



International Journal of Information Science
and Technological Applications-UAS

IJISTA

ISSN: 3122-4474

<https://revistas.uas.edu.mx/index.php/IJISTA>

Junio 2026

Vol. II.

Número. II



Herramientas gratuitas para desarrolladores principiantes de videojuegos




Free tools for beginner video game developers


Emmet Crespo-Rojas¹, José Armando Villalobos-Puga¹, Gerson Gibran Ramirez-Holguin¹,
Félix David Duran-Hernandez¹, Luis David Sanchez-Fragoso¹





¹Facultad de Informática Mazatlán, Universidad Autónoma de Sinaloa, México.




 <https://orcid.org/0009-0008-8462-2291>
joarvipu2003@gmail.com

 <https://orcid.org/0009-0000-6949-6743>
gersonxw@gmail.com

 <https://orcid.org/0009-0000-8384-8944>
duranfelix53@gmail.com

 <https://orcid.org/0009-0005-5170-0255>
ls9860866@gmail.com

Autor de Correspondencia: Crespo-Rojas Emmet, emmetv.cr@gmail.com

 <https://orcid.org/0009-0002-9328-1763>



CREATIVE COMMONS

Recibido: abril 2026
Publicado: junio 2026

Este es un artículo de acceso abierto distribuido bajo los términos de la Licencia Creative Commons Atribución-No Comercial-Compartir igual (CC BY-NC-SA 4.0), que permite compartir y adaptar siempre que se cite adecuadamente la obra, no se utilice con fines comerciales y se comparta bajo las mismas condiciones que el original.

Resumen:

El desarrollo de videojuegos ha experimentado una transformación en los últimos años. Este proceso se ha visto favorecido por la disponibilidad de herramientas gratuitas y de código abierto que cubren todos los aspectos del proceso creativo. Este artículo analiza herramientas accesibles para desarrolladores principiantes, abarcando motores de videojuegos, software de diseño visual, composición de audio y bibliotecas de recursos (assets) libres de regalías, todas ellas bajo licencias del Instituto Tecnológico de Massachusetts (por sus siglas en inglés, MIT) y/o Licencia Pública General de GNU (por sus siglas en inglés, GNU GPL) que garantizan su uso sin restricciones comerciales independientemente del alcance del proyecto. A diferencia de soluciones propietarias como Unity o Unreal Engine, cuyo modelo de licenciamiento puede representar una barrera económica ante el éxito comercial, las herramientas aquí presentadas reducen dicha barrera, haciendo del tiempo y esfuerzo del desarrollador uno de los principales recursos requeridos. Los trabajos relacionados revisados respaldan la viabilidad del desarrollo independiente de videojuegos sin inversión económica, y evidencian que la ingeniería de software para videojuegos es un campo en crecimiento y maduración. Se concluye que existe un ecosistema completo y funcional de herramientas gratuitas que permite a desarrolladores principiantes llevar a cabo proyectos de videojuegos de principio a fin.

Palabras Clave:

Desarrollo de videojuegos, herramientas de código abierto, licencias de software libre, desarrollo independiente, motores de videojuegos.

Abstract:

Video game development has undergone a transformation in recent years. This process has been favored by the availability of free and open-source tools that cover all aspects of the creative process. This article analyzes accessible tools for beginner developers, encompassing game engines, visual design software, audio composition, and royalty-free resource (assets) libraries, all under Massachusetts Institute of Technology (MIT) and/or GNU General Public License (GNU GPL) licenses that guarantee unrestricted commercial use regardless of the project's scope. Unlike proprietary solutions such as Unity or Unreal Engine, whose licensing models may impose economic barriers upon commercial success, the tools presented here reduce such barriers, making time and developer effort the main resources required. Related works reviewed support the feasibility of independent video game development without financial investment, and demonstrate that software engineering for video games is a growing and maturing field. It is concluded that a complete and functional ecosystem of free tools exists, enabling beginner developers to carry out video game projects from start to finish.

Keywords:

Video game development, open-source tools, free software licenses, indie development, game engines.

1. Introducción

La industria de los videojuegos ha experimentado un crecimiento dentro de la economía creativa global, con una tasa de crecimiento sostenida que atrae tanto a grandes corporaciones como a desarrolladores independientes [1]. Esta expansión no ha sido homogénea: mientras que los grandes estudios continúan apostando por producciones de alto presupuesto conocidas como títulos AAA (triple A), el segmento independiente ha experimentado un auge particularmente notable, impulsado por la proliferación de plataformas de distribución digital como Steam, itch.io y las tiendas de aplicaciones móviles [1].

Historicamente, el desarrollo de videojuegos ha sido dominado por grandes estudios con recursos para sustentar una producción prolongada, un equipo numeroso y múltiples capas de burocracia corporativa. Este modelo, imponía barreras de entrada significativas para desarrolladores sin respaldo institucional. Sin embargo, la evolución tecnológica y la madurez alcanzada por herramientas de desarrollo accesibles han transformado este panorama, haciendo posible que equipos pequeños o desarrolladores individuales produzcan y distribuyan videojuegos a escala global [2].

En este contexto, el desarrollo independiente de videojuegos ha ganado una relevancia creciente, tanto desde el punto de vista cultural como económico, un fenómeno que se observa en la aceptación de los videojuegos como tópico de periodismo especializado [3]. Títulos como *Stardew Valley*, *Hollow Knight*, *Undertale* y, más recientemente, *Balatro* desarrollado íntegramente con *Love2D* han demostrado que la calidad y el éxito comercial no son exclusivos de los grandes estudios. No obstante, uno de los desafíos persistentes para los desarrolladores independientes es la selección de herramientas adecuadas a sus capacidades técnicas, artísticas y, especialmente, financieras. En este sentido, el modelo de licenciamiento del software de desarrollo se convierte en un factor crítico, pues puede representar un riesgo económico significativo conforme el proyecto escala en popularidad.

A diferencia de soluciones comerciales como Unity o Unreal Engine, cuyo modelo de licenciamiento puede generar costos inesperados o modificaciones unilaterales de términos una vez que un proyecto alcanza determinado nivel de éxito, existe en la actualidad un ecosistema de herramientas de código abierto o de uso gratuito que garantizan condiciones estables independientemente del alcance comercial del proyecto [4]. Parte de estas herramientas opera bajo la licencia MIT, que permite su uso, redistribución y modificación sin restricciones; otra parte se distribuye bajo licencias GNU, que garantizan su uso completamente gratuito sin limitaciones comerciales. A diferencia de alternativas propietarias, estas herramientas garantizan que los

términos no cambiarán independientemente del alcance del proyecto.

Estas herramientas de acceso libre cubren gran parte del flujo de trabajo en el desarrollo de videojuegos. Para el desarrollo del juego sin necesidad de construir un motor desde cero, se emplean herramientas como Godot, un motor con licencia MIT que ha experimentado un crecimiento significativo en su adopción durante los últimos años y que soporta de forma nativa C++, C# y GDScript, su propio lenguaje de scripting inspirado en Python; así como *Love2D*, un framework basado en Lua que ha probado su viabilidad en proyectos comerciales de alto impacto [4]. En el ámbito visual, la elección de herramientas depende del estilo artístico que se busca lograr: *Krita* para ilustración 2D, *Pixelorama* para arte en pixel art, *Blockbench* para modelado 3D low poly, y *Blender* para producciones de mayor complejidad geométrica [5]. En el ámbito del audio, los desarrolladores utilizan herramientas como *Audacity* para la edición y producción de efectos de sonido, y en *FamiStudio* o *Bosca Ceoil Blue* para la composición de música original. Adicionalmente, existen repositorios de assets de uso libre como los *Sonniss GDC Bundles*, que ponen a disposición de la comunidad efectos de sonido libre de regalías (royalty-free) para su integración en proyectos comerciales.

A pesar de la relevancia práctica de este ecosistema de herramientas, existe una carencia notable de estudios académicos que lo documenten y analicen de manera sistemática. La mayoría de los recursos disponibles se concentran en tutoriales en línea o en foros de comunidades de práctica, pero pocas investigaciones formales han abordado la caracterización y comparación de estas alternativas en el contexto de desarrollo independiente. Este trabajo busca contribuir a subsanar esa brecha, ofreciendo un panorama estructurado del ecosistema de herramientas libres y gratuitas disponibles para el desarrollo independiente de videojuegos, con atención a sus modelos de licenciamiento, capacidades técnicas y casos de uso representativos.

El resto del artículo se organiza de la siguiente manera: la sección 2 presenta una revisión de la literatura relevante; la sección 3 describe la metodología empleada; la sección 4 presenta los resultados; la sección 5 expone el análisis de resultados; y la sección 6 presenta las conclusiones.

2. Trabajos Relacionados

Los trabajos relacionados se clasifican en cuatro categorías para su análisis comparativo que estructuran el estado del arte sobre el ecosistema de herramientas para el desarrollo independiente de videojuegos: (1) la industria y cultura del desarrollo independiente, (2) la ingeniería de software aplicada a videojuegos, (3) las

herramientas libres y de código abierto, y (4) las herramientas en contextos educativos y de accesibilidad.

2.1. Industria y Cultura del Desarrollo Independiente de Videojuegos

La consolidación del desarrollador indie como figura relevante en la industria ha motivado un creciente cuerpo de investigación. Garda y Grabarczyk [1] argumentan que la independencia en el desarrollo de videojuegos no constituye una condición binaria, sino un espectro que involucra dimensiones económicas, creativas y distributivas, lo que implica que la elección de herramientas de desarrollo tiene implicaciones que van más allá de lo meramente técnico. En línea con esto, Garda y Grabarczyk [1] documentan la precariedad estructural que caracteriza al sector indie, señalando que estos desarrolladores operan en condiciones de incertidumbre financiera que los expone particularmente a variaciones en los costos operativos, incluidos los modelos de licenciamiento de herramientas.

Desde una perspectiva cultural e histórica, Sotamaa [2] analiza las culturas de desarrollo de videojuegos en el contexto finlandés mediante un enfoque de tres capas: regional, laboral y de estudio que permite contextualizar cómo factores históricos, regulatorios y organizacionales moldean las prácticas cotidianas de producción. El autor señala que el modelo clásico de publicación dependía fuertemente de herramientas propietarias, lo que restringía el acceso de estudios pequeños a los canales controlados por los grandes actores de la industria. De forma complementaria, Berg Marklund [6] et al. realizan una revisión de investigación empírica sobre desarrollo de videojuegos, identificando que los estudios académicos disponibles aún no cubren de manera exhaustiva las dimensiones prácticas del proceso productivo en contextos de pequeña escala. Por su parte, Moradzadeh [7] et al. presentan una revisión sistemática de la literatura sobre videojuegos en el ámbito del trabajo cooperativo asistido por computadora (CSCW), evidenciando que los videojuegos constituyen un dominio de estudio multidimensional cuyas dinámicas de colaboración y producción merecen atención académica continua.

En cuanto a los aspectos legales y contractuales del sector, Gonçalves y Paim [8] analizan la integración de licencias mecánicas para videojuegos en contextos normativos de distintos países, destacando la complejidad que enfrentan los desarrolladores independientes al gestionar derechos de uso de contenido, lo que refuerza la importancia de contar con herramientas con modelos de licenciamiento claros y accesibles.

2.2. Ingeniería de Software para Videojuegos y Motores de Juego

El desarrollo de videojuegos ha madurado como disciplina técnica. Actualmente constituye un campo con características propias dentro de la ingeniería de software. Chueca et al. [9] presentan una revisión sistemática de la literatura sobre Ingeniería de Software para Videojuegos (GSE, por sus siglas en inglés) a escala industrial, confirmando que se trata de un dominio independiente y en crecimiento, con una producción investigativa cuatro veces mayor que antes de 2009, y con temas emergentes como arquitectura, diseño y gestión de calidad. Scacchi [10] ofrece una panorámica de las prácticas y tecnologías en la ingeniería de software para videojuegos, describiendo una amplia variedad de kits de desarrollo, motores y marcos de trabajo tanto comerciales como de código abierto orientados a distintos géneros y plataformas. El autor identifica que los motores de videojuegos representan uno de los casos de éxito más representativos en materia de reutilización de software, aunque también advierte que algunos estudios evitan adoptar motores comerciales disponibles por temor a que sus características limiten la originalidad del producto final.

Desde una perspectiva comparativa, Politowski [11] et al. exploran si los motores de videojuegos pueden considerarse marcos de software (frameworks) a partir de tres perspectivas: literatura, de código y humana. Analizando 282 motores de código abierto en GitHub, concluyen que existen diferencias cualitativas pero no cuantitativas significativas respecto a los frameworks tradicionales, siendo los motores generalmente más grandes, complejos y con comunidades más pequeñas. En una línea complementaria, Ullmann [12] et al. realizan un análisis anatómico comparativo de motores de videojuegos de código abierto, identificando patrones arquitectónicos comunes y diferencias estructurales entre distintas categorías de motores. Toftedahl [4] y Engström, por su parte, proponen una taxonomía de motores de videojuegos dividida en cuatro tipos: motor central, motor de juego, motor de propósito general y motor de propósito especial y analizan su adopción en plataformas de distribución como Steam e Itch.io, encontrando que Unity concentra el 47.3% de los juegos independientes en esta última plataforma.

Sobota y Pietriková [13] abordan el rol de los motores de videojuegos tanto en el proceso de desarrollo como en la enseñanza de ciencias de la computación, destacando que su uso en entornos educativos facilita la comprensión de conceptos complejos de ingeniería de software. En el ámbito de las pruebas de software, Politowski, Petrillo y Guéhéneuc [14] realizan un estudio exhaustivo sobre las prácticas de prueba en videojuegos, identificando que este es uno de los procesos menos estandarizados en la industria y que la mayoría de las herramientas disponibles aún no están integradas de

forma nativa en los principales motores. Asimismo, Kamienski y Bezemer [15] realizan un estudio empírico sobre el uso de sitios de preguntas y respuestas como Stack Overflow por parte de desarrolladores de videojuegos, encontrando que estas comunidades constituyen un recurso fundamental de soporte técnico para quienes trabajan con motores tanto comerciales como de código abierto.

2.3. Herramientas Libres, de Código Abierto y Contenido Procedural

La investigación específica sobre herramientas libres para el desarrollo de videojuegos es aún escasa, pero creciente. Hall, Dogbe y Said en [16]. presentan una revisión sistemática de la literatura sobre desarrollo de videojuegos de código abierto, identificando las principales herramientas disponibles, sus casos de uso y los factores que inciden en su adopción por parte de desarrolladores independientes. Los autores señalan que, a pesar del crecimiento sostenido del ecosistema de herramientas libres, la literatura académica sobre sus características, limitaciones y modelos de gobernanza sigue siendo insuficiente.

En el ámbito de la generación de contenido, Zamorano, Cetina y Sarro [17] presentan una revisión exhaustiva de técnicas de generación procedural de contenido asistida por búsqueda (SBPCG) aplicadas a videojuegos, clasificando las herramientas y enfoques disponibles según el tipo de contenido generado. Esta área resulta relevante para el ecosistema de herramientas libres, dado que muchos de estos métodos se implementan como bibliotecas de código abierto. Sweetser [18] realiza un análisis preliminar del estado de la investigación sobre la integración de modelos de lenguaje de gran escala (LLMs) en videojuegos, abarcando desde la generación de contenido narrativo hasta la asistencia en el proceso de desarrollo mismo, lo cual abre nuevas posibilidades para herramientas accesibles basadas en inteligencia artificial. Hu [19] et al. complementan este panorama con una revisión sobre el uso de videojuegos como entornos de entrenamiento y evaluación para sistemas de inteligencia artificial, destacando la importancia de plataformas abiertas y reproducibles para la investigación en este campo.

2.4. Herramientas en Contextos Educativos y Accesibilidad

La literatura sobre tecnología educativa aporta evidencia relevante acerca del impacto que tiene la disponibilidad de herramientas accesibles en la participación y producción creativa. Christopoulos, Conrad y Shukla [5] analizan cómo entornos virtuales accesibles y de bajo costo inciden directamente en los niveles de participación y producción creativa de los

estudiantes, un hallazgo extrapolable al contexto del desarrollo independiente de videojuegos.

De manera más específica, Sáez-López et al. [20] evalúan una propuesta pedagógica para la formación inicial de docentes que combina el entorno de programación visual por bloques Scratch con el motor Unity, utilizando actividades progresivas en C# como puente entre ambos entornos. Los resultados muestran mejoras estadísticamente significativas tanto en el rendimiento como en las actitudes hacia la programación, lo que refuerza la viabilidad de introducir herramientas de desarrollo de videojuegos incluyendo el uso de motores gratuitos en contextos educativos formales. Este estudio resulta especialmente relevante para el presente trabajo, ya que Unity constituye uno de los motores gratuitos de mayor adopción en el contexto indie y su uso en educación ilustra el potencial democratizador de las herramientas de acceso libre.

De la revisión presentada se desprende que, si bien existe un conjunto amplio de investigaciones sobre motores comerciales, culturas de desarrollo y prácticas de ingeniería de software para videojuegos, la caracterización sistemática del ecosistema de herramientas libres y gratuitas con atención a sus modelos de licenciamiento, capacidades técnicas por área de producción y pertinencia para el contexto indie actual constituye un vacío que este estudio busca contribuir a cerrar.

3. Metodología

El presente trabajo adopta un enfoque de revisión estructurada y análisis comparativo, orientado a caracterizar el ecosistema de herramientas libres y gratuitas disponibles para el desarrollo independiente de videojuegos. La metodología se organiza en tres etapas: definición de criterios de selección, recopilación y clasificación de herramientas, y análisis comparativo por área de producción.

3.1. Criterios de Selección

Para delimitar el conjunto de herramientas a analizar se establecieron los siguientes criterios de inclusión:

- La herramienta debe ser de uso gratuito, ya sea bajo una licencia de código abierto (MIT, GNU GPL, Apache) o mediante un modelo freeware sin restricciones comerciales.
- Debe cubrir al menos una de las áreas del flujo de trabajo en el desarrollo de videojuegos: motor de juego, arte 2D, arte 3D, audio o gestión de assets.
- Debe contar con documentación oficial activa y una comunidad de usuarios verificable, a través de repositorios públicos o foros oficiales.

- Debe haber sido utilizada en al menos un proyecto de distribución pública, en plataformas como Steam, itch.io, repositorios públicos o el sitio web oficial del proyecto, ya sea comercial o no comercial.
- Se excluyeron herramientas con modelos freemium que imponen restricciones comerciales al superar ciertos umbrales de ingresos, así como aquellas cuya documentación oficial se encuentra descontinuada o sin mantenimiento activo.

3.2. Recopilación y Clasificación de Herramientas

La recopilación de herramientas se realizó a partir de tres fuentes principales: literatura académica revisada en la sección 2, documentación oficial de plataformas de distribución de videojuegos independientes (Steam e itch.io), y repositorios públicos en GitHub. Las herramientas identificadas fueron clasificadas según el área de producción a la que corresponden, dando lugar a cinco categorías, como se muestra en la Tabla 1.

Tabla 1. Categorías de herramientas libres para el desarrollo independiente de videojuegos.

Fuente: elaboración propia.

Categoría	Descripción	Ejemplos Identificados
Motor de juego	Entorno principal de desarrollo y ejecución	Godot, Love2D
Arte 2D	Ilustración, animación y pixel art	Krita, Pixelorama
Arte 3D	Modelado y animación tridimensional	Blender, Blockbench
Audio	Edición de sonido y composición musical	Audacity, FamiStudio, Bosca Ceoil Blue
Assets libres	Repositorios de recursos reutilizables	Sonniss GDC Bundles

3.3. Análisis Comparativo

Se analizó cada herramienta con base en cuatro dimensiones: (1) modelo de licenciamiento y sus implicaciones comerciales, (2) capacidades técnicas principales, (3) plataformas de destino soportadas, y (4) casos de uso representativos en proyectos publicados. Este análisis se sintetiza en la sección 4 mediante tablas comparativas y se complementa con una discusión sobre las fortalezas y limitaciones de cada herramienta en el contexto del desarrollo indie. El flujo general de la metodología se ilustra en la Fig. 1.



Figura 1. Flujo metodológico del estudio. Elaboración propia

3.4. Evaluación de Modelos de Licenciamiento

Dado que uno de los ejes centrales de este trabajo es el impacto del licenciamiento en la viabilidad económica del desarrollo independiente, se realizó un análisis específico de los términos legales de cada herramienta identificada, evaluando cuatro aspectos: restricciones de uso comercial, condiciones de redistribución, restricciones sobre la modificación del código fuente y los términos ante el éxito comercial del proyecto. Para ello se consultaron directamente los textos de licencia disponibles en los repositorios oficiales y en la documentación publicada por cada proyecto. Las licencias fueron clasificadas en tres categorías según el grado de libertad que otorgan sobre el uso comercial y la redistribución de proyectos y código derivado.

Las licencias permisivas, como MIT y Apache 2.0, permiten el uso, modificación y distribución comercial sin restricciones ni obligación de publicar el código fuente del producto derivado. Este tipo de licencia representa la opción de menor riesgo contractual para el desarrollador independiente, ya que garantiza estabilidad en los términos independientemente del éxito comercial del proyecto.

Las licencias copyleft, como GNU GPL y GNU LGPL, garantizan el uso completamente gratuito sin limitaciones comerciales, pero imponen condiciones sobre la redistribución del código. En la práctica, esto significa que, si el desarrollador modifica la herramienta y distribuye el resultado, está obligado a publicar esas modificaciones bajo la misma licencia.

Los modelos freeware sin código abierto corresponden a herramientas gratuitas cuyo código fuente no está disponible públicamente. Si bien no implican un costo directo, generan una dependencia total del proveedor para actualizaciones, correcciones y continuidad del proyecto, lo que representa un riesgo a largo plazo comparable al de las soluciones comerciales.

Este análisis permite al desarrollador independiente no solo identificar qué herramientas son gratuitas en el momento de adoptarlas, sino también anticipar qué condiciones seguirán vigentes conforme el proyecto escale en popularidad o ingresos, reduciendo el riesgo de enfrentar modificaciones unilaterales de términos como las documentadas en herramientas comerciales de amplia adopción [4].

4. Resultados

A continuación, se presentan los datos obtenidos del análisis comparativo de las herramientas libres identificadas, organizados por categoría de producción. Para cada categoría se sintetizan el modelo de licenciamiento, las capacidades técnicas principales, las plataformas de destino soportadas y los casos de uso representativos en proyectos publicados.

Tabla 2. Comparativa de motores de videojuego libres para desarrollo independiente.

Fuente: elaboración propia.

Herramienta	Licencia	Lenguajes	Plataformas destino	Caso de uso representativo
Godot 4	MIT	GScript, C#, C++	Windows, macOS, Linux, Android, iOS, Web	Usado en títulos publicados en Steam e itch.io; base de 1,5 M de usuarios activos según la propia comunidad
Love2D	MIT	Lua	Windows, macOS, Linux, Android, iOS	LuaBalatro (2024), uno de los juegos indie más vendidos de su año, desarrollado íntegramente con este framework

Tabla 3. Comparativa de herramientas libres para arte 2D.

Fuente: elaboración propia.

Herramienta	Licencia	Especialidad	Formatos de exportación	Caso de uso representativo
Krita	GNU GPL v3	Ilustración y pintura digital	PNG, PSD, SVG, TIFF, WebP	Creación de concept art, sprites y fondos para proyectos 2D; adoptada por ilustradores en títulos indie publicados en plataformas digitales
Pixelorama	MIT	Pixel art y animación frame-by-frame	PNG, GIF, APNG, spritesheet	Diseño de sprites animados para juegos de plataformas y RPGs; integración directa con Godot mediante el formato de proyecto compartido

Tabla 4. Comparativa de herramientas libres para arte 3D.

Fuente: elaboración propia.

Herramienta	Licencia	Especialidad	Formatos de exportación	Caso de uso representativo
Blender	GNU GPL v2+	Modelado, rigging, animación y render	FBX, OBJ, GLTF/GLB, DAE, USD	Producción de personajes, escenarios y cinemáticas; adoptado en producciones indie de complejidad media-alta con exportación directa a Godot y Unity
Blockbench	MIT	Modelado low poly y voxel	OBJ, FBX, GLTF, JSON (Bedrock)	Diseño de modelos para juegos de estilo minimalista y mods de Minecraft; flujo de trabajo orientado a principiantes con interfaz simplificada

Tabla 5. Comparativa de herramientas libres para audio.

Fuente: elaboración propia.

Herramienta	Licencia	Especialidad	Formatos de exportación	Caso de uso representativo
Audacity	GNU GPL v2	Edición y producción de audio	WAV, MP3, OGG, FLAC, AIFF	Grabación, limpieza y procesamiento de efectos de sonido para integración en motores de videojuegos
FamiStudio	MIT	Composición de música chiptune (NES)	WAV, OGG, NSF, ROM NES	Composición de bandas sonoras con estética retro para juegos de plataformas y RPGs de 8 bits
Bosca Ceoil Blue	MIT	Composición de música electrónica y chiptune	WAV, OGG, MID	Creación de música original para proyectos de jam de videojuegos; curva de aprendizaje reducida orientada a desarrolladores sin formación musical

Tabla 6. Repositorios de assets libres de regalías identificados.

Fuente: elaboración propia.

Recurso	Licencia	Tipo de contenido	Uso comercial	Caso de uso representativo
Sonniss GDC Bundles	Royalty-free (custom)	Efectos de sonido	Sí, sin restricciones	Bibliotecas de efectos de sonido distribuidas anualmente durante la Game Developers Conference; utilizadas en proyectos indie publicados comercialmente

Tabla 7. Distribución de herramientas por tipo de licencia.

Fuente: elaboración propia.

Tipo de licencia	Herramientas incluidas	Permite uso comercial sin restricciones	Requiere publicar modificaciones
Permisiva (MIT)	Godot, Love2D, Pixelorama, Blockbench, FamiStudio, Bosca Ceoil Blue	Sí	No
Copyleft (GNU GPL)	Krita, Blender, Audacity	Sí	Solo si se redistribuye la herramienta modificada
Royalty-free (custom)	Sonniss GDC Bundles	Sí	No aplica

5. Análisis de Resultados

Los resultados presentados en la sección anterior sugieren la existencia de un ecosistema completo de herramientas libres que cubre integralmente el flujo de trabajo en el desarrollo independiente de videojuegos, desde la programación del motor hasta la producción de audio, sin requerir inversión económica significativa en licencias. A continuación, se interpreta el significado de estos hallazgos en relación con el contexto planteado en la introducción.

En primer lugar, el análisis del modelo de licenciamiento (Tabla 7) revela que la mayoría de las herramientas identificadas operan bajo licencias MIT, consideradas las más permisivas del espectro del software libre. Esta distribución es especialmente relevante en el contexto indie: a diferencia de licencias copyleft como GNU GPL, las licencias MIT no imponen condiciones sobre el código propietario del videojuego resultante, reduciendo la incertidumbre legal al escalar un proyecto hacia la distribución comercial. Este hallazgo es consistente con el riesgo documentado en motores propietarios como Unity, cuyas modificaciones unilaterales de términos en 2023 generaron una crisis de confianza en el sector independiente [4], y refuerza la pertinencia de las alternativas aquí analizadas.

En cuanto a los motores de videojuego (Tabla 2), los datos sugieren que Godot y Love2D representan opciones viables para proyectos de distinto perfil, considerando el soporte multiplataforma, disponibilidad de documentación y una comunidad activa. Godot destaca por su arquitectura basada en nodos, su soporte nativo para exportación multiplataforma y su crecimiento sostenido en adopción desde 2020, impulsado en parte por el éxodo de desarrolladores que abandonaron Unity tras las controversias de licenciamiento mencionadas. Love2D, por su parte, ha demostrado viabilidad comercial a través de títulos como Balatro o Move or Die, casos que sugieren que un framework de menor escala puede sustentar productos comerciales exitosos, aunque no

constituyen evidencia generalizable. En conjunto, ambas herramientas cubren un amplio espectro de necesidades de un desarrollador principiante: Godot ofrece un entorno visual integrado más accesible, mientras que Love2D favorece un enfoque más programático orientado a quienes ya dominan la lógica de código.

Respecto a las herramientas visuales (Tablas 3 y 4), se observa una complementariedad natural entre las opciones identificadas. En el ámbito 2D, Krita y Pixelorama atienden estilos artísticos diferenciados: ilustración de alta fidelidad y pixel art, respectivamente, lo que permite al desarrollador seleccionar la herramienta según la estética del proyecto. En el ámbito 3D, la distinción entre Blender y Blockbench refleja una jerarquía de complejidad técnica: Blender es la herramienta de referencia para producciones de mayor exigencia geométrica y animación, mientras que Blockbench reduce la barrera de entrada para quienes se aproximan al modelado 3D por primera vez. Esta distribución es coherente con el enfoque del trabajo, centrado en desarrolladores principiantes que pueden progresar dentro del mismo ecosistema libre conforme crecen sus capacidades técnicas.

En el área de audio (Tabla 5), los resultados muestran que las tres herramientas identificadas cubren las dos dimensiones principales de la producción sonora para videojuegos: edición técnica de audio (Audacity) y composición musical original (FamiStudio, Bosca Ceoil Blue). La especialización chiptune de FamiStudio y Bosca Ceoil Blue no debe interpretarse como una limitación, sino como una ventaja contextual para el perfil de desarrollador principiante: la música de 8 bits reduce la complejidad compositiva al trabajar con síntesis paramétrica acotada, lo que facilita la producción de bandas sonoras originales sin formación musical previa. Audacity, por su parte, es suficiente para el procesamiento de cualquier efecto de sonido en proyectos de escala indie, cubriendo tareas desde la eliminación de ruido hasta la mezcla multicanal básica.

Finalmente, la inclusión de los Sonniss GDC Bundles como repositorio de assets libres (Tabla 6) aborda una dimensión práctica frecuentemente ignorada en la literatura: la disponibilidad de recursos de alta calidad que pueden integrarse directamente sin necesidad de producirlos desde cero. Esto es especialmente relevante en las etapas tempranas del desarrollo, cuando el desarrollador principiante puede priorizar la mecánica del juego sobre la producción de todos sus activos. En conjunto, los resultados respaldan la hipótesis central de este trabajo: existe un ecosistema funcional y maduro de herramientas libres que, tomadas en su conjunto, proporcionan al desarrollador independiente todo lo necesario para llevar un proyecto de videojuego de la concepción a la

distribución comercial sin incurrir en costos de licenciamiento.

6. Conclusiones

El presente trabajo ha caracterizado un ecosistema de nueve herramientas libres y gratuitas distribuidas en cinco categorías de producción (motores de videojuego, arte 2D, arte 3D, audio y assets), todas ellas bajo licencias que favorecen el uso comercial sin restricciones ni riesgo de modificaciones unilaterales de términos. El análisis realizado permite formular las siguientes conclusiones.

Primero, el ecosistema de herramientas libres para el desarrollo de videojuegos ha alcanzado un nivel de madurez suficiente para sustentar proyectos comerciales de impacto, como lo demuestra los casos de Balatro o Move or Die con Love2D, y Buckshot Roulette, Brotato o Webfishing con Godot. Esto representa un cambio estructural respecto al paradigma descrito en la introducción, en el que los grandes estudios concentraban el acceso a las herramientas de producción profesional.

Segundo, la predominancia de licencias MIT en el ecosistema analizado constituye una ventaja diferencial frente a las alternativas propietarias: garantiza estabilidad contractual independientemente del éxito comercial del proyecto, reduciendo el riesgo documentado en motores como Unity. Esta estabilidad es un factor crítico para el desarrollador indie, cuya precariedad financiera lo hace especialmente vulnerable a cambios en los términos de licenciamiento [1].

Tercero, el conjunto de herramientas identificado cubre gran parte del flujo de trabajo en el desarrollo de videojuegos, desde el motor y el diseño visual hasta la producción de audio y la gestión de assets, lo que convierte al tiempo y esfuerzo del desarrollador en los principales recursos requeridos para ejecutar un proyecto de principio a fin.

Como trabajo futuro, se proponen tres líneas de investigación. En primer lugar, la realización de estudios empíricos comparativos que midan el impacto del uso de herramientas libres frente a soluciones propietarias sobre la productividad, la calidad del producto final y la viabilidad económica de proyectos indie reales. En segundo lugar, la evaluación de la integración de inteligencia artificial generativa, particularmente modelos de lenguaje de gran escala, en el flujo de trabajo con herramientas libres, dado el creciente interés académico en este campo [18]. En tercer lugar, la extensión de este análisis a contextos educativos formales, explorando el potencial de este ecosistema para democratizar la enseñanza del desarrollo de videojuegos en instituciones con recursos tecnológicos limitados, en línea con los hallazgos de Christopoulos et al. [5] y Sáez-López et al. [20] sobre el impacto positivo

de herramientas accesibles en la participación y producción creativa de los estudiantes.

7. Referencias

- [1] Garda, M. B., & Grabarczyk, P. (2016). Is every indie game independent? Towards the concept of indie game. *Game Studies*, 16(1). <http://gamestudies.org/1601/articles/gardagrabczyk>
- [2] Sotamaa, O. (2021). Studying game development cultures. *Games and Culture*. <https://doi.org/10.1177/15554120211005242>
- [3] Foxman, Maxwell & Nieborg, David. (2016). Between a Rock and a Hard Place: Games Coverage and its Network of Ambivalences. *Journal of Games Criticism*. 3. https://www.researchgate.net/publication/296484250_Between_a_Rock_and_a_Hard_Place_Games_Coverage_and_its_Network_of_Ambivalences
- [4] Toftedahl, M., & Engström, H. (2019). A taxonomy of game engines and the tools that drive the industry. *DiGRA Conference Proceedings*. <https://doi.org/10.26503/dl.v2019i1.1077>
- [5] Christopoulos, A., Conrad, M., & Shukla, M. (2018). Increasing student engagement through virtual and gamified environments. *Virtual Reality* 22(4). <https://doi.org/10.1007/s10055-017-0330-3>
- [6] Berg Marklund, B., Engström, H., Hellkvist, M., & Backlund, P. (2019). What empirically based research tells us about game development. *The Computer Games Journal*, 8, 179–198. <https://doi.org/10.1007/s40869-019-00085-1>
- [7] Moradzadeh, S., Zhang, Z., Gui, X., & Kou, Y. (2025). The state of video game research in computer-supported cooperative work: A systematic literature review. *Computer Supported Cooperative Work*, 34, 587–638. <https://doi.org/10.1007/s10606-025-09519-z>
- [8] Gonçalves, L. R., & Paim, B. W. (2023). How to integrate mechanical licences for video games in the United States, the United Kingdom and China. *Journal of Gaming & Virtual Worlds*, 15, 161–175. https://doi.org/10.1386/jgvw_00078
- [9] Chueca, J., Verón, J., Font, J., Pérez, F., & Cetina, C. (2024). The consolidation of game software engineering: A systematic literature review of software engineering for industry-scale computer games. *Information and Software Technology*, 165. <https://doi.org/10.1016/j.infsof.2023.107330>
- [10] Scacchi, W. (2017). Practices and technologies in computer game software engineering. *IEEE Software*, 34(1), 110–116. <https://doi.org/10.1109/MS.2017.20>
- [11] Politowski, C., Petrillo, F., Montandon, J. E., Valente, M. T., & Guéhéneuc, Y.-G. (2020). Are game engines software frameworks? A three-perspective study. *Journal of Systems and Software*, 171. <https://doi.org/10.1016/j.jss.2020.110846>
- [12] Ullmann, G. C., Politowski, C., Guéhéneuc, Y.-G., & Petrillo, F. (2022). Game engine comparative anatomy. *arXiv:2207.06473* [cs.SE]. <https://doi.org/10.48550/arXiv.2207.06473>
- [13] Sobota, B., & Pietriková, E. (2023). The role of game engines in game development and teaching. In *Computer Science for Game Development and Game Development for Computer Science*. IntechOpen. <https://doi.org/10.5772/intechopen.1002257>

- [14] Politowski, C., Petrillo, F., & Guéhéneuc, Y.-G. (2021). A survey of video game testing. arXiv:2103.06431 [cs.SE]. <https://doi.org/10.48550/arXiv.2103.06431>
- [15] Kamienski, A., & Bezemer, C.-P. (2021). An empirical study of Q&A websites for game developers. *Empirical Software Engineering*, 26, 115. <https://doi.org/10.1007/s10664-021-10014-4>
- [16] Hall, H., Dogbe, A., & Said, H. (2025). Open-source game development: A systematic literature review. In M. Jones (Ed.), *Proceedings of InSITE 2025: Informing Science and Information Technology Education Conference*, Article 20. Informing Science Institute. <https://doi.org/10.28945/5563>
- [17] Zamorano, M., Cetina, C., & Sarro, F. (2023). The quest for content: A survey of search-based procedural content generation for video games. arXiv:2311.04710 [cs.SE]. <https://doi.org/10.48550/arXiv.2311.04710>
- [18] Sweetser, P. (2024). Large language models and video games: A preliminary scoping review. arXiv:2403.02613 [cs.HC]. <https://doi.org/10.48550/arXiv.2403.02613>
- [19] Hu, C., Zhao, Y., Wang, Z., Du, H., & Liu, J. (2024). Games for artificial intelligence research: A review and perspectives. arXiv:2304.13269 [cs.AI]. <https://doi.org/10.48550/arXiv.2304.13269>
- [20] Sáez-López, J. M., González-Calero, J. A., Cózar-Gutierrez, R., & del Olmo-Muñoz, J. (2023). Scratch and unity design in elementary education: A study in initial teacher training. *Journal of Computer Assisted Learning*, 39(5), 1528–1538. <https://doi.org/10.1111/jcal.12815>