 1.4. Creación de un nuevo proyecto

En esta ventana debemos indicar el nombre que tendrá nuestro proyecto, en qué lugar del disco lo vamos a guardar, si va a ser un proyecto 3D o 2D y si queremos añadir paquetes o servicios adicionales. Cualquiera de estos valores puede ser cambiado con posterioridad, por lo que no debemos preocuparnos si cometemos cualquier error.

Una vez cumplimentada esta información pulsaremos sobre el botón **Create Project**. Unity creará un nuevo directorio con el nombre del proyecto en la ubicación que le hemos indicado, generando una estructura de carpetas y ficheros básica para el proyecto. Tras ello se abrirá el editor.

## 1.5 APERTURA DE UN PROYECTO EXISTENTE

---

Para abrir un proyecto utilizaremos la opción **Open** de la ventana, con lo que Unity nos mostrará el explorador del disco para seleccionar la carpeta donde se encuentra el proyecto.

También es posible abrir un proyecto haciendo doble clic en uno de sus ficheros de escena. Una vez abierto un proyecto, éste aparecerá en el listado de proyectos frecuentes y bastará hacer clic en el mismo para volver a abrirlo en cualquier momento.

## 1.6 ESTRUCTURA DE CARPETAS DE UN PROYECTO

---

Al crear un nuevo proyecto, Unity genera una serie de carpetas y ficheros dentro de la carpeta del proyecto.

La estructura de carpetas inicial es la siguiente:

- **Assets:** esta es la carpeta raíz de todos los recursos de nuestro juego. En general trabajaremos con esta carpeta a través del editor de Unity, ya que es el lugar donde se copiarán todos los recursos que importemos, y donde se desplegarán los distintos *plugins* y paquetes que instalemos. Junto a cada recurso almacenado en esta carpeta, Unity generará también un fichero con el mismo nombre y extensión *.meta*, que almacenará las opciones de importación del recurso, por lo que en ningún caso debemos eliminarlos o modificarlos.
- **Project Settings:** Unity dispone de un conjunto de herramientas denominadas *Settings Managers*. Cada una de estas herramientas gestiona algún aspecto de configuración de nuestro proyecto, como pueden ser los controles del juego, los tipos de objetos físicos, la calidad gráfica o el nombre e icono del juego, entre otras muchas opciones. El resultado

de estas configuraciones se almacena en la carpeta Project Settings. Es por tanto una carpeta importante ya que almacena gran parte de la configuración de nuestro juego y entorno de trabajo.

- ▀ **Library:** esta carpeta almacena versiones procesadas y optimizadas de los distintos recursos que guardamos en la carpeta Assets. Los recursos de la carpeta Assets nunca se usarán tal cual en la versión ejecutable del juego, si no que Unity los procesará y optimizará para adecuarlos al *hardware* o plataforma de destino. Por ejemplo, Unity nos permite importar ficheros de Photoshop en un proyecto, pero este formato no es adecuado para el “renderizado” de imágenes en una tarjeta gráfica, por lo que Unity lo procesará, comprimiéndolo y optimizándolo adecuadamente. Este procesamiento se realizará en todos y cada uno de los recursos de la carpeta Assets, independientemente de su formato. En ningún momento debemos modificar estos ficheros a mano, aunque si podremos borrarlos sin preocupación si lo necesitamos, ya que Unity volverá a generarlos de nuevo al abrir el proyecto.
- ▀ **Temp:** en esta carpeta Unity almacena ficheros temporales necesarios para su funcionamiento interno. Entre otros, en esta carpeta se crearán ficheros que almacenarán el estado actual de la escena en la que estemos trabajando. En general nunca modificaremos o eliminaremos estos ficheros.
- ▀ **Otras carpetas:** aunque inicialmente el proyecto solo contará con las carpetas anteriormente mencionadas, cuando editemos *scripts* por primera vez se generarán dos carpetas adicionales, “obj” y “bin”, además de varios ficheros con distintas extensiones como .sln, .csproj o .unityproj. Todas estas carpetas y ficheros son utilizados durante las compilaciones de proyectos, o por el editor de código fuente, y en general nunca queremos abrirlos o modificarlos manualmente.



#### TRUCO

A la hora de mantener tu proyecto bajo control de versiones o de realizar copias de seguridad del mismo, solo deberás mantener versiones o copiar las carpetas Assets y Project Settings de la carpeta raíz del proyecto, asegurando que copias todo su contenido incluyendo los ficheros .meta que acompañan a cada recurso. El resto de carpetas y ficheros puedes mantenerlos fuera del control de versiones u omitirlos a la hora de realizar una copia de seguridad, ya que Unity los volverá a generar automáticamente al abrir el proyecto.

---

## 1.7 EDITOR DE CÓDIGO FUENTE

---

Una de las herramientas más importantes a la hora de desarrollar un juego es el editor de código fuente, también denominado IDE<sup>1</sup> (*Integrated Development Environment*). El IDE nos permite programar *scripts* con los que definiremos el comportamiento de los distintos elementos de nuestro juego. Unity no dispone de un IDE incrustado en el editor, si no que utiliza un editor externo con el que se integra y sincroniza automáticamente.

El editor de código por defecto es MonoDevelop, herramienta que se instalará automáticamente junto con Unity. Si bien en la versión de Unity para Windows también tendremos la posibilidad de instalar y utilizar Microsoft Visual Studio en su lugar. Aunque cada una de estas herramientas tiene sus ventajas e inconvenientes, ambos permiten desarrollar exactamente los mismos *scripts*. En este libro utilizaremos MonoDevelop debido a que está disponible tanto en Windows como en Mac.

---

1 IDE: un IDE (en español, Entorno de Desarrollo Integrado), es un sistema de *software* diseñado para desarrollar *software* y que incluye una gran variedad de herramientas como editor de código, compilador, depurador, etc. Existen muchos tipos de IDE para distintos lenguajes de programación o plataformas, como Visual Studio para C# y NetBeans para Java.