UNIVERSITAT POLITÈCNICA DE VALÈNCIA

_ TELECOM ESCUELA TÉCNICA VLC SUPERIOR DE INGENIERÍA DE TELECOMUNICACIÓN

UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Escuela Técnica Superior de Ingeniería de Telecomunicación

Desarrollo de un gemelo digital de un fonógrafo portátil de los años 20 del siglo XX para el Museo de Historia de la Telecomunicación Vicente Miralles Segarra.

Trabajo Fin de Grado

Grado en Tecnología Digital y Multimedia

AUTOR/A: Martínez Quivén, Adrián

Tutor/a: Bachiller Martin, Maria Carmen

Cotutor/a: Rey Solaz, Beatriz

Cotutor/a: Monzó Ferrer, José María

CURSO ACADÉMICO: 2024/2025

Resumen

El objetivo del proyecto es el desarrollo de un gemelo digital de un gramófono portátil, Ica-fono, de los años 20 del siglo XX. Este gramófono se halla en los fondos del Museo de Historia de la Telecomunicación Vicente Miralles Segarra. La pieza se halla en perfecto estado de funcionamiento y es capaz de reproducir discos de pizarra, no obstante, para su preservación no se aconseja que sea utilizada, además se halla en exposición de forma permanente en el Museo. Se entiende que la experiencia del funcionamiento de este tipo de piezas es una parte fundamental de la experiencia didáctica del Museo, por tanto, resulta de gran valor para el mismo poder mostrarla a los visitantes. El desarrollo de un gemelo digital permitiría que esta funcionalidad se pudiera experimentar sin dañar la pieza original. Para ello se procederá al escaneo 3D de la pieza y su modelado en Blender. A continuación, se la dotará de funcionalidad (animación interactiva) mediante el uso de la herramienta Unity. El objetivo es obtener un gemelo digital para su uso en gafas de realidad virtual Meta Quest 2 y Meta Quest 3.

Resum

L'objectiu del projecte és el desenvolupament d'un bessó digital d'un fonògraf portàtil, Ica-fono, dels anys 20 del segle XX. Aquest fonògraf es troba en els fons del Museu d'Història de la Telecomunicació Vicente Miralles Segarra. La peça es troba en perfecte estat de funcionament i és capaç de reproduir discos de pissarra, no obstant això, per a la seua preservació no s'aconsella que siga utilitzada, a més es troba en exposició de manera permanent en el Museu. S'entén que l'experiència del funcionament d'esta mena de peces és una part fonamental de l'experiència didàctica del Museu, per tant, resulta de gran valor per al mateix poder mostrar-la als visitants. El desenvolupanet d'un bessó digital permetria que aquesta funcionalitat es poguera experimentar sense danyar la peça original. Per a això es procedirà a l'escaneig 3D de la peça i el seu modelatge en Blender. A continuació, li la dotarà de funcionalitat (animació interactiva) mitjançant l'ús de la ferramenta Unity. L'objectiu és obtindre un bessó digital per al seu ús en ulleres de realitat virtual Quest 2 i Meta Quest 3.

Abstract

The aim of the project is the development of a digital twin of a portable phonograph, Ica-fono, from the 1920s. This phonograph is in the collections of the Vicente Miralles Segarra Museum of the History of Telecommunications. The piece is in perfect working order and can play slate records, however, for its preservation it is not recommended that it be used, and it is on permanent display in the Museum. It is understood that the experience of the operation of this type of piece is a fundamental part of the didactic experience of the Museum, therefore, it is of great value for the Museum to be able to show it to visitors. The development of a digital twin would allow this functionality to be experienced without damaging the original piece. To this end, the piece will

be 3D scanned and modelled in Blender. It will then be given functionality (interactive animation) using the Unity tool. The aim is to obtain a digital twin for use in Meta Quest 2 and Meta Quest 3 virtual reality glasses.

RESUMEN EJECUTIVO

La memoria del TFG del Grado en Tecnología Digital y Multimedia debe desarrollar en el texto los siguientes conceptos, debidamente justificados y discutidos, centrados en el ámbito de la tecnologías digitales y multimedia

CONCEPT (ABET)	**CONCEPTO (traducción)**	**¿Cumple? (S/N)**	**¿Dónde? (páginas)**
1. IDENTIFY:	1. IDENTIFICAR:		
1.1. Problem statement and opportunity	1.1. Planteamiento del problema y oportunidad	S	1
1.2. Constraints (standards, codes, needs, requirements & specifications)	1.2. Toma en consideración de los condicionantes (normas técnicas y regulación, necesidades, requisitos y especificaciones)	S	1
1.3. Setting of goals	1.3. Establecimiento de objetivos	S	1
2. FORMULATE:	2. FORMULAR:		
2.1. Creative solution generation (analysis)	2.1. Generación de soluciones creativas (análisis)	S	6-8
2.2. Evaluation of multiple solutions and decision-making (synthesis)	2.2. Evaluación de múltiples soluciones y toma de decisiones (síntesis)	S	7-12
3. SOLVE:	3. RESOLVER:		
3.1. Fulfilment of goals	3.1. Evaluación del cumplimiento de objetivos	S	34
3.2. Overall impact and significance (contributions and practical recommendations)	3.2. Evaluación del impacto global y alcance (contribuciones y recomendaciones prácticas)	S	34

Índice

Capítulo 1. Introducción 1
1.1 Objetivos 1
1.2 Justificación y motivación 2
1.3 Metodología empleada 2
1.4 Estructura del trabajo 3
Capítulo 2. Contexto histórico y museológico 4
2.1 El gramófono y su evolución técnica 4
2.2 Características del gramófono portátil Ica-fono 4
2.3 El Museo de Historia de la Telecomunicación Vicente Miralles Segarra 5
2.4 La conservación del patrimonio tecnológico y cultural 5
2.5 Patrimonio inmaterial: registro y conservación sonora 5
Capítulo 3. Estado del arte 6
3.1 Gemelos digitales: concepto, arquitectura y aplicaciones 6
3.2 Realidad virtual aplicada al patrimonio 7
3.3 Herramientas tecnológicas utilizadas 7
3.3.1 Escaneado 3D con LiDAR 7
3.3.2 Modelado y texturizado con Blender 8
3.3.3 Desarrollo interactivo con Unity 8
Capítulo 4. Desarrollo del gemelo digital 9
4.1 Preparación y escaneado del gramófono 9
4.2 Procesado y modelado 3D 9
4.3 Grabación de su sonido 11
4.4 Implementación de animaciones y funcionalidades interactivas 13
4.5 Desarrollo de la aplicación de realidad virtual 13
4.6 Scripts desarrollados 16
4.7 Resultados 23
Capítulo 5. Conservación y difusión del patrimonio mediante realidad virtual 32
5.1 Aportaciones del gemelo digital a la conservación preventiva 32
5.2 Valor educativo y experiencia para los visitantes 32
5.3 Limitaciones y oportunidades futuras 32

Capítulo 6. Conclusiones 34
6.1 Conclusiones generales del proyecto 34
6.2 Propuestas de mejora y trabajo futuro 34
Capítulo 7. Objetivos de desarrollo sostenible de la agenda 2030 36
Capítulo 8. Bibliografía 37
Capítulo 9. Anexo: Scripts del gramófono 39

Índice de figuras

Figura 1. Fotografía del gramófono Ica-fono. Fuente: [2] 4
Figura 2. Sensor 8
Figura 3. Estructura principal del gramófono en Blender. 9
Figura 4. Disco de pizarra modelado en Blender. 10
Figura 5. Aguja, diafragma y cavidad resonante modelada en Blender. 10
Figura 6. Brazo modelado en Blender. 11
Figura 7. Manivela modelada en Blender. 11
Figura 8. Caratulas de los discos grabados. 12
Figura 9. Proceso de registro sonoro en estudio de grabación. 12
Figura 10. Configuración del proyecto en Unity. 13
Figura 11. Package Manager de Unity. 14
Figura 12. Disposición inicial de la sala del gemelo digital en Unity. 14
Figura 13. Máquina de estados del gramófono. 15
Figura 14. Gramófono sobre la mesa. 23
Figura 15. Apertura del gramófono. 24
Figura 16. Colocación de manivela. 24
Figura 17. Colocación de fieltro sobre plato. 25
Figura 18. Elegir un disco. 25
Figura 19. Colocación del disco sobre fieltro. 26
Figura 20. Cargar la cuerda. 26
Figura 21. Quitar freno. 27
Figura 22. Bajada de brazo. 27
Figura 23. Bajada de Aguja. 28
Figura 24. Escuchar reproducción de disco. 28
Figura 25. Quitar aguja. 29
Figura 26. Puesta de freno. 29
Figura 27. Retirada de disco. 30
Figura 28. Retirada de fieltro. 30
Figura 29. Retirada de manivela. 31
Figura 30. Cerrar gramófono. 31

Capítulo 1. Introducción

En el contexto del patrimonio cultural, la preservación y difusión de objetos históricos ha expandido sus horizontes con el uso de nuevas tecnologías digitales. Este trabajo se centra en el desarrollo de un gemelo digital, es decir, una réplica virtual interactiva de un objeto físico que simula su apariencia y comportamiento real, con fines de visualización, análisis o conservación, de un gramófono portátil de los años 20 del siglo XX, que pertenece al Museo de Historia de la Telecomunicación Vicente Miralles Segarra. El objetivo es contribuir a la conservación del patrimonio material e inmaterial asociado a este dispositivo.

Este gramófono representa un testimonio valioso de la evolución tecnológica de los dispositivos de reproducción de sonido y constituye, por tanto, un objeto de gran interés histórico, técnico y cultural, pues se trata, de un aparato emblemático en la historia de las comunicaciones y la música.

Al digitalizarlo, no solo capturamos su estado físico, sino que también simulamos sus funciones y el trasfondo cultural que lo envuelve. En este sentido, la creación de un gemelo digital (una réplica virtual precisa del objeto real abre nuevas posibilidades para su estudio, conservación, exposición interactiva. [1]

Este proyecto combina distintas disciplinas y herramientas, incluyendo el escaneo 3D mediante LiDAR, el modelado 3D usando Blender como software y el desarrollo de entornos virtuales interactivos para gafas de realidad virtual en el motor Unity. También se investiga el patrimonio cultural y cómo digitalizarlo, usando métodos de la ingeniería y la museología.

El objetivo principal de este trabajo es diseñar y desarrollar un gemelo digital que reproduzca fielmente la apariencia y funcionalidad del gramófono, permitiendo su visita interactiva y sirviendo como base para futuras iniciativas educativas y museográficas. De esta manera se podrá enseñar y experimentar la puesta en marcha del gramófono sin riesgo de dañar la pieza real, que es delicada y requiere de conocimientos específicos para usarse.

1.1 Objetivos

El objetivo principal de este Trabajo de Final de Grado es desarrollar de un gemelo digital de un gramófono portátil de los años 20, perteneciente a la colección del Museo de Historia de la Telecomunicación Vicente Miralles Segarra. Este proyecto tiene como finalidad la digitalizar, reconstruir en 3D e implementar de forma interactiva el dispositivo original en un entorno de realidad virtual, con el propósito de contribuir a su conservación, difusión y valorización como patrimonio tecnológico.

Para alcanzar este objetivo general, se plantean una serie de objetivos específicos. En primer lugar, se documentará el estado físico y funcional del gramófono mediante técnicas de escaneado tridimensional y observación directa, con el fin de obtener unos conocimientos adecuados para su reproducción digital. Finalmente, este modelo se integrará en un espacio interactivo creado con Unity, donde el usuario podrá examinar la apariencia, el uso y el sonido del gramófono desde distintos ángulos.

Adicionalmente, se desarrollarán animaciones que simulen el funcionamiento básico del aparato, como el giro del plato, el desplazamiento del brazo fonocaptor y el manejo de la manivela, con el fin de representar de forma didáctica sus mecanismos básicos. Al mismo tiempo, se realizará una investigación orientada a la protección del patrimonio cultural, grabando el sonido que emite

dicho equipo al reproducir discos de su época. Finalmente, se evaluará el potencial del gemelo digital como herramienta de divulgación y soporte para museos.

1.2 Justificación y motivación

Digitalizar el patrimonio se ha convertido en una parte clave del proceso de conservación de piezas históricas en los museos de la actualidad. En este contexto, el presente proyecto surge con el objetivo de conservar y difundir el gramófono portátil Ica-fono, pieza del Museo de Historia de la Telecomunicación Vicente Miralles Segarra mediante la creación de un gemelo digital interactivo accesible en realidad virtual.

Elegir hacer un gemelo digital con carácter didáctico responde a la creciente necesidad por parte de los museos de preservar objetos frágiles sin renunciar a poder difundir conocimiento sobre ellos. Asegurando su integridad física y facilitando su conocimiento y disfrute por parte del público. En el caso del presente gramófono, el uso de un gemelo digital tiene todo el sentido debido a razones de conservación, pues la pieza se encuentra en un estado delicado. En esta situación la creación de un gemelo digital es una solución ideal para poder seguir divulgado la importancia histórica de este aparato, que representó un avance tecnológico significativo para su época.

El proyecto se alinea con las tendencias actuales en museografía, que apuestan por experiencias interactivas, inmersivas y accesibles, capaces de enriquecer la visita al museo y de adaptarse a entornos virtuales o remotos. Gracias a la realidad virtual, se permite al visitante explorar, manipular y escuchar el funcionamiento del gramófono de forma segura, sin poner en riesgo la pieza original.

Cabe destacar que el carácter multidisciplinar de este proyecto no es solo un desafío técnico y creativo, sino también una forma de homenajear a los avances tecnológicos del pasado, uniendo la tradición con la innovación. convirtiéndolo en un ejemplo práctico de cómo las tecnologías emergentes pueden aplicarse al ámbito cultural para conservar, comunicar y poner en valor nuestro legado histórico.

1.3 Metodología empleada

Para llevar a cabo este Trabajo Final de Grado se ha seguido una metodología interdisciplinar que combina investigación documental, digitalización tridimensional, modelado 3D y desarrollo de entornos interactivos.

En primer lugar, se realizó una labor de documentación histórica sobre el gramófono original, contextualizándolo en la línea temporal de la evolución de la telecomunicación y la reproducción sonora en el siglo XX. Esta fase permitió establecer las bases para una reproducción virtual fiel y significativa. Asimismo, se estudió y se puso a prueba su operativa, probando su funcionalidad y grabando en estudio los sonidos que este produce al reproducir discos contemporáneos a su época de fabricación.

Posteriormente, se procedió a la digitalización del objeto físico mediante un escaneado tridimensional utilizando tecnología LiDAR, obteniendo así un modelo tridimensional que sirvió como referencia para la reconstrucción digital del dispositivo en software de modelado 3D.

A continuación, se corrigió y optimizó el modelo en Blender, teniendo en cuenta tanto la fidelidad visual como la eficiencia para su integración en el entorno virtual. Este modelo fue incorporado posteriormente en un entorno interactivo desarrollado con Unity, donde se implementaron

funcionalidades específicas para la visualización y manipulación del objeto en un entorno de realidad virtual.

Finalmente, se realizaron pruebas funcionales y de experiencia de usuario, con el objetivo de corroborar que el entorno desarrollado es intuitivo y resulta didáctico.

1.4 Estructura del trabajo

Este Trabajo Final de Grado tiene como objetivo principal el desarrollo de un gemelo digital de un gramófono portátil de los años 20 del siglo XX, integrando tecnologías como la digitalización 3D, el modelado tridimensional y la realidad virtual. El documento se ha estructurado en distintos capítulos que detallan cada fase del proyecto.

Para empezar, tenemos una sección introductoria donde se desglosan las metas del trabajo, las razones técnicas, de interés social y académicas que lo motivan, la estrategia utilizada a lo largo de su elaboración y el esquema general del documento.

A continuación, se dedica un capítulo al estudio histórico y contextual del gramófono como dispositivo, revisando su relevancia en la evolución de los sistemas de reproducción de sonido y su importancia como pieza de patrimonio. Igualmente, se abordan conceptos esenciales sobre la digitalización del patrimonio y las oportunidades que brinda la realidad virtual en espacios de museos.

Seguidamente, se explica cómo se digitalizó el gramófono físico usando tecnología de escaneo 3D con sensores LiDAR. Se exponen las acciones llevadas a cabo para lograr una copia digital fiel y se debaten los problemas que surgieron en esta fase.

Posteriormente, se pasa al proceso de modelado y mejora del objeto tridimensional utilizando el programa Blender. Aquí se explican las elecciones de diseño tomadas, la reconstrucción de partes ausentes o malformadas y la preparación del modelo para usarlo en entornos interactivos.

El siguiente bloque se centra en el desarrollo del entorno en Unity, plataforma en la que se construye la experiencia inmersiva. En esta parte se explican los aspectos técnicos y de diseño relacionados con el movimiento, la interacción y la presentación del gramófono en un entorno de museo virtual.

Para terminar, se analizan los resultados conseguidos y se reflexiona sobre las futuras aplicaciones de estas tecnologías en el campo de la museología digital.

Capítulo 2. Contexto histórico y museológico

2.1 El gramófono y su evolución técnica

El gramófono surgió del ingenio de Emile Bernier, quien, interesado por la grabación y reproducción de audio, patentó este nuevo instrumento en 1888, permitiendo la grabación y reproducción de sonido sobre un disco plano, característica principal que lo diferencia del fonógrafo, cuyo soporte debe ser cilíndrico.

Este aparato tiene una calidad sonora superior a sus predecesores, y además era más económico de fabricar, lo que hizo que se popularizara rápidamente, sin embargo, tenía una importante desventaja respecto a instrumentos como el fonógrafo, pues el registro de audio a un nivel casero era mucho más complicado, pues se requería de equipo especializado para ello, lo cual provocó que los usuarios en su mayoría lo utilizaran exclusivamente para la reproducción de sonido.

A lo largo de las décadas, este dispositivo experimentó importantes avances técnicos, como la transición de sistemas mecánicos a eléctricos, la mejora de la calidad sonora o la incorporación de motores de cuerda más eficientes. Estos cambios facilitaron su uso doméstico y convirtieron al gramófono en un objeto común en muchos hogares durante varias décadas del siglo XX. [2]

2.2 Características del gramófono portátil Ica-fono

El gramófono utilizado en el presente TFG se trata de un gramófono portátil fabricado alrededor de 1915 por la marca Ica-fono (Figura 1). Este está fabricado con material tales como metal y madera, protegida con una fibra que imita la piel. La pieza fue diseñada buscando la practicidad, destaca por su estructura compacta, que incluye una caja de transporte que aloja todos los elementos funcionales, como el plato giratorio, el brazo fonocaptor y el mecanismo de cuerda. Además, es fácilmente transportable gracias a su asa de transporte y peso contenido, que, combinado con su capacidad de funcionar sin electricidad, pues el plato gira al darle cuerda, hacen de este gramófono un producto muy interesante en su contexto histórico. [3]

El diseño del Ica-fono responde a la demanda de dispositivos que pudieran usarse en cualquier lugar, sin necesidad de instalaciones eléctricas, algo fundamental en la Europa de principios del siglo XX. A pesar de su reducido tamaño, el sistema de amplificación mecánica mediante bocina interna le permite emitir sonido con una potencia suficiente para uso doméstico o social. Conserva el sistema de reproducción típico de los gramófonos de la época: un brazo articulado que sostiene la cápsula fonocaptora con una aguja que recorre los surcos del disco, transmitiendo las vibraciones a un diafragma que amplifica el sonido de forma totalmente analógica.

Figura 1. Fotografía del gramófono Ica-fono. Fuente: [2]

2.3 El Museo de Historia de la Telecomunicación Vicente Miralles Segarra

El Museo de la Telecomunicación Vicente Miralles Segarra, perteneciente a la Escueta Técnica Superior de Ingeniería de Telecomunicación de la *Universitat Politècnica de València,* alberga una gran colección de dispositivos vinculados a la evolución de la telecomunicación. Entre su inventario podemos encontrar piezas como alguno de los primeros telégrafos, televisores, centralitas telefónicas o el gramófono sobre el cual realizamos el presente trabajo. [4]

2.4 La conservación del patrimonio tecnológico y cultural

La conservación del patrimonio tecnológico plantea retos que divergen de los más conocidos como pueden ser los atribuidos a la conservación de obras de arte. Pues a diferencia de estas, el patrimonio tecnológico, generalmente, se trata de algún tipo de dispositivo de uso cuotidiano o intensivo, lo que añade un aspecto de funcionalidad a su preservación. Estos aparatos contienen partes mecánicas, por tanto, se producen desgastes físicos con el uso y el paso del tiempo, siendo, en algunas ocasiones muy complicado encontrar a alguien capaz de arreglar o de operar correctamente dicho dispositivo. Esto hace que la conservación no se limite a guardar el objeto, sino también a documentar su uso, su contexto histórico y su funcionamiento. [5]

En estas situaciones, se ve el potencial de las tecnologías digitales como la realidad virtual aplicadas a la conservación del patrimonio, permitiendo recrear el aspecto y comportamiento de estos dispositivos sin poner en riesgo el original, de forma que el público puede aprender sobre el funcionamiento de estos objetos históricos haciendo más accesible y atractiva la difusión del patrimonio.

2.5 Patrimonio inmaterial: registro y conservación sonora

En la *Convención para la Salvaguardia del Patrimonio Cultural Inmaterial* (París, 17 de octubre de 2003), la UNESCO consolidó el termino de “patrimonio imaterial” [6], reconociendo la importancia que constituye, pues el patrimonio no se limita únicamente a lo que podemos tocar o ver físicamente. Existen expresiones culturales que forman parte esencial de nuestra historia y que, sin embargo, son intangibles. En el caso del presente proyecto, el sonido, específicamente, las grabaciones musicales y de voz registradas en discos conservados junto a nuestro gramófono, es un elemento clave del patrimonio inmaterial que merece conservarse al igual que el propio dispositivo.

Para preservar su sonoridad, hemos registrado el sonido que producen discos contemporáneos al gramófono al reproducirlos con él. El proceso se llevó a cabo en el estudio de grabación de la ETSIT, donde se montó cuidadosamente el dispositivo y se realizaron grabaciones de alta calidad. El objetivo era integrar ese sonido original en el gemelo digital, permitiendo que la réplica virtual no solo se vea, sino que también suene como el objeto físico real. Esta forma de conservación no solo protege el contenido, sino que también ayuda a entender cómo se percibía en su momento.

Capítulo 3. Estado del arte

3.1 Gemelos digitales: concepto, arquitectura y aplicaciones

La idea del gemelo digital tuvo sus inicios en la industria y el sector aeroespacial a principios de siglo, buscando optimizar el diseño, la producción, el soporte y la supervisión de sistemas intrincados. Se le da crédito a Michael Grieves, quien lo sugirió en el contexto de la gestión del ciclo de vida del producto. Desde ese momento, la idea ha evolucionado de forma notable, gracias al avance de tecnologías como el modelado 3D, el Internet de las Cosas, la inteligencia artificial y la computación en la nube. [7]

Un gemelo digital se entiende como una copia virtual de un objeto, sistema o procedimiento físico. Esta copia no es fija, sino dinámica e interactiva: se crea para simular, monitorear y analizar el comportamiento del original en tiempo real o bajo circunstancias específicas. Su puesta en marcha hace posible observar el estado de un objeto sin tener que intervenir directamente, y a veces, adelantarse a futuros problemas o ensayar posibles situaciones alternativas.

Al principio el uso de gemelos digitales se concentraba en sectores como la industria manufacturera, la automoción o la aeronáutica. Sin embargo, su uso se ha extendido poco a poco a muchos otros campos, incluyendo la medicina, la logística, la energía y, últimamente, el patrimonio cultural y tecnológico. Esta extensión de miras ha requerido reinterpretar la idea para ajustarla a nuevas realidades, en las que el papel del gemelo no siempre es técnico u operativo, sino también comunicativo, educativo o de conservación.

En su versión más completa, un gemelo digital abarca varias capas de información y funcionalidad: desde la representación tridimensional de la forma del objeto, hasta la unión de sensores que hacen posible la recopilación y el análisis de datos al instante. Estos datos se pueden tratar a través de algoritmos de IA para generar simulaciones dinámicas, detectar errores o aconsejar acciones. Además, los sistemas de almacenamiento en la nube permiten que este saber esté al alcance desde cualquier sitio. [8]

Por otro lado, no todos los gemelos digitales precisan de este nivel de complejidad. De hecho, en contextos como el del patrimonio histórico o museológico, el objetivo principal no es tanto controlar o predecir el desempeño de un sistema activo, sino conservar, reinterpretar y difundir bienes culturales que, en muchos casos, ya no se usan o cuyo manejo directo podría representar un peligro para su integridad. Es en estas aplicaciones donde el gemelo digital se vuelve clave para cuidar nuestro legado, tanto tangible como intangible, ya que permite generar réplicas digitales precisas de objetos históricos mediante técnicas como el escaneado 3D, la fotogrametría o el modelado digital manual.

En este proyecto en particular, el gemelo digital busca mostrar con detalle y de forma interactiva un gramófono portátil de los años 20, el Ica-fono. Para lograrlo, se hizo un modelo 3D a partir de escaneos LiDAR, que se pulió y se animó en Blender, para luego integrarlo en un ambiente inmersivo usando Unity. A este modelo se le sumaron grabaciones auténticas del sonido del gramófono, además de una interfaz para manejarlo virtualmente en un entorno recreado, ya sea en la pantalla o con realidad virtual.

Si bien este gemelo digital no está enlazado a sensores físicos ni recibe información al instante, conserva lo principal de la idea original: crear una representación virtual que ayuda a entender, vivir y transmitir las características de un objeto físico desde lejos. Esta adaptación del modelo del gemelo digital al mundo de la cultura demuestra lo adaptable y útil que es, y abre nuevas puertas para mantener vivo nuestro patrimonio de forma activa.

3.2 Realidad virtual aplicada al patrimonio

La realidad virtual ya no es solo usada en videojuegos; se ha transformado en una tecnología con un futuro prometedor en áreas tan diversas como la formación, la sanidad o la protección de nuestro legado. Precisamente, la realidad virtual nos da la posibilidad de diseñar espacios inmersivos que recrean sitios, artefactos y experiencias que, de otra manera, serían imposibles de disfrutar para la mayoría.

Una de las mayores ventajas de usar la realidad virtual en el ámbito del patrimonio es que nos permite examinar objetos históricos sin poner en peligro su integridad. Hay museos y centros de estudio que ya están empleando esta tecnología para exhibir versiones digitales de objetos delicados o de difícil acceso, e incluso para reconstruir escenarios históricos desaparecidos.

Un ejemplo destacado es el British Museum, que ha desarrollado experiencias en VR para permitir al visitante explorar digitalmente tumbas egipcias o piezas del periodo romano sin moverse de casa. [9] Por su parte, el Museo Arqueológico Nacional (MAN) de España ha incorporado recorridos virtuales en los que se pueden observar piezas arqueológicas en detalle, con información contextual y reconstrucciones tridimensionales. [10]

Por si fuera poco, la realidad virtual hace más fácil compartir el saber de una forma más íntima y vivencial. No solo “vemos” la pieza, sino que nos sentimos dentro del ambiente histórico, entendemos mejor su función o incluso oímos sonidos relacionados con su uso original. Esto puede ser muy valioso en la educación, donde el objetivo es despertar el interés del usuario de manera más activa y comprometida.

Iniciativas como "Scan The World" o "Google Arts & Culture" han digitalizado miles de obras y espacios culturales, permitiendo su visualización en 3D o VR a través de navegadores o visores. Esto democratiza el acceso a la cultura y ofrece oportunidades únicas para la conservación, la divulgación y la educación. [11]

3.3 Herramientas tecnológicas utilizadas

Para desarrollar el presente gemelo digital del gramófono portátil Ica-fono se han usado diversas tecnologías que permiten capturar, modelar, texturizar e implementar una experiencia interactiva. A continuación, se describen las herramientas utilizadas en cada fase del proyecto:

3.3.1 Escaneado 3D con LiDAR

La fase inicial del proyecto consistió en transformar el gramófono físico en un modelo digital, aprovechando la tecnología LiDAR, (acrónimo de Light Detection and Ranging), que permitió conseguir una representación tridimensional muy precisa del objeto real. Esta tecnología funciona enviando pulsos de láser que rebotan en la superficie del objeto, estos pulsos vuelven al LiDAR, y, medir el tiempo que tarda el pulso en volver al sensor nos permite calcular la distancia que recorre el láser, recreando así las formas del objeto con precisión. Es una tecnología parecida al radar, pero a diferencia de este, emplea luz para medir en lugar de ondas.

Concretamente usamos como dispositivo LiDAR el iPhone15 Pro, que posee un escáner de este tipo. Esta tecnología fue incluida en los dispositivos móviles de Apple con el objetivo de mejorar la experiencia fotográfica y de realidad aumentada (AR). Sin embargo, la inclusión de esta

tecnología ha supuesto una gran ayuda para pequeños proyectos como este, pues el LiDAR (Figura 2) en el iPhone permite capturar escenas con una calidad suficientemente alta para usos como el modelado 3D de objetos de pequeño y mediano tamaño, reconstrucción de espacios o documentación de patrimonio. 2[12]

Figura 2. Sensor. Fuente: [12]

3.3.2 Modelado y texturizado con Blender

Una vez obtenida la nube de puntos, se procedió con la creación de un modelo 3D limpio, optimizado y adecuado para su integración en entornos virtuales. Para ello se utilizó Blender, una herramienta de software libre y código abierto ampliamente utilizada en las industrias del diseño gráfico, la animación y la visualización 3D. [13]

3.3.3 Desarrollo interactivo con Unity

Finalmente se desarrolla la experiencia interactiva usando Unity, un motor de desarrollo multiplataforma ampliamente utilizado en la creación de videojuegos, simuladores y aplicaciones interactivas en tiempo real. Unity permitió integrar el modelo 3D del gramófono en un entorno digital inmersivo, donde el usuario puede interactuar con el instrumento en un ambiente controlado. [14]

Capítulo 4. Desarrollo del gemelo digital

4.1 Preparación y escaneado del gramófono

Para empezar con la digitalización del gramófono, se usaron las instalaciones de restauración del patrimonio de la UPV, allí, se puso la pieza sobre una mesa circular giratoria para facilitar el proceso de escaneo. Para ello, se empleó un iPhone 15 Pro, que integra un sensor LiDAR, tecnología que permitió capturar con precisión la geometría de objetos mediante la emisión de pulsos de luz láser que miden la distancia a la superficie de los objetos. El software empleado fue ProdAR - Escáner 3D, descargado desde la App Store.

Se recorrió todo el objeto escaneando su volumen principal desde distintos ángulos, asegurando que la pieza se viera correctamente desde todas las perspectivas y sin zonas ocultas. Algunas piezas específicas, como el brazo fonocaptor o la manivela se escanearon por separado, permitiendo así obtener una malla mejor detallada en estas piezas más complejas y pequeñas.

Durante este proceso se tomaron precauciones para evitar reflejos y diferencias en la iluminación, manteniendo una iluminación constante y difusa. Sin embargo, al tener esta pieza abundantes partes metálicas, algunos errores en el escaneo fueron inevitables y se requería de correcciones en postproducción.

Una vez finalizado el escaneo, los modelos se exportaron desde la aplicación en formato USDZ, el cual fue posteriormente convertido a USDC para poder abrirlo correctamente en Blender.

4.2 Procesado y modelado 3D

Una vez terminado el proceso de escaneo, se procedió a su importación en Blender, herramienta de software libre ampliamente utilizada para la edición y modelado 3D. El objetivo de esta fase fue unificar los distintos escaneos, corregir errores geométricos como ruido, o geometría fragmentada y ajustar las texturas para obtener un modelo limpio y coherente, véase Figura 3.

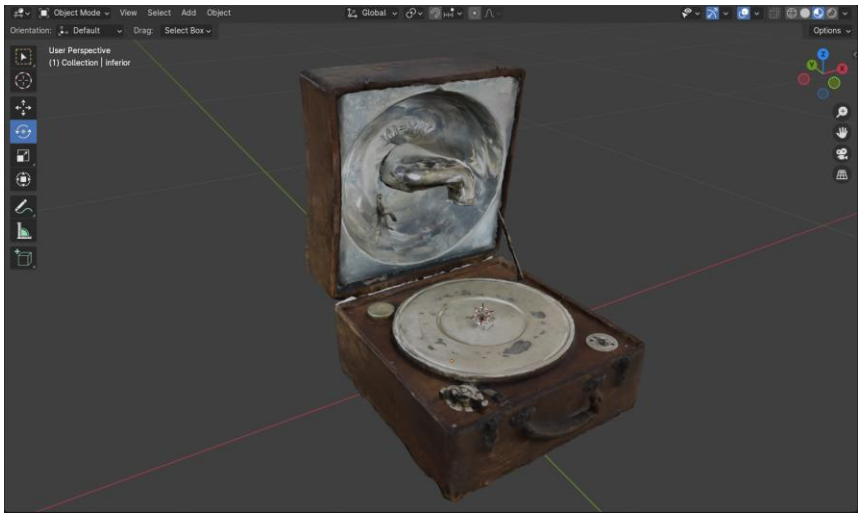

Figura 3. Estructura principal del gramófono en Blender.

A su vez, se modelaron discos de goma laca (Figura 4), conocidos como discos de pizarra, que el Ica-fono es capaz de reproducir y se les aplicaron texturas obtenidas a partir de fotografías reales de sus carátulas y etiquetas centrales.

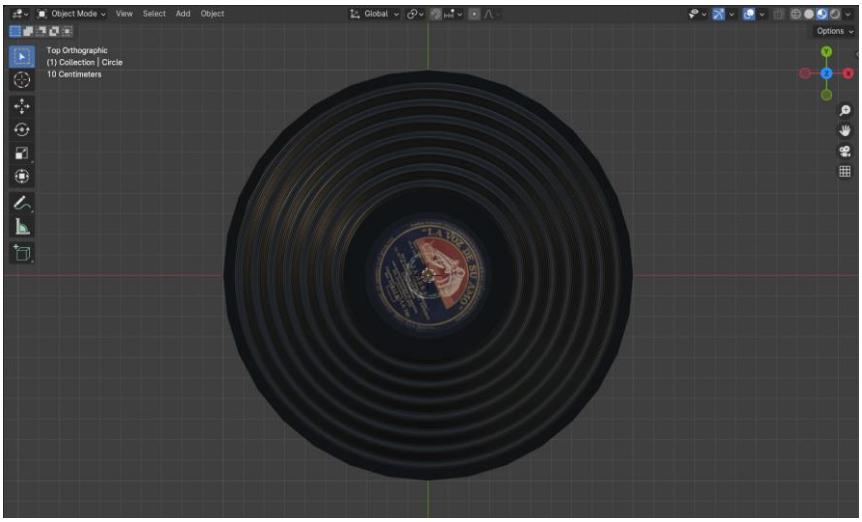

Figura 4. Disco de pizarra modelado en Blender.

Sin embargo, a pesar de cuidado con el que se realizó el escaneo, algunas piezas pequeñas como la manivela han tenido que ser modelas de cero (Figura 5, Figura 6, Figura 7), pues la malla obtenida del LiDAR era inutilizable.

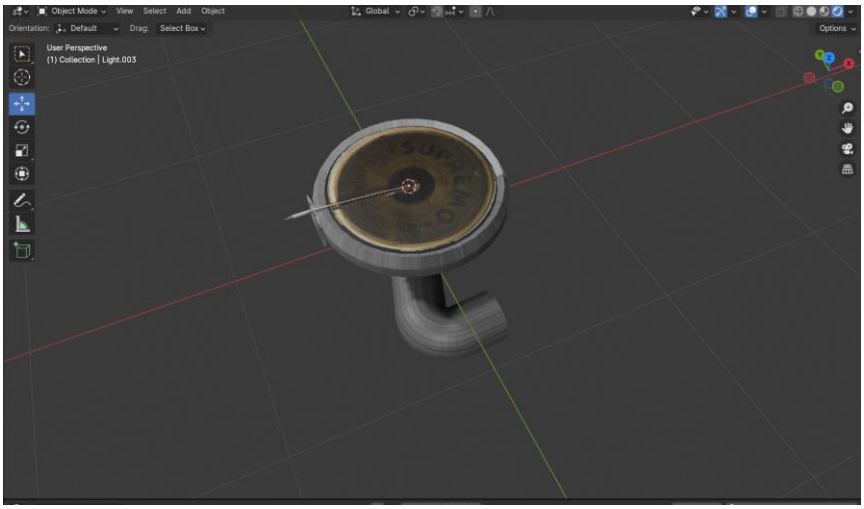

Figura 5. Aguja, diafragma y cavidad resonante modelada en Blender.

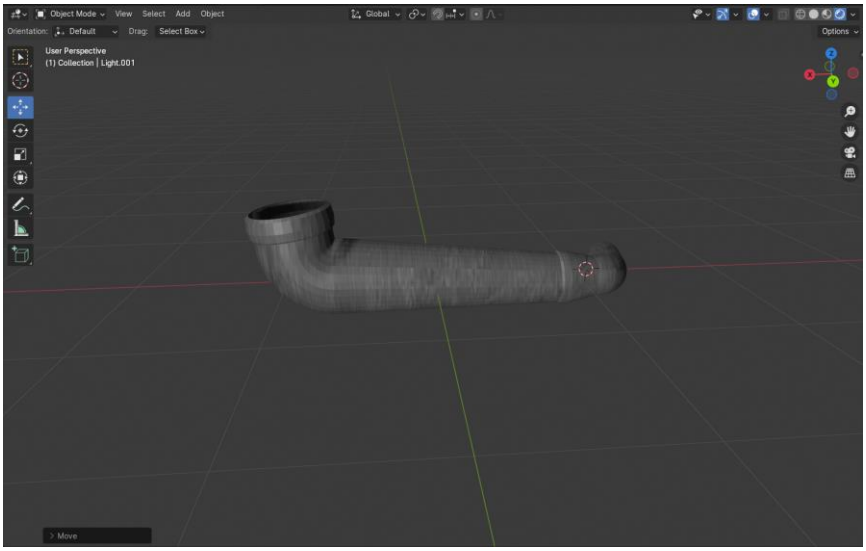

Figura 6. Brazo modelado en Blender.

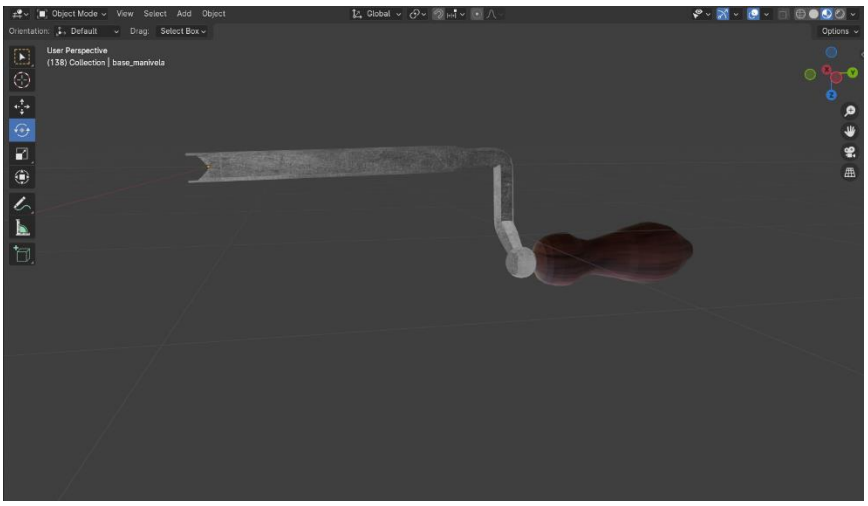

Figura 7. Manivela modelada en Blender.

4.3 Grabación de su sonido

Con el fin de preservar la sonoridad original del gramófono, se realizaron grabaciones de discos de época reproducidos con el dispositivo real, lo cual permitió capturar su sonoridad auténtica. Para registrar el audio, se utilizó el micrófono MPM-2000U de la marca Marantz Professional y el programa de uso libre Audacity, lo que permitió obtener una señal limpia, sin distorsiones, y fiel al sonido original del gramófono. Se procuró minimizar el ruido ambiente y capturar tanto las frecuencias graves del motor como los matices agudos de la aguja sobre el disco.

Además, se fotografiaron las caratulas de los discos para integrarlos en el entorno interactivo. Véase Figura 8.

Figura 8. Caratulas de los discos grabados.

Este proceso tuvo lugar en el estudio de grabación de la ETSIT (Figura 9), lo cual nos permitió obtener un audio de buena calidad para integrarlo en el gemelo digital y así ofrecer una experiencia más auténtica, que no solo reproduce su apariencia, sino también cómo se escuchaba en su tiempo.

Figura 9. Proceso de registro sonoro en estudio de grabación.

4.4 Implementación de animaciones y funcionalidades interactivas

Se han implementado animaciones en las partes móviles de la pieza, véase el plato giratorio, freno, manivela, caja de transporte y el brazo que sostiene la cavidad resonante, el diafragma y la aguja. Para ello se ha utilizado el componente *Animator*, de Unity, usando fotogramas clave como técnica de animación. Estas se han realizado dentro del editor de Unity utilizando la ventana *Animation*, donde se han registrado manualmente los valores de rotación y posición de los objetos a lo largo del tiempo para generar las secuencias deseadas. Cada animación fue vinculada a un parámetro dentro del *Animator Controller*, permitiendo activarlas o desactivarlas mediante scripts en C#. Estas se sincronizan con los eventos de la máquina de estados, activando las animaciones en respuesta a acciones del usuario, como insertar la manivela o quitar el freno.

El desarrollo se ha hecho buscando la interactividad con la pieza, para ello se han emulado los movimientos que las partes movibles de la pieza harían si se tratase del gramófono real, lo cual nos permite representar fielmente la forma y aspecto del objeto original además de dotar al gemelo digital de interactividad y de un comportamiento dinámico.

4.5 Desarrollo de la aplicación de realidad virtual

Finalmente se procedió a la última fase de este proyecto, que consiste en el desarrollo de una aplicación interactiva en realidad virtual. Esta se desarrolló utilizando el motor de videojuegos Unity en su versión 2022.2.21f1 y los paquetes XR Interaction Toolkit y XR Plugin Management, que proporciona un conjunto de herramientas para la creación de entornos de realidad virtual. Este paquete proporciona componentes ya integrados para detectar y responder a eventos como la selección de objetos, ray casting, detección de colisiones etc. En la escena principal se configuró un entorno básico con un Complete XR Origin Set Up, que incluye componentes esenciales como el Input Action Manager, el XR Interaction Manager y el Event System, al que se le añadió el módulo XR UI Input Module para permitir interacciones con interfaces gráficas en VR.

El sistema de locomoción se configuró con proveedores de movimiento y rotación como Snap Turn Provider, Continuous Turn Provider, Dynamic Move Provider, y un Character Controller que permite desplazarse libremente por el entorno virtual. [15][16]

En la figura 10 se pueden observar los *Project Settings* de Unity y en la figura 11 se muestran los paquetes importados para realizar el presente proyecto.

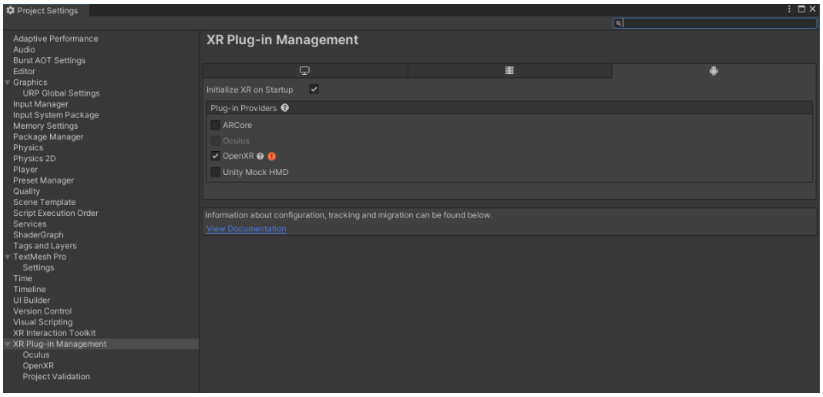

Figura 10. Configuración del proyecto en Unity.

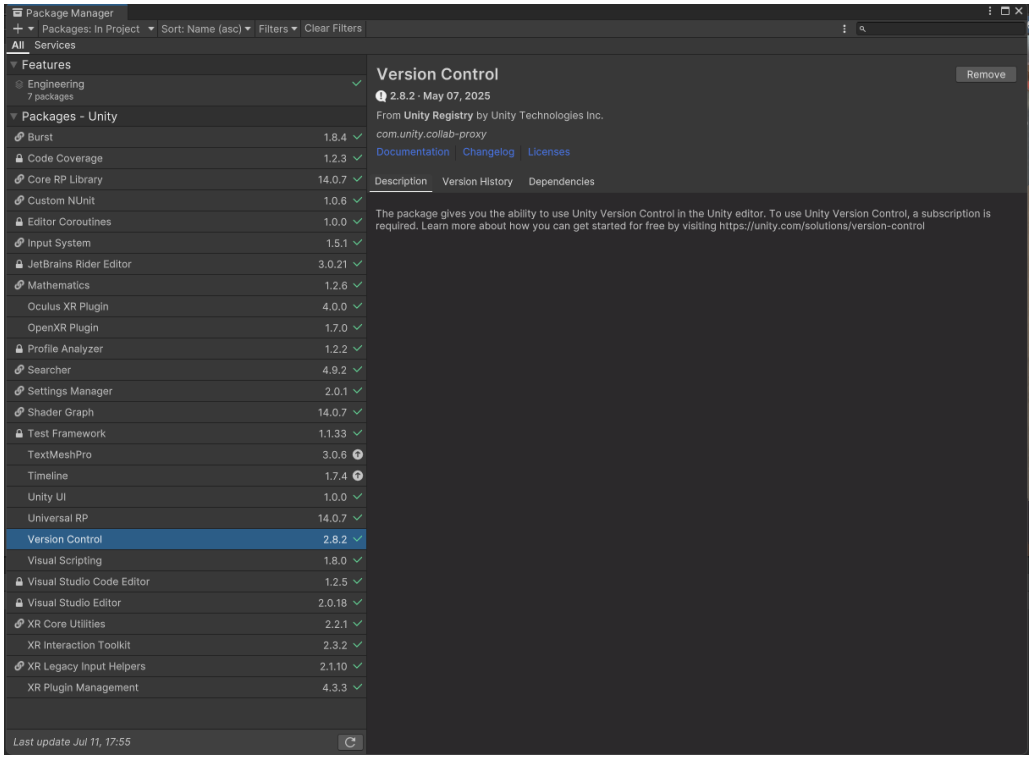

Figura 11. Package Manager de Unity.

Con la configuración finalizada de creó y diseñó la sala del gramófono (Figura 12)

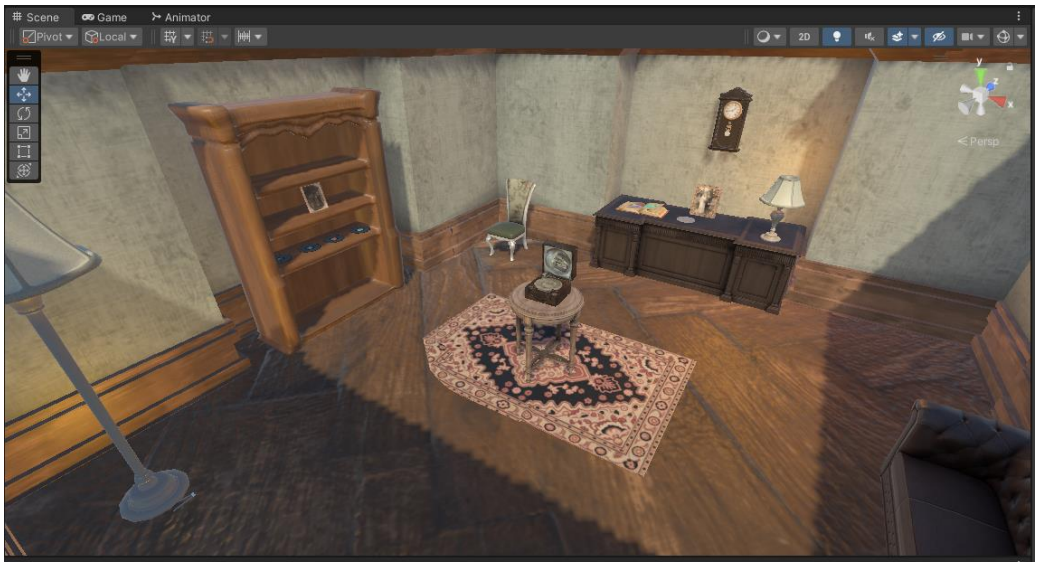

Figura 12. Disposición inicial de la sala del gemelo digital en Unity.

Se han usado algunos objetos importados desde la *Asset Store* de Unity para decorar el entorno.

Para estructurar el comportamiento del gramófono de forma coherente e interactiva, se implementó una máquina de estados, véase Figura 13. Esta controla los distintos estados en los que se puede encontrar el gramófono, limitando las acciones que el usuario puede realizar en

función del estado en que se encuentre. De esta forma se asegura que el dispositivo virtual se opere siguiendo el orden lógico de funcionamiento del original.

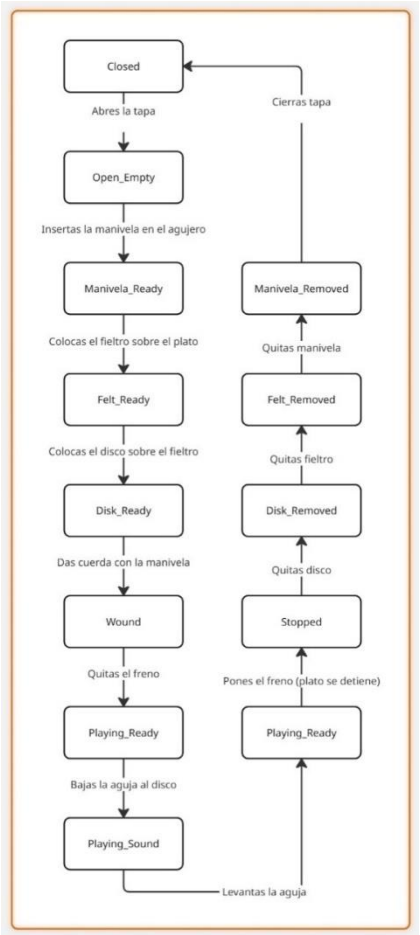

Figura 13. Máquina de estados del gramófono.

Para permitir una interacción directa con los componentes móviles del gramófono, se utilizaron los scripts *XRGrabInteractable* y *XRBaseInteractable*. Esto permitió que el usuario pueda:

- Abrir la tapa del gramófono.
- Insertar la manivela en el lateral.
- Colocar el fieltro y el disco en su sitio.
- Dar cuerda girando la manivela.
- Quitar el freno y girar el plato.
- Colocar brazo y aguja sobre el disco y comenzar la reproducción.

Cada acción disponible está condicionada al estado actual del dispositivo, evitando errores y acciones erróneas.

Seguidamente, para mejorar la experiencia del usuario se implementaron efectos sonoros al abrir el gramófono, dar cuerda, girar el plato, y al bajar la aguja sobre el disco, se activa un clip de audio real del disco elegido. Asimismo, se muestran mensajes en pantalla mediante un sistema de diálogo que guía al usuario paso a paso durante el proceso de interacción con el dispositivo. Este se ha desarrollado usando un componente *TextMeshPro* y animaciones de escritura progresiva

para que se sienta fluida la experiencia. El cuadro de dialogo informa al usuario del estado actual del gramófono y de la siguiente acción posible.

4.6 Scripts desarrollados

En esta sección se va a exponer el código desarrollado para poder implementar la experiencia de realidad virtual.

EstadoGramofono:

Representa los distintos estados del gramófono en su ciclo completo de uso.

```
public enum EstadoGramofono
{
    // El gramófono está cerrado, no se puede interactuar todavía
    Closed,

    // El gramófono está abierto pero no tiene ninguna pieza colocada
    Open_Empty,

    // La manivela ha sido colocada
    Manivela_Ready,

    // El fieltro (plato giratorio) ha sido colocado
    Felt_Ready,

    // El disco ha sido colocado sobre el fieltro
    Disk_Ready,

    // El gramófono ha sido cargado girando la manivela
    Wound,

    // Todo está montado y listo para reproducir
    Playing_Ready,

    // El gramófono está reproduciendo sonido
    Playing_Sound,

    // La reproducción ha sido detenida mediante el freno
    Stopped,

    // El disco ha sido retirado tras detenerse la reproducción
    Disk_Removed,

    // El fieltro ha sido retirado
    Felt_Removed,

    // La manivela ha sido retirada, el gramófono está completamente
desmontado
    Manivela_Removed
```

```
}
```

EstadoGramofonoController:

Permite simular paso a paso el montaje y desmontaje del gramófono, actúa de máquina de estados.

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

// Script que controla la lógica de estados del gramófono en una
experiencia VR interactiva.
// Permite simular el montaje y desmontaje del gramófono paso a paso,
gestionar el disco insertado,
// cambiar estados y reproducir audio asociado a cada disco.

public class EstadoGramofonoController : MonoBehaviour
{
    // Estado actual del gramófono
    public EstadoGramofono estadoActual = EstadoGramofono.Closed;

    // Referencia al sistema de UI que muestra mensajes al usuario
    public DialogueUI dialogueUI;

    // Fuente de audio general (no usada directamente en reproducción de
disco)
    public AudioSource audioSource;

    // Disco actualmente colocado
    private Disco discoActual;

    // Cambia el estado del gramófono y muestra mensaje contextual
    private void CambiarEstado(EstadoGramofono nuevoEstado)
    {
        EstadoGramofono estadoAnterior = estadoActual;
        estadoActual = nuevoEstado;

        if (dialogueUI != null)
            dialogueUI.MostrarDialogoConContexto(nuevoEstado.ToString(),
estadoAnterior.ToString());
    }

    // Abre la tapa del gramófono si está cerrado
    public void AbrirTapa()
    {
        if (estadoActual == EstadoGramofono.Closed)
        {
            Debug.Log("Tapa abierta");
            CambiarEstado(EstadoGramofono.Open_Empty);
```

```
        }
        else
        {
            Debug.LogWarning("No puedes abrir la tapa en este estado: " +
estadoActual);
        }
    }

    // Inserta la manivela si el gramófono está abierto y vacío
    public void InsertarManivela()
    {
        if (estadoActual == EstadoGramofono.Open_Empty)
        {
            Debug.Log("Manivela insertada");
            CambiarEstado(EstadoGramofono.Manivela_Ready);
        }
        else
        {
            Debug.LogWarning("No puedes insertar la manivela en este
estado.");
        }
    }

    // Coloca el fieltro si la manivela está lista
    public void ColocarFieltro()
    {
        if (estadoActual == EstadoGramofono.Manivela_Ready)
        {
            Debug.Log("Fieltro colocado");
            CambiarEstado(EstadoGramofono.Felt_Ready);
        }
    }

    // Coloca un disco sobre el fieltro y lo guarda como disco actual
    public void ColocarDisco(GameObject discoGO)
    {
        if (estadoActual == EstadoGramofono.Felt_Ready)
        {
            discoActual = discoGO.GetComponent<Disco>();
            if (discoActual != null)
            {
                Debug.Log("Disco colocado: " + discoActual.nombreDisco);
                dialogueUI.MostrarDialogoPersonalizado("Disco
seleccionado: " + discoActual.nombreDisco);
            }

            CambiarEstado(EstadoGramofono.Disk_Ready);
        }
```

```
    }

    // Da cuerda al gramófono si hay disco colocado
    public void DarCuerda()
    {
        if (estadoActual == EstadoGramofono.Disk_Ready)
        {
            Debug.Log("Cuerda dada");
            CambiarEstado(EstadoGramofono.Wound);
        }
    }

    // Quita el freno para permitir que el plato gire
    public void QuitarFreno()
    {
        if (estadoActual == EstadoGramofono.Wound)
        {
            Debug.Log("Freno quitado");
            CambiarEstado(EstadoGramofono.Playing_Ready);
        }
    }

    // Baja la aguja sobre el disco y comienza la reproducción
    public void BajarAguja()
    {
        if (estadoActual == EstadoGramofono.Playing_Ready)
        {
            Debug.Log("Aguja bajada: empieza el sonido");
            CambiarEstado(EstadoGramofono.Playing_Sound);
        }
    }

    // Sube la aguja y detiene el sonido, sin parar el plato
    public void SubirAguja()
    {
        if (estadoActual == EstadoGramofono.Playing_Sound)
        {
            Debug.Log("Aguja levantada");
            CambiarEstado(EstadoGramofono.Playing_Ready);
        }
    }

    // Vuelve a poner el freno y detiene el plato
    public void PonerFreno()
    {
        if (estadoActual == EstadoGramofono.Playing_Ready)
        {
            Debug.Log("Freno puesto: el plato se detiene");
```

```
            CambiarEstado(EstadoGramofono.Stopped);
        }
    }

    // Quita el disco después de detener el plato
    public void QuitarDisco()
    {
        if (estadoActual == EstadoGramofono.Stopped)
        {
            Debug.Log("Disco quitado");
            CambiarEstado(EstadoGramofono.Disk_Removed);
        }
    }

    // Quita el fieltro después de quitar el disco
    public void QuitarFieltro()
    {
        if (estadoActual == EstadoGramofono.Disk_Removed)
        {
            Debug.Log("Fieltro quitado");
            CambiarEstado(EstadoGramofono.Felt_Removed);
        }
    }

    // Quita la manivela después del fieltro
    public void QuitarManivela()
    {
        if (estadoActual == EstadoGramofono.Felt_Removed)
        {
            Debug.Log("Manivela quitada");
            CambiarEstado(EstadoGramofono.Manivela_Removed);
        }
    }

    // Cierra la tapa del gramófono después de desmontarlo por completo
    public void CerrarGramofono()
    {
        if (estadoActual == EstadoGramofono.Manivela_Removed)
        {
            Debug.Log("Gramófono cerrado");
            CambiarEstado(EstadoGramofono.Closed);
        }
    }

    // Devuelve el estado actual del gramófono
    public EstadoGramofono ObtenerEstado()
    {
        return estadoActual;
```

```
    }

    // Devuelve el disco actualmente colocado
    public Disco ObtenerDiscoActual() => discoActual;

    // Fuente de audio específica para reproducir el disco
    private AudioSource audioSourceDisco;

    // Reproduce el sonido asociado al disco actual
    public void ReproducirDisco()
    {
        if (discoActual != null && discoActual.clipSonido != null)
        {
            if (audioSourceDisco == null)
                audioSourceDisco =
gameObject.AddComponent<AudioSource>();

            audioSourceDisco.clip = discoActual.clipSonido;
            audioSourceDisco.loop = true;
            audioSourceDisco.playOnAwake = false;
            audioSourceDisco.spatialBlend = 0f;

            audioSourceDisco.Play();
            Debug.Log("Reproduciendo disco: " + discoActual.nombreDisco);
        }
    }

    // Detiene la reproducción del disco
    public void DetenerDisco()
    {
        if (audioSourceDisco != null && audioSourceDisco.isPlaying)
        {
            audioSourceDisco.Stop();
            Destroy(audioSourceDisco);
            audioSourceDisco = null;
            Debug.Log("Disco detenido.");
        }
    }
}
```

Además, se han implementado los siguientes scripts:

AbrirGramofono: Al entrar en el entorno, encontramos una habitación en la que tenemos, junto a algunos muebles de decoración, el gemelo digital del gramófono. Este se encuentra cerrado, si interactuamos con él se abrirá.

AbrirTapaConBrazo: Controla la animación del brazo, para que siga a la tapa durante la animación de apertura

BrazoSigueTapa: El movimiento del brazo puede estar restringido o no al de la tapa del gramófono, desde este script se controla.

ManivelaSlot: Se controla la inserción de la manivela en el lateral derecho del gramófono.

FieltroSlot: Controla el colocar y retirar el fieltro que se coloca sobre el plato del gramófono.

DiscoSlot: Al igual que el fieltro, controla la colocación y retirada del disco sobre el gramófono.

DiscoSeleccionado: Este script muestra un mensaje en el cuadro de dialogo informando del disco escogido.

DarCuerda: Permite dar cuerda al gramófono, detecta el giro de la manivela y avisa al jugador cuando está suficientemente cargada.

Frenar: Se encarga de controlar como interactuamos con el freno del gramófono.

MoverBrazo: Para reproducir Sonido necesitamos llevar el brazo y la aguja al disco, este script gestiona la interacción con el brazo.

EncajarAguja: Gestiona el comportamiento de la aguja para ponerla sobre el disco y retirarla.

GirarDisco: Hace girar al disco una vez el plato del gramófono se libera del freno

Disco: Se asocia a cada disco de la escena, dándole nombre y un clip de audio a cada disco

CerrarGramofono: Script que permite cerrar el gramófono después de montarlo y desmontarlo.

DialogueUI: Gestiona la aparición de mensajes en la interfaz de la sala del gramófono.

El detalle de estos scripts se puede encontrar en el Anexo *Scripts del proyecto* al final del documento.

4.7 Resultados

Esta sección muestra imágenes de cada funcionalidad desarrollada en el presente proyecto, correspondientes a las diferentes fases del uso del gramófono.

A continuación, se muestra el gramófono colocado sobre una mesa, en su estado inicial antes de comenzar la interacción (Figura 14).

Figura 14. Gramófono sobre la mesa.

Para empezar, se debe abrir la tapa del gramófono para poder interactuar con sus componentes (Figura 15).

Figura 15. Apertura del gramófono.

Una vez abierto, el usuario debe insertar la manivela en el lateral del gramófono, como se ilustra en la Figura 16.

Figura 16. Colocación de manivela.

Posteriormente, se debe colocar el fieltro sobre el plato, lo cual se muestra en la Figura 17.

Figura 17. Colocación de fieltro sobre plato.

Seguidamente, como vemos en la Figura 18, se debe seleccionar uno de los discos disponibles para su reproducción.

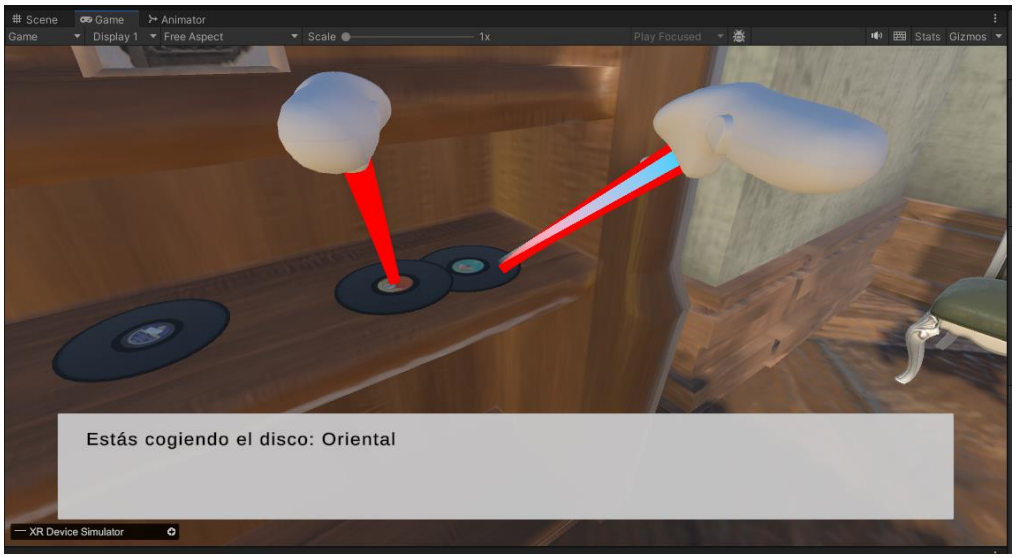

Figura 18. Elegir un disco.

Una vez elegido el disco, este se coloca sobre el fieltro (Figura 19).

Figura 19. Colocación del disco sobre fieltro.

El usuario debe entonces girar la manivela describiendo movimientos circulares con el mando que agarra la manivela para cargar la cuerda del mecanismo, como se ve en la Figura 20.

Figura 20. Cargar la cuerda.

Después, se retira el freno para que el plato comience a girar (Figura 21).

Figura 21. Quitar freno.

Se baja el brazo fonocaptor para prepararlo para la reproducción (Figura 22).

Figura 22. Bajada de brazo.

Luego, se baja la aguja sobre el disco para iniciar la reproducción de sonido (Figura 23).

Figura 23. Bajada de Aguja.

A continuación, el gramófono reproduce el audio del disco seleccionado (Figura 24).

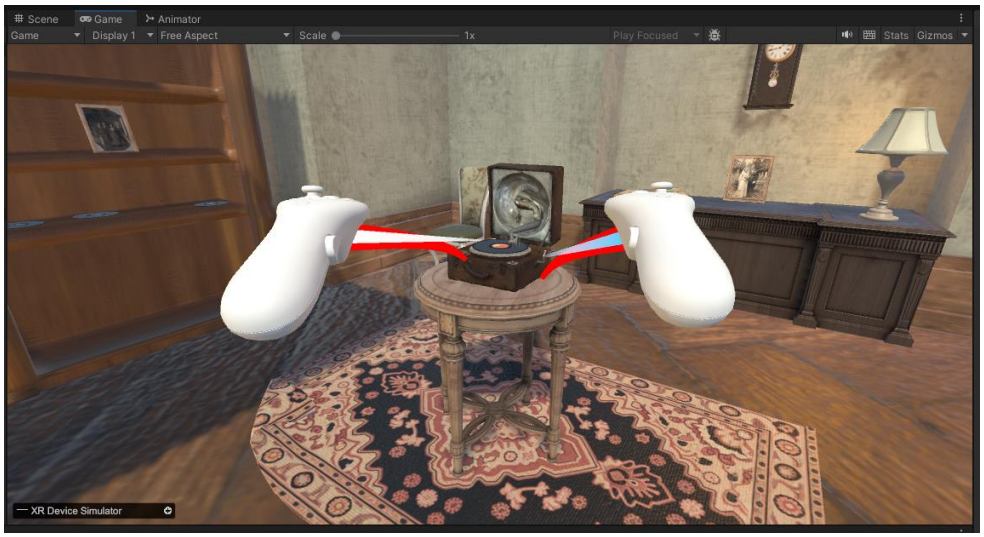

Figura 24. Escuchar reproducción de disco.

Para dejar de utilizar el gramófono, primero se levanta la aguja (Figura 25).

Figura 25. Quitar aguja.

Después se pone el freno, lo cual detiene el giro del plato (Figura 26).

Figura 26. Puesta de freno.

Se retira el disco del plato, como se puede apreciar en la Figura 27.

Figura 27. Retirada de disco.

Posteriormente, se retira el fieltro que estaba debajo del disco (Figura 28).

Figura 28. Retirada de fieltro.

Luego se retira la manivela (Figura 29).

Figura 29. Retirada de manivela.

Finalmente, se cierra el gramófono, completando así el ciclo interactivo (Figura 30).

Figura 30. Cerrar gramófono.

Capítulo 5. Conservación y difusión del patrimonio mediante realidad virtual

Como ya se ha comentado, el uso de nuevas tecnologías está revolucionando el ámbito del patrimonio, abriendo nuevas vías no exploradas para su conservación, divulgación y documentación. Es en este contexto que ha aparecido la tecnología de la realidad virtual, ofreciendo un enfoque innovador que facilita el acceso las piezas de patrimonio histórico, permitiendo que se muestre y se divulgue sobre el patrimonio de una forma distinta a la habitual.

5.1 Aportaciones del gemelo digital a la conservación preventiva

El gemelo digital creado en el presente proyecto sirve como un instrumento útil para la preservación preventiva, entendida como el conjunto de acciones tomadas para prevenir o reducir el desgaste de los bienes culturales. Inicialmente, la captura tridimensional del gramófono asegura la preservación precisa de su forma, tamaño y apariencia de la superficie, crucial si el elemento original está dañado, alterado o perdido.

Además, al permitir interactuar de forma virtual con la pieza, se reduce la necesidad de interactuar con el objeto real en exposiciones o en actividades educativas. Esto permite mantener su integridad física, pues no la expone a desgaste innecesario. Por otro lado, el gemelo digital permite la simulación de la funcionalidad de la pieza, que podría ser imposible demostrar con el objeto real debido a su delicadeza o a su mal estado.

Por otro lado, desde un punto de vista documental, el modelo incorpora detalles adicionales, como audio, gráficos y un sistema fácil de usar, que amplía la comprensión del sujeto, creando una valiosa herramienta multifacética para investigación, restauración y gestión del patrimonio.

5.2 Valor educativo y experiencia para los visitantes

Uno de los objetivos principales de la iniciativa es mejorar el aspecto del aprendizaje y conocimiento de la historia humana a través de actividades atractivas e inmersivas. Gracias a la implementación de la realidad virtual en Unity, el usuario no solo puede ver el gramófono de cerca, sino también comprometerse con él de manera realista y sistemática, siguiendo el procedimiento original de uso: dar cuerda con la manivela, colocar el disco, accionar la aguja y reproducir el sonido.

Esta experiencia mejora enormemente la comprensión que el usuario puede llegar a tener del objeto y su contexto, especialmente con usuarios no especializados, y más si no están habituados al uso de la realidad virtual, para este tipo de usuario encontrarse así con un objeto en exposición es una experiencia inolvidable. Al involucrar directamente nuestros sentidos, creamos una conexión más cercana con el patrimonio, lo que aumenta su valor y alienta a pasarlo a la siguiente generación.

En términos generales, este tipo de proyectos proporcionan una experiencia envolvente y didáctica, potencialmente implementable en distintos museos o centros educativos, pues permitiría que superasen barreras físicas o geográficas, facilitando el acceso al patrimonio a personas con movilidad reducida.

5.3 Limitaciones y oportunidades futuras

Aunque conlleva numerosos beneficios, usar gemelos digitales y entornos de realidad virtual en patrimonio, también presenta ciertas limitaciones. Por una parte, la calidad del escaneado y cómo de fiel sea el modelo dependen de los recursos que se tengan para la compra de las herramientas

de captura y de la habilidad del modelador. Y, aunque se puede acceder al entorno de realidad virtual con dispositivos como Meta Quest 2, es necesario tener algo de equipo técnico y saber usarlo. Observándolo desde un punto de vista más general, es importante recordar que un gemelo digital, aunque sea muy fiel, no puede replicar lo que se siente al estar frente al objeto real. Debe verse como algo que complementa la exhibición y el cuidado físico, no como un sustituto.

No obstante, las posibilidades futuras son muy amplias, pues a medida que las herramientas se abaraten y se aumente la accesibilidad a las tecnologías de realidad virtual, se podrán crear catálogos virtuales más grandes, exposiciones interactivas en línea e incluso reconstrucciones contextuales completas que muestren el objeto en su contexto original.

Además, si se añaden técnicas de inteligencia artificial, como el reconocimiento de gestos o la traducción automática de textos, se podría mejorar aún más la experiencia y adaptarla a distintos tipos de usuarios. En resumen, este proyecto no es solo un uso aislado de la realidad virtual en el patrimonio, sino que también marca el camino hacia una nueva manera de entender, proteger y dar a conocer la cultura material del pasado en la era digital.

Capítulo 6. Conclusiones

En este capítulo se presenta un resumen de los resultados obtenidos a lo largo del desarrollo del presente proyecto, así como una reflexión acerca de lo aquello que se ha aprendido y las posibles mejoras para la evolución futura de este proyecto.

6.1 Conclusiones generales del proyecto

El proyecto finalmente ha logrado cumplir su meta: el desarrollo de un gemelo digital funcional e interactivo de un gramófono portátil Ica-fono de los años 20 del siglo XX, que pertenece al Museo de Historia de la Telecomunicación Vicente Miralles Segarra. Combinando técnicas de escaneado 3D, modelado, programación y diseño de entornos interactivos, se ha conseguido crear un gemelo digital fiel visual, sonora y funcionalmente. A través del desarrollo de un gemelo digital interactivo del gramófono Ica-fono, ha sido posible no solo preservar digitalmente un objeto físico con alto valor histórico, sino también ofrecer una experiencia inmersiva que facilita su comprensión y apreciación por parte del público.

Técnicamente, ha supuesto un reto pues como ya hemos comentado se han usado distintas tecnologías, empezando por el escaneado 3D, mediante LiDAR y, el modelado en Blender, la implementación de animaciones y el entorno virtual en Unity y la programación de una aplicación de realidad virtual compatible con dispositivos como Meta Quest 2. Y, más allá de los aspectos técnicos, este proyecto pone en evidencia el potencial de las tecnologías digitales como herramientas al servicio del patrimonio. Al igual que ocurre en entornos industriales, donde los gemelos digitales se utilizan para monitorizar, optimizar o prever fallos, en el ámbito cultural permiten conservar, documentar y hacer accesibles objetos históricos que, por su fragilidad o valor, no siempre pueden manipularse físicamente.

A modo de conclusión, el desarrollo de este Trabajo de Fin de Grado ha sido toda una experiencia, que más allá de mejorar mis habilidades de modelado y desarrollo de entornos virtuales, me ha servido también para conocer de forma más intensa el patrimonio material e inmaterial que tenemos, lo que me ha permitido apreciarlo más de lo que ya hacía, además, contribuir a la conservación de este ha sido algo que nunca olvidaré.

6.2 Propuestas de mejora y trabajo futuro

Aunque el presente gemelo digital cumpla con los objetivos que se planearon inicialmente, hay, sin dudas, diversas líneas de mejora que podrían mejorar la experiencia final. Una de las mejoras más evidentes podría ser el refinamiento de la simulación de los mecanismos físicos internos del gramófono, especialmente el sistema de cuerda y el desplazamiento del brazo reproductor. Pues, al representar estos sistemas con mayor realismo, se podría entender con precisión el funcionamiento del dispositivo.

Para lograr esto habría que modelar con atención al detalle el sistema de engranajes que se encuentra en la parte inferior del aparato, además de programar una serie de scripts que controlasen el movimiento de estos, imitando su comportamiento original.

En cuanto a la interfaz de usuario, se podría mejorar la experiencia educativa, implementando subtítulos y una narración que explique la historia del gramófono, así como su funcionamiento mientras el usuario interactúa con él.

También se propone la adaptación de la aplicación para navegadores web, lo que la haría accesible desde cualquier dispositivo sin necesidad de usar unas gafas de realidad virtual

Otra mejora significativa sería implementar un modo multiusuario en red, que permitiera a varios usuarios interactuar con el gramófono simultáneamente, o atender a una explicación en la que una persona experta explique el funcionamiento del objeto mientras un grupo observa, cada uno con una perspectiva única del gramófono, ideal para entornos educativos o visitas guiadas en museos donde se quiera fomentar la colaboración o el aprendizaje grupal.

Finalmente, una posible línea de trabajo futuro sería la extensión del proyecto a otros dispositivos del Museo de Historia de la Telecomunicación Vicente Miralles Segarra, con el fin de construir un catálogo completo de gemelos digitales interactivos que ayuden a divulgar y conservar el patrimonio tecnológico.

Capítulo 7. Objetivos de desarrollo sostenible de la agenda 2030

El Trabajo de Fin de Grado presentado se alinea con varios Objetivos de Desarrollo Sostenible (ODS) definidos por la Agenda 2030 de las Naciones Unidas. A través del diseño y desarrollo del proyecto, se han tenido en cuenta principios de sostenibilidad, inclusión, innovación y responsabilidad social.

- ODS 4: Educación de calidad.
  El desarrollo de este trabajo promueve la adquisición de competencias tecnológicas, el pensamiento crítico y el aprendizaje práctico, contribuyendo a una educación más inclusiva y equitativa. Si el proyecto se orienta a una herramienta educativa, se refuerza aún más este objetivo.
- ODS 9: Industria, innovación e infraestructura
  Este proyecto fomenta la innovación mediante el uso de tecnologías emergentes (por ejemplo, realidad virtual) así como la mejora de infraestructuras digitales. Se demuestra cómo la tecnología puede ser aplicada de forma creativa para resolver problemas prácticos.

Capítulo 8. Bibliografía

[1] Europa, E. V. (2024, 13 noviembre). *Virtualización del patrimonio cultural*. EVE Museos + Innovación. https://evemuseografia.com/2024/11/13/virtualizacion-del-patrimonio-cultural/

[2] *Gramófono – Museo ETSIT*. (s. f.). https://museotelecomvlc.webs.upv.es/gramofono-2/

[3] Centro de Documentación Musical de Andalucía. (s.f.). *Gramófono*. Junta de Andalucía. https://www.centrodedocumentacionmusicaldeandalucia.es/-/506-gramofono?

[4] *Museo ETSIT – Museo de la Telecomunicación Vicente Miralles Segarra*. (s. f.). https://museotelecomvlc.webs.upv.es/

[5] Marta, C. L. (2024, June 25). *La importancia del patrimonio tecnológico. Proceso de conservación y restauración de una máquina de proyección Pathé-Baby*. https://riunet.upv.es/entities/publication/1f041c8f-6995-4351-9e8c-7d32ca979a7b

[6] Instituto Andaluz del Patrimonio Histórico. (s.f.). *¿Qué es el patrimonio cultural inmaterial?* Junta de Andalucía. https://www.iaph.es/export/sites/default/galerias/patrimonio-cultural/documentos/gestion-informacion/que_es_patrimonio_inmateria.pdf

[7] Ibm. (2025, 7 marzo). Gemelo Digital. *¿Qué es un gemelo digital?* https://www.ibm.com/es-es/think/topics/what-is-a-digital-twin

[8] *Gemelos digitales: qué es, tipos y ejemplos | Repsol*. (2025, 11 junio). REPSOL. https://www.repsol.com/es/energia-futuro/tecnologia-innovacion/gemelos-digitales/index.cshtml

[9] BBC News. (2015, August 4). *British Museum offers virtual reality tour of Bronze Age*. https://www.bbc.com/news/technology-33772694

[10] *La realidad virtual llega al Museo Arqueológico Nacional*. (2017, May 20). Ministerio De Cultura. https://www.cultura.gob.es/actualidad/2017/05/20170508-virtual.html

[11] *Scanning the world and democratising culture through 3D | Europeana PRO*. (n.d.). Europeana PRO. https://pro.europeana.eu/post/scanning-the-world-and-democratising-culture-through-3d

[12] OpenTopography. (s. f.). *iPhone lidar with applications for the geosciences | OpenTopography*. https://opentopography.org/blog/iphone-lidar-applications-geosciences

[13] Blender Foundation. (n.d.). *Features — Blender.org*. blender.org. https://www.blender.org/features/

[14] Unity. (n.d.). *Plataforma de desarrollo en tiempo real de Unity | Motor de VR, AR, 3D y 2D*. https://unity.com/es

[15] *XR Interaction Toolkit | XR Interaction Toolkit | 3.2.0*. (s. f.). https://docs.unity3d.com/Packages/com.unity.xr.interaction.toolkit@3.2/manual/index.html

[16] Technologies, U. (s. f.). *Unity - Manual: XR Plugin Management*. https://docs.unity3d.com/6000.1/Documentation/Manual/com.unity.xr.management.html

Capítulo 9. Anexo: Scripts del gramófono

AbrirGramofono:

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.XR.Interaction.Toolkit;

// Este script se encarga de abrir la tapa del gramófono cuando el usuario interactua
con él
public class AbrirGramofono : MonoBehaviour
{
    // Referencia al componente Animator para reproducir la animación de apertura
    public Animator animator;

    // Referencia al controlador de la máquina de estados del gramófono
    public EstadoGramofonoController fsm;

    // Componente XR que permite que el objeto sea interactuable en VR
    private XRBaseInteractable interactable;

    // Variable booleana para evitar múltiples aperturas
    private bool yaAbierta = false;

    void Awake()
    {
        // Se obtiene el componente XRBaseInteractable
        interactable = GetComponent<XRBaseInteractable>();

        // Se registra la función OnSelect para cuando el objeto sea seleccionado
        interactable.selectEntered.AddListener(OnSelect);
    }

    // Esta función se ejecuta cuando el usuario selecciona el objeto
    private void OnSelect(SelectEnterEventArgs args)
    {
        // Si la tapa no se ha abierto aún, y el estado actual del gramófono es
"Closed"
        if (!yaAbierta && fsm != null && fsm.estadoActual == EstadoGramofono.Closed)
        {
            // Se lanza el trigger de la animación de apertura
            animator.SetTrigger("Abrir");

            // Se marca como abierta para evitar que se vuelva a ejecutar
            yaAbierta = true;

            // Se notifica a la máquina de estados que se ha abierto la tapa
```

```
            fsm.AbrirTapa();
        }
    }
}
```

Abrir tapa con brazo:

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
//Controla la animación de apertura de la tapa y coordinar cu□ndo el brazo debe
seguirla
public class AbrirTapaConBrazo : MonoBehaviour
{
    public Animator animator;
    public BrazoSigueTapa brazoScript;

    public void AbrirTapa()
    {
        if (brazoScript != null)
            brazoScript.ActivarConstraint();

        if (animator != null)
            animator.SetTrigger("Abrir");
        StartCoroutine(DesactivarTrasRetraso(1.5f)); //tiempo duración
    }

    private System.Collections.IEnumerator DesactivarTrasRetraso(float segundos)
    {
        yield return new WaitForSeconds(segundos);

        if (brazoScript != null)
            brazoScript.DesactivarConstraint();
    }
}
```

BrazoSigueTapa:

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.Animations;

// Gestiona el Parent Constraint del brazo del gramófono para que pueda activarse o
desactivarse dinámicamente
public class BrazoSigueTapa : MonoBehaviour
{
```

```
    // Referencia al componente ParentConstraint (usado para que el brazo siga a la
tapa)
    public ParentConstraint constraint;

    // Método para activar el constraint (hace que el brazo siga el movimiento del
objeto al que está vinculado)
    public void ActivarConstraint()
    {
        if (constraint != null)
            constraint.constraintActive = true;
    }

    // Método para desactivar el constraint (el brazo deja de seguir el movimiento)
    public void DesactivarConstraint()
    {
        if (constraint != null)
            constraint.constraintActive = false;
    }
}
```

ManivelaSlot:

```
using UnityEngine;
using UnityEngine.XR.Interaction.Toolkit;

// Este script gestiona la inserción y extracción de la manivela del gramófono.
// Permite encajarla en una posición determinada y controla su comportamiento en
función del estado del sistema.

public class ManivelaSlot : MonoBehaviour
{
    public Transform posicionInsertada;             // Posición fija donde debe
encajar la manivela
    public EstadoGramofonoController fsm;           // Referencia a la FSM del
gramófono

    private XRGrabInteractable manivelaGrab;        // Componente interactivo XR de
la manivela
    private Rigidbody manivelaRb;                   // Rigidbody de la manivela para
controlar físicas

    private GameObject manivela;                    // Referencia al objeto de la
manivela
    private bool manivelaInsertada = false;         // Flag que indica si la manivela
ya está encajada
    private bool listaParaQuitar = false;           // Flag que permite su extracción
solo en el estado adecuado
```

```
// Detecta cuándo la manivela entra en la ranura
private void OnTriggerEnter(Collider other)
{
    // Si ya hay una manivela insertada, ignorar
    if (manivelaInsertada) return;

    // Solo reaccionar si el objeto tiene el tag "Manivela"
    if (!other.CompareTag("Manivela")) return;

    // Solo se permite insertar en los estados apropiados
    if (fsm.estadoActual != EstadoGramofono.Open_Empty &&
        fsm.estadoActual != EstadoGramofono.Manivela_Ready)
    {
        Debug.LogWarning("No puedes insertar la manivela en este estado.");
        return;
    }

    // Guardamos referencias a componentes necesarios
    manivela = other.gameObject;
    manivelaGrab = manivela.GetComponent<XRGrabInteractable>();
    manivelaRb = manivela.GetComponent<Rigidbody>();

    // Si el jugador la tiene agarrada, se procede a insertarla automáticamente
    if (manivelaGrab != null && manivelaGrab.isSelected)
    {
        InsertarManivela();
    }
}

// Inserta la manivela en la posición correcta y cambia su comportamiento físico
private void InsertarManivela()
{
    // Libera la manivela del interactor que la tenía agarrada
    var interactor = manivelaGrab.GetOldestInteractorSelecting();
    if (interactor != null)
    {
        manivelaGrab.interactionManager.SelectExit(interactor, manivelaGrab);
    }

    // Coloca la manivela en la posición y orientación definidas
    manivela.transform.position = posicionInsertada.position;
    manivela.transform.rotation = posicionInsertada.rotation;

    // Desactiva físicas para mantenerla fija
    if (manivelaRb != null)
    {
        manivelaRb.isKinematic = true;
        manivelaRb.useGravity = false;
```

```
        }

        // Restringe movimiento: solo puede rotar en su sitio
        manivelaGrab.trackPosition = false;
        manivelaGrab.trackRotation = true;
        manivelaGrab.movementType = XRGrabInteractable.MovementType.Kinematic;

        manivelaInsertada = true;
        listaParaQuitar = false;

        // Informa a la máquina de estados del cambio
        if (fsm != null)
        {
            fsm.InsertarManivela();
        }
    }

    private void Update()
    {
        // Solo se puede retirar la manivela cuando se ha quitado el fieltro (estado
Felt_Removed)
        if (manivelaInsertada && fsm.estadoActual == EstadoGramofono.Felt_Removed)
        {
            listaParaQuitar = true;
        }

        // Si está lista y el usuario la agarra
        if (listaParaQuitar && manivelaGrab != null && manivelaGrab.isSelected)
        {
            // Reactiva físicas para permitir retirarla con naturalidad
            if (manivelaRb != null)
            {
                manivelaRb.isKinematic = false;
                manivelaRb.useGravity = true;
            }

            // Permite de nuevo que se mueva libremente
            manivelaGrab.trackPosition = true;
            manivelaGrab.movementType =
XRGrabInteractable.MovementType.VelocityTracking;

            // Informa a la FSM del cambio
            fsm.QuitarManivela();

            // Actualiza flags internos
            manivelaInsertada = false;
            listaParaQuitar = false;
```

```
            Debug.Log("Manivela retirada");
        }
    }
}
```

FieltroSlot:

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.XR.Interaction.Toolkit;

// Este script gestiona la colocación y retirada del fieltro en el gramófono.
// Solo se puede colocar el fieltro cuando el estado de la FSM es 'Manivela_Ready'.
//Solo se puede retirar cuando el estado es 'Disk_Removed'.
public class FieltroSlot : MonoBehaviour
{
    public Transform posicionInsertada; // Transform que indica la posición final
donde debe colocarse el fieltro
    public EstadoGramofonoController fsm; // Referencia al controlador de la máquina
de estados del gramófono

    private GameObject fieltroColocado; // Referencia al fieltro actualmente en el
slot
    private XRGrabInteractable fieltroGrab; // Componente XRGrabInteractable del
fieltro (para habilitar/deshabilitar interacción)
    private Rigidbody fieltroRb; // Rigidbody del fieltro (para controlar la física)

    /// <summary>
    /// Detecta cuando un objeto entra en el área del slot.
    /// Si el objeto es un fieltro válido y el estado lo permite, lo posiciona
correctamente y lo bloquea.
    /// </summary>
    private void OnTriggerEnter(Collider other)
    {
        // Solo aceptar un fieltro si aún no hay uno colocado
        if (fieltroColocado != null) return;

        // Solo permitir colocar el fieltro si el gramófono está listo tras colocar
la manivela
        if (fsm.estadoActual != EstadoGramofono.Manivela_Ready) return;

        // Solo reaccionar ante objetos con la etiqueta "Fieltro"
        if (!other.CompareTag("Fieltro")) return;

        // Obtener referencias al componente interactable y al rigidbody del objeto
        fieltroGrab = other.GetComponent<XRGrabInteractable>();
        fieltroRb = other.GetComponent<Rigidbody>();
```

```
        // Si el jugador está sujetando el fieltro, forzar que lo suelte
        if (fieltroGrab != null && fieltroGrab.isSelected)
        {
            var interactor = fieltroGrab.interactorsSelecting.Count > 0 ?
fieltroGrab.interactorsSelecting[0] : null;
            if (interactor != null)
            {
                fieltroGrab.interactionManager.SelectExit(interactor, fieltroGrab);
            }
        }

        // Posicionar el fieltro exactamente en su lugar final
        other.transform.position = posicionInsertada.position;
        other.transform.rotation = posicionInsertada.rotation;

        // Desactivar física e interacción mientras el fieltro esté colocado
        if (fieltroRb != null) fieltroRb.isKinematic = true;
        if (fieltroGrab != null) fieltroGrab.enabled = false;

        // Guardar referencia al fieltro colocado y actualizar estado FSM
        fieltroColocado = other.gameObject;
        fsm.ColocarFieltro();

        Debug.Log("Fieltro colocado correctamente.");
    }

    /// <summary>
    /// Durante la fase Disk_Removed, se permite volver a interactuar con el fieltro
para retirarlo.
    /// Si el jugador lo toma, se actualiza el estado y se limpia la referencia.
    /// </summary>
    private void Update()
    {
        // Si hay un fieltro colocado y estamos en el estado correcto para retirarlo
        if (fieltroColocado != null && fsm.estadoActual ==
EstadoGramofono.Disk_Removed)
        {
            // Rehabilitar interacción si no está activa
            if (fieltroGrab != null && !fieltroGrab.enabled)
            {
                fieltroGrab.enabled = true;
                if (fieltroRb != null) fieltroRb.isKinematic = false;
                Debug.Log("Fieltro habilitado para ser retirado.");
            }

            // Si el jugador ha recogido el fieltro, limpiar y actualizar FSM
            if (fieltroGrab != null && fieltroGrab.isSelected)
```

```
            {
                fsm.QuitarFieltro(); // Cambia el estado a Fieltro_Removed (o el que
corresponda)

                GetComponent<Collider>().enabled = false; // Desactiva el trigger del
slot para evitar nuevas detecciones

                Debug.Log("Fieltro retirado.");

                // Limpiar referencias
                fieltroColocado = null;
                fieltroGrab = null;
                fieltroRb = null;
            }
        }
    }
}
```

DiscoSlot:

```
using System.Collections;
using UnityEngine;
using UnityEngine.XR.Interaction.Toolkit;

/// <summary>
/// Este script gestiona la colocación y retirada del disco en el gramófono.
/// Solo permite colocar un disco si el slot está vacío.
/// Solo se puede retirar el disco cuando el gramófono ha sido detenido (estado:
'Stopped').
/// </summary>
public class DiscoSlot : MonoBehaviour
{
    public Transform posicionInsertada; // Posición donde se encaja el disco al
colocarlo
    public EstadoGramofonoController fsm; // Referencia a la máquina de estados del
gramófono

    private GameObject discoColocado; // Referencia al disco actualmente en el slot
    private XRGrabInteractable discoGrab; // Componente XRGrabInteractable del disco
(para habilitar/deshabilitar interacción)
    private Rigidbody discoRb; // Rigidbody del disco (para controlar la física)

    /// <summary>
    /// Update se encarga de verificar si el disco puede retirarse,
    /// lo que solo ocurre cuando el gramófono ha sido detenido (estado 'Stopped').
    /// </summary>
    private void Update()
    {
```

```
        // Solo actuar si hay un disco colocado y el gramófono ha sido detenido
        if (discoColocado != null && fsm.estadoActual == EstadoGramofono.Stopped)
        {
            // Habilita la interacción si aún está desactivada
            if (discoGrab != null && !discoGrab.enabled)
            {
                discoGrab.enabled = true;
                if (discoRb != null) discoRb.isKinematic = false;
                Debug.Log("Disco habilitado para ser retirado");
            }

            // Si el jugador ha recogido el disco, actualizar FSM y limpiar
referencias
            if (discoGrab != null && discoGrab.isSelected)
            {
                fsm.QuitarDisco(); // Cambia el estado de la FSM a 'Disk_Removed'

                //Desactivar el collider del slot para evitar recolocación automática
del disco
                GetComponent<Collider>().enabled = false;

                // Limpiar referencias del disco
                discoColocado = null;
                discoGrab = null;
                discoRb = null;
            }
        }
    }

    /// <summary>
    /// Detecta cuándo un disco entra en el área del slot.
    /// Si es un objeto válido y no hay otro colocado, lo bloquea en su lugar.
    /// </summary>
    private void OnTriggerEnter(Collider other)
    {
        // Si ya hay un disco, ignorar nuevas entradas
        if (discoColocado != null) return;

        // Solo aceptar objetos etiquetados como "Disco"
        if (!other.CompareTag("Disco")) return;

        // Obtener componentes del objeto entrante
        discoGrab = other.GetComponent<XRGrabInteractable>();
        discoRb = other.GetComponent<Rigidbody>();

        // Si el disco está siendo sostenido, forzar que el jugador lo suelte
        if (discoGrab != null && discoGrab.isSelected)
        {
```

```
            var interactor = discoGrab.interactorsSelecting.Count > 0 ?
discoGrab.interactorsSelecting[0] : null;
            if (interactor != null)
            {
                discoGrab.interactionManager.SelectExit(interactor, discoGrab);
            }
        }

        // Posicionar y rotar el disco exactamente en el slot
        other.transform.position = posicionInsertada.position;
        other.transform.rotation = posicionInsertada.rotation;

        // Bloquear el disco en su sitio (sin física ni interacción)
        if (discoRb != null) discoRb.isKinematic = true;
        if (discoGrab != null) discoGrab.enabled = false;

        // Guardar referencia al disco y actualizar la FSM
        discoColocado = other.gameObject;
        fsm.ColocarDisco(discoColocado); // Cambia estado a 'Playing_Ready'

        Debug.Log("Disco colocado correctamente en el slot");
    }
}
```

DiscoSeleccionado:

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.XR.Interaction.Toolkit;

// Este script se encarga de mostrar un mensaje personalizado en pantalla
// cuando el jugador recoge un disco. Utiliza el sistema de interacción XR de Unity.
[RequireComponent(typeof(XRGrabInteractable))] // Garantiza que este GameObject tenga
un XRGrabInteractable
public class DiscoSeleccionado : MonoBehaviour
{
    private XRGrabInteractable grabInteractable; // Referencia al componente de
interacción del disco
    private EstadoGramofonoController fsm; // Referencia a la máquina de estados del
gramófono

    void Awake()
    {
        // Cargar el componente de interacción del propio objeto
        grabInteractable = GetComponent<XRGrabInteractable>();
    }
```

```
    void Start()
    {
        // Buscar automáticamente el controlador del gramófono en la escena (solo
uno)
        fsm = FindObjectOfType<EstadoGramofonoController>();
    }

    /// <summary>
    /// Suscribirse al evento de "grab" cuando el objeto se activa
    /// </summary>
    private void OnEnable()
    {
        grabInteractable.selectEntered.AddListener(OnGrabbed);
    }

    /// <summary>
    /// Cancelar la suscripción al evento cuando el objeto se desactiva
    /// (buenas prácticas para evitar fugas de memoria o errores)
    /// </summary>
    private void OnDisable()
    {
        grabInteractable.selectEntered.RemoveListener(OnGrabbed);
    }

    /// <summary>
    /// Método llamado automáticamente cuando el disco es cogido por un controlador
    /// </summary>
    private void OnGrabbed(SelectEnterEventArgs args)
    {
        // Asegurarse de que la FSM y el UI estén correctamente inicializados
        if (fsm != null && fsm.dialogueUI != null)
        {
            // Obtener los datos del disco (nombre, etc.)
            var disco = GetComponent<Disco>();
            if (disco != null)
            {
                // Mostrar mensaje personalizado con el nombre del disco
                fsm.dialogueUI.MostrarDialogoPersonalizado($"Estás cogiendo el disco:
{disco.nombreDisco}");
            }
        }
    }
}
```

DarCuerda:

```
using System.Collections;
```

```
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.XR.Interaction.Toolkit;

// Script que detecta el giro de la manivela en una experiencia VR para dar cuerda al
gramófono.
// Al girar una cierta cantidad de grados, cambia el estado del gramófono y activa la
animación correspondiente.
public class DarCuerda : MonoBehaviour
{
    // Referencia al controlador de estados del gramófono
    public EstadoGramofonoController fsm;

    // Grados necesarios para considerar que se ha dado la cuerda completamente
    public float gradosParaCuerda = 360f;

    // Referencia al Animator que controla la animación de la manivela
    public Animator animator;

    // Componente XR para detectar cuándo se agarra el objeto
    private XRGrabInteractable grab;

    // Acumulador de grados girados
    private float gradosAcumulados = 0f;

    // Marca si ya se ha dado la cuerda para evitar repetir
    private bool cuerdaDada = false;

    // Interactor que está girando la manivela actualmente
    private IXRSelectInteractor interactorActual;

    // Rotación inicial al comenzar a girar
    private Quaternion rotacionInicial;

    // Se ejecuta al iniciar el script (cuando se instancia el objeto)
    private void Awake()
    {
        // Obtenemos el componente de agarre
        grab = GetComponent<XRGrabInteractable>();

        // Registramos los eventos para detectar cuándo se agarra o suelta
        grab.selectEntered.AddListener(OnGrab);
        grab.selectExited.AddListener(OnSuelta);
    }

    // Limpia los eventos registrados al destruir el objeto
    private void OnDestroy()
    {
```

```
        grab.selectEntered.RemoveListener(OnGrab);
        grab.selectExited.RemoveListener(OnSuelta);
    }

    // Se llama cuando el usuario agarra la manivela
    private void OnGrab(SelectEnterEventArgs args)
    {
        interactorActual = args.interactorObject;
        rotacionInicial = interactorActual.transform.rotation;
        gradosAcumulados = 0f;
        cuerdaDada = false;

        // Reinicia la animación de la manivela y la pausa (se controla manualmente)
        animator.Play("ManivelaGira", 0, 0f);
        animator.speed = 0f;
    }

    // Se llama al soltar la manivela
    private void OnSuelta(SelectExitEventArgs args)
    {
        interactorActual = null;
        animator.speed = 0f; // Pausa la animación
    }

    // Se ejecuta cada frame
    private void Update()
    {
        // Verificamos si el gramófono está listo, si ya se dio cuerda o si no hay
interacción
        if (fsm.estadoActual != EstadoGramofono.Disk_Ready || cuerdaDada ||
interactorActual == null) return;

        // Obtenemos la rotación actual del controlador
        Quaternion rotacionActual = interactorActual.transform.rotation;

        // Calculamos la diferencia de rotación desde la última posición
        Quaternion delta = rotacionActual * Quaternion.Inverse(rotacionInicial);
        float deltaX = delta.eulerAngles.x;

        // Corregimos el valor para obtener un rango entre -180 y 180 grados
        if (deltaX > 180f) deltaX -= 360f;

        // Sumamos el valor absoluto del giro acumulado
        gradosAcumulados += Mathf.Abs(deltaX);
        rotacionInicial = rotacionActual;

        // Avanza la animación en proporción al progreso de la cuerda
        float progreso = Mathf.Clamp01(gradosAcumulados / gradosParaCuerda);
```

```
        animator.Play("ManivelaGira", 0, progreso);
        animator.speed = 0f; // Pausamos para fijar el frame exacto según progreso

        // Si se ha girado lo suficiente, se da la cuerda
        if (gradosAcumulados >= gradosParaCuerda)
        {
            cuerdaDada = true;
            fsm.DarCuerda(); // Llamamos al cambio de estado en el controlador
            Debug.Log("¡Cuerda dada!");
        }
    }
}
```

Frenar:

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.XR.Interaction.Toolkit;

// Script que gestiona el comportamiento del freno del gramófono en una experiencia
VR.
// Según el estado actual del gramófono, al hacer clic en el freno se activa o
desactiva el giro del plato.
// También se actualiza la animación del freno y se cambia el estado de la FSM.
public class Frenar : MonoBehaviour
{
    // Referencia al controlador de estados del gramófono
    public EstadoGramofonoController fsm;

    // Animator del freno, controla su animación
    public Animator animatorFreno;

    // Lista de scripts que hacen girar el disco y el fieltro
    public List<GirarDisco> giradores = new List<GirarDisco>();

    // Interactable del freno para detectar interacciones VR
    private XRBaseInteractable interactable;

    private void Awake()
    {
        // Obtenemos el componente de interacción y registramos el evento de click
(grab/select)
        interactable = GetComponent<XRBaseInteractable>();
        interactable.selectEntered.AddListener(OnClickFreno);
    }

    private void OnDestroy()
```

```
    {
        // Quitamos el listener al destruir el objeto
        interactable.selectEntered.RemoveListener(OnClickFreno);
    }

    // Método que se ejecuta al hacer clic o agarrar el freno
    private void OnClickFreno(SelectEnterEventArgs args)
    {
        if (fsm.estadoActual == EstadoGramofono.Wound)
        {
            // Estamos en proceso de montaje → quitar el freno para que comience a
girar

            animatorFreno.SetTrigger("ActivarFreno"); // Animación de quitar el freno

            // Activamos el giro en todos los objetos (disco y fieltro)
            foreach (var g in giradores)
            {
                g.ActivarGiro(true);
            }

            fsm.QuitarFreno(); // Cambia el estado a Playing_Ready
            Debug.Log("Freno quitado: Fieltro y vinilo girando...");
        }
        else if (fsm.estadoActual == EstadoGramofono.Playing_Ready)
        {
            // Estamos desmontando → volver a frenar

            // No se activa animación, solo se fuerza el estado Idle del freno
            animatorFreno.Play("IdleFreno"); // Asegura que la animación vuelva al
inicio

            // Detenemos el giro
            foreach (var g in giradores)
            {
                g.ActivarGiro(false);
            }

            fsm.PonerFreno(); // Cambia el estado a Stopped
            Debug.Log("Freno puesto: plato detenido.");
        }
        else
        {
            // En otros estados, no está permitido usar el freno
            Debug.LogWarning("No puedes usar el freno en este estado: " +
fsm.estadoActual);
        }
    }
```

```
}
```

MoverBrazo:

```
using System.Collections;
using UnityEngine;
using UnityEngine.Animations;
using UnityEngine.XR.Interaction.Toolkit;

// Script que mueve el brazo del gramófono de forma suave hacia la posición de
reproducción cuando el usuario lo selecciona.
// Solo permite el movimiento si el gramófono está en el estado Playing_Ready y evita
que se repita.
// También desactiva cualquier ParentConstraint que esté manteniendo el brazo en una
posición inicial.

public class MoverBrazo : MonoBehaviour
{
    // Transform del objetivo donde queremos que llegue el brazo (posición final)
    public Transform brazoObjetivo;

    // Duración de la animación del movimiento
    public float duracion = 1f;

    // Referencia al controlador de estados del gramófono
    public EstadoGramofonoController fsm;

    // Constraint que puede mantener el brazo "pegado" a una posición al inicio
    private ParentConstraint constraint;

    // Para asegurarnos de que el brazo solo se mueve una vez
    private bool yaMovido = false;

    private void Awake()
    {
        // Obtenemos el constraint si existe
        constraint = GetComponent<ParentConstraint>();
        if (constraint == null)
        {
            Debug.LogWarning("No hay ParentConstraint en el brazo!");
        }
    }

    // Método público que debe conectarse al evento onSelectEntered del XR Simple
Interactable en el brazo
    public void OnClickBrazo()
    {
```

```
        // Solo permitimos mover si el gramófono está en el estado adecuado y aún no
lo hemos movido
        if (fsm.estadoActual == EstadoGramofono.Playing_Ready && !yaMovido)
        {
            Debug.Log("Click detectado en brazo");

            // Si el constraint existe, lo desactivamos para permitir el movimiento
libre
            if (constraint != null)
            {
                constraint.constraintActive = false;
                constraint.enabled = false;
                Debug.Log("ParentConstraint desactivado");
            }
            else
            {
                Debug.LogWarning("Constraint no encontrado o ya desactivado");
            }

            // Iniciamos la corrutina que mueve el brazo
            StartCoroutine(Mover());

            // Marcamos que ya fue movido para evitar que se repita
            yaMovido = true;
        }
    }

    // Corrutina que interpola suavemente la posición y rotación del brazo hasta su
objetivo
    private IEnumerator Mover()
    {
        // Guardamos la posición y rotación iniciales
        Vector3 startPos = transform.position;
        Quaternion startRot = transform.rotation;

        // Posición y rotación destino (donde debe quedar el brazo)
        Vector3 targetPos = brazoObjetivo.position;
        Quaternion targetRot = brazoObjetivo.rotation;

        float t = 0f;
        while (t < 1f)
        {
            t += Time.deltaTime / duracion;

            // Interpolamos posición y rotación progresivamente
            transform.position = Vector3.Lerp(startPos, targetPos, t);
            transform.rotation = Quaternion.Lerp(startRot, targetRot, t);
```

```
            yield return null;
        }

        Debug.Log("Brazo movido a posición objetivo");
    }
}
```

EncajarAguja:

```
using System.Collections;
using UnityEngine;
using UnityEngine.XR.Interaction.Toolkit;

// Este script gestiona el comportamiento de la aguja del gramófono:
// - Cuando se acerca al disco, se coloca automáticamente en la posición final,
//   inicia la reproducción y bloquea la interacción temporalmente.
// - Si el usuario la vuelve a agarrar mientras suena el disco, se detiene la música
//   y el brazo vuelve suavemente a su posición de reposo.
public class EncajarAguja : MonoBehaviour
{
    // Transform que indica dónde debe posicionarse la aguja al encajarla sobre el
disco
    public Transform posicionFinalAguja;

    // Transform de la posición de reposo (cuando la aguja está levantada fuera del
disco)
    public Transform posicionReposoBrazo;

    // Referencia al controlador de estados del gramófono
    public EstadoGramofonoController fsm;

    // Reproductor de sonido del disco (aunque no se usa directamente aquí, es útil
tenerlo referenciado)
    public AudioSource sonidoDisco;

    // Transform del brazo que contiene la aguja (usado para animar el retorno a
reposo)
    public Transform extensionBrazo;

    // Componente que permite agarrar la aguja
    private XRGrabInteractable grabInteractable;

    // Flag que indica si la aguja está reproduciendo (puesta sobre el disco)
    private bool reproduciendo = false;

    private void Awake()
    {
```

```
        // Obtenemos el componente XRGrabInteractable y registramos el evento al
agarrar la aguja
        grabInteractable = GetComponent<XRGrabInteractable>();
        grabInteractable.selectEntered.AddListener(OnAgujaAgarrada);
    }

    private void OnDestroy()
    {
        // Nos aseguramos de eliminar el listener para evitar errores si el objeto se
destruye
        grabInteractable.selectEntered.RemoveListener(OnAgujaAgarrada);
    }

    // Cuando la aguja entra en el trigger del disco
    private void OnTriggerEnter(Collider other)
    {
        // Verificamos que haya entrado en el trigger correcto, que el estado sea
adecuado y que aún no esté reproduciendo
        if (other.CompareTag("TriggerAguja") && fsm.estadoActual ==
EstadoGramofono.Playing_Ready && !reproduciendo)
        {
            // Posicionamos la aguja exactamente sobre el disco
            transform.position = posicionFinalAguja.position;
            transform.rotation = posicionFinalAguja.rotation;

            // Cambiamos el estado para bajar la aguja y comenzar la reproducción
            fsm.BajarAguja();
            fsm.ReproducirDisco();

            // Marcamos que estamos reproduciendo
            reproduciendo = true;

            // Desactivamos la interacción temporalmente para evitar movimiento
accidental
            if (grabInteractable != null)
            {
                // Cancelamos la interacción si está agarrada
                grabInteractable.interactionManager?.CancelInteractableSelection((IXR
SelectInteractable)grabInteractable);

                grabInteractable.enabled = false;

                // Rehabilitamos la interacción tras un pequeño delay
                Invoke(nameof(ActivarGrab), 0.1f);
            }

            Debug.Log("Aguja colocada sobre el disco. Reproducción iniciada.");
        }
```

```
    }

    // Método que vuelve a permitir agarrar la aguja (después de bloquearla
momentáneamente)
    private void ActivarGrab()
    {
        if (grabInteractable != null)
        {
            grabInteractable.enabled = true;
        }
    }

    // Este método se llama automáticamente cuando el usuario agarra la aguja
    private void OnAgujaAgarrada(SelectEnterEventArgs args)
    {
        // Solo respondemos si está reproduciendo y el estado es Playing_Sound
        if (reproduciendo && fsm.estadoActual == EstadoGramofono.Playing_Sound)
        {
            Debug.Log("Aguja agarrada. Deteniendo reproducción y levantando aguja.");

            // Cambiamos los estados correspondientes
            fsm.DetenerDisco();
            fsm.SubirAguja();

            // Iniciamos la animación para devolver el brazo a su posición original
            StartCoroutine(VolverABrazoReposo());

            // Ya no está reproduciendo
            reproduciendo = false;
        }
    }

    // Corrutina que mueve suavemente el brazo desde la posición actual hacia la
posición de reposo
    private IEnumerator VolverABrazoReposo()
    {
        Vector3 startPos = extensionBrazo.position;
        Quaternion startRot = extensionBrazo.rotation;

        Vector3 targetPos = posicionReposoBrazo.position;
        Quaternion targetRot = posicionReposoBrazo.rotation;

        float duracion = 1f;
        float t = 0f;

        while (t < 1f)
        {
            t += Time.deltaTime / duracion;
```

```
            // Interpolamos posición y rotación suavemente
            extensionBrazo.position = Vector3.Lerp(startPos, targetPos, t);
            extensionBrazo.rotation = Quaternion.Slerp(startRot, targetRot, t);

            yield return null;
        }

        // Nos aseguramos de llegar exactamente al destino
        extensionBrazo.position = targetPos;
        extensionBrazo.rotation = targetRot;

        Debug.Log("Extensión del brazo volvió a la posición de reposo.");
    }
}
```

GirarDisco:

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

// Este script hace girar un objeto (como un disco de gramófono) cuando se activa.
// El giro se realiza en el eje Y con una velocidad configurable.
// Puede activarse o detenerse mediante el método `ActivarGiro`.
public class GirarDisco : MonoBehaviour
{
    // Velocidad de rotación en grados por segundo
    public float velocidad = 100f;

    // Indica si el disco está girando actualmente
    public bool girando = false;

    // Se ejecuta cada frame
    void Update()
    {
        // Si el flag está activado, rota el objeto sobre el eje Y (Vector3.up)
        if (girando)
        {
            transform.Rotate(Vector3.up, velocidad * Time.deltaTime);
        }
    }

    // Método público que activa o desactiva el giro
    public void ActivarGiro(bool estado)
    {
        girando = estado;
```

```
    }
}
```

Disco:

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

// Este script define un "Disco" con un nombre identificativo y un clip de audio.
// Se utiliza para representar los diferentes vinilos en la experiencia del gramófono
VR.
public class Disco : MonoBehaviour
{
    // Nombre del disco, puede mostrarse como texto en la experiencia
    public string nombreDisco;

    // Clip de sonido que se reproducirá cuando el disco esté en el gramófono
    public AudioClip clipSonido;
}
```

CerrarGramofono:

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.XR.Interaction.Toolkit;

// Este script se encarga de cerrar la tapa del gramófono cuando el jugador la
selecciona en VR.
// Durante el cierre, activa una animación, hace que el brazo siga el movimiento de
la tapa,
// y cambia el estado de la máquina de estados del gramófono.

public class CerrarGramofono : MonoBehaviour
{
    public Animator animator;                       // Referencia al Animator que
ejecuta la animación de la tapa
    public EstadoGramofonoController fsm;           // Controlador del estado actual
del gramófono
    public BrazoSigueTapa brazoScript;              // Script que permite al brazo
seguir la tapa al cerrarse

    private XRBaseInteractable interactable;        // Componente que permite
detectar selección en VR
```

```
    private bool yaCerrada = false;                 // Bandera para evitar múltiples
cierres

    void Awake()
    {
        // Obtenemos el componente de interacción del objeto
        interactable = GetComponent<XRBaseInteractable>();

        // Registramos el método que se ejecuta al seleccionar el objeto
        interactable.selectEntered.AddListener(OnSelect);
    }

    // Se ejecuta cuando el usuario selecciona la tapa en VR
    private void OnSelect(SelectEnterEventArgs args)
    {
        // Solo permite cerrar si aún no se ha cerrado y el estado del gramófono es
el adecuado
        if (!yaCerrada && fsm != null && fsm.estadoActual ==
EstadoGramofono.Manivela_Removed)
        {
            // Activa el constraint que hace que el brazo siga la tapa
            if (brazoScript != null)
                brazoScript.ActivarConstraint();

            // Lanza la animación de cierre de la tapa
            if (animator != null)
                animator.SetTrigger("Cerrar");

            // Marca que la tapa ya ha sido cerrada
            yaCerrada = true;

            // Cambia el estado de la FSM a Closed
            fsm.CerrarGramofono();

            // Desactiva el constraint después del tiempo que dura la animación
            StartCoroutine(DesactivarConstraintTrasRetraso(1.5f));
        }
    }

    // Espera un tiempo y luego desactiva el constraint del brazo
    private IEnumerator DesactivarConstraintTrasRetraso(float segundos)
    {
        // Espera la cantidad de segundos especificada
        yield return new WaitForSeconds(segundos);

        // Desactiva el constraint si el script del brazo está presente
        if (brazoScript != null)
            brazoScript.DesactivarConstraint();
```

```
    }

    void OnDestroy()
    {
        // Elimina el listener cuando se destruye el objeto para evitar fugas de
memoria
        if (interactable != null)
            interactable.selectEntered.RemoveListener(OnSelect);
    }
}
```

DialogueUI:

```
using System.Collections;
using UnityEngine;
using TMPro;

// Este script gestiona los mensajes de texto que aparecen en la interfaz VR
// dependiendo del estado del gramófono. Muestra diálogos con efecto de máquina de
escribir
// y permite personalizarlos según el contexto.

public class DialogueUI : MonoBehaviour
{
    public TextMeshProUGUI textComponent;         // Componente de texto en pantalla
    public GameObject dialogueBox;                // Caja de diálogo que se
activa/desactiva
    public float textSpeed = 0.05f;               // Velocidad con la que aparece
cada letra
    public float tiempoVisible = 5f;              // Tiempo que permanece visible
cada línea

    private int index = 0;                        // Índice de la línea actual en el
diálogo
    private string[] lines;                       // Array de líneas a mostrar

    void Start()
    {
        textComponent.text = string.Empty;        // Limpia el texto al inicio
        dialogueBox.SetActive(false);             // Oculta la caja de diálogo
inicialmente
    }

    // Muestra un diálogo básico según el estado del gramófono
    public void MostrarDialogo(string estado)
    {
```

```
        lines = ObtenerTextoPorEstado(estado);    // Obtiene las líneas para ese
estado
        if (lines.Length == 0) return;            // Si no hay texto, no hace nada

        StopAllCoroutines();                      // Detiene cualquier diálogo
anterior
        dialogueBox.SetActive(true);              // Muestra la caja de diálogo
        index = 0;                                // Reinicia el índice
        StartCoroutine(MostrarLineas());          // Comienza a mostrar el diálogo
    }

    // Devuelve el texto correspondiente a cada estado del gramófono
    private string[] ObtenerTextoPorEstado(string estado)
    {
        switch (estado)
        {
            case "Closed":
                return new string[] {"¡Felicidades, ya sabes como operar el
gramófono!"};
            case "Open_Empty":
                return new string[] {"¡Felicidades! Has abierto el Gramófono. Ahora
inserta la manivela en el lateral derecho de la base"};
            case "Manivela_Ready":
                return new string[] {"Has colocado la manivela en posición. Ahora
debes colocar un fieltro sobre el plato"};
            case "Felt_Ready":
                return new string[] {"¡Genial! Escoge un disco y colocalo sobre el
fieltro. ¡Tienes varios discos disponibles en la estantería de la izquierda!"};
            case "Disk_Ready":
                return new string[] {"¡Bien hecho! Ahora gira la manivela hasta que
tengamos suficiente cuerda"};
            case "Wound":
                return new string[] {"Ahora que la cuerda está dada debes quitar el
freno" };
            case "Playing_Ready":
                return new string[] {"El plato está girando, para reproducir el audio
debes colocar el brazo fonocaptor en posición y la aguja sobre el disco"};
            case "Playing_Sound":
                return new string[] {"¡Reproduciendo música!"};
            case "Stopped":
                return new string[] {"El gramófono se ha detenido"};
            default:
                return new string[0]; // Estado desconocido
        }
    }

    // Muestra cada línea del array con efecto de escritura
    private IEnumerator MostrarLineas()
```

```
    {
        while (index < lines.Length)
        {
            textComponent.text = "";                        // Limpia el texto actual
            yield return StartCoroutine(EscribirLinea(lines[index])); // Escribe la
línea
            yield return new WaitForSeconds(tiempoVisible);           // Espera antes
de continuar
            index++;                                         // Pasa a la siguiente
línea
        }

        dialogueBox.SetActive(false);                       // Oculta la caja de diálogo
al finalizar
    }

    // Escribe una línea letra por letra con retardo
    private IEnumerator EscribirLinea(string linea)
    {
        foreach (char c in linea.ToCharArray())
        {
            textComponent.text += c;                        // Añade una letra
            yield return new WaitForSeconds(textSpeed);  // Espera antes de añadir la
siguiente
        }
    }

    // Muestra un mensaje personalizado (fuera del flujo de estados)
    public void MostrarDialogoPersonalizado(string mensaje)
    {
        StopAllCoroutines();                                // Detiene cualquier diálogo
anterior
        dialogueBox.SetActive(true);                        // Muestra la caja
        textComponent.text = "";                            // Limpia el contenido
        StartCoroutine(EscribirLinea(mensaje));             // Escribe el mensaje
        StartCoroutine(CerrarDialogoPasadoTiempo());        // Oculta la caja tras unos
segundos
    }

    // Espera un tiempo y oculta la caja de diálogo
    private IEnumerator CerrarDialogoPasadoTiempo()
    {
        yield return new WaitForSeconds(tiempoVisible);  // Espera
        dialogueBox.SetActive(false);                       // Oculta la caja
    }

    // Muestra texto teniendo en cuenta el cambio de estado anterior -> nuevo
    public void MostrarDialogoConContexto(string estadoNuevo, string estadoAnterior)
```

```
    {
        lines = ObtenerTextoPorEstadoConContexto(estadoNuevo, estadoAnterior); //
Lógica contextual
        if (lines.Length == 0) return;

        StopAllCoroutines();                            // Detiene otros diálogos
        dialogueBox.SetActive(true);                    // Muestra la caja
        index = 0;                                      // Reinicia índice
        StartCoroutine(MostrarLineas());                // Comienza el nuevo diálogo
    }

    // Devuelve el texto específico cuando hay una transición entre dos estados
concretos
    private string[] ObtenerTextoPorEstadoConContexto(string estadoNuevo, string
estadoAnterior)
    {
        if (estadoNuevo == "Playing_Ready" && estadoAnterior == "Playing_Sound")
        {
            return new string[] {"Para seguir desmontando el gramófono, pon el freno
para detener el plato"};
        }
        if (estadoNuevo == "Stopped" && estadoAnterior == "Playing_Ready")
        {
            return new string[] {"Para dejar de utilizar el gramófono, pon el freno
para detener el plato"};
        }
        if (estadoNuevo == "Disk_Removed" && estadoAnterior == "Stopped")
        {
            return new string[] {"Perfecto, el disco ha sido retirado. Ahora puedes
quitar el fieltro"};
        }
        if (estadoNuevo == "Felt_Removed" && estadoAnterior == "Disk_Removed")
        {
            return new string[] {"El fieltro ha sido retirado. Ahora puedes retirar
la manivela"};
        }
        if (estadoNuevo == "Manivela_Removed" && estadoAnterior == "Felt_Removed")
        {
            return new string[] {"Has retirado la manivela. guárdala y ya puedes
cerrar la caja"};
        }

        // Si no hay contexto específico, devuelve el diálogo básico del nuevo estado
        return ObtenerTextoPorEstado(estadoNuevo);
    }
}
```