



Aplicación de realidad virtual para reforzar temas de programación y lógica virtual

Reality application to reinforce programming and logic topics

Cristhian Ivan Garcia Maldonado¹

cgarciam021@alumno.uaemex.mx

ORCID: 0009-0000-8371-855X

Yaroslaf Aarón Albarrán Fernández

yaalbarranf@uaemex.mx

ORCID: 0000-0002-2615-4740

Resumen

La realidad virtual se basa en una tecnología inmersiva la cual emula entornos apegados a la realidad por medio de una visión digital, esta interacción se da por una serie de dispositivos como lentes VR, guantes hápticos y una serie de transductores de movimiento. En la actualidad las aplicaciones de RV están en áreas como la educación, entretenimiento, trabajo, salud, exploración y en el metaverso, han generado experiencias como gráficas en 4k/8k, aplicaciones con integración de IA generativa y alta eficiencia en la conectividad. Es por eso que este trabajo tiene la finalidad

¹ Autores de la Ingeniería en Sistemas Inteligentes, Centro Universitario UAEM Nezahualcóyotl, Universidad Autónoma del Estado de México.



de integrar esta tecnología, pero una de las problemáticas encontradas es que todavía hay áreas inhóspitas que no han implementado este tipo de experiencias, esto nos lleva a diseñar un proyecto, donde usaremos una plataforma que nos permita combinar una experiencia inmersiva como Google Cardboard fácil y rápida de implementar, en donde se propone realizar una aplicación con realidad virtual acerca de un juego de preguntas generales de programación, conceptos y ejercicios de lógica para poner a prueba sus conocimientos de una manera divertida. El propósito de esta práctica es simular un examen o test en un entorno intuitivo donde por medio de un escenario relajante con objetos que sean extremadamente atractivos esto para mejorar la concentración a la hora de contestarlo y demostrar una diferencia entre los resultados sobre lo tradicional a lo innovador. Se utilizaron softwares como Blender que sirve para el modelado en 3D o 2D, iluminaciones y renderizado. De igual manera el uso de Unity que es un motor de desarrollo de videojuegos y de aplicaciones móviles el cual es más utilizado hoy en día y finalmente el manejo del lenguaje de programación C# para agregar líneas de código ofreciendo funcionalidades a la aplicación.

Palabras claves: Realidad virtual, C#, Cardboard, Unity, Programación, Lógica, Gaze.

Abstract

Virtual reality is based on immersive technology that emulates realistic environments through digital vision. This interaction is achieved through a series of devices such as VR headsets, haptic gloves, and a series of motion transducers. Currently, VR applications are found in areas such as education, entertainment, work, healthcare, exploration, and the metaverse. They have generated



experiences such as 4k/8k graphics, applications with generative AI integration, and highly efficient connectivity. That is why this work aims to integrate this technology. However, one of the problems encountered is that there are still inhospitable areas that have not implemented this type of experience. This led us to design a project where we will use a platform that allows us to combine an immersive experience like Google Cardboard, which is easy and quick to implement. The project proposes creating a virtual reality application based on a game of general programming questions, concepts, and logic exercises to test your knowledge in a fun way. The purpose of this practice is to simulate an exam or test in an intuitive environment, using a relaxing setting with highly engaging objects to improve concentration while answering the test and demonstrate the difference between traditional and innovative results. Software such as Blender, which is used for 3D or 2D modeling, lighting, and rendering, was used. Unity, the most widely used development engine for video games and mobile applications, was also used today. Finally, the C# programming language was used to add lines of code that offer functionality to the application.

Keywords: Virtual Reality, C#, Cardboard, Unity, Programming, Logic, Gaze

Fecha de envío: 20/05/2025

Fecha de aprobación: 18/07/2025

Fecha de publicación: 01/09/2025



Introducción

La realidad virtual ha impactado de manera favorable en el ámbito tecnológico, ya que era algo que solo se veía en las novelas de ciencia ficción como Isaac Asimov (yo, Robot) o Arthur C. Clark (2001, Una odisea espacial), esto nos hace ver que la realidad avanza hacia la ciencia ficción.

Al ofrecer experiencias sensoriales en entornos tridimensionales llamando la atención al público por su innovación, sin embargo, esta resulta ser una buena alternativa para fines educativos ya que nos ayuda a tener una mejor retención de la información.

Como señala Hu-Au y Lee (2018), esta herramienta nos ofrece una exploración e interacción para desplazarnos dentro de escenarios simulados, generando emociones que enriquecen la experiencia de usuario, “proporciona una forma novedosa de aprendizaje para los estudiantes que puede ser más atractiva e impactante que los métodos tradicionales” (p.112). Esto resulta ser más efectivo que los típicos métodos que conocemos hoy en día. Gracias a su fácil capacidad de inmersión, se ha convertido en una solución innovadora que mejora la formación y la creatividad en diversos campos para los desarrolladores principiantes. Donde se buscó una herramienta, de bajo costo y fácil de utilizar para cualquier persona sin necesidad de tomar alguna capacitación, Wang y Chan (2024) destaca que “Google Cardboard ofrece una experiencia VR accesible y motivador para los estudiantes, aunque presenta retos técnicos como la calidad de la pantalla o el mareo”.

Este mismo necesita de su particular configuración que nos brinda su página oficial y un visor de esta marca construido de cartón junto con un teléfono inteligente de gama media necesarios para la visualización, además, permite un despliegue rápido y un rendimiento



equilibrado por ende lo convierte en una alternativa ideal para proyectos escolares o para desafíos dependiendo de los requerimientos.

El presente artículo tiene como propósito desarrollar una prueba intuitiva utilizando Unity para que programadores de nivel intermedio de la carrera de ingeniería en sistemas inteligentes puedan reforzar sus conocimientos acerca de conceptos básicos y avanzados vistos durante toda su carrera. Se busca realizar una aplicación eficiente con compatibilidad para dispositivos convencionales preferiblemente Android para brindar una sensación diferente a la hora de realizar un examen teórico. Se presentará visualmente un espacio agradable con texturas, colores, objetos en 3D, luces, textos grandes, luces y además de elementos significativos de la prueba dentro de un escenario sin llamar la atención del jugador.

Con el objetivo de elevar la concentración y evaluación de los usuarios dentro de un entorno simulado de forma divertida y dinámica para que puedan participar en múltiples intentos para mejorar la retención de información, según estudios por parte de Roediger, H. L III, & Karpicke, J. D. (2006). Test-enhanced learning: Taking memory tests improves long-term retention. *Psychological Science*, 17(3), 249-255. Para su desarrollo, se realizó una investigación sobre los requerimientos básicos para la instalación de los programas a usar. Por lo tanto, se utilizó el lenguaje de C# por defecto para la implementación de funciones interactivas, eventos que aporten realismo y funcionalidad a la aplicación.

Asimismo, se usó conceptos vistos en clase, como librerías, variables, algoritmos, estructuras de control, clases, objetos y funciones, brindando una base sólida en programación orientada a objetos. Durante la construcción, el lector será capaz de ver lo fácil que es implementar una aplicación en menos de 2 horas, aplicando conocimientos prácticos en un entorno accesible de código abierto, utilizando recursos gratuitos de assets gracias a su gran comunidad. Finalmente,



esto permitirá reforzar temas esenciales acerca de la programación y sus ramas hasta poder dominar los conceptos por completo así mismo creando confianza a los participantes para desenvolverse en entrevistas de trabajo o pruebas técnicas para aplicar en los siguientes roles de Programadores como Junnior y un Mid-Level. Esto impulsara una idea más clara sobre el perfil que requiere un programador en el ámbito de la realidad virtual.

Antecedentes

En 1980, la valoración de habilidades de programación era de pruebas escritas en lenguajes como Fortran o COBOL, posteriormente surgieron nuevas plataformas de tipo estáticas para mejorar las habilidades por medio de retos o desafíos ayudando a desarrolladores y programadores en la actualidad.

La realidad virtual es una herramienta realmente útil en nuestras vidas. Ya que nos brinda sumergirnos en mundos virtuales interactivos con visores comerciales. Actualmente desarrollar una app con RV te garantiza nuevas oportunidades laborales partiendo de resolver tareas básicas a lo más complejo, sin embargo, la proyección o el alcance para proyectos con realidad virtual ha sido orientada en el área de salud. El mercado de RV educativa está dominado por medicina (35%) y ciencias naturales (30%), con solo un 7% en programación". Informe de Marlet Research Future (2023). Esto inicia una oportunidad para desarrollar entornos inmersivos que fortalezcan habilidades en informática o ciencias de la computación.

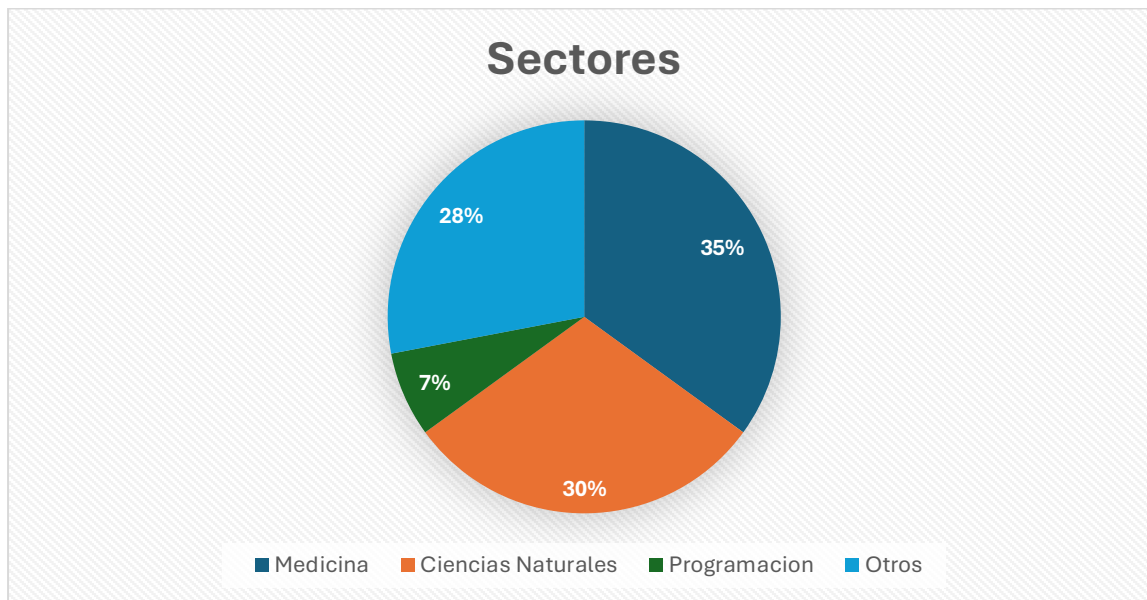


Figura 1. Grafica pastel del uso de la VR

Materiales y métodos

Se desarrollaron múltiples prototipos utilizando los siguientes elementos de software: Unity 2021.3.45 LTS, un motor de desarrollo de juegos en 3D, y C# para codificar funciones interactivas. Se utilizaron pasos oficiales sobre el uso del SDK, y se empleó Blender para generar modelos tridimensionales. En cuanto al hardware, simplemente se utilizó un conjunto de VR genéricos y una computadora de marca HP con las siguientes especificaciones que son un procesador Intel i3-1005G y 16G de RAM. Tenía gráficos integrados, pero se recomienda una tarjeta gráfica y considerar 8+ GB de RAM para una ejecución fluida del programa. Se realizó una sesión de práctica con una clase de ingeniería de sistemas inteligentes de semestres intermedios de entre estudiantes de 21 y 24 años. Luego se pidió a los participantes que completaran un cuestionario para evaluar su familiaridad con la herramienta antes de avanzar al primer ensayo en su intento de



mejorar sus habilidades básicas de programación y aspectos esenciales de Unity. El objetivo era utilizar un entorno virtual para enseñar conceptos, algoritmos y lógica de programación en un juego entretenido. Se descargó un entorno de desarrollo de la tienda oficial, gratuito y listo para ser importado al proyecto, con objetos, texturas, colores, sonidos dentro de una escena principal; sería la escena más usada a través del código. Los participantes utilizaron el manual de Unity para poder implementar la tarea. El visor Google Cardboard permite emular la realidad con un dispositivo inteligente para experimentar de un mundo virtual intuitivo, estos cascos generalmente son fáciles de usar para cualquier persona y accesibles para cada uno de nuestros bolsillos sin arriesgar nuestro presupuesto.



Figura 2. Visor de Google Cardboard.



Figura 3. Uso de la aplicación.

Se sugiere usar la versión más actual, junto con los siguientes módulos listados para descargar:

- Microsoft Visual Studio Community 2022



- Soporte de compilación para Android
- OpenJDK
- Herramientas de Android SDK y NDK
- Soporte de compilación WebGL
- Text Mesh Pro

Se empezó de cero al crear un nuevo proyecto generado a partir de una carpeta vacía con un nombre de aplicación en mi caso, test vr (con un renderizado construido). En la ventana del administrador de paquetes se abrió dentro de la interfaz de Unity y fue posible descargar el repositorio Hello Cardboard. Esto es muy versátil a lo hora de importar ya que tiene múltiples opciones, la más conocida es por GitHub. Los archivos principales que son la jerarquía del proyecto y las carpetas están separados por secciones como: Plugin (donde están los archivos de configuración), Samples (son compuesto por materiales y objetos), Scripts (donde se encuentran los códigos en C#) y XR Scenes (que es, por el momento, la escena principal predeterminada). El proyecto será para Android al configurar la plataforma de destino en configuración de construcción lo que puede tardar uno o dos minutos mientras se vuelven a importar todos los archivos que necesita para ejecutar.

He modificado la configuración de realidad virtual en el apartado del jugador, todas las demás configuraciones son predeterminadas. La orientación predeterminada se estableció en horizontal izquierda como es usual en VR y la tasa de fotogramas se aumentó para una mejor capacidad de respuesta. Las otras configuraciones fueron OpenGL ES3 para la API Gráfica, el nivel de API de Android se estableció en 34, IL2CPP fue el backend de scripting esto proporciona un mejor rendimiento, se activó una arquitectura que soportaba tanto ARM32 como ARM64. Los

archivos en Plugin son utilizados por Cardboard, y en la configuración del player, se habilito la plantilla gradle principal personalizada y la plantilla de propiedades gradle personalizada para agregar dependencias de SDK en la configuración, en la cuales unicamente se tenía que agregar unas líneas de código manualmente y guardar los cambios en cada uno.

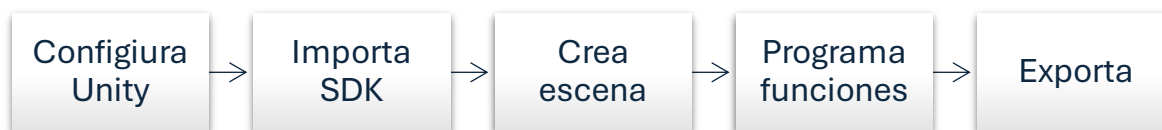


Figura 4. Secuencia para desarrollar el proyecto

Se analizaron varios puntos antes de empezar a la construcción las cuales son cruciales para el desarrollo, por lo que se tuvo que plantear objetos que serán representados, como datos de entradas, salidas y funciones a ocupar. Teniendo una idea clara se empezó a realizar mockups en el diseño para una mejorar la experiencia de usuario a la hora de visualizarlo. Finalmente tocaba agregar funcionalidades a cada objeto para indicarle que acción hará y el objetivo de esa tarea. Se programo la siguiente lógica, tener una lista de preguntas y que tengan unicamente un texto string y una respuesta con valores de verdadero o falso, una función aleatoria para que, dependiendo del número de preguntas, elija las que yo quisiera por ejemplo 15. También una tarea que a la hora de contestar con la mirada donde tendrá un gaze este guarde el resultado y que tenga un contador asignado con un valor de cero para que sume uno cada vez que se cumpla correctamente, para el final se colocó un panel que dentro del muestre cuantas tuvimos bien en la prueba.

Se uso el script de “inicio” que inicia desde cero y se configuro el sistema predeterminado una vista de cámara en tercera persona para ayudar a navegar por el sistema como si estuvieras en



la realidad. Después de agregar todo, el dispositivo Android se conectaría al visor para emular VR. Una vez construido, el proyecto se guardó como un archivo APK y se colocó en un directorio con el nombre de la aplicación. Se verificó que la aplicación cumpliera con los requisitos deseados y que los scripts funcionaran al realizar la “test vr”. Al probar, la aplicación se instaló en un dispositivo compatible con Android para disfrutar de manera satisfactoria y sin errores de compilación esta experiencia de inmersión.

Resultados

Se observó que, al ejecutar la aplicación test de programación, la mayoría de los estudiantes logró contestar correctamente 20 preguntas aleatorias en un tiempo promedio de 10 a 12 minutos. Esto sugiere un nivel aceptable de conocimiento general para estudiantes de semestres intermedios que han trabajado en distintos momentos a lo largo de su carrera, por lo cual se buscó abarcar un poco de todo para cubrir mayor parte de conocimientos de diferentes materias.

Las preguntas abordaban temas importantes como conceptos básicos, paradigmas de programación, estructuras de datos, bases de datos, comandos de Linux, sentencias y ejercicios de lógica. La estructura de las preguntas fue de tipo verdadero o falso, ya que este formato es más sencillo de implementar dentro de listas o arreglos en el código, mediante variables tipo string y un valor booleano que indica la respuesta correcta. Durante el proceso de despliegue, que fue largo debido a la necesidad de verificar y compilar línea por línea, internamente se verifica detenidamente las librerías, elementos 3D y módulos esenciales. Al final, se generó un archivo APK que fue guardado en el equipo de cómputo para su posterior prueba de renderizado y



rendimiento. Se realizaron ajustes en la ventana de Player Settings para desactivar procesos innecesarios que afectaban negativamente el rendimiento del prototipo.

En cuanto a la compatibilidad, no se presentaron fallos graves, aunque se recomienda el uso en dispositivos Android de gama media como mínimo, ya que en dispositivos de baja gama la aplicación puede tardar en iniciarse o tener mayor latencia. La interfaz resultó ser atractiva para los usuarios al estar en un entorno de realidad virtual, ofreciendo un 100% de experiencia de inmersión.

Los visores utilizados se adquirieron por internet y funcionaron correctamente en las pruebas físicas. Aunque están fabricados en cartón duro, ofrecieron una buena experiencia y demostraron ser resistentes por su calidad-precio. Su tamaño compacto también facilita su transporte por lo ligero que son. Algunos participantes que no obtuvieron inicialmente los resultados esperados mostraron una mejor retención de conceptos después de jugar la aplicación varias veces (hasta 10 intentos).

Este tipo de repetición fue beneficiosa, ya que permitió reforzar contenidos previamente vistos, lo cual era uno de los objetivos del artículo. Finalmente, no se presentó ninguna irregularidad con la configuración de realidad virtual, ya que se siguió correctamente la documentación oficial de Google Cardboard.

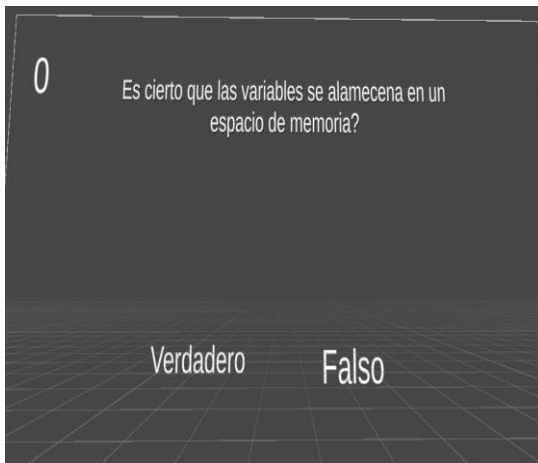


Figura 5. Panel de preguntas T-Mesh



Figura 6. Vista del Playmode en Unity

Discusión

La información nos indica que la realidad virtual es conocida, sin embargo, no la plataforma de Google Cardboard que podrían ser por razones de falta de información o construcción de materiales reciclados y con poca interacción comparando a sus competidores que son más sofisticados y por ende más costosos para su adquisición. Esto nos da una idea de donde partir para que las personas descubran y se desenvuelvan en esta herramienta aplicando sus conocimientos vistos en la escuela relacionados a la programación, especialmente en algoritmos y estructuras de datos. Se considera que la inmersión es gratamente positiva para el aprendizaje porque mejora la retención del conocimiento mucho más que leer o repasar incluso en un mundo inmersivo por que produce sensaciones sensoriales logrando mejorar el razonamiento del participante.

Una limitación crucial fue a la hora de elegir OPENGL y VULKAN ya que en unos casos puede causar problemas más adelante ya que no comparten mismos requisitos, aunque los dos son funcionales. También el equipo de cómputo que vayas a utilizar es importante ver los requisitos



mínimos para ejecutar los programas ya que unos ocupan distintos recursos o demasiado espacio en tu disco duro de tu equipo de cómputo. Por último, debes asegurarte de que hayas descargado los módulos necesarios ya que es muy común que los principiantes tengan problemas en la compilación a la hora de configurar el SDK por tenerlo desactualizado.

Existen alternativas como Unreal Engine, una plataforma enfocada en el desarrollo de proyectos con alta calidad gráfica y un nivel de realismo alto. Sin embargo, esta potencia visual requiere mayores recursos de hardware, siendo más adecuada para consolas y computadoras de alta gama. Es por eso por lo que, Unity se caracteriza por contar con una interfaz más amigable para el usuario, lo que facilita el desarrollo rápido, especialmente para quienes están iniciando recientemente en esta rama. Además, de que ofrece un buen rendimiento y es ampliamente utilizado hoy en día para la creación de videojuegos para dispositivos móviles y para experiencias de realidad virtual (RV)

Conclusiones

En este trabajo, utilizamos la realidad virtual para crear una prueba simple y rápida sobre conceptos de programación abarcando diferentes ramas importantes dirigida a la carrera de sistemas, que se pueden aplicarse de ayuda de una aplicación intuitiva y construida desde cero en menos de dos horas con mecánicas de juego como por ejemplo la mirada conocida por el nombre de Gaze. En comparación con otras tecnologías como Godot o Unreal, se sugiere usar Unity para desarrolladores principiantes que quieren imitar la realidad o crear videojuegos ya fáciles de aprender, para proponer funciones que se pueden resolver tareas en diversas áreas. Hay muchas herramientas que explorar e investigar en este gran camino para sobresalir en este mercado. Para



su despliegue era importante la simplicidad a la hora de asignar fácilmente funciones a objetos 3D como capsulas, esferas, cubos y vacíos a lo largo del código, así como entrelazar esto componentes de enseñanza sobre el lenguaje de programación C#. También se incluyó la instalación de algunos programas para llevar los proyectos al siguiente nivel gráficamente como: Blender que es uno de los más usados en la actualidad u optar por descargar elementos ya creados de su tienda oficial en la puedes encontrar lo que necesites de manera gratuita, esto ahorra tiempo de desarrollo para darle más prioridad a otras cosas.

La intención del artículo es cambiar ligeramente la forma tradicional de ver la típica interfaz de usuario cuadrada, por una de interacción a través de actividades físicas y sensoriales en recorridos virtuales o 3D. El objetivo es ofrecer una respuesta atractiva al usuario final con la experiencia de estar en un modo de inmersión. Esta aplicación ayuda a las generaciones más jóvenes a tomar autoconfianza en aspectos de desarrollo como escenarios, herramientas auxiliares, elementos e interacciones sin abusar, sino empleando realmente lo necesario.

Con esto podemos aplicar en una amplia gama de industrias como la como la medicina, como ayuda de terapias y rehabilitaciones; en la educación, donde nos permite aprender de manera más dinámica y finalmente en el entretenimiento, donde nos sumerge en videojuegos emocionantes. La configuración adecuada del SDK demostró ser efectiva al reducir errores de compilación al crear tu primera apk segura para tu propio ordenador para descargarlo de manera exitosa y ponerlo a prueba con un celular inteligente para al final disfrutar de la experiencia.

En resumen, se obtuvo un resultado favorable y amigable con la interfaz a la hora del uso por el usuario con la aplicación. Las pruebas internamente del proyecto fueron buenas obteniendo un bajo consumo de recursos y un peso ligero a la hora de descargar en un dispositivo Android de gama media. Se busco la aprobación y valoración de los usuarios reales para recaudar información



y contar con precedentes para tomar en cuenta las posibles mejoras del proyecto con el propósito de generar interés a todo tipo de personas que quisieran reforzar temas de una manera diferente en la que estamos acostumbrados a practicar inclinándonos por una experiencia atractiva y memorable con la realidad virtual. Se cumplió el objetivo planteado a la hora de trabajar con personas reales por un método de varias repeticiones para fomentar una mayor retención de información que es excepcional a largo plazo.



Referencias

Mosca, G., & Tipler, P. A. (2005). Física 2C: Para la ciencia y la tecnología. Física moderna. Reverté.

Gutiérrez Sánchez, M. de J. (2019). La Gamificación en Los Ambientes De Realidad Virtual Móvil. Pistas Educativas, (133).

Serrano, B. (2012). Realidad Virtual y Virtualidad Aumentada. Eae Editorial Academia española.

Google Developers. (s.f). Guía de inicio rápido para Unity. <https://developers.google.com/cardboard/develop/unity/quickstart?hl=es-419>

Martos, J. (2022). Crea tu primera App de Realidad Virtual (VR) para Cardboard [Curso en línea]. Udemy. <https://www.udemy.com/course/crea-tu-primer-app-de-realidad-virtual-vr-para-cardboard/>

Google. (2024, octubre 9). *Quickstart for Google VR SDK for Android*. <https://developers.google.com/vr/develop/android/get-started>

Microsoft. (s. f.). Documentación de C#. <https://learn.microsoft.com/es-es/dotnet/csharp/>

Hu-Au, E., & Lee, J. J. (2018). Virtual reality in education: A tool for learning in the experience age. *International Journal of Innovation in Education*, 5(2), 111–126. <https://doi.org/10.1504/IJIE.2018.10013664>

Wang, Z., & Chan, M.-T. (2024). A systematic review of Google Cardboard used in education. *Computers & Education: X Reality*, 4, 100046. <https://doi.org/10.1016/j.cexr.2023.100046>



Roediger, H. L., III, & Karpicke, J. D. (2006). *Test-enhanced learning: Taking memory tests improves long-term retention*. *Psychological Science*, 17(3), 249–255.
<https://doi.org/10.1111/j.1467-9280.2006.01693.x>