# Registro de programas

El software, al igual que la música o las novelas, está protegido por derechos de autor. Estos derechos son:

*-Derecho a publicar*

*-El derecho a ejecutar o reproducir*

*-El derecho de comercialización*

*-El derecho a distribuir sus obras*

*-El derecho a reclamar pagos por ellas*

Estos derechos son parte de la propiedad intelectual, y son intangibles. Después de todo, el software es un producto intangible. Se puede distribuir fácilmente un programa por Internet, pero los derechos de autor se encargan de evitar posibles malos usos de ese software.

Sin embargo, a veces los derechos de autor no bastan para proteger un programa adecuadamente. A veces es necesario una patente industrial. Este tema es motivo de debate en diversos ámbitos.

Mientras que los derechos de autor están pensados para evitar plagio, la patente industrial es más compleja, y está pensada para evitar que alguien use una idea nuestra, sin importar su representación.

Imaginemos un programa A, que hace una cierta tarea. Los programas B y C también hacen esa misma tarea, pero se crearon después que A. Los derechos de autor de A solo previenen el plagio del código de A, pero no de su idea, por lo que si los programas B y C no tienen código que haya sido copiado literalmente de A, no se considera plagio. Imaginemos ahora que el creador de A tiene una patente industrial sobre su programa. En este caso, B y C sí que estarían infringiendo la ley, ya que la idea de A pertenece a su creador, y no pueden representarla, aunque sea de un modo distinto.

Teniendo en cuenta lo dicho, podría parecer que sería sencillo inventar un programa rápidamente sin optimizar, pedir la patente industrial, e irlo mejorando poco a poco, lo cual impediría tener competencia alguna. Aquí está la diferencia clave entre los derechos de autor y la patente. Mientras que los derechos de autor existen desde el momento que se tiene la idea, la patente solo se puede obtener cuando se tiene un producto que realice algún tipo de beneficio, y aun así el producto es sometido a un riguroso examen con costes económicos y temporales.

Los derechos de autor son claramente la opción más fácil de proteger nuestro software, ya que las dificultades de obtener una patente en muchos casos son infranqueables para todos excepto para grandes corporaciones.

# Uso de recursos

Vamos a exponer la definición de uso de recursos:

**Definición:** *“Un recurso es un elemento multimedia que realiza una función específica en el proyecto mejorando sus características o añadiendo nuevas.”*

Tras la definición de uso de recursos, vamos a tratar las diferentes características que forman estos recursos como:

## Tipos de recursos.

En esta sección diferenciamos los tipos de recursos que podemos usar:

* Modelos 3D
* Texturas
* Tiles
* Música
* Librerías
* Fondos e imágenes
* Videos
* Herramientas

## Licencias

### ¿Se pueden utilizar libremente los recursos accesibles en Internet?

La mera navegación y ojeo de contenidos en Internet no suele implicar una explotación de derechos de propiedad intelectual, sin embargo, un uso posterior de dichos contenidos, aunque estén libremente accesibles, deberá respetar lo que el titular de los derechos establezca. Si el contenido aparece protegido por el símbolo del copyright © acompañado de la expresión “todos los derechos reservados”, no se puede dar a tal contenido más uso que el permitido por la ley.

Sin embargo, cada vez con más frecuencia se pueden encontrar contenidos con licencias de uso más permisivas, llamadas licencias libres o abiertas. En esos casos se debe respetar la voluntad del titular en cuanto a usos consentidos y condiciones establecidas.

Ejemplo de ello son las licencias Creative Commons.

Si no hay ninguna indicación sobre las condiciones de uso de una obra, ha de considerarse que tiene todos los derechos reservados. Los derechos de la obra pueden pertenecer a su autor o autora original o a otro titular a quien le han sido cedidos, y en cada caso se ha de determinar a quién hay que dirigirse para solicitar permiso al autor.

No necesitamos pedir permiso en los casos siguientes:

### Materiales con licencias libres

Generalmente se puede utilizar cualquier material que se difunde con una licencia libre, como mínimo sin finalidad comercial, pero siempre hay que reconocer al autor/a y mostrar el aviso de la situación legal. Hay que ver en cada caso las condiciones que ha establecido el titular de la obra para que ésta se pueda utilizar.

### Material perteneciente al dominio público

Se trata de material del cual se ha agotado el período de protección de derechos de explotación (generalmente, 70 años después de la muerte del autor/a). No obstante, hay que citar al autor/a original y respetar la integridad de la obra. Hay que ir con mucho cuidado al determinar si un material es de dominio público, ya que las modificaciones de la Ley prevén singularidades a la hora de determinar la caducidad de los derechos de autor. Para ahorrarnos problemas podemos acceder a determinados espacios donde encontraremos materiales con licencias libres que nos permitirán la utilización.

### Material que la Ley nos permite usar

Es decir, nos acogemos a alguna de las excepciones previstas por la Ley de Propiedad Intelectual y que limitan los derechos de autor. En este caso hay que determinar claramente si el uso que queremos hacer queda recogido por excepciones como la copia privada, la cita o la ilustración, entre otras.

Ahora enumeraremos las licencias más importantes que se usan en la actualidad:

* Copyright: Derecho exclusivo de un autor, editor o concesionario para explotar una obra literaria, científica o artística durante cierto tiempo.

Ejemplo:

"el copyright de un libro; existe un copyright sobre plantas y semillas logradas por selección genética; el copyright se indica con el signo © delante de la persona física o jurídica que posee el derecho de explotación"

* Software libre: El término software libre refiere el conjunto de software (programa informático) que, por elección manifiesta de su autor, puede ser copiado, estudiado, modificado, utilizado libremente con cualquier fin y redistribuido con o sin cambios o mejoras.
  + Tipos de licencias:
    - GPL (Licencia Pública General).
    - LGPL (Licencia Pública General Reducida).
    - AGPL (Licencia Pública General de Affero).
    - BSD.
    - PSFL (Licencia de Software de la Fundación Python).
    - MIT (Instituto Tecnológico de Massachusetts).
    - MPL (Licencia Publica de Mozilla).
    - CDDL (Licencia Común de Desarrollo y Distribución).
    - Copyleft.
    - EPL (Licencia Pública Eclipse).
    - APSL (Licencia Pública de Apple).
    - Apache.
    - PHP.
* Creative Commons: Organización que permite usar y compartir tanto la creatividad como el conocimiento a través de una serie de instrumentos jurídicos de carácter gratuito.

## ¿Dónde encontrarlos?

* Modelos 3D
  + <http://tf3dm.com/>
  + <http://archive3d.net/>
  + <http://www.3dmodelfree.com/>
  + <http://www.3dxtras.com/>
  + <http://artist-3d.com/>
  + <https://sketchfab.com/>
* Texturas
  + <http://www.textures.com/>
  + <http://www.textureking.com/>
  + <http://mayang.com/textures/>
  + <http://texturez.com/>
* Tiles
  + <http://opengameart.org/>
  + <http://oryxdesignlab.com/>
  + <http://rltiles.sourceforge.net/>
  + <http://www.zincland.com/powder/>
* Música
  + <https://soundcloud.com/>
  + <http://freesoundtrackmusic.com/>
  + <http://freemusicarchive.org/>
  + <https://www.freesound.org/browse/tags/sound-effects/>
  + <http://soundbible.com/free-sound-effects-1.html>
  + <http://www.jamendo.com/>
  + <http://freeplaymusic.com>
  + <http://www.sounddogs.com/>
* Librerías
  + <http://box2d.org/>
  + <http://www.sfml-dev.org/>
  + <http://irrlicht.sourceforge.net/>
  + <http://www.libgosu.org/>
  + <http://liballeg.org/>
* Fondos e imágenes
  + <https://pixabay.com/es/>
  + <https://500px.com/creativecommons>
  + <http://es.freeimages.com/>
  + <http://www.gettyimages.es/>
  + <http://unplash.com>
  + <http://flickr.com>
  + <http://gratisography.com>
  + <http://deathtothestockphoto.com>
* Herramientas
  + <https://unity3d.com/>
  + <http://www.yoyogames.com/gamemaker>
  + <https://www.scirra.com/>
  + <https://www.blender.org/>
  + <http://www.autodesk.es/products/3ds-max/overview>
  + <http://www.autodesk.es/products/maya/overview>
  + <http://www.adobe.com/es/products/illustrator.html>
  + <http://www.adobe.com/es/products/photoshop.html>
  + <http://www.gimp.org.es/>
  + <http://www.adobe.com/es/products/aftereffects.html>
  + <http://www.vegascreativesoftware.com/es/>
  + <https://www.steinberg.net/en/products/cubase/start.html>
  + <https://www.avid.com/es/pro-tools>
  + <http://www.mapeditor.org/>

# Usos de recursos destinados a videojuegos.

## Modelos 3D.

Los modelos son las figuras, cuerpos u objetos que usamos en los videojuegos (entre otros) para ambientar la escena y decorar los escenarios.

Estos, se pueden conseguir de dos formas: la primera es creándolos nosotros mismos usando un software de modelado y aplicando nuestras habilidades para obtener el modelo deseado; la segunda es acudiendo a webs las cuales contienen gran cantidad de modelos acabados y los cuales se pueden usar hasta cierto punto, dependiendo de las licencias que posean, en nuestros trabajos.

## Texturas.

Las texturas son aquello que se la aplica al modelo en la superficie para dar sensación de realidad, de que el modelo está compuesto por algún material real.

Hay distintos tipos de texturas: están las texturas que son en 3D, que no necesitan un mapeado detallado porque así dan la sensación de que son más realistas, ya que crean imperfecciones y arrugas; y las texturas planas que pueden originar algún problema si se ponen a algún objeto en 3 dimensiones. Al igual que los modelos, se pueden crear u obtener de páginas web.

Para la creación de estas se usan herramientas de diseño como Photoshop o Illustrator.

## Fondos e imágenes.

Las imágenes son elementos usados para representar visualmente objetos, escenarios o cualquier otro elemento del videojuego. Se suelen usar de muchas formas como elementos de la interfaz de usuario, iconos en el menú o en la interfaz, también se pueden usar como fondos prerenderizados para ahorrar el coste que supone tener un escenario 3D en memoria, como parte de algún elemento del escenario (un poster, un mapa de la zona, etc…).

En juegos 2D también se suelen usar imágenes llamadas en este caso sprites para representar a los personajes y enemigos como a objetos. En páginas como OpenGameArt.org, spriters-resource.com o spritedatabase.net podemos encontrar mucha cantidad de estos para poder usarlos en nuestros juegos.

## Tiles.

La traducción de tile al español es baldosa. Se llama así a la forma de construir escenarios para ciertos videojuegos, normalmente los juegos arcade, y consiste en usar una herramienta de edición gráfica o un software más específico para la creación de estos escenarios. Uno de los más famosos es Tiled, en el cual a través de una cuadrícula prediseñada por colores lisos o mezcla de colores en cada tile, se "pinta" un mapa para un videojuego.

Además del Tiled como herramienta con la cual crear los escenarios con tiles, también podemos incluir el RPG Maker o el GameMaker: Studio que también usan un sistema basado en tiles para el diseño de sus escenarios / mapas. Además, incluyen una serie de materiales y librerías que puedes usar para desarrollar tu videojuego.

En páginas como OpenGameArt.org se pueden encontrar diferentes colecciones de tilesets con diferentes licencias para poderlas incorporar en tus diseños.

## Música y sonido

En los videojuegos se suele usar la música y/o los efectos de sonido para crear un cierto ambiente y lograr que el jugador sienta que está dentro del juego.

A la hora de crear un videojuego, es importante saber que música usar. Algunas webs permiten encontrar música con licencias creative commons para usar en nuestro proyecto. Dichas webs suelen tener un motor de búsqueda que permite usar etiquetas (guitarra, ambiente....) para encontrar la música perfecta.

En el ámbito de la música de videojuegos, es de especial importancia la licencia de creative commons "Sin obra derivada". Algunos juegos tienen canciones que se reutilizan de un modo totalmente distinto por medio de un remix. Esta licencia impediría hacer esto con esas canciones, por lo que sería conveniente que esas canciones solo se usaran una vez, o siempre del mismo modo.

Casi igual de importante que la música son los efectos de sonido. En un videojuego, todo debe crearse desde cero, y eso incluye todo tipo de efectos de sonido, desde los pasos del personaje hasta los disparos.

Al igual que con la música, hay diversas webs dedicadas a recopilar efectos de sonido de muchos tipos, especialmente los más extravagantes (romper cristales, explosión de granada...). Aunque, también sobra decir que algunos efectos de sonido muy simples se pueden crear fácilmente por nuestra cuenta.

## Librerías y Motores3D.

Las librerías son algo importante a la hora de hacer un videojuego, ya que nos pueden solucionar muchos de los problemas que se nos plantean: la física, representar gráficos por pantalla, hacer interfaces, etc… Es por eso que se suelen usar librerías especializadas en alguno de estos ámbitos, ya sean de pago o libres, o incluso en cierto punto, cuando ninguna satisface las necesidades del videojuego planteado, se crean sus propias libreras para manejar alguna de estas características.

Las librerías de pago no suelen tener mucha información ya que pertenecen, la gran mayoría, a las empresas que las crean y algunas se comercializan para sacar beneficio. Por ello nos vamos a centrar en las librerías gratuitas y/o de código libre.

Sobre las librerías gratuitas tenemos desde las que trae los compiladores c y c++ (<stdio.h>, <string.h>, <stdlib.h>, <math.h>, etc…) hasta más específicas de dibujar gráficos por pantalla, tratar con música, trabajar con físicas…

Ahora nos centraremos en algunas librerías que hemos ido usando:

Box2D es una biblioteca libre que implementa un motor físico en dos dimensiones. Está programada en C++ por Erin Catto, y se distribuye bajo la licencia zlib.

Box2D es usado como motor de físicas para tratar la simulación de esta y resolver el tratado de colisiones entre objetos. Es una librería que ha sido usada en muchos juegos.

Ejemplos:

* 6 Dimensions
* Angry Birds
* Crayon Physics Deluxe
* Fantastic Contraption
* Incredibots
* Rolando
* Tiny Wings
* Transformice

Box2D también ha sido integrado en motores de videojuegos como Unity o Cocos2Dx.

Bullet es un motor de físicas que simula colisiones y detección de estas de cuerpos blandos y rígidos. Ha sido usada en videojuegos como también en efectos especiales en las películas. Erwin Coumans es su autor.

La licencia de la librería de Bullet es gratuita y open-source sujeta a los términos de la zlib License.

Bullet está integrado en el motor de físicas de Havok, el cual es un motor privado que presenta licencias de pago para poder usarlo. Este motor se ha usado en muchos juegos.

También está integrado en programas de 3D como Blender o Cinema4D e incluso se puede añadir a otras librerías como OGRE o Irrlicht.

Juegos con Bullet:

* *Toy Story 3: The Video Game publicado por Disney Interactive Studios.*
* *Grand Theft Auto IV, Grand Theft Auto V y Red Dead Redemption creado por Rockstar Games.*
* *Trials HD creado por RedLynx.*
* *Free Realms creado por Sony Online Entertainment.*
* *HotWheels: Battle Force 5.*
* *Gravitronix.*
* *Madagascar Kartz publicado por Activision.*
* *Regnum Online creado por ngd Studios. Un MMORPG que ha sustituido su motor físico por Bullet en su última gran actualización.*
* *Blood Drive publicado por Activision.*
* *Hydro Thunder Hurricane.*
* *Serie DiRT*
* *Rocket League.*

Havok es un motor físico que se usa para la simulación dinámica de objetos por lo que detecta colisiones, gravedad, masa y velocidad en tiempo real para recrear ambientes y lugares mucho más realistas.

Havok es uno de los motores más usados en la industria de los videojuegos por su gran versatilidad y creatividad a la hora de usarlo. Muchos juegos del mercado actual están hechos con este motor de físicas:

* Guild Wars 2
* Half-Life: Source
* Half-Life Deathmatch: Source
* Half-Life 2
* Half-Life 2: Deathmatch
* Half-Life 2: Episode One
* Half-Life 2: Episode Two
* Call of Duty 4: Modern Warfare
* Call of Duty: Black Ops II
* Call of Duty: Black Ops Declassified
* Company of Heroes
* Counter-Strike: Source
* Crackdown
* Crash Nitro Kart
* Darkwatch
* Dark Messiah
* Dark Souls

Irrlicht es un motor 3D gratuito y de código abierto basado en la licencia de zlib/libpng, escrito en C++, el cual puede ser usado tanto en C++ como con lenguajes .Net.

Irrlicht soporta el renderizado 3D mediante OpenGL, DirectX 8, 9 y 11 (con extensiones) también tiene librerías para renderizar materiales standard y uso de shaders para ellos.

Irrlicht es usado para representar objetos en un mundo 3D y hacer que interactúen con otros objetos. También tiene nodos para hacer animaciones de personajes e incluye una librería para crear interfaces 2D.

Irrlicht se ha usado en muchos juegos como motor gráfico como, por ejemplo:

* *Amulet of tricolor*, un juego 2D
* *Bolzplatz 2006* creado por Xenoage Software, un juego de fútbol.
* *Eve*, un simulador de robot creado por el *Biologically Inspired Robotics Group* (BIRG) en el *Swiss Federal Institute of Technology* de Lausanne.
* *Galactic Dream: Rage of War*, un juego de estrategia espacial comercial creado por Evolution Vault
* Gekkeiju Online, a gratuitous fantasy multi user (?) role playing game (un juego de fantasia y rol (RPG) multijugador gratuito)
* H-Craft Championship, un juego de carreras Sci-Fi open source.
* Minetest, un juego de mundo abierto similar a Minecraft
* Octodad: Dadliest Catch, un cómico simulador de pulpo.
* *Sokoban Challenge*, un remake en 3D del clásico juego de puzzles Sokoban para Android
* SuperTuxKart (desde la versión 0.7), un juego de carreras de karts en 3D; desde la versión 0.8.2 usa una versión muy modificada de Irrlicht llamada "Antarctica"

## Herramientas

Las herramientas como recurso para el desarrollo de software concretamente de videojuegos. Como es normal cada obra necesita de sus utensilios para poder efectuarse en el caso de la creación o diseño de un videojuego es lo mismos los desarrollares usan una serie de programas para poder materializar sus ideas. Este software tiene diferentes usos y hay una gran cantidad de ellos.

El software empleado para el desarrollo existe en diferentes licencias (de las ya comentadas) y cada equipo de desarrollo debe elegir cual elegir tanto por el coste de estas licencias o por las facilidades que puedan aportar. Por ejemplo, hay programas libres que permiten hacen muchas cosas que como desarrollador necesitas, pero igual no facilitan la exportación o que no te permiten conseguir resultados profesionales en el diseño o para conseguir estos hay que invertir muchas más horas y por lo tanto igual sería más conveniente utilizar uno de pago que te facilite esas tareas.

Como he comentado existen muchos tipos de programas y se utilizan muchos diferentes. Estos programas son utilizados para el modelaje 3d o tratamiento de imágenes, como entornos de trabajo integrados donde le es más accesible al desarrollador realizar su trabajo o para muchas más cosas. Ahora vamos a nombrar algunos de los programas más utilizados y comentar un poco su uso.

* Programas para la edición fotográfica o el dibujo.

1. Gimp: programa de edición de imágenes digitales en forma de mapa de bits, tanto dibujos como fotografías. Forma parte del proyecto GNU y está disponible bajo la Licencia pública general de GNU y GNU Lesser General Public License.
2. Adobe Photoshop: editor de gráficos rasterizados desarrollado por Adobe Systems Incorporated. Usado principalmente para el retoque de fotografías y gráficos, su nombre en español significa literalmente "taller de fotos". Dispone una licencia de uso te tipo propietario.
3. Adobe illustrator: es un editor de gráficos vectoriales en forma de taller de arte que trabaja sobre un tablero de dibujo y está destinado a la creación artística de dibujo y pintura para ilustración. Es desarrollado y comercializado por Adobe Systems.
4. Inkscape: editor de gráficos vectoriales gratuito y de código libre. Inkscape puede crear y editar diagramas, líneas, gráficos, logotipos, e ilustraciones complejas. El formato principal que utiliza el programa es Scalable Vector Graphics (SVG). Licencia pública general de GNU.

* Programas para el modelaje 3D.

1. Autodesk 3ds-max: es un programa de creación de gráficos y animación 3D para el sistema operativo Windows. Licencia propietaria.
2. Autodesk Maya: es un programa informático dedicado al desarrollo de gráficos 3D por ordenador, efectos especiales y animación. El programa posee diversas herramientas para modelado, animación, renderización, simulación de ropa y cabello, dinámicas (simulación de fluidos), entre otras. Licencia propietaria.
3. Blender: es un programa dedicado especialmente al modelado, iluminación, renderizado, animación y creación de gráficos tridimensionales. También de composición digital utilizando la técnica procesal de nodos, edición de vídeo, escultura (incluye topología dinámica) y pintura digital. En Blender, además, se puede desarrollar vídeojuegos ya que posee un motor de juegos interno. Su licencia es GPL.
4. ZBrush: software de modelado 3d, escultura y pintura digital. Licencia propietaria.

* Edición y creación de sonido y Música.
  1. Pro Tools: es una estación de trabajo de audio digital o AEE, una plataforma de grabación, edición y mezcla multipista de audio y midi, que integra hardware y software. Licencia de pago y uso de llave validadora USB ILok.
  2. Ardour: programa de grabación multipista de audio y MIDI a disco duro. Generalmente es usado como un entorno de masterizado de audio. Código abierto, distribuido bajo la licencia GNU General Public License.
  3. Cubase: serie de aplicaciones informáticas para editar audio digital, MIDI y un secuenciador de música. Licencia propietaria.
* Herramientas de tratamiento, edición y posproducción de videos.
  1. Sony vegas: es un Sistema de edición no lineal diseñado para PC. Ofrece edición de vídeo y audio en tiempo real en múltiples pistas, soporte de audio en 24-bit/192 kHz, mezclas de audio en calidad Dolby Digital, y soporte para complementos tales como DirectX y VST. Licencia propietaria.
  2. Adobe aftereffects: es una aplicación que tiene forma de estudio destinado para la creación o aplicación en una composición, así como realización de gráficos profesionales en movimiento y efectos especiales, que desde sus raíces han consistido básicamente en la superposición de capas. Licencia propietaria.
* Entornos de desarrollo integrados (IDEs).
  1. NetBeans: es un entorno de desarrollo integrado libre, hecho principalmente para el lenguaje de programación Java. Permite que las aplicaciones sean desarrolladas a partir de un conjunto de componentes de software llamados módulos. También admite otros leguajes de programación. Sus licencias son CDDL y GPL2.
  2. Eclipse: es una plataforma de software compuesto por un conjunto de herramientas de programación de código abierto multiplataforma para desarrollar el proyecto. Esta plataforma, típicamente ha sido usada para desarrollar entornos de desarrollo integrados, pero también se puede usar para otros tipos de aplicaciones cliente. Su licencia es Eclipse Public License.
  3. Microsoft Visual Studio: es un entorno de desarrollo integrado para sistemas operativos Windows. Soporta múltiples lenguajes de programación tales como C++, C#, Visual Basic .NET, F#, Java, Python, Ruby, PHP. Visual Studio permite a los desarrolladores crear sitios y aplicaciones web, así como servicios web en cualquier entorno que soporte la plataforma .NET. A partir del 2013 tiene dos licencias una llamada Community gratuita y otra de pago.
  4. Code::Blocks: es un entorno de desarrollo integrado libre y multiplataforma para el desarrollo de programas en lenguaje C y C++. Está basado en la plataforma de interfaces gráficas WxWidgets, lo cual quiere decir que puede usarse libremente en diversos sistemas operativos, y está licenciado bajo la Licencia pública general de GNU.
* Editores de texto.
  1. Sublime Text es un editor de texto y editor de código fuente está escrito en C++ y Python para los plugins. Se puede descargar y evaluar de forma gratuita pero su uso no es de software libre o de código abierto y se debe obtener una licencia para su uso continuado, aunque la versión de evaluación es plenamente funcional y no tiene fecha de caducidad.
  2. Atom: editor de texto que se auto completa con plugins diseñado por github. Es un producto OpenSource y su licencia está basada en una propia de GitHub.
  3. Notepad++: es un editor de texto y de código fuente libre con soporte para varios lenguajes de programación. De soporte nativo a Microsoft Windows. Se distribuye bajo los términos de la Licencia Pública General de GNU.
* Editores “Multiherramienta” y motores de videojuego.
  1. GameMaker: Studio es una herramienta de desarrollo rápido de aplicaciones, basada en un lenguaje de programación interpretado y un kit de desarrollo de software (SDK) para desarrollar videojuegos, creado en el lenguaje de programación Delphi, y orientado a usuarios novatos o con pocas nociones de programación. El programa es gratuito, aunque existe una versión comercial ampliada con características adicionales.
  2. Construct: es un editor de juegos 2D basado en HTML5, desarrollado por Scirra Ltd. Está dirigido principalmente a no programadores, permitiendo la creación rápida de juegos en una forma de arrastrar y soltar usando un editor visual y un sistema lógico basado en el comportamiento. Licencia propietaria.
  3. Unity: es un motor de videojuego multiplataforma creado por Unity Technologies. Unity está disponible como plataforma de desarrollo para Microsoft Windows, OS X. La plataforma de desarrollo tiene soporte de compilación con diferentes tipos de plataformas. Unity incorpora muchas otras herramientas para su uso en el desarrollo del videojuego. Hay dos licencias principales para desarrolladores: Unity personal y Unity Professional. La versión Pro tiene características adicionales, tales como render a textura, determinación de cara oculta, iluminación global y efectos de postprocesamiento. La versión gratuita, por otro lado, muestra una pantalla de bienvenida (en juegos independientes) y una marca de agua (en los juegos web) que no se puede personalizar o desactivar.
  4. Unreal Engine: es un motor de juego de PC y consolas creados por la compañía Epic Games. La versión actual está programada en C++ y es compatible tanto con OpenGL como DirectX 11 y 12, siendo compatible con varias plataformas como PC (Microsoft Windows, GNU/Linux), Apple Macintosh (Mac OS X) y la mayoría de consolas (Xbox One y PlayStation 4). Unreal Engine también ofrece varias herramientas de gran ayuda para diseñadores y artistas facilitando la visualización de entornos o de construcciones. Su licencia es gratuita, pero a partir de cierta cantidad de beneficios, la empresa que lo use debe pagar un 5% a Epic Games.
* Otras herramientas.
  1. GitHub Desktop: Plataforma de escritorio para seguir tus desarrollos en GitHub y además facilita el manejo de git.
  2. Tiled: editor de mapas por tiles. De libre uso.

# Software libre

Ya hemos hablado antes del software libre. Ahora vamos a profundizar un poco más.

El software libre es el que respeta todas las libertades de los usuarios, por lo que ellos pueden usarlo, modificarlo y compartirlo como les plazca. Por lo general, el software es libre si tiene las siguientes cuatro libertades esenciales:

*-La libertad de ejecutar el programa como se desea, con cualquier propósito (libertad 0).*

*-La libertad de estudiar cómo funciona el programa, y cambiarlo para que haga lo que usted quiera (libertad 1). El acceso al código fuente es una condición necesaria para ello.*

*-La libertad de redistribuir copias para ayudar a su prójimo (libertad 2).*

*-La libertad de distribuir copias de sus versiones modificadas a terceros (libertad 3). Esto le permite ofrecer a toda la comunidad la oportunidad de beneficiarse de las modificaciones. El acceso al código fuente es una condición necesaria para ello.*

Un software que no cumpla estas condiciones no es libre, sino privativo. Más adelante hablaremos del software privativo.

Como se puede observar, el concepto clave del software libre es la libertad que proporciona a todos los que lo usan. Esto quiere decir que cualquiera puede coger un programa, mejorarlo, y distribuir dicha versión mejorada, para ayudar a los demás. Otras personas pueden a su vez coger dicha versión y mejorarla aún más para trabajo propio.

Hay que tener cuidado a la hora de hacer que un software sea libre. El modo más sencillo sería hacerlo de dominio público, sin derechos de autor. Aunque esto le daría mucha libertad al software, no se garantiza la libertad de futuras versiones mejoradas, por lo que una empresa podría coger el programa original, mejorarlo y protegerlo con copyright para venderlo, perdiendo los usuarios de dicha versión la libertad que tenía originalmente.

Para impedir que esto pase, se ha creado el concepto de copyleft, como contrapartida al copyright. El copyleft hace que el software pueda distribuirse y modificarse, pero cualquier versión distribuida debe contar con las mismas libertades que la versión original.

# Software privativo

El software privativo es lo opuesto del software libre explicado anteriormente. El software privativo no tiene todas las libertades esenciales necesarias para ser libre. El software privativo está protegido por los derechos de autor, mencionados al principio de este documento.

Por lo general, la principal diferencia entre el software libre y el privativo es que el software privativo limita las posibilidades de distribución y mejora. El software privativo no suele permitir acceso al código fuente, lo que impide cualquier mejora o corrección del software, quedando el usuario a merced del desarrollador. Además, hay límites sobre la distribución del software. El usuario debe tener una licencia para poder usar el software. Las licencias pueden variar en sus características, pero por lo general tienen un coste y limitan el uso del software a un cierto número de ordenadores (algunas veces lo limita a un solo ordenador). Para poder usar el software en otro ordenador una vez alcanzado el límite, o bien hay que desactivar el software en uno de los ordenadores, o bien hay que comprar otra licencia.

# Nuevas formas de uso

Hoy en día están apareciendo nuevas formas de uso de software, más permisivas y sencillas de utilizar.

Ya hemos hablado antes de software libre y copyleft. Otro concepto interesante son las licencias Creative Commons. Las licencias Creative Commons están pensadas para compartir fácilmente contenidos en Internet manteniendo un cierto control sobre el uso que se le da. Creative Comons proporciona cuatro licencias, que se pueden combinar de distintas formas para obtener la licencia adecuada para nosotros:

*-Reconocimiento: En cualquier explotación de la obra autorizada por la licencia hará falta reconocer la autoría.*

*-No Comercial: La explotación de la obra queda limitada a usos no comerciales.*

*-Sin obras derivadas: La autorización para explotar la obra no incluye la transformación para crear una obra derivada.*

*-Compartir Igual: La explotación autorizada incluye la creación de obras derivadas siempre que mantengan la misma licencia al ser divulgadas.*

Como se puede ver, las licencias Creative Commons están pensadas para poder ser entendidas fácilmente, sin necesidad de tener conocimientos legales. Eso permite que cualquier usuario medio de Internet pueda usarlas con facilidad.

# Bibliografía

<https://es.wikipedia.org/wiki/Biblioteca_estándar_de_C>

<https://es.wikipedia.org/wiki/Box2D>

<https://en.wikipedia.org/wiki/Box2D>

<https://es.wikipedia.org/wiki/Irrlicht>

<http://crai.ub.edu/es/que-ofrece-el-crai/derechos-de-autor-y-propiedad-intelectual-y-acceso-abierto/uso-recursos-informacion-ajenos>

<http://www.uc3m.es/ss/Satellite/Biblioteca/es/TextoDosColumnas/1371213998525/>

<http://www.irrlicht3d.org/wiki/index.php?n=Main.IrrlichtPoweredGames>

<https://es.wikipedia.org/wiki/Licencia_de_software>

<https://en.wikipedia.org/wiki/Bullet_(software)>

<https://es.wikipedia.org/wiki/Havok_(software)>

<https://es.wikipedia.org/wiki/Software_libre>

<https://es.wikipedia.org/wiki/Mapeado_de_texturas>

<http://zbrushcore.com/>

<http://www.ardour.org/>

<https://atom.io/>

<https://desktop.github.com/>

<https://www.unrealengine.com/what-is-unreal-engine-4>

https://www.sublimetext.com/3

<https://www.microsoft.com/es/sam/intproperty.aspx>

<http://www.alvarezramosabogados.com/la-propiedad-intelectual-del-software/>

<http://www.protectia.eu/patentes/se-puede-patentar-un-software/>

<https://www.gnu.org/philosophy/free-sw.es.html>

<https://www.gnu.org/copyleft/copyleft.html>

<https://jalvarezm.wordpress.com/2006/09/30/el-software-privativo/>

<http://karlospg1.blogspot.es/>

<http://www.definicionabc.com/tecnologia/software-propietario.php>

<http://es.creativecommons.org/blog/licencias/>