



Universidad de las Ciencias Informáticas
Facultad 4

Trabajo de diploma para optar por el título de
Ingeniero en Ciencias Informáticas

Catálogo virtual interactivo para la visualización de
contenidos bibliográficos usando Kinect 2

Autor:

Leudis Rafael Estrada González

Tutores:

Dr.C. Omar Correa Madrigal
Ms.C. Yadira Ramírez Rodríguez

La Habana, Cuba
Noviembre, 2022

Frase Motivacional

"Los hombres me han llamado loco; pero aún no está determinada la cuestión de si la locura es o no la más excelsa inteligencia, si mucho de lo que es gloria, si todo aquello que es profundo, no brota de la enfermedad del pensamiento, de modos de pensar exaltados respecto del intelecto general. Aquellos que sueñan de día son conocedores de muchas cosas que se les escapan a los que únicamente sueñan de noche".

Edgar Allan Poe

"Hay una fuerza motriz más poderosa que el vapor, la electricidad y la energía atómica: la voluntad".

Albert Einstein

Dedicatoria

El fracaso y el dolor forman parte de nuestras vidas y hay que aprender a lidiar con ello todos los días de este mundo. En todo este tiempo, los errores que cometí, me hicieron más sabio; las pérdidas que sufrí, me volvieron más fuerte; las amistades que encontré, me hicieron dichoso; el apoyo y la confianza que gané, impulsaron mi espíritu. Finalmente, la voluntad férrea de seguir adelante me hizo cumplir esta meta. Doy gracias a la vida, a mi familia, a mi hermana y a mi madre, por todo. A ustedes va dedicada, de manera especial; esta tesis.

Agradecimientos

A mi madre

De niño me recuerdo estando siempre a tu lado, inocente, inquieto, feliz. Contigo realicé las primeras lecturas, escribí las primeras palabras, los primeros números. Cultivaste en mí el deseo de aprendizaje, me enseñaste a no rendirme nunca, a valorar lo poco que tenía y dar en todo momento lo mejor de mí. Cuando dormía en tu lecho, me sentía seguro. La vida nos impuso mil batallas, las libramos juntos, y siempre supiste mantenerte firme e inquebrantable. Eso, unido a tu voluntad férrea, tu alegría y tus deseos de vivir, te convirtieron en mi modelo a seguir. Por eso, aunque hoy no estés presente, quiero que sepas que me enorgullecerá siempre el hecho de haber sido tu hijo, por todo el afecto invaluable y las enseñanzas que obtuve de ti. Gracias por haber sido, la mejor de todas las madres. Con amor incondicional, Leudis.

A mi hermana

Cuando nos quedamos solos me sentí perdido, traicionado por la vida, fuiste tú la que señaló el camino. Gracias por estar ahí en todos los momentos, por mantenerte firme, por tu cariño, por esos dos sobrinos que me diste, por apoyarme y confiar en mí.

A mi familia

A ustedes debo todos mis valores. El amor que siempre me han profesado no se puede medir. Agradecido siempre estaré por sus consejos y por convertirme en el hombre que soy. Siempre estaré con ustedes, aunque vaya lejos. Mis pensamientos, los mejores deseos y bendiciones, para todos.

A mi segunda familia (Carlos, David, Dariel, Javier, Daimita)

Que sería de mí sin ustedes. Me conocen desde siempre y solo ante sus presencias he llegado a revelar todo mi ser. Mis amigos incondicionales de toda la vida, juntos todos en tristezas, juntos todos en alegrías. Mucha salud, prosperidad y bendiciones en cada uno de sus días.

A mis compañeros de aula

En todos estos años que hemos convivido juntos, se han vuelto muy importantes para mí. Gracias por la ayuda solidaria, el apoyo, la comprensión y el respeto. Éxitos a todos en sus vidas y carreras como profesionales.

A mis profesores

Un buen profesor es capaz de entregarse en cuerpo y alma a su trabajo. Como un padre que atiende a sus hijos en todo momento, aportan sus experiencias y nos nutren con conocimiento. A todos los educadores que me guiaron por este camino, mil gracias por su tiempo y que la dicha, la salud y todo el amor del mundo los acompañe siempre, en todo momento.

A mis tutores (Omar y Yadira)

Dicen que lo mejor no se guarda para el final, pero ustedes lo son para mí. Cuando los conocí a ambos mi vida enfiló abruptamente. Sin el apoyo y la confianza que en mí han depositado hoy no estaría aquí presente. Valoraré siempre su comprensión y sus destrezas a lo largo de esta etapa que ha significado tanto en mi vida. Por todo el empeño y la disposición que han mostrado siempre que los he necesitado, le estaré simplemente, eternamente agradecido.

Declaración de autoría

Queda declarada la autoría de la presente tesis titulada: “Catálogo virtual interactivo para la visualización de contenidos bibliográficos usando Kinect 2”, y se reconoce a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmo la presente a los 16 días del mes de noviembre del año 2022.

Autor



Leudis Rafael Estrada González

Tutor



Dr.C. Omar Correa Madrigal

Tutor



Ms.C. Yadira Ramírez Rodríguez

Resumen

El reconocimiento de gestos en las ciencias de la computación tiene como principal objetivo, interpretar los movimientos y el lenguaje corporal de los humanos a través de algoritmos matemáticos. Las **Tecnologías de Interfaz Humano-Computadora** que se desarrollan en la Universidad de las Ciencias Informáticas, apuestan por integrar esta solución en un catálogo virtual que permite la visualización e interacción dinámica con contenidos bibliográficos, haciendo uso del sensor de gestos **Kinect 2** y el motor de juegos **Unity-3D**. Dicho catálogo ofrece una experiencia positiva a los usuarios que necesitan explorar el fondo bibliográfico de la universidad, al integrar elementos que permiten enriquecer el constante proceso de enseñanza y aprendizaje, en aras de alcanzar una mejor formación profesional. El proceso de desarrollo, desde el análisis hasta las pruebas, emplea como guía fundamental la metodología ágil **Extreme Programming (XP)**. Se definen las tareas de ingeniería correspondientes a cada **HU** y se aplican las pruebas unitarias y de aceptación, para comprobar el correcto funcionamiento de la aplicación y su aceptación por el cliente.

Palabras Clave: catálogo virtual, reconocimientos de gestos, Kinect 2, contenidos bibliográficos.

Índice general

Introducción	- 1 -
Capítulo 1. Fundamentación Teórica.....	- 5 -
Introducción	- 5 -
1.1 Catálogos Públicos de Acceso en Línea (OPAC).....	- 5 -
1.2 Interfaces Naturales de Usuario (NUI)	- 6 -
1.2.1 Interfaces Basadas en gestos	- 6 -
1.3. Aplicaciones basadas en bibliotecas virtuales.....	- 7 -
1.3.1 Análisis de las aplicaciones basadas en bibliotecas virtuales	- 7 -
1.4 Gestos	- 10 -
1.4.1 Aplicaciones basadas en gestos	- 11 -
1.4.2 Análisis de las aplicaciones basadas en gestos con Kinect 2.....	- 12 -
1.5 Selección del dominio de aplicación a partir del análisis de aplicaciones homólogas	- 14 -
1.6 Método de Ingeniería de Software	- 16 -
1.7 Herramientas y Tecnologías	- 17 -
1.7.1 Kinect for Windows SDK 2.0.....	- 17 -
1.7.2 Kinect 2.....	- 17 -
1.7.3 Herramienta CASE	- 18 -
1.7.4 Motor Gráfico	- 19 -
1.7.5 Visual Studio Code.....	- 20 -
1.7.6 Lenguaje de Programación C#.....	- 21 -
Conclusiones parciales del capítulo.....	- 22 -
Capítulo 2. Características y diseño de la solución.....	- 23 -
Introducción	- 23 -
2.1 Propuesta de solución	- 23 -
2.2 Fase I: Planificación.....	- 23 -

2.2.1 Historias de usuario (HU)	- 24 -
2.2.2 Estimación del esfuerzo por Historia de Usuario	- 29 -
2.2.3 Requisitos no funcionales.....	- 29 -
2.2.4 Desarrollo del plan de iteraciones	- 31 -
2.2.5 Plan de duración de las iteraciones.....	- 31 -
2.2.6 Plan de entregas	- 32 -
2.3 Fase II: Diseño del Sistema	- 32 -
2.4 Patrones de diseño	- 34 -
2.5 Tarjetas CRC.....	- 36 -
Conclusiones parciales del capítulo.....	- 43 -
Capítulo 3. Implementación y pruebas del sistema.....	- 44 -
Introducción	- 44 -
3.1 Fase III: Desarrollo.....	- 44 -
3.1.1 Tareas de Ingeniería	- 44 -
3.2 Fase IV: Pruebas	- 48 -
3.2.1 Pruebas unitarias	- 48 -
3.2.2 Pruebas de aceptación.....	- 50 -
3.2.3 Análisis de las pruebas de aceptación.....	- 54 -
Conclusiones parciales del capítulo.....	- 56 -
Conclusiones generales	- 57 -
Recomendaciones.....	- 58 -
Referencias bibliográficas	- 59 -
Anexos.....	- 63 -
Análisis del Catálogo Virtual Interactivo	- 63 -
Principales funcionalidades y comportamiento actual.....	- 63 -
Tarjetas CRC.....	- 65 -
Tareas de ingeniería asociadas a Historias de Usuario	- 71 -
Tareas de ingeniería para la Iteración I	- 71 -

Tareas de ingeniería para la Iteración II	- 74 -
Tareas de ingeniería para la Iteración III	- 74 -
Pruebas unitarias.....	- 76 -
Pruebas de aceptación	- 76 -
Pruebas de aceptación para la Iteración I	- 76 -
Pruebas de aceptación para la Iteración II	- 80 -

Índice de ilustraciones

Ilustración 1.Ciberoteca. Proyecto de la obra social Bancaja.	- 8 -
Ilustración 2. Portal Web de la Red de Bibliotecas y Archivos del CSIC.	- 8 -
Ilustración 3.Portal Web de la Biblioteca de Literatura Infantil y Juvenil.....	- 9 -
Ilustración 4.Interactive AR Catalogue for Colian Group.....	- 9 -
Ilustración 5. Zumoko AR-VR Catalogue.	- 10 -
Ilustración 6.Gesture Based UI Development.	- 12 -
Ilustración 7.Gesture Recognition App.	- 13 -
Ilustración 8. Kinect 2 Examples with MS-SDK an Nuitrack.....	- 13 -
Ilustración 9. Control Basics WPF.	- 14 -
Ilustración 10.Dispositivo Kinect 2.	- 18 -
Ilustración 11.Arquitectura del catálogo virtual interactivo.	- 32 -
Ilustración 12. Fragmento de la prueba unitaria para la clase AppManager (Modo Edición).....	- 49 -
Ilustración 13. Fragmento de prueba unitaria para la clase BookSelector (Modo Juego)	- 50 -
Ilustración 14. Resultados obtenidos en las pruebas de aceptación.....	- 55 -
Ilustración 15.Representación de los gestos incluidos en la solución.	- 64 -
Ilustración 16.Flujo de trabajo en el catálogo virtual interactivo.	- 64 -
Ilustración 17.Representación de las pruebas unitarias en Modo Editor y en Modo Juego.....	- 76 -

Índice de tablas

Tabla 1. Comparación de las aplicaciones de acuerdo a las variables	14 -
Tabla 2. Características principales. Dispositivo Kinect 2	18 -
Tabla 3. Historia de Usuario #1	24 -
Tabla 4. Historia de Usuario #2	25 -
Tabla 5. Historia de Usuario #3	26 -
Tabla 6. Historia de Usuario #4	26 -
Tabla 7. Historia de Usuario #5	27 -
Tabla 8. Historia de Usuario #6	28 -
Tabla 9. Historia de Usuario #7	28 -
Tabla 10. Estimación de esfuerzo por Historia de Usuario	29 -
Tabla 11. Plan de duración de las iteraciones	31 -
Tabla 12. Plan de entrega de versiones	32 -
Tabla 13. Prototipo de tarjeta CRC.....	36 -
Tabla 14. Tarjeta CRC # 1.....	37 -
Tabla 15. Tarjeta CRC # 2.....	38 -
Tabla 16. Tarjeta CRC # 3.....	38 -
Tabla 17. Tarjeta CRC # 4.....	38 -
Tabla 18. Tarjeta CRC # 5.....	39 -
Tabla 19. Tarjeta CRC # 6.....	39 -
Tabla 20. Tarjeta CRC # 7.....	40 -
Tabla 21. Tarjeta CRC # 8.....	41 -
Tabla 22. Tarjeta CRC # 9.....	42 -
Tabla 23. Tarea de Ingeniería #1	44 -
Tabla 24. Tarea de Ingeniería #2	45 -
Tabla 25. Tarea de Ingeniería #5	45 -
Tabla 26. Tarea de Ingeniería #6	45 -
Tabla 27. Tarea de Ingeniería #12	46 -
Tabla 28. Tarea de Ingeniería #14	46 -
Tabla 29. Tarea de Ingeniería #16	47 -
Tabla 30. Tarea de Ingeniería #18	47 -
Tabla 31. Tarea de Ingeniería #21	47 -
Tabla 32. Prueba de aceptación #1	51 -
Tabla 33. Prueba de aceptación #2.....	51 -
Tabla 34. Prueba de aceptación #3.....	52 -
Tabla 35. Prueba de aceptación #4.....	52 -

Tabla 36. Prueba de aceptación #11	- 53 -
Tabla 37. Prueba de aceptación #13.....	- 54 -
Tabla 38. Tarjeta CRC # 10.....	- 65 -
Tabla 39. Tarjeta CRC # 11.....	- 65 -
Tabla 40. Tarjeta CRC # 12.....	- 66 -
Tabla 41. Tarjeta CRC # 13.....	- 66 -
Tabla 42. Tarjeta CRC # 14.....	- 67 -
Tabla 43. Tarjeta CRC # 15.....	- 67 -
Tabla 44. Tarjeta CRC # 16.....	- 69 -
Tabla 45. Tarjeta CRC # 17.....	- 70 -
Tabla 46. Tarea de Ingeniería #3	- 71 -
Tabla 47. Tarea de Ingeniería #4	- 71 -
Tabla 48. Tarea de Ingeniería #7	- 72 -
Tabla 49. Tarea de Ingeniería #8	- 72 -
Tabla 50. Tarea de Ingeniería #9	- 72 -
Tabla 51. Tarea de Ingeniería #10	- 73 -
Tabla 52. Tarea de Ingeniería #11	- 73 -
Tabla 53. Tarea de Ingeniería #13	- 73 -
Tabla 54. Tarea de Ingeniería #15	- 74 -
Tabla 55. Tarea de Ingeniería #17	- 74 -
Tabla 56. Tarea de Ingeniería #19	- 74 -
Tabla 57. Tarea de Ingeniería #20	- 75 -
Tabla 58. Tarea de Ingeniería #22	- 75 -
Tabla 59. Prueba de aceptación #5.....	- 76 -
Tabla 60. Prueba de aceptación #6.....	- 77 -
Tabla 61. Prueba de aceptación #7.....	- 77 -
Tabla 62. Prueba de aceptación #8.....	- 78 -
Tabla 63. Prueba de aceptación #9.....	- 78 -
Tabla 64. Prueba de aceptación #10.....	- 79 -
Tabla 65. Prueba de aceptación #12.....	- 80 -

Introducción

En los últimos años la tecnología ha evolucionado considerablemente. Incontables avances en computación gráfica y visión por computador han contribuido al desarrollo de dispositivos como **MS-Kinect**, **Leap Motion**¹ y **MyO**² para el trabajo con interfaces basadas en gestos. Estos equipos han marcado un hito en la proliferación de una nueva generación de aplicaciones con las que los usuarios interactúan sin la necesidad de establecer contacto físico, gracias al uso de algoritmos de **visión artificial**³ para interpretar los gestos.

Esto ha permitido prescindir de los ya tradicionalmente empleados periféricos mouse y teclado, haciendo más atractivo e intuitivo este proceso y ofreciendo una comunicación más natural entre hombre y computador. Dicho proceso se ha vuelto cada vez más eficiente y productivo, permitiendo interactuar de forma nativa, todo ello inspirado en la forma en la que el ser humano se comunica.

Lo anteriormente mencionado, unido al constante auge de las tecnologías de la información, producto de la aparición y uso de Internet, ha revolucionado los sistemas de información tradicionales, volviéndolos cada vez más inmersivos e interactivos con los usuarios. En Cuba como parte del constante proceso de informatización que vive la sociedad, se hace imprescindible la integración de estas tecnologías en cada uno de los sectores de desarrollo estratégico de la nación, de manera progresiva, a fin de lograr una mayor soberanía tecnológica.

En este sentido, la Universidad de las Ciencias Informáticas apoya de manera constante el **Proceso de Enseñanza y Aprendizaje (PEA)**, combinando la innovación con el conocimiento de un amplio claustro de profesionales a fin de formar ingenieros cada vez más preparados en el área de la informática. Para apoyar esta tarea, la Dirección de Información Científico Técnica propone el empleo de librerías virtuales, las cuales constituyen un recurso necesario para el acceso y manejo de información digitalizada. Las mismas cuentan con **catálogos automatizados de acceso público en línea**, por sus siglas **OPAC** que permiten a los usuarios introducir etiquetas, puntuaciones y comentarios en los registros bibliográficos, permitiéndoles además seleccionar documentos como favoritos, organizarlos de una manera personalizada y

1 Módulo de seguimiento óptico de las manos que captura los movimientos de las mismas con una precisión inigualable.

2 Controlador gestual que desencadena una serie de acciones en el ordenador basadas en las contracciones de tus músculos y los movimientos de tu brazo.

3 Disciplina científica que trata la capacidad de percepción y comprensión de imágenes en un ordenador para que actúe de una forma determinada.

compartirlos con otros usuarios. Con este nuevo recurso los docentes pueden alcanzar una noción sobre los textos y otras materias en las que deseen indagar, luego de haberlas consultado de acuerdo a sus propios criterios de búsqueda.

De igual forma, se necesitan nuevas vías que faciliten el acceso a los contenidos y sean capaces de satisfacer las nuevas realidades y retos a los que se enfrenta el proceso docente-educativo. De ahí que el empleo del catálogo virtual pretenda resolver deficiencias asociadas a:

- Dificultad en la interacción usuario-información.
- Ineficiencias en el acceso de los usuarios a la información.
- Interfaces poco intuitivas para la exposición de la información.
- Limitada cantidad de contenidos, que beneficien la experiencia del usuario sin afectar su atención.

Teniendo en cuenta la situación problemática antes descrita, se identifica el siguiente **problema de investigación**: ¿cómo contribuir al proceso de interacción con la información para lograr un mejor aprovechamiento del fondo bibliográfico existente en la universidad?

A partir del problema de investigación formulado, se define como **objeto de estudio**: los catálogos digitales, específicamente los catálogos digitales basados en gestos.

En este sentido, para darle solución al problema planteado, se define como **objetivo general**: desarrollar un catálogo virtual interactivo para la visualización de contenidos bibliográficos usando **Kinect 2**.

Lo anteriormente planteado permite identificar como **campo de acción**: catálogos digitales basados en gestos para la interacción con la información.

Para el desarrollo de la investigación serán ejecutadas las siguientes tareas:

- Elaboración del marco teórico de la investigación a través del estudio del estado del arte para una mejor comprensión de la investigación.
- Identificación de la técnica y métodos de computación basada en gestos para mejorar la interacción en un catálogo digital.
- Caracterización de los catálogos digitales y las aplicaciones basadas en gestos realizadas tanto a nivel nacional como internacional, identificando los principales elementos que las componen y seleccionando aquellas funcionalidades que puedan incluirse en la solución propuesta.
- Caracterización de las herramientas y tecnologías que se utilizarán en el desarrollo de solución.

- Generación de los artefactos ingenieriles de acuerdo a la metodología seleccionada.
- Implementación de las interfaces del catálogo según los artefactos ingenieriles realizados.
- Validación del catálogo mediante las herramientas y métodos científicos determinados en la investigación.

Para darle solución a los objetivos trazados en la investigación se emplearon los siguientes **Métodos científicos**:

➤ **Métodos teóricos:**

Histórico-lógico: se utiliza para realizar un estudio del estado del arte acerca del desarrollo de bibliotecas virtuales para la educación en general, analizando los antecedentes y las tendencias actuales en el área, las aplicaciones y demás elementos tratados en la investigación y basado en estos datos, complementar las características necesarias y deseables para la solución que se propone.

Analítico-sintético: se emplea para realizar un estudio de las teorías y documentos más relevantes sobre el uso e importancia de las bibliotecas virtuales utilizadas en la educación, permitiendo así, extraer los elementos más importantes sobre las mismas.

Revisión documental: se utiliza para fundamentar el propósito de la investigación y permitir el desarrollo del marco teórico y/o conceptual de la misma.

Modelación: se emplea para la confección del catálogo virtual para brindar apoyo al proceso de enseñanza y aprendizaje de estudiantes y docentes en el área de las ciencias computacionales., al tener en cuenta los elementos teóricos y prácticos que lo caracterizan a partir del sustento teórico que aporta el tema.

➤ **Métodos empíricos:**

Consulta de información proveniente de fuentes confiables: se emplea para la elaboración del marco teórico de la investigación.

Observación: se emplea como método referencial al observar distintas aplicaciones que sirvieron como objeto de análisis y comparación para establecer las características y elementos fundamentales que debía cumplir la propuesta que plantea el autor.

Entrevista: se utiliza para establecer una comunicación con los especialistas para obtener información acerca de los procesos de acceso a la información y los usuarios escogidos para la realización de las pruebas de aceptación del catálogo virtual.

El presente documento está compuesto por tres capítulos en los que se relacionan todo lo referente a la investigación, estructurados de la siguiente forma:

Capítulo 1: Marco teórico referencial sobre el catálogo virtual interactivo para la visualización de contenidos bibliográficos. En este capítulo se definen los elementos teóricos necesarios para el desarrollo de la investigación y los principales conceptos que se emplearán durante todo el trabajo. Se realiza un análisis de algunas soluciones similares y se selecciona la metodología de desarrollo de software, así como las herramientas y tecnologías a utilizar.

Capítulo 2: Características y diseño del catálogo virtual interactivo para la visualización de contenidos bibliográficos. En este capítulo se describe el procedimiento seguido en las etapas de planificación y diseño que propone la metodología **XP**. Se especifican las **Historias de Usuarios (HU)**, el plan de iteraciones y el plan de entregas. También se describe la propuesta de solución y se mencionan los patrones de diseño utilizados en la implementación del catálogo.

Capítulo 3: Implementación y prueba del catálogo virtual interactivo para la visualización de contenidos bibliográficos. En esta etapa se definen las tareas de ingenierías correspondientes a cada **HU**. También se aplican las pruebas unitarias y de aceptación que permiten comprobar el correcto desarrollo de las funcionalidades implementadas y la aceptación del software por el cliente. De esta forma se obtiene un producto funcional probado y listo para entregar al final de cada iteración, de acuerdo con lo propuesto por la metodología **XP**.

Capítulo 1. Fundamentación Teórica

Introducción

En este capítulo se definen los principales elementos que sustentan el dominio de la investigación. Para ello se expone una valoración del estado del arte sobre el estudio realizado, referente a la interacción basada en gestos y se analizan algunas técnicas, métodos y herramientas que pueden aplicarse en la solución. Se describen, además, las principales tecnologías que se utilizaron para llevar a cabo la implementación.

1.1 Catálogos Públicos de Acceso en Línea (OPAC)

El **Catálogo de Acceso Público en Línea (OPAC)** es una de las herramientas para recuperar las fuentes de información. Sus características se inclinan más hacia los requisitos de la función de la biblioteca, como la adquisición, la catalogación, la circulación, el control de series y la búsqueda aleatoria, lo cual resulta estratégico para la recuperación de información de recursos y servicios. El **OPAC** cambia todo el panorama del sistema y sus metodologías como una revolución en el centro bibliotecario, las características del desarrollo informático que lo hacen popular son su uso, almacenamiento, recuperación, precisión, consistencia y secuencia automática de funcionamiento [1], [2].

La infraestructura de la **World Wide Web**⁴ ha traído consigo un cambio en la organización y presentación de la información. La web es un medio poderoso a través del cual toda persona u organización puede publicar mensajes para el consumo o uso de cualquiera que tenga acceso a través del **WEBOPAC**⁵. Se utiliza tanto para recibir información como para proporcionar información e interactuar con los usuarios.

En el caso de los catálogos digitales, esta tecnología se puede utilizar para proporcionar recorridos virtuales interactivos de cualquier servicio de información que tenga una prevalencia de colecciones de información en cualquier disciplina [2].

De acuerdo a lo anterior, se hace necesario que la solución propuesta opte por un diseño llamativo para los usuarios, que integre con calidad las imágenes de los contenidos y muestre la información de la forma más clara y concisa posible, manteniendo el orden de la representación en todo momento.

⁴ Sistema global interconectado de páginas públicas accesibles a través de Internet.

⁵ Web Online Public Access Catalog referente a los catálogos públicos de acceso en línea.

1.2 Interfaces Naturales de Usuario (NUI)

La forma en que se interactúa con las computadoras, ha sufrido cambios importantes desde el inicio de la computación; que van desde la utilización de tarjetas perforadas, hasta el desarrollo de interfaces cada vez más interactivas e intuitivas.

A este tipo de interacción, se le ha dado el nombre de: **Interfaces Naturales de Usuario (NUI)** por sus siglas en inglés. Un primer acercamiento a este tipo de interfaces sería: *“aquellas que permiten a los usuarios de hardware y software, interactuar con ellos en la misma manera cómo interactúan con el mundo real”*. Wigdor y Wixon, en su libro *“Brave NUI World”* [3], reflejan que la meta de estas interfaces es construir una experiencia de uso, que le sea natural al usuario [4].

Una interfaz de este tipo debe brindar la sensación de ser una extensión del cuerpo y ser natural tanto para usuarios expertos, como para los nuevos usuarios [5]. Para su creación se deben tener en cuenta las correctas metáforas, indicaciones visuales, retroalimentación y métodos de entrada y salida [6]. Todo ello enfocado en brindar una experiencia que sea auténtica al medio, a fin de evitar caer en la trampa de copiar los paradigmas de interfaz de usuario existentes [7].

1.2.1 Interfaces Basadas en gestos

Adquieren su nombre producto del uso exclusivo de gestos para operar sobre una interfaz. Su componente principal, formado por el reconocimiento de gestos, hace uso de algoritmos avanzados de visión artificial, reconocimiento de patrones, y otras técnicas de aprendizaje propias de la **Inteligencia Artificial**⁶ que permiten a un sistema responder ante determinadas acciones o intenciones del usuario. Funcionan, en gran medida, asistidos por el empleo de dispositivos externos como sensores, cámaras y micrófonos que permiten recolectar los datos, que luego son procesados e interpretados para finalmente generar un evento determinado [8].

Por esto, durante la interacción, el cuerpo humano se convierte en el agente principal para establecer la comunicación. Su empleo abarca un amplio espectro de aplicaciones orientadas, principalmente, al campo de la medicina, el comercio, la industria del entretenimiento y el sector automotriz. En el peculiar caso de la solución que se propone, será un ejemplo vivo de lo que se puede lograr también en el campo de la educación [9].

⁶ Disciplina que intenta replicar y desarrollar la inteligencia y sus procesos implícitos a través de computadoras.

1.3. Aplicaciones basadas en bibliotecas virtuales

Existe, hoy en día una amplia variedad de sistemas orientados al trabajo con bibliotecas virtuales, los cuales brindan infinidad de servicios y disponen de suficientes medios materiales. Estos servicios, se consideran recursos materiales a los efectos de este proceso, ya que van destinados a ofrecer servicios de información científica, técnica y profesional de apoyo al aprendizaje, la docencia, la investigación y la gestión de las instituciones educativas [10].

Entre las más destacadas a nivel mundial, ya sea por el amplio volumen de obras que poseen o visitas que reciben a diario, unido a si están o no especializadas en un área determinada, se encuentran aquellas asociadas directamente a organizaciones internacionales, como la **ONU**⁷ y la **UNESCO**⁸. Estas conservan valiosas colecciones que muestran la diversidad de las culturas del mundo y que tienen su origen precisamente en las contribuciones de otros cientos de organizaciones más. También las hay famosas por zonas, dependiendo de su área geográfica, antigüedad y valía de materiales en áreas del conocimiento como: las ciencias, la cultura, la religión, la historia, la política, la economía y sociedad, que han sido digitalizadas con el paso del tiempo. Otras se enfocan plenamente en la literatura infantil y juvenil y cuentan con amplios catálogos de trabajos, incluyendo fonotecas de obras clásicas, todo ello desplegado en una gran variedad de idiomas diferentes [11] [12].

La gran ventaja que han demostrado estas aplicaciones, es que suelen ser muy intuitivas y completas, por lo que los usuarios suelen estar más dispuestos a aprender. No obstante, existen inconvenientes, relacionados con las habilidades necesarias para la búsqueda y procesamiento de la información [2].

1.3.1 Análisis de las aplicaciones basadas en bibliotecas virtuales

Se realiza un estudio de homólogos sobre un conjunto de aplicaciones basadas en bibliotecas virtuales para identificar las potencialidades que puedan ayudar en la creación de la propuesta de solución.

Ciberoteca: Es un proyecto de la Obra Social Bancaja, cuyo fin principal es la catalogación de los mejores recursos culturales disponibles a través de Internet. Considerada como la Biblioteca virtual más grande del mundo, Ciberoteca brinda acceso gratuito a miles de textos literarios, científicos y técnicos, y se relaciona directamente

⁷ Organización de Naciones Unidas.

⁸ Organización de las Naciones Unidas para la Cultura, las Ciencias y la Educación.

con otros cientos de bibliotecas virtuales disponibles en Internet, por lo que puede definirse como una biblioteca de bibliotecas [2].



Ilustración 1. Ciberoteca. Proyecto de la obra social Bancaja.

Red de Bibliotecas y Archivos del CSIC: El servicio de Información de la Red de Bibliotecas del Consejo Superior de Investigaciones Científicas es un sistema horizontal de información científica que permite la selección, organización, difusión, acceso y conservación de los recursos de información con el fin de ser un instrumento útil para la investigación. Ofrece soporte a la excelencia científica garantizando el acceso intelectual, físico y electrónico a la información, las ideas y los servicios para una comunidad de investigación multidisciplinar como la del **CSIC**⁹ mediante la utilización, en modo local o remoto, de infraestructuras y servicios de calidad [13].



Ilustración 2. Portal Web de la Red de Bibliotecas y Archivos del CSIC.

Biblioteca de Literatura Infantil y Juvenil: La Biblioteca de Literatura Infantil y Juvenil contribuye a divulgar un fondo histórico parcialmente conocido que sirve para delimitar el campo de estudio de la Literatura Infantil y Juvenil, y al mismo tiempo cuenta con un material con el que poder establecer los contactos del niño y del joven con la

⁹ Consejo Superior de Investigaciones Científicas.

lectura a partir de obras clásicas de la historia literaria. Representa un espacio de referencia que contiene un catálogo virtual de autores españoles e hispanoamericanos con obras infantiles y juveniles, revistas, cuentos, bibliotecas de autor, videoteca y fonoteca de obras clásicas, talleres y enlaces de interés, todo ello dirigido al mundo de la educación, edición, formación e investigación [14].



Ilustración 3. Portal Web de la Biblioteca de Literatura Infantil y Juvenil.

Interactive AR Catalogue for Colian Group: Proyecto dedicado a los representantes de ventas del grupo **Colian**¹⁰, uno de los mayores productores de **FMCG**¹¹ de Polonia. Consiste en un catálogo promocional interactivo con el uso de tecnología de realidad aumentada. La aplicación **AR**¹² permite a los representantes de ventas mostrar a sus clientes un catálogo moderno que presenta nuevos productos en 3D junto con sus principales ventajas. Además, permite ver varios estantes de ventas disponibles para **POS**¹³ en **AR** [15].



Ilustración 4. Interactive AR Catalogue for Colian Group.

¹⁰ Compañía productora de refrescos y dulces en Polonia.

¹¹ Fast Moving Consumer Goods o bienes de consumo de rápido movimiento.

¹² Augmented Reality o Realidad Aumentada. Tecnología que permite indexar contenidos virtuales en el mundo real mediante dispositivos de enlace tales como teléfonos móviles o tabletas.

¹³ Point of Sale o Terminal de Venta.

Zumoko AR-VR Catalogue: Presenta varios casos de uso para el despliegue de tecnología **AR** y **VR**¹⁴. Incluye un visualizador de automóviles para la personalización de automóviles de tamaño real en Realidad Aumentada. También posee un módulo de visualización de bienes raíces e interacciones personalizadas para personalizar cualquier necesidad arquitectónica, ofreciendo un recorrido y la personalización de una casa de ensueño en Realidad Virtual. Muestra además una representación del Sistema Solar en Realidad Aumentada, donde el usuario puede explorar y aprender más sobre el tema, accediendo a información interesante sobre cada planeta [16].

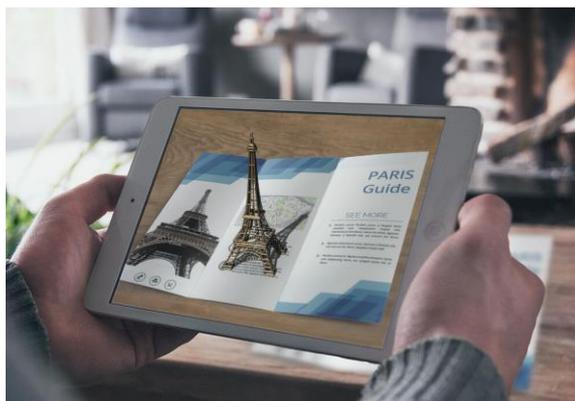


Ilustración 5. Zumoko AR-VR Catalogue.

El análisis anterior permitió identificar un grupo de soluciones desplegadas con fines específicos que responden a la necesidad principal del manejo de información para satisfacer las necesidades propias del mercado y sus clientes. Esto hace posible tomarlas como referencia a la hora de definir los contenidos necesarios y su modo de representación en la solución.

1.4 Gestos

Los gestos son empleados por los seres humanos como parte de su comunicación no verbal con otras personas, pero su uso puede ser llevado a la línea de comunicación entre hombre y computadora. Para ello se debe revisar la definición de gestos en el contexto de una conversación, ya que estos pueden definirse como etiquetas para acciones que poseen características de expresividad deliberadamente manifestada y que los participantes de la interacción pueden reconocer. Por lo tanto, un usuario puede

¹⁴ Entorno de escenas y objetos simulados de apariencia real que crean en el usuario la sensación de estar inmerso en él.

realizar movimientos con sus manos y al unísono estas acciones ser reconocidas por la computadora para ejecutar alguna orden a través del sensor de gestos **Kinect 2**¹⁵.

1.4.1 Aplicaciones basadas en gestos

El uso de gestos como una forma de interacción con ordenadores ha sido estudiado desde hace más de 30 años. Sin embargo, recientemente ha cobrado mayor relevancia gracias al surgimiento de nuevos dispositivos que facilitan el desarrollo de aplicaciones de este tipo.

Entre los equipos que han contribuido a la proliferación de aplicaciones basadas en gestos se encuentran **MS-Kinect**, **Leap Motion** y **MyO**, ya descritos anteriormente. Empleados de manera exclusiva para la detección, estos dispositivos han sido utilizados en múltiples soluciones para áreas que van más allá del entretenimiento [17], como la robótica suave; dando tratamiento a tareas recurrentes como seleccionar, maximizar, arrastrar, copiar y pegar. De igual forma, hacen posible la inclusión de la preservación cultural y la computación incluyente que involucra grupos poblacionales particulares como niños, adultos mayores y personas con capacidades limitadas [18].

Existe una cantidad importante de investigaciones que muestran la utilidad de las aplicaciones basadas en gestos en el ámbito educativo [19]. En todos estos trabajos el usuario, mediante movimientos de su cuerpo o comandos de voz, controla las interfaces y los objetos de los sistemas correspondientes. En este sentido, las aplicaciones basadas en gestos para la educación, pueden ayudar a aumentar el rendimiento y estimular el aprendizaje de los estudiantes, facilitando de alguna manera la manipulación del material didáctico educativo, lo que ayudaría a crear una clase interactiva, interesante y más entretenida [20].

De manera general la investigación y el desarrollo tecnológico de métodos y herramientas que permitan la comunicación entre el humano y el ordenador, a través de gestos, es un tema vigente, de mucho interés tanto para la comunidad académica como para la industria [6], por lo que resulta de gran importancia lograr una visión que permita desarrollar soluciones capaces de afrontar los desafíos existentes e integrarse exitosamente en los entornos reales donde se requiera su aplicación [19].

¹⁵ El sensor Kinect es un producto heredado que se lanzó originalmente para Xbox 360, y luego para la serie Xbox One con el adaptador Kinect. Kinect tiene una cámara integrada que permite jugar a juegos interactivos, hacer fotos y mucho más.

1.4.2 Análisis de las aplicaciones basadas en gestos con Kinect 2

De igual forma se realiza un estudio sobre un grupo de aplicaciones basadas en gestos con **Kinect 2**, en su mayoría soluciones de prueba. Esto permite analizar el flujo de trabajo y los estilos de interacción que utilizan a fin de ayudar a enriquecer los propios que se deben integrar en la propuesta de solución.

Gesture Based UI Development: Una aplicación de Unity desarrollada con **Kinect 2** que incorpora el uso único de varios gestos y patrones de voz en dos juegos separados unidos por una interfaz de usuario controlada por gestos. Desarrollada en conjunto por un grupo de investigadores asociados al tema, busca explorar todo el potencial del **SDK¹⁶** para **Kinect 2** [21].



Ilustración 6. Gesture Based UI Development.

Gesture Recognition App: Consiste en una aplicación de escritorio de Windows para el reconocimiento de gestos y su asignación a varias entradas de mouse y teclado. La aplicación **WPF (Windows Presentation Foundation)** tiene como objetivo controlar las entradas del mouse y el teclado con gestos de manos mapeados de manera innovadora (tanto continuos como discretos). La aplicación autentica y también autoriza dinámicamente a los usuarios a usar gestos para el control interactivo del sistema. El diseño intuitivo de la interfaz ayuda a los usuarios a establecer la sensibilidad y les permite crear perfiles de autorización de forma dinámica [22].

¹⁶ Software Development Kit o Kit de Desarrollo de Software. Consiste en un grupo de librerías de código especialmente destinadas para la creación de aplicaciones.

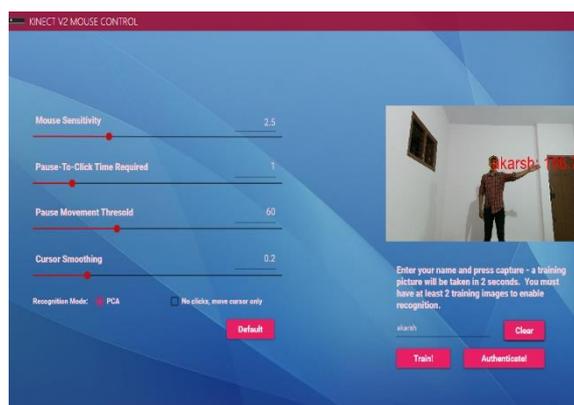


Ilustración 7. Gesture Recognition App.

Kinect 2 Examples with MS-SDK and NuiTrack: Consiste en una colección de muestras realizadas con **Kinect 2**, su **SDK** y el framework **Nuitrack**¹⁷; integrada por una amplia variedad de recursos, así como scripts y escenas de demostración para diferentes casos de uso. Permite a los desarrolladores adquirir una rápida noción del trabajo con estas tecnologías y con ello aumentar la rapidez y calidad con que se realizan las soluciones. Aunque es de pago, puede ser accesible para el uso académico [23].



Ilustración 8. Kinect 2 Examples with MS-SDK an NuiTrack.

Control Basics WPF: Aplicación de muestra que viene de la mano del paquete de demostración que posee la **SDK de Kinect 2**. Se centra en demostrar el uso de controles en una experiencia de interacción con este dispositivo. Excelente para un primer acercamiento al desarrollo de soluciones con esta tecnología [24].

¹⁷ NuiTrack™ es un middleware de seguimiento 3D desarrollado por 3DiVi Inc. para el trabajo con sensores como el Kinect.

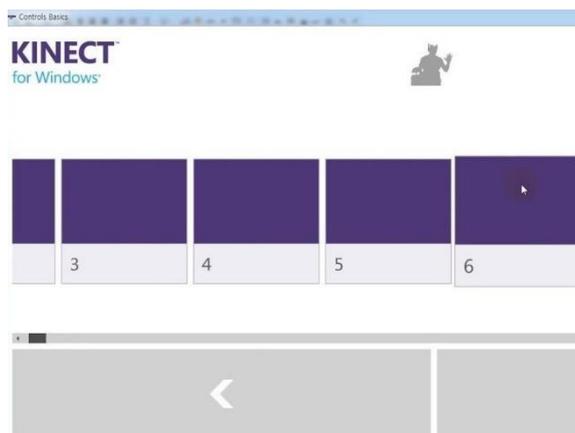


Ilustración 9. Control Basics WPF.

El análisis anterior permitió identificar un grupo de aplicaciones orientadas al trabajo con gestos, que responden a la necesidad de interacción con el entorno o los contenidos que puedan ofrecer de acuerdo a las necesidades propias del mercado y sus clientes. Esto hace posible tomarlas como referencia a la hora de definir los estilos de interacción necesarios para aplicar en la solución.

1.5 Selección del dominio de aplicación a partir del análisis de aplicaciones homólogas

Se realiza una comparación (**ver Tabla 1**) entre todas las aplicaciones homólogas para seleccionar el dominio de la aplicación propuesta como solución. Las variables seleccionadas, teniendo en cuenta algunas características funcionales de los sistemas analizados anteriormente, se definen como: **nivel de acceso a la información, lúdica, idioma, nivel de enseñanza, licencia, y estilo de interacción**. Las mismas están enfocadas al apoyo del trabajo en las áreas de visualización e interacción con contenidos a fin de mejorar el proceso de enseñanza y aprendizaje.

Leyenda:

1-Ciberoteca. 2- Red de Bibliotecas de CSIC. 3- Biblioteca de Literatura Infantil y Juvenil. 4- Interactive Catalogue for AR Colian Group. 5-Zumoko AR-VR Catalogue. 6-Gesture Based UI Development. 7-Gesture Recognition App. 8-Kinect 2 Examples with MS-SDK and NuiTrack. 9- Control Basics WPF.

Tabla 1. Comparación de las aplicaciones de acuerdo a las variables

Variables	Categorías	Aplicaciones								
		1	2	3	4	5	6	7	8	9
Plataforma	Web	X	X	X						
	Escritorio						X	X	X	X

	Móvil				X	X				
Catálogo de contenidos				X	X	X				
Nivel de enseñanza	Primaria			X						
	Secundaria	X		X	X	X				
	Nivel Medio-Superior	X	X	X	X	X			X	
	Nivel Superior		X				X	X	X	X
Lúdica		X			X	X	X	X	X	X
Medio de acceso a la información	Online	X	X	X	X	X				
	Offline						X	X	X	X
Interacción	Interacción Directa	X	X	X	X					
	Basado en gestos						X	X	X	X
Tipo de licencia	Pública	X	X	X			X	X	X	X
	Privada				X	X				
Idioma	Español	X	X	X	X	X				
	Inglés		X		X	X	X	X	X	X

Después de interpretar la información recogida en la tabla anterior, se determina que ninguna herramienta incluye en su totalidad las variables definidas en la investigación, a tratar en las áreas sobre las que se va a trabajar en la solución. El primer grupo de aplicaciones web varían significativamente en cuanto al nivel de enseñanza y en su mayoría no poseen un catálogo de contenidos. En cambio, el segundo grupo compuesto por aplicaciones móviles, explora el tema de los catálogos digitales sobre entornos **VR** y **AR**. Estas soluciones muestran sus productos o servicios al cliente haciendo uso de estas tecnologías, siendo privativos a la hora de su uso.

En cuanto a las aplicaciones de escritorio están compuestas, en su mayoría; por soluciones de prueba que, aunque aplican un modelo común de interacción basado en gestos requieren de conocimientos técnicos elevados demasiado engorrosos para su aplicación a todos los niveles de enseñanza, variando además en los tipos de licencia requeridos para su uso. Entre las soluciones, existen diferencias notables en el manejo de los patrones de interacción, ya sea mediante interacción directa o con gestos, así como en las vías empleadas por los usuarios para acceder a estos servicios, ya sea por vía online u offline.

Teniendo en cuenta lo anterior, se concluye que ninguno de los sistemas tratados es capaz de cumplir en su totalidad con los objetivos propuestos, por lo que la solución deberá contar con las variables definidas anteriormente y con características comunes entre todas las aplicaciones homólogas, siendo la más importante: la creación de un catálogo virtual que refleje una variedad de materiales bibliográficos accesibles mediante estilos de interacción definidos, para lograr una experiencia que beneficie el aprendizaje de los usuarios.

Como resultado de esta evaluación, se determinan las principales características que no deben faltar en el desarrollo de la solución:

- Interfaz intuitiva, amigable y dinámica.
- Sonido de fondo.
- Acceso a la información a través del catálogo de contenidos.
- Acceso a los materiales digitales del catálogo.
- Interacción dinámica con el contenido, combinando el uso de gestos o la interacción directa, según el usuario desee.

Como se puede apreciar, luego de analizar los tres tipos de aplicaciones tratados con antelación y con el fin de obtener los mejores resultados posibles en términos de visualización, interacción con los contenidos y experiencia de usuario; se determina desarrollar un catálogo virtual basado en gestos de uso formativo que dé solución al problema de investigación en cuestión.

1.6 Método de Ingeniería de Software

El desarrollo de esta aplicación se manifiesta de forma diferente a un proceso de desarrollo de software común. De ahí que las actividades ingenieriles que se deben ejecutar durante este proceso generen resultados diferentes a un proyecto de software de cualquier otro dominio de aplicación.

Tomando como punto de partida la idea anterior, se determina hacer uso de los pasos que define la metodología **XP**¹⁸ para guiar el proceso de desarrollo, atendiendo al esquema que se propone en [25] y adoptando sus 4 fases (**Planificación, Diseño, Desarrollo y Prueba**), producto al estatus de sobresaliente que posee por encima de los otros procesos ágiles de desarrollo de software. De ahí que su fuente de éxito se deba, principalmente al empeño que destina a la satisfacción del cliente. Esta metódica es capaz de facultar a los desarrolladores para responder con seguridad a las necesidades cambiantes de los clientes, incluso en etapas tardías del ciclo de vida del producto. También enfatiza el trabajo en equipo, por lo que los jefes de proyecto, clientes y desarrolladores son socios iguales en un ambiente estrecho de colaboración que se autoorganiza en torno al problema a resolver de la forma más eficiente posible.

XP mejora un proyecto de software en cinco aspectos esenciales: la comunicación, la sencillez, la retroalimentación, el respeto y el valor. Los programadores extremos se comunican constantemente con sus clientes y colegas de trabajo, manteniendo durante el proceso un diseño elemental y limpio [26]. Al mismo tiempo, reciben retroalimentación probando su software desde del primer día y entregan el sistema a los clientes tan pronto

¹⁸ *Extreme Programming* o Programación extrema. Metodología ágil para el desarrollo de software.

como sea posible con el propósito de implementar cambios como y cuando se sugiera. Cada pequeño éxito profundiza el respeto por las contribuciones únicas de cada uno de los miembros del equipo, formando una base que permite a los programadores extremos enfrentar con valentía y audacia las cambiantes necesidades del mercado y las tecnologías [25].

1.7 Herramientas y Tecnologías

Un aspecto imprescindible en el desarrollo de software, viene constituido por la descripción de las herramientas a utilizar, las cuales contribuyen en buena medida a mejorar la efectividad, así como el tiempo de desarrollo de la solución.

1.7.1 Kinect for Windows SDK 2.0

Se emplea el **Kit de Desarrollo de Software 2.0 de Windows para Kinect, SDK 2.0** por sus siglas en inglés debido a que constituye la herramienta gratuita de Microsoft que permite a desarrolladores crear aplicaciones que soportan el reconocimiento de gestos y voz utilizando el Kinect. Incluye los controladores necesarios para utilizar este dispositivo con Windows, así como la **API**¹⁹ y ejemplos de código. Con esta herramienta se puede registrar y seguir el movimiento de las personas, determinar distancias entre un objeto y el sensor, así como capturar audio y encontrar la ubicación de la fuente, entre otros.

Sus **requerimientos mínimos** incluyen:

- Sistema operativo Windows 8, Windows 8.1 o superior.
- 4 GB de memoria o más.
- Procesador Dual-Core de 3.1 GHz o mejor.
- Controlador USB 3.0 para el sensor de Kinect 2.0.
- Adaptador gráfico con DirectX 11.
- Sensor Kinect 2.0 de Microsoft.

1.7.2 Kinect 2

Es el dispositivo que se utiliza para llevar a cabo este trabajo (ver Ilustración 1). Permite a los usuarios controlar y comunicarse con una consola o PC sin necesidad de establecer contacto físico con un periférico de entrada tradicional, mediante una interfaz natural de usuario que reconoce gestos, comandos de voz y objetos. Su funcionamiento

¹⁹ Application Programming Interface o Interfaz de Programación de Aplicaciones. Consiste en un conjunto de definiciones y protocolos que se usan para diseñar e integrar el software de las aplicaciones.

se basa en recoger distintos tipos de imágenes, detectando así a las personas que se encuentren delante de la cámara. Con esta información y con los programas que incluye su **SDK** versión **2.0**, es posible identificar los gestos que se emplean en la solución. La siguiente tabla hace referencia a las principales características de este dispositivo.

Tabla 2. Características principales. Dispositivo Kinect 2

Dispositivo	Kinect v2
Video	1920x1080@30fps Alta Definición
Profundidad	512x424 Distancias 0,5 a 4,5 m
Rastreo del cuerpo	Capacidad para detectar 6 cuerpos, todos completamente rastreados. Por cuerpo rastreado, es capaz de identificar 25 articulaciones.
Motor de inclinación	No. Tiene mayor ángulo de visión. En caso de que se requiera, se puede graduar manualmente su inclinación.
USB	3.0
Sistema Operativo	Windows 8.1 o superior (64bits)



Ilustración 10. Dispositivo Kinect 2.

1.7.3 Herramienta CASE

Visual Paradigm for UML (versión 15.1) es una herramienta **CASE**²⁰ que soporta el modelado mediante **UML**²¹ y proporciona asistencia a los analistas, ingenieros de software y desarrolladores, durante todos los pasos del Ciclo de Vida de desarrollo de un Software [27].

²⁰ Computer Aided Software Engineering o Ingeniería de Software Asistida por Computadora.

²¹ Unified Modified Language o Lenguaje de Modelado Unificado.

Algunas de las ventajas que proporciona son:

- **Dibujo:** facilita el modelado de **UML**, proporcionando herramientas específicas para ello, lo cual permite la estandarización de la documentación, ya que la misma se ajusta al estándar soportado por la herramienta.
- **Corrección sintáctica:** controla que el modelado con **UML** sea correcto.
- **Coherencia entre diagramas:** al disponer de un repositorio común, es posible visualizar el mismo elemento en varios diagramas, evitando duplicidades.
- **Interoperabilidad con otras aplicaciones:** permite integrarse con otras aplicaciones, como herramientas ofimáticas, lo cual aumenta la productividad.
- **Trabajo multiusuario:** permite el trabajo en grupo, proporcionando herramientas de compartición de trabajo.
- **Reutilización:** facilita la reutilización, ya que dispone de una herramienta centralizada donde se encuentran los modelos utilizados para otros proyectos.
- **Generación de código:** permite generar código de forma automática, reduciendo los tiempos de desarrollo y evitando errores en la codificación del software.
- **Generación de informes:** permite generar diversos informes a partir de la información introducida en la herramienta.

En cuanto al **Lenguaje de Modelado Unificado (UML) (versión 2.5)**, Pressman lo define como: *“un lenguaje estándar para escribir diseños de software, el cual puede usarse para visualizar, especificar, construir y documentar los artefactos de un sistema de software intensivo”* [28]. Está pensado para utilizarse en todos los métodos de desarrollo, etapas del ciclo de vida de un software, dominios de aplicación y medios. Es un sistema notacional destinado a los sistemas de modelado que utilizan conceptos orientados a objetos. Representa un estándar mundial ampliamente utilizado por desarrolladores, autores y proveedores de Herramientas para Ingeniería de Software Asistida por Computación (**CASE**) [29].

1.7.4 Motor Gráfico

Un motor de videojuego es un término que hace referencia a una serie de librerías de programación que permiten el diseño, la creación y la representación de un videojuego. El aspecto más destacado a la hora de elegir un motor de videojuegos entre todos los disponibles que hay en el mercado son las capacidades gráficas, ya que son las encargadas de mostrar las imágenes 2D y 3D en pantalla, así como calcular algunos aspectos como los polígonos, la iluminación, las texturas. Otras características para tener en cuenta a la hora de la elección son la facilidad de aprender a usar el motor de

videojuegos y la facilidad para exportar el juego a diferentes plataformas. Una representación de los motores más importantes que existen en la actualidad viene formada por: **Unreal Engine**, **GameMaker** y **Unity 3D**, por solo mencionar algunos de ellos.

Unity 3D

Se selecciona para el desarrollo de la solución **Unity 3D (versión 2020.2.7f1)** producto a su potencia como motor **cross-platform** (multiplataforma) para realizar proyectos tanto 2D como 3D. Fácil de manejar para los principiantes, permite crear videojuegos y aplicaciones para multitud de plataformas [30]. Posee las **stores** (tiendas), que son bibliotecas de **assets** (recursos) comerciales y gratuitos creados por **Unity Technologies** y miembros de la comunidad. Hay una gran cantidad de **assets** disponibles, desde texturas, modelos, animaciones y extensiones del editor [31]. Otros factores determinantes a la hora de la selección fueron:

- Soporte completo para el dispositivo **Kinect 2**, mediante empleo del plugin **KinectForWindows** y la integración con **Kinect for Windows SDK 2.0**.
- Facilidad para realizar videojuegos tanto 2D como 3D.
- Curva de aprendizaje sencilla. Posee una gran comunidad de creadores de **plugins**²² y recursos, así como muchos manuales oficiales para el aprendizaje rápido.
- Las herramientas de edición gráfica son muy buenas y pueden ser mejoradas con **plugins**.
- Soporta multitud de formatos de recursos y los convierte de manera automática a los formatos más óptimos dependiendo de la plataforma objetivo.
- Licencia gratuita (desde **Unity 5**), que cubre prácticamente todas las necesidades del desarrollador.
- Soporta oficialmente dos tipos de lenguaje de programación: **C#**, **UnityScript** (básicamente **Javascript**). El más generalizado en la comunidad es **C#**.
- Ahorro y reutilización de código, puesto que se incluyen ciertas partes del mismo.

1.7.5 Visual Studio Code

Se utiliza **Visual Studio Code (versión 1.72.2)** producto a la capacidad que ofrece como editor de código para combinar su sencillez con potentes herramientas para

²² Complementos que añaden funcionalidades extra o mejoras a los programas.

desarrolladores como la finalización y depuración de código **IntelliSense**²³, destinadas en esta ocasión al trabajo con **C#** [32]. VS Code utiliza una arquitectura de servicio de herramientas que le permiten integrarse con muchas de las mismas tecnologías que potencian Visual Studio, como **Roslyn** para **.NET** y **TypeScript**, su motor de depuración. Permite el trabajo con **Git** y otros proveedores de **SCM**²⁴, mediante acciones simples llevadas a cabo directamente desde el editor como revisiones, preparación de archivos y commits entre otras. **Code** es rápido, gratuito, potente y presenta numerosas funcionalidades. Posee compatibilidad con **Unity** incorporada, mediante el trabajo con extensiones. Al mismo tiempo, mejora la precisión de los flujos de trabajo, permitiendo a los desarrolladores realizar sus tareas con eficiencia y sin contratiempos, a la vez que se integran sin problemas con las numerosas partes del proceso [33].

1.7.6 Lenguaje de Programación C#

Se determina hacer uso de **C#** o **C Sharp (versión 10.0)**, como lenguaje de programación para dar tratamiento a la solución porque presenta una sintaxis sencilla, lo que simplifica al desarrollador a la hora de escribir código. Se puede integrar fácilmente a otros lenguajes que compilen con **.NET** y mejora la gestión de memoria al disponer de un recolector de basura que destruye los objetos que no se usan.

Sin embargo, el motivo fundamental por el que se aplica viene dado porque es empleado significativamente por **Unity 3D** para compilar los scripts. **C#** es orientado a objetos, lo cual facilita el trabajo con esta herramienta ya que todo componente o elemento del sistema es un objeto o clase. Esto hace que soporte las características propias de este paradigma como son: la encapsulación, la herencia y el polimorfismo. También es orientado a componentes [34], lo que permite definir propiedades sin necesidad de crear métodos, como en Java o usar eventos sin tratar con punteros a funciones. Permite la definición de una o más clases dentro de un mismo espacio de nombres. Por último, pero no por ello menos importante resulta el hecho que dispone de un rango más amplio y definido de tipos de datos que otros lenguajes [35].

²³ Herramienta empleada por Visual Studio Code para el autocompletado de código, capaz de proporcionar elementos inteligentes basados en tipos de variables, definiciones de funciones y módulos importados.

²⁴ System Control Management o Manejo de Control de Versiones.

Conclusiones parciales del capítulo

Durante el desarrollo del capítulo se arriba a las siguientes conclusiones parciales:

- Se identifican los estilos de interacción como directa o basada en gestos, a los cuales deberá responder el componente de interacción en la solución.
- Se realiza el estudio de homólogos sobre un grupo de aplicaciones para identificar las principales características que deben estar presentes en el desarrollo de la solución.
- Atendiendo a las características que debe cumplir la solución propuesta, se determina crear un catálogo virtual como medio interactivo, lúdico y llamativo que apoye el uso del fondo bibliográfico de la institución.
- El marco de trabajo ingenieril para el proceso de desarrollo justifica el uso de la metodología **XP** porque garantiza la generación de artefactos que son necesarios para una mayor comprensión de la solución a desarrollar.
- Por poseer un precedente de desarrollo de videojuegos comprobado, se seleccionan para el desarrollo de la investigación herramientas y tecnologías como: **Visual Studio Code** como editor de código, **Unity 3D** como motor de desarrollo de videojuegos, **Visual Paradigm for UML** como herramienta **CASE** y el **Kinect for Windows SDK 2.0** como el conjunto de librerías necesarias para el trabajo con el dispositivo de gestos **Kinect 2**.

Finalmente se está en condiciones de desarrollar el catálogo virtual interactivo para la visualización de contenidos bibliográficos, utilizando las herramientas y tecnologías seleccionadas.

Capítulo 2. Características y diseño de la solución

Introducción

El presente capítulo describe la propuesta de solución que se propone alcanzar mediante el desarrollo de la investigación. Se mencionan los roles que intervienen en el proceso y se especifican los principales aspectos que define el Marco de Trabajo Ingenieril, enfocado en la metodología **XP**, para la construcción de la solución. Dichos aspectos son: las historias de usuario, la planificación de entregas de versiones del producto y el diseño del sistema de acuerdo a las fases que **XP** propone. Lo anterior permitirá identificar las necesidades del producto a fin de ayudar al desarrollador a comprender mejor la naturaleza del software que se debe construir.

2.1 Propuesta de solución

Después de analizar las necesidades del sistema y seleccionar las herramientas para la implementación, se definen los elementos a desarrollar para dar solución al problema planteado.

Catálogo de Contenidos: Permitirá el acceso de los usuarios a un entorno virtual donde podrán visualizar e interactuar con una amplia variedad de materiales (libros, revistas, artículos científicos, tesis de grado y eventos) propios del área de la informática y la universidad. Pretende brindar una mejor representación de la información que puede ser utilizada para apoyar el proceso de enseñanza y aprendizaje.

Módulo de Gestión de Contenidos: Responderá a las necesidades del catálogo de mantenerse actualizado con los materiales más recientes por área de interés, pertenecientes a las secciones de: libros, revistas, artículos científicos, tesis y eventos. Garantizará la disponibilidad de los contenidos en todo momento para las consultas de información por parte de los usuarios del sistema.

Módulo de Reconocimiento de Gestos: Permitirá la integración de un reconocedor que asistido por el empleo del sensor Kinect 2, facilitará la interacción mediante gestos entre los usuarios y la aplicación. Para una lograr una mayor comprensión, se propone al lector consultar los [anexos](#) de la investigación.

Para el desarrollo de los módulos propuestos se siguen los pasos que establece la metodología ágil **XP**, donde se respetará el esquema que explica [26].

2.2 Fase I: Planificación

Constituye la fase inicial de la metodología **XP**, donde se establece una comunicación continua entre el equipo de desarrollo y el cliente, para obtener los requisitos del

sistema. Además, permite establecer el alcance general del proyecto, así como las fechas de entrega del sistema, tomando en cuenta la prioridad y el tiempo estimado para el desarrollo de cada historia de usuario **(HU)**²⁵. Partiendo de lo anteriormente mencionado, una primera entrega debería obtenerse en un plazo no mayor a tres meses [25], [26].

2.2.1 Historias de usuario (HU)

Las **HU** representan la técnica utilizada por **XP** para especificar los requisitos de software. Las mismas deben ser programadas en un período de tiempo comprendido entre una y tres semanas aproximadamente. Si, por el contrario, la estimación es superior a las tres semanas, se debe dividir la historia en dos o más **HU** y si es menos de una semana, se debe combinar con otra respectivamente. A la hora de estimar el esfuerzo asociado a la implementación de las historias, los programadores deben utilizar como medida de referencia el punto. Un punto es equivalente a una semana ideal de programación (5 días hábiles). Las historias valen, generalmente; de 1 a 3 puntos en esta escala. A continuación, se describen las **HU** definidas para llevar a cabo el desarrollo de los módulos propuestos en la solución.

Tabla 3. Historia de Usuario #1

Historia de Usuario	
Número: 1	Nombre: Mostrar vista principal
Usuario: Usuario	
Prioridad en negocio: Alta	Riesgo en desarrollo: Media
Puntos estimados: 1.5	Iteración asignada: 1
Programador responsable: Leudis Rafael Estrada González	
Descripción: Ofrece a los usuarios la interfaz inicial de la aplicación, desde la cual podrán interactuar y navegar por los diferentes escenarios del catálogo, siendo capaces de acceder a las secciones de ayuda, configuración de interacción o sonido, o de salida de la aplicación.	
Observaciones: Acceso público a todos los usuarios del sistema.	
Prototipo de Interfaz: Vista principal	

²⁵ Historias de Usuario



Tabla 4. Historia de Usuario #2

Historia de Usuario	
Número:2	Nombre: Mostrar vista de contenidos
Usuario: No aplica	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Puntos estimados: 2	Iteración asignada: 1
Programador responsable: Leudis Rafael Estrada González	
Descripción: Carga los contenidos bibliográficos, así como toda la información asociada, volviéndolos disponibles en todo momento para las consultas de información por parte de los de usuarios del sistema.	
Observaciones: No aplica	
Prototipo de Interfaz: Vista de contenidos	

Tabla 5. Historia de Usuario #3

Historia de Usuario	
Número:3	Nombre: Mostrar información de contenidos
Usuario: Usuario	
Prioridad en negocio: Alta	Riesgo en desarrollo: Media
Puntos estimados: 1	Iteración asignada: 1
Programador responsable: Leudis Rafael Estrada González	
Descripción: Permite a cada usuario acceder a todos los contenidos por temas previamente clasificados, los cuales podrán consultar para obtener información necesaria como: el título, la descripción, el autor(es), el idioma, la fecha de publicación, el número de páginas, el código, entre otros.	
Observaciones: Una vez seleccionado un material de la colección, se muestra un panel que recoge en su interior toda la información necesaria sobre el contenido.	
Prototipo de Interfaz: Panel de información de contenidos	
 <p>The screenshot shows a web interface titled 'Contenidos bibliográficos'. It features a header with the UCI logo and a search icon. The main content area displays information for 'Global Game Jam 2023'. On the left is a colorful poster for the event. On the right, there is a 'Descripción' section with text about the event, and several blue buttons containing key information: 'Ubicación: La Habana, Cuba', 'Fecha: 30 Enero - 5 Febrero 2023', 'Universidad de las Ciencias Informáticas', and 'Status: Por realizar'.</p>	

Tabla 6. Historia de Usuario #4

Historia de Usuario	
Número:4	Nombre: Controlar opciones
Usuario: Usuario	
Prioridad en negocio: Alta	Riesgo en desarrollo: Medio
Puntos estimados: 2	Iteración asignada: 2
Programador responsable: Leudis Rafael Estrada González	
Descripción: Permite al usuario configurar las opciones de interacción o sonido según lo desee. En el caso de caso de la primera, podrá activar o desactivar la interacción con Kinect 2, pasando de gestual a manual y viceversa. En cuanto al sonido, este podrá ser regulado a conveniencia, según se desee.	

Observaciones: El usuario selecciona el botón de “**Opciones**” y se muestra un panel que le permite configurar a su gusto los métodos de interacción y el volumen del sistema.

Prototipo de Interfaz: Opciones

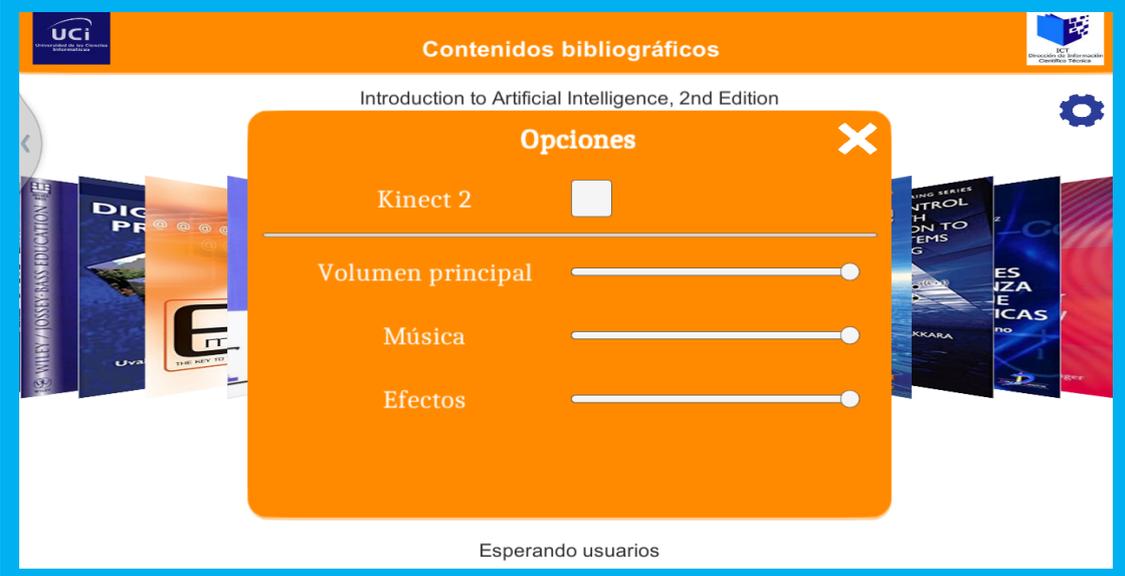


Tabla 7. Historia de Usuario #5

Historia de Usuario	
Número: 5	Nombre: Consultar ayuda
Usuario: Usuario	
Prioridad en negocio: Media	Riesgo en desarrollo: Bajo
Puntos estimados: 1	Iteración asignada: 2
Programador responsable: Leudis Rafael Estrada González	
Descripción: Permite acceder al usuario a la sección de “ Ayuda ”, donde puede obtener información general sobre la aplicación.	
Observaciones: Una vez seleccionado el botón de “ Ayuda ” se muestra un panel que brinda información general sobre la aplicación.	
Prototipo de Interfaz: Ayuda	



Tabla 8. Historia de Usuario #6

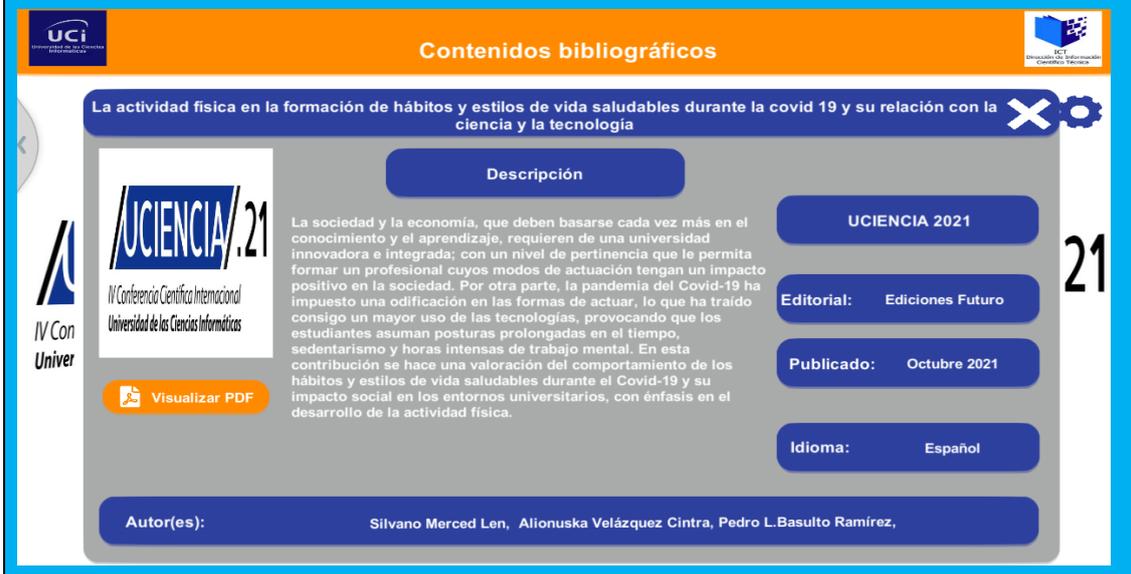
Historia de Usuario	
Número:6	Nombre: Generar reconocedor de gestos
Usuario: No aplica	
Prioridad en negocio: Alta	Riesgo en desarrollo: Medio
Puntos estimados: 3	Iteración asignada: 3
Programador responsable: Leudis Rafael Estrada González	
Descripción: Permite la integración de un reconocedor que facilita la interacción a través de gestos entre los usuarios y la aplicación, asistido por el empleo del sensor Kinect 2.	
Observaciones: El componente de interacción con Kinect 2 del que se hará uso en la aplicación será posible gracias a la integración de un reconocedor que interpretará las acciones de los usuarios.	
Prototipo de Interfaz: No aplica.	

Tabla 9. Historia de Usuario #7

Historia de Usuario	
Número:7	Nombre: Visualizar PDF
Usuario: Usuario	
Prioridad en negocio: Alta	Riesgo en desarrollo: Medio
Puntos estimados: 1	Iteración asignada: 3
Programador responsable: Leudis Rafael Estrada González	
Descripción: Permite la visualización del material digital del contenido seleccionado en formato pdf.	

Observaciones: El usuario selecciona el botón con el icono correspondiente a un archivo pdf y acto seguido, se abre el navegador del sistema, mostrando el material digital, si el usuario dispone de acceso a la red.

Prototipo de Interfaz: Visualizar PDF



2.2.2 Estimación del esfuerzo por Historia de Usuario

Se realiza la estimación de esfuerzo que arroja cada **HU**, de manera tal que se pueda obtener un correcto desarrollo del sistema. Finalmente, para una mayor organización se decide asignar a cada iteración el conjunto de historias agrupadas en correspondencia con el módulo al cual representan. A continuación, se muestra la estimación realizada.

Tabla 10. Estimación de esfuerzo por Historia de Usuario

Historias de Usuario		Puntos estimados(semanas)
1	Mostrar vista principal	1.5
2	Mostrar vista de contenidos	2
3	Mostrar información de contenidos	1
4	Controlar opciones	2
5	Consultar ayuda	1
6	Generar reconocedor de gestos	3
7	Visualizar PDF	1

2.2.3 Requisitos no funcionales

Se definen los requisitos no funcionales de la propuesta de solución como una serie de cualidades, restricciones y características [36] que limitan las **HU** tratadas con anterioridad.

RnF 1: Jugabilidad.

- **RnF 1.1:** el usuario se desplaza e interactúa con el entorno virtual, haciendo uso del ratón o mediante el empleo del sensor de gestos **Kinect 2**.
- **RnF 1.2:** el usuario podrá acceder a cualquier escena que desee sin restricción alguna para ello.

RnF 2: Usabilidad.

- **RnF 2.1:** el catálogo virtual interactivo está diseñado en función del aprendizaje, y puede ser empleado por todas las personas que accedan a la biblioteca de la universidad.
- **RnF 2.2:** el tiempo para la transición entre las diferentes acciones no excederá los tres segundos.
- **RnF 2.3:** para hacer uso del catálogo no se requieren conocimientos técnicos avanzados sobre el área de la informática o la educación en general.

RnF 3: Apariencia o Interfaz externa.

- **RnF 3.1:** en las diferentes interfaces predominan los colores: azul oscuro, blanco y naranja propios del sitio web de la biblioteca y la universidad.
- **RnF 3.2:** el tamaño de la letra oscila entre los 18 y los 28 puntos; mientras que la tipografía usada comprende los tipos: Arial, Arial-Bold y Caladea-Regular.
- **RnF 3.3:** la distribución de los botones debe proporcionar consistencia en las interfaces.

RnF 4: Software.

- **RnF 4.1:** la aplicación está disponible para el sistema operativo **Windows** a partir de su versión **8.0**.
- **RnF 4.2:** se requiere la instalación del **Kit de Desarrollo de Software 2.0 de Windows para Kinect**, a fin de lograr un correcto funcionamiento de este dispositivo en la aplicación.
- **RnF 4.3:** se requiere la instalación de los **drivers** (controladores) de video y sonido del ordenador en el que se despliegue la aplicación, para obtener un mejor rendimiento de esta durante su ejecución.

RnF 5: Hardware.

- **RnF 5.1:** es necesario, para el correcto funcionamiento del sistema como mínimo: un procesador **Intel(R) Core (TM) i3-3120M CPU @2.50GHz (4 CPUs)**, ~2.5GHz, con **4GB** de memoria **RAM**, tarjeta de video **Intel(R) HD Graphics 4000**, así como unos altavoces.
- **RnF 5.2:** el empleo del dispositivo **Kinect 2** puede ser opcional (en caso de que no se cuente con uno), pues la aplicación puede funcionar de manera convencional como cualquier otra.

RnF 6: Herramientas.

- **RnF 6.1:** librerías: **Kinect for Windows SDK 2.0**.
- **RnF 6.2:** motor gráfico: **Unity 3D** en su versión (2020.2.7f1).
- **RnF 6.3:** editor de código: **Visual Studio Code** en su versión (1.72.2).
- **RnF 6.4:** sensor de gestos: **Kinect 2**.
- **RnF 6.5:** lenguaje de programación: **C#** en su versión (10.0).
- **RnF 6.6:** herramientas **CASE: Visual Paradigm** en su versión (15.1).

2.2.4 Desarrollo del plan de iteraciones

Una vez definidas las **HU** y realizada la estimación de esfuerzos, se procede a planificar la etapa de implementación del sistema. En este espacio se crea el plan de iteraciones, que se encarga de especificar el nivel de prioridad con el que se desarrollarán las **HU** organizadas por iteraciones. Teniendo en cuenta el esfuerzo asociado a las **HU** y a las prioridades del cliente, se define una versión que sea de valor para este.

2.2.5 Plan de duración de las iteraciones

Se muestra el plan de duración de las iteraciones correspondiente, cuyo fin es mostrar la duración de cada iteración y el orden en que serán implementadas las **HU** en cada nueva iteración, lo cual se muestra en la tabla siguiente:

Tabla 11. Plan de duración de las iteraciones

Iteración	Historias de Usuario	Duración(semanas)
1	Mostrar vista principal	4.5
	Mostrar vista de contenidos	
	Mostrar información de contenidos	
2	Controlar opciones	3
	Consultar ayuda	

3	Generar reconocedor de gestos	4
	Visualizar PDF	
Total	Semanas	11.5

2.2.6 Plan de entregas

El plan de entrega mostrado a continuación, hace una propuesta de las versiones (**releases**) del sistema, por lo que cada versión será generada al finalizar una iteración.

Tabla 12. Plan de entrega de versiones

Entregable	Iteración 1	Iteración 2	Iteración 3
Módulos	Versión 0.1	Versión 0.1.2	Versión 0.1.3
Fecha	02/06/2022	23/06/2022	20/07/2022

2.3 Fase II: Diseño del Sistema

Partiendo del estudio realizado, se propone una arquitectura basada en los principios del patrón arquitectónico: arquitectura basada en capas, con componentes ordenados dentro de esta, de manera tal que queden estructurados los elementos necesarios para una aplicación de plataformas de este tipo. Los componentes de cada capa se comunican con aquellos que aparecen en otras capas a través de interfaces definidas o instancias de clases. La imagen siguiente muestra la distribución de las capas presentes en la arquitectura propuesta.

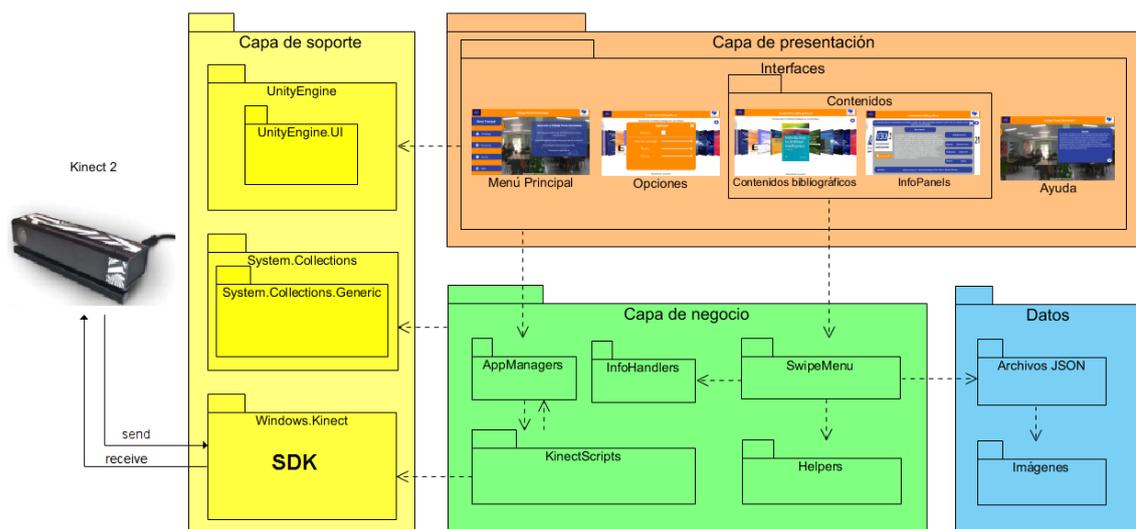


Ilustración 11. Arquitectura del catálogo virtual interactivo.

Según Pressman, desde un punto de vista orientado a objetos, un componente es un conjunto de clases que colaboran, por lo que en cada **script** de la arquitectura propuesta

se definen las clases requeridas para identificarlo como un componente. De ahí que las clases se definan para cada uno de estos componentes y se confeccionen para que incluyan todos los métodos y acciones relevantes para su implementación.

El análisis anterior permite elaborar los detalles de cada uno de los componentes utilizados para que puedan brindar información suficiente en el momento de la implementación. Teniendo en cuenta su reutilización, se describen las funciones que realizan, sus interfaces, así como la comunicación que requieren y la colaboración que establecen entre sí [29].

Como parte de la solución, se determinan todas las interfaces que permiten a las clases comunicarse y colaborar entre sí, dependiendo de la capa donde se encuentren. Las capas se definen de la siguiente manera:

Capa de Presentación: En esta capa se encuentran los diferentes escenarios, así como los materiales bibliográficos y sus contenidos asociados.

Capa de Negocio: Está formada por los principales componentes del sistema, entre los que se encuentran: el grupo de controladores principales de la aplicación **AppManagers**, para el trabajo con el sonido (**Sound Manager**), los gestos (**KinectManager**), las escenas (**LevelManager**) y las opciones (**AppSettingsManager**).

- **SoundManager:** maneja el sonido del sistema, controlando todos los elementos o acciones que generen un evento de su tipo.
- **LevelManager:** da tratamiento a las transiciones que ocurren entre las escenas de la aplicación.
- **KinectManager:** es el eslabón que permite la interacción con gestos mediante el empleo del sensor **Kinect 2**. Utiliza el paquete **KinectScripts** para aprovechar las funcionalidades propias de la **SDK** y mantener la comunicación en tiempo real con el dispositivo.
- **AppSettingsManager:** maneja los estados de configuración en la aplicación, salvando y cargando los ajustes realizados por el usuario.

En esta capa se encuentran, además, los controladores de animación y actualización de las posiciones de los elementos dentro del catálogo (**SwipeMenu**), que trabajan en conjunto con los **Helpers** para garantizar esta función. También emplean los

manejadores (**InfoHandlers**) para procesar los datos que son recibidos de los archivos en formato **JSON**²⁶ y alistarlos para cuando se realicen las consultas de información.

Capa de Soporte: En esta capa se encuentran las librerías “**System.Collection**” y “**System.Collection.Generic**”, con características propias del lenguaje C#. Las librerías “**UnityEngine**” y sus derivadas, que contienen funciones de **Unity**. Se emplea, además, la biblioteca “**Windows.Kinect**” que aprovecha los recursos de la **SDK** instalada, facilitando la comunicación con el dispositivo y el trabajo con gestos en la solución.

Datos: En esta capa se encuentran los archivos en formato **JSON** con los datos de los diferentes materiales bibliográficos y su correspondiente repositorio de imágenes empleadas en las portadas.

2.4 Patrones de diseño

Los patrones de diseño se encargan de dar solución a los problemas repetitivos que se presentan en situaciones particulares durante el diseño de software. Identifican clases, instancias, roles, y las colaboraciones que ocurren entre estas, por lo que los desarrolladores los usan como una forma para reutilizar la experiencia, clasificando las soluciones con términos de común denominación que les permitan formar un amplio repertorio de principios generales y expresiones; guiándolos durante el proceso de creación [28]. De ahí que estos patrones en su conjunto representen una descripción de las clases y objetos que se comunican entre sí, adaptada para resolver un problema de diseño general en un contexto particular [37].

El diseño de la solución hizo posible identificar algunos de los **Patrones Generales de Asignación de Responsabilidades (GRASP)**, por sus siglas en inglés) así como determinados patrones de la **Pandilla de los Cuatro (GOF)**, por sus siglas en inglés). Dichos elementos se describen continuación:

1. Patrones GRASP

- **Experto:** clase que tiene la responsabilidad de ejecutar una acción determinada y cuenta con el acceso a los datos necesarios para ello. Aparece reflejado en la interfaz que muestra el listado de elementos del catálogo. Esta clase busca toda la información que debe ser visible a la hora de mostrar los contenidos, obteniendo datos como el título, la descripción, el idioma, o el código ISBN por cada uno de los elementos seleccionados respectivamente. Lo anterior contribuye a la

²⁶ Acrónimo de Javascript Object Notation o Notación de Objeto de Javascript. Constituye un formato de texto sencillo para el intercambio de datos.

existencia de un “**Bajo Acoplamiento**” y una “**Alta Cohesión**” porque cada clase tiene asignada una única responsabilidad y no existe dependencia entre ninguna de ellas. Como ejemplo de lo anterior se tiene a la clase **BookSelector**.

- **Creador:** se encarga de crear instancias de una clase o de un objeto. Se implementa en las clases encargadas de gestionar los aspectos principales de la aplicación, permitiendo que se creen objetos de otras clases asociadas a ellas. Por ejemplo: clases **Menu**, **MenuItemController** entre otras.
- **Alta Cohesión:** aplicado a clases que realizan pocas funciones.
- **Bajo Acoplamiento:** típico de las clases que no dependen de muchas otras.
- **Controlador:** clase que se encarga de manejar los objetos del sistema. Aplicado en la clase **SceneManager** para tratar la gestión de todos los eventos del sistema.

2. Patrones GOF

- **Creacionales:** los patrones creacionales tratan con las formas de crear instancias de objetos. Su objetivo es abstraer el proceso de instanciación y ocultar los detalles de cómo los objetos son creados o inicializados. Ejemplos de estos son los siguientes (**Fábrica abstracta, Constructor, Método de fabricación, Prototipo, Instancia única**).
- **Estructurales:** los patrones estructurales describen como las clases y objetos pueden ser combinados para formar grandes estructuras y proporcionar nuevas funcionalidades. Estos objetos adicionales pueden ser incluso objetos simples u objetos compuestos. Ejemplos de estos son (**Adaptador, Puente, Compuesto, Decorador, Fachada, Peso ligero**).
- **Comportamiento:** los patrones de comportamiento ayudan a definir la comunicación e iteración entre los objetos de un sistema. El propósito de este patrón es reducir el acoplamiento entre los objetos. Ejemplo de estos patrones son (**Cadena de responsabilidad, Orden, Intérprete, Iterador, Mediador, Observador, Estado, Estrategia, Método plantilla, Visitante**).

De los patrones **GOF** anteriormente mencionados, sobresalen por su empleo en la solución los siguientes:

- **Observador:** este patrón define una dependencia de uno a muchos entre objetos, de modo que cuando el estado de un objeto cambie, se les notifica el cambio a todos los que dependen de él y se actualizan de forma automática. En la solución se usará para notificar a las clases que empleen este patrón, como **SwipeHandler**, que deben actualizarse. Un ejemplo de su uso vendría apoyado por la actualización de las posiciones de los elementos

(**MenuItems**) dentro del catálogo, en el mismo instante en que ocurre el evento **Drag** (arrastrar) en la interacción.

- **Fachada (Facade)**: es el que eleva el nivel de abstracción de un determinado sistema para ocultar ciertos detalles de implementación y hacer más sencillo su uso. Provee una interfaz a una serie de clases para evitar el exceso de instancias de las mismas, dejando a las subclases accesibles para ser usadas directamente. Se implementa en la clase **Menu**, encargada de gestionar los aspectos principales de los **MenuItems** del catálogo, permitiendo el acceso al resto de las clases asociadas a ella o convirtiendo al **Menu** en la fachada para todos los **MenuItems** en la capa de presentación de la solución.
- **Instancia única (Singleton)**: el patrón de diseño singleton se asegura de que una clase solo tenga una instancia, y provee un punto de acceso global a ella. Para la propuesta de solución se usó este patrón en la clase **AppManager**, dado que dicha clase es la controladora de la mayoría de los aspectos del catálogo, por lo que para optimizar código y evitar sobrecarga de recursos cargando demasiadas instancias de la misma clase se necesita un único acceso a la misma.

Una vez definidos los patrones de diseño, se describen las clases utilizadas en la solución. Siguiendo la metodología **XP**, se analizan las **HU** y luego se descomponen en tareas independientes.

2.5 Tarjetas CRC

En este apartado, las **HU** son evaluadas y divididas en tareas, cada una de las cuales representa una característica del sistema, permitiendo el diseño de una prueba de unidad a modo de verificación. Estas tareas se representan mediante las tarjetas **Clase-Responsabilidad-Colaborador (CRC)**.

Tabla 13. Prototipo de tarjeta CRC

Nombre de la clase	
Responsabilidades	Colaboradores

Como se puede observar en la tabla anterior, cada tarjeta contiene información del nombre de la clase, la descripción de las responsabilidades o métodos asociados a la esta, así como la lista de las clases con las que se relaciona o que colaboran con ella. A continuación, se describen las tarjetas definidas para la implementación de la solución. El resto puede ser consultado en los [anexos](#) de la investigación.

Tabla 14. Tarjeta CRC # 1

Clase: <i>Menu</i>	
Responsabilidad	Colaboradores
<p>La clase del menú principal. Maneja la animación y la actualización de las posiciones de los elementos dentro del catálogo, su ángulo respectivo y la distancia entre ellos.</p> <ul style="list-style-type: none"> ➤ Awake(): Comprobaciones previas al inicio de la animación. ➤ Update(): Acciones realizadas en cada instante de tiempo en que esté activa la aplicación. ➤ MoveLeftRightByAmount(): Configura el movimiento para todos los elementos en conjunto dentro del menú de selección del catálogo, ya sea hacia la derecha o hacia la izquierda. ➤ UpdateMenuItemsPositionInWorldSpace(): Actualiza la posición de los elementos del menú en el espacio tridimensional de la escena. ➤ LocktoClosest(): Encuentra el elemento del menú más cercano al centro y lo desplaza para que ocupe dicha posición. ➤ HideMenus(): Deshabilita todos los elementos del menú. ➤ ShowMenus(): Habilita todos los elementos del menú. ➤ ActivateSelectedMenuItem(item): Invoca al evento OnClick para el elemento del menú especificado. ➤ MenuCentred(item): Identifica si el elemento del menú especificado está centrado o no. ➤ GetClosestMenuIndex(): Retorna el índice del menú más cercano al centro. ➤ CalculateMenuItemRotation(offsetx): Determina la rotación del elemento del menú correspondiente en ese momento. 	<p>MenuItem</p>

Tabla 15. Tarjeta CRC # 2

Clase: MenuItem	
Responsabilidad	Colaboradores
<p>Clase adjunta a cada uno de los elementos del catálogo. Define su comportamiento a la hora de ser seleccionado mediante algún tipo de interacción.</p> <ul style="list-style-type: none"> ➤ Update(): Se encarga de mostrar el título del elemento del catálogo que esté centrado en un instante de tiempo determinado, mientras la aplicación está activa 	<p>Menu</p> <p>Booklist</p>

Tabla 16. Tarjeta CRC # 3

Clase: SliderMenuAnimation	
Responsabilidad	Colaboradores
<p>Clase adjunta a objetos del tipo Menu de la aplicación.</p> <ul style="list-style-type: none"> ➤ ShowHideMenu(): Maneja la animación del panel del menú de manera tal que pueda mostrarse u ocultarse según la interacción del usuario con el botón correspondiente. ➤ CloseMenu(): Cierra el panel del menú. ➤ OpenOptionsPanel(): Abre el panel de "Opciones". ➤ CloseOptionsPanel(): Cierra el panel de "Opciones". ➤ OpenInfoPanel(): Abre el panel informativo correspondiente a los contenidos del catálogo. ➤ CloseInfoPanel(): Cierra el panel informativo. 	<p>No aplica</p>

Tabla 17. Tarjeta CRC # 4

Clase: SwipeHandler	
Responsabilidad	Colaboradores
<p>Clase adjunta a objetos del tipo Menu de la aplicación. Maneja el deslizamiento y el desplazamiento. Incluye soporte para el ratón y el móvil. También controla parámetros importantes de la animación de los ítems del</p>	<p>Menu</p> <p>MenuItem</p>

<p>menú como la fuerza máxima permitida para la interacción y la inercia, entre otras.</p> <ul style="list-style-type: none"> ➤ isSwiping (): Identifica si un elemento está o no moviéndose adecuadamente. ➤ HandleMouseSwipe(): Define el comportamiento de los desplazamientos, luego de la interacción con el ratón de la computadora. 	
--	--

Tabla 18. Tarjeta CRC # 5

Clase: MenuCatalogController	
Responsabilidad	Colaboradores
<p>Permite la navegación por las diferentes áreas de contenidos dentro del catálogo.</p> <ul style="list-style-type: none"> ➤ newMenu(newMenu): Verifica los índices de menú, y si son diferentes, realiza el cambio. ➤ menuChange(newPage): Realiza el cambio de menú. 	<p>Menu</p>

Tabla 19. Tarjeta CRC # 6

Clase: BookSelector	
Responsabilidad	Colaboradores
<p>Actualiza la información en la sección “Libros” del catálogo de contenidos. Una vez que es seleccionado un elemento de este tipo, se muestra la misma en el panel informativo correspondiente.</p> <ul style="list-style-type: none"> ➤ SelectedTitle(): Envía la información del título del libro seleccionado para mostrarse en el panel informativo correspondiente. ➤ SelectedPages(): Envía la información del no. de páginas del libro seleccionado para mostrarse en el panel informativo correspondiente. ➤ SelectedISBN(): Envía la información del código ISBN del libro seleccionado para que se muestre en el panel informativo correspondiente. 	<p>MenuItem</p>

<ul style="list-style-type: none"> ➤ SelectedShortDescription(): Envía la información de la descripción del libro seleccionado para que se muestre en el panel informativo correspondiente. ➤ SelectedEditorial(): Envía la información de la editorial del libro seleccionado para que se muestre en el panel informativo correspondiente. ➤ SelectedPublishedDate(): Envía la información de la fecha de publicación del libro seleccionado para que se muestre en el panel informativo correspondiente. ➤ SelectedAuthors(): Envía la información del autor(es) del libro seleccionado para que se muestre en el panel informativo correspondiente. ➤ SelectedCategories(): Envía la información de las categorías a las que pertenece el libro seleccionado para que se muestre en el panel informativo correspondiente. ➤ SelectedIdiom(): Envía la información del idioma que posee el libro seleccionado para que se muestre en el panel informativo correspondiente. 	
--	--

Tabla 20. Tarjeta CRC # 7

Clase: SerieCSelector	
Responsabilidad	Colaboradores
<p>Actualiza la información en la sección “Serie Científica” del catálogo de contenidos. Una vez que es seleccionado un elemento de este tipo, se muestra la misma en el panel informativo correspondiente.</p> <ul style="list-style-type: none"> ➤ SelectedTitle(): Envía la información del título del artículo seleccionado para mostrarse en el panel informativo correspondiente. ➤ SelectedVolumen(): Envía la información del volumen al que pertenece el artículo seleccionado en la revista científica para mostrarse en el panel informativo correspondiente. 	<p>MenuItem</p>

<ul style="list-style-type: none"> ➤ SelectedShortDescription(): Envía la información de la descripción del artículo seleccionado para que se muestre en el panel informativo correspondiente. ➤ SelectedEditorial(): Envía la información de la editorial del artículo seleccionado para que se muestre en el panel informativo correspondiente. ➤ SelectedPublishedDate(): Envía la información de la fecha de publicación del artículo seleccionado para que se muestre en el panel informativo correspondiente. ➤ SelectedAuthors(): Envía la información del autor(es) del artículo seleccionado para que se muestre en el panel informativo correspondiente. ➤ SelectedIdiom(): Envía la información del idioma que posee el artículo seleccionado para que se muestre en el panel informativo correspondiente. ➤ SelectedPDFURL(): Envía la información con la dirección del enlace en el que se encuentra el formato digital del artículo seleccionado para que sea accesible a través del panel informativo correspondiente. 	
---	--

Tabla 21. Tarjeta CRC # 8

Clase: <i>HelpManager</i>	
Responsabilidad	Colaboradores
<p>Se encarga de manejar la información dispuesta en la sección “Ayuda” del catálogo de contenidos.</p> <ul style="list-style-type: none"> ➤ Start(): Se llama a este método antes de la actualización del primer fotograma. Inicializa la cola de sentencias. ➤ StartHelpDialogue(dialogue): Inicia la descripción de la información que se presenta en el panel de “Ayuda” del catálogo. ➤ DisplayNextSentence(): Verifica si existen sentencias disponibles o no para mostrar en el 	<p><i>HelpDialogue</i> <i>HelpDialogue_Trigger</i></p>

<p>panel y realiza la acción correspondiente, mostrándolas o terminando esta operación.</p> <ul style="list-style-type: none"> ➤ TypeSentence(): Maneja la salida de las sentencias en el panel. ➤ EndDialogue(): Cierra el panel de “Ayuda” del catálogo. 	
--	--

Tabla 22. Tarjeta CRC # 9

Clase: LevelManager (Singleton)	
Responsabilidad	Colaboradores
<p>Facilita el cambio de escenas en la aplicación. Maneja el cambio de nombre de escena y el efecto de Fade que se lleva a cabo durante la transición de estas.</p> <ul style="list-style-type: none"> ➤ Start(): Verifica que existe la instancia requerida de GUIManager para actualizar el nombre de la escena activa y producir el efecto de FadeIn durante la transición de escenas. ➤ GoToLevel(levelname): Dirige al usuario hacia la escena deseada. ➤ GoToLevelCo(levelname): Espera un pequeño periodo de tiempo, luego realiza el cambio hacia la escena seleccionada. 	<p>GUIManager GameManager</p>

Conclusiones parciales del capítulo

En este capítulo se han abordado los aspectos referentes a la concepción del producto a desarrollar y sus características funcionales, lo cual permitió arribar a las siguientes conclusiones:

- A partir de la definición de las **HU** se pudieron identificar las principales funcionalidades a desarrollar en correspondencia con los módulos: **Catálogo de Contenidos, Gestión de Contenidos y Reconocimiento de Gestos**.
- La estimación del tiempo para la implementación de las **HU** definidas, permitió calcular una entrega final del producto en tres meses.
- La arquitectura en tres capas propició una buena organización a la hora de implementar los componentes y facilitó la integración del sistema con el sensor de gestos **Kinect 2**.
- Fue posible identificar los patrones **GRASP** y **GOF** para el desarrollo de la solución.
- Se exponen los artefactos generados en las fases de Planificación y Diseño que establece la metodología **XP**.

Con los aspectos arquitectónicos y de diseño a utilizar definidos, queda establecida la vía para llevar a cabo la implementación y prueba del catálogo virtual interactivo para la visualización de contenidos bibliográficos con **Kinect 2**.

Capítulo 3. Implementación y pruebas del sistema

Introducción

En este capítulo se detallan las iteraciones realizadas durante la etapa de construcción de la solución propuesta. Además, se exponen las tareas de ingeniería generadas para cada **HU** que fueron definidas, así como las pruebas de aceptación planificadas para el sistema. De esta forma se obtiene un producto funcional probado y listo para entregar al cliente al final de cada iteración, de acuerdo con lo propuesto por la metodología **XP**.

3.1 Fase III: Desarrollo

Durante esta fase, **XP** determina que las **HU** seleccionadas para ser implementadas se realizan durante el transcurso de la iteración a la que pertenecen. De acuerdo con lo anterior, se lleva a cabo una revisión del plan de iteraciones y se modifican en caso de ser necesario. Como parte de este plan se descomponen las **HU** en **tareas de ingeniería** [25].

3.1.1 Tareas de Ingeniería

Las tareas de ingeniería se pueden describir mediante un lenguaje técnico y no necesariamente entendible por el cliente. Su objetivo consiste definir cada una de las actividades que dan cumplimiento a las **HU**, haciendo más entendible las funciones del sistema y facilitando su construcción. Para una mayor organización, se definen en correspondencia con las iteraciones definidas como se muestra a continuación.

Tareas de ingeniería para la Iteración I

Para la primera iteración, se definieron un total de 13 tareas de ingeniería, correspondientes con las **HU** propias de esta iteración: **Mostrar vista principal**, **Mostrar vista de contenidos** y **Mostrar información de contenidos**. A continuación, se describen algunas de las tareas realizadas en esta iteración. El resto puede ser consultado en los **anexos** de la investigación.

HU Mostrar vista principal

Tabla 23. Tarea de Ingeniería #1

Tarea	
Número de Tarea:1	Número de Historia de Usuario:1

Nombre de la tarea: Desarrollar vista principal de la aplicación.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.2
Fecha de inicio: 25 de abril del 2022	Fecha de fin: 26 de abril del 2022
Programador responsable: Leudis Rafael Estrada González	
Descripción: Se desarrolla una interfaz que permite mostrar al usuario la vista principal de la aplicación.	

Tabla 24. Tarea de Ingeniería #2

Tarea	
Número de Tarea: 2	Número de Historia de Usuario: 1
Nombre de la tarea: Desarrollar menú principal desplegable.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.5
Fecha de inicio: 26 de abril del 2022	Fecha de fin: 29 de abril del 2022
Programador responsable: Leudis Rafael Estrada González	
Descripción: Se desarrolla un panel que permite mostrar al usuario el menú en forma de desplegable en la vista principal de la aplicación. Permite el acceso a las secciones: Catálogo, Opciones, Ayuda o Salir de la aplicación. Integra un componente de animación para el despliegue.	

HU Mostrar vista de contenidos

Tabla 25. Tarea de Ingeniería #5

Tarea	
Número de Tarea: 5	Número de Historia de Usuario: 2
Nombre de la tarea: Desarrollar vista para mostrar los contenidos bibliográficos.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.1
Fecha de inicio: 5 de mayo del 2022	Fecha de fin: 6 de mayo del 2022
Programador responsable: Leudis Rafael Estrada González	
Descripción: Se desarrolla una interfaz que muestra los contenidos bibliográficos, así como su información asociada, por área de interés; volviéndolos disponibles en todo momento para las consultas de información por parte de los de usuarios del sistema	

Tabla 26. Tarea de Ingeniería #6

Tarea	
Número de Tarea: 6	Número de Historia de Usuario: 2
Nombre de la tarea: Implementar funcionalidad para cargar la información en formato JSON.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.4
Fecha de inicio: 9 de mayo del 2022	Fecha de fin: 11 de mayo del 2022
Programador responsable: Leudis Rafael Estrada González	

Descripción: Se implementa una funcionalidad que carga los datos del archivo en formato JSON, volviéndolos disponibles para su posterior uso en las consultas de información por parte de los usuarios del sistema.

HU Mostrar información de contenidos

Tabla 27. Tarea de Ingeniería #12

Tarea	
Número de Tarea: 12	Número de Historia de Usuario: 3
Nombre de la tarea: Implementar funcionalidad para generar información a partir de los datos en formato JSON.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.5
Fecha de inicio: 26 de mayo del 2022	Fecha de fin: 28 de mayo del 2022
Programador responsable: Leudis Rafael Estrada González	
Descripción: Se implementa una funcionalidad que transforma los datos del archivo en formato JSON a información disponible para cada uno de los materiales del catálogo.	

Tareas de ingeniería para la Iteración II

En una segunda iteración, se elaboraron un total de 4 tareas de ingeniería. La mayoría, desglosadas a partir de las **HU** correspondientes a: **Controlar opciones** y **Consultar ayuda**. Algunas se describen a continuación. El resto puede ser consultado en los [anexos](#) de la investigación.

HU Controlar opciones

Tabla 28. Tarea de Ingeniería #14

Tarea	
Número de Tarea: 14	Número de Historia de Usuario: 4
Nombre de la tarea: Desarrollar vista para controlar las opciones de configuración de la aplicación.	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha de inicio: 2 de junio del 2022	Fecha de fin: 8 de junio del 2022
Programador responsable: Leudis Rafael Estrada González	
Descripción: Se desarrolla una interfaz que permite al usuario configurar las opciones de interacción o sonido presentes en la aplicación. Permite activar o desactivar la interacción con Kinect 2 , pasando de gestual a manual y viceversa. En cuanto al sonido, este puede ser regulado a conveniencia del usuario. Integra un componente de animación para el despliegue	

HU Consultar ayuda

Tabla 29. Tarea de Ingeniería #16

Tarea	
Número de Tarea:16	Número de Historia de Usuario:5
Nombre de la tarea: Desarrollar vista para acceder a la sección de “Ayuda” de la aplicación.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.5
Fecha de inicio: 16 de junio del 2022	Fecha de fin: 20 de junio del 2022
Programador responsable: Leudis Rafael Estrada González	
Descripción: Se desarrolla una interfaz que permite al usuario acceder a la sección de “Ayuda”, donde puede obtener información general sobre la aplicación.	

Tareas de ingeniería para la Iteración III

Para la tercera y última iteración, se definieron un total de 5 tareas de ingeniería. Todas, desglosadas a partir de las **HU** correspondientes a: **Generar reconocedor de gestos** y **Visualizar PDF**, algunas de las cuales se describen a continuación. El resto puede ser consultado en los [anexos](#) de la investigación.

HU Generar reconocedor de gestos

Tabla 30. Tarea de Ingeniería #18

Tarea	
Número de Tarea:18	Número de Historia de Usuario:6
Nombre de la tarea: Integrar complemento (plugin) para el trabajo con el sensor de gestos Kinect 2 en Unity.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.5
Fecha de inicio: 24 de junio del 2022	Fecha de fin: 28 de junio del 2022
Programador responsable: Leudis Rafael Estrada González	
Descripción: Se integra un plugin a la aplicación que permite el trabajo con el sensor de gestos Kinect 2 , directamente desde el motor gráfico Unity.	

HU Visualizar PDF

Tabla 31. Tarea de Ingeniería #21

Tarea	
Número de Tarea:21	Número de Historia de Usuario:7
Nombre de la tarea: Desarrollar interfaz para acceder al material digital del contenido seleccionado en formato PDF.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.4

Fecha de inicio: 14 de julio del 2022	Fecha de fin: 18 de julio del 2022
Programador responsable: Leudis Rafael Estrada González	
Descripción: Se desarrolla una interfaz que permite al usuario acceder al material digital del contenido seleccionado en formato PDF.	

Con las tareas de ingeniería definidas, se hace necesario establecer un grupo de pruebas que comprueben la calidad de la solución implementada. Luego, se analizan estos casos de prueba y se ejecutan, lo que permite medir el nivel de cumplimiento de acuerdo a los objetivos de implementación trazados y el nivel de satisfacción del cliente, según lo propuesto por la metodología **XP**.

3.2 Fase IV: Pruebas

Las pruebas consisten en un grupo de actividades que pueden planificarse de forma adelantada y realizarse sistemáticamente. Por esta razón, es necesario definir las en el proceso de la ingeniería del software. Esto contribuye a elevar la calidad del producto desarrollado y aumenta la seguridad de los desarrolladores a la hora de introducir cambios o modificaciones.

La metodología **XP** segmenta las pruebas en dos grupos: **pruebas unitarias**, diseñadas por los programadores para verificar el código de forma automática, y las **pruebas de aceptación**, destinadas para evaluar si al final de una iteración se obtuvo como salida la funcionalidad requerida, además de comprobar que dicha funcionalidad sea la esperada por el cliente [26].

Se procede a realizar ambas (pruebas unitarias y pruebas de aceptación) al catálogo digital implementado para verificar que el sistema cumpla con los requisitos establecidos por el usuario. Esto permite obtener el grado de satisfacción del cliente y, por consecuencia el índice de éxito de la solución.

3.2.1 Pruebas unitarias

En la medida en la que un proyecto de software crece, y con ello el número de *scripts*, clases y métodos asociados, puede ser difícil asegurar que un cambio en una parte del código no cree problemas en otras secciones. Las pruebas automatizadas pueden ayudar a comprobar que todas las partes del código funcionen como se esperaba. También ahorran tiempo al identificar los problemas tan pronto como se producen durante el desarrollo, en lugar de depender de las pruebas manuales, o incluso peor, de los informes de usuarios finales [38].

El paquete **Unity Test Framework** (antes "**Unity Test Runner**") es una herramienta de **Unity** que permite probar el código tanto en modo **Edición** como en modo **Juego**,

para cualquier tipo de plataformas de destino ya sean **Standalone**, **Android** o **iOS**. A continuación, se muestra una representación de las pruebas unitarias que se realizaron para comprobar el correcto funcionamiento del código en la aplicación. La salida generalizada de este proceso puede ser consultada en los [anexos](#) de la investigación.

Modo Editor

```
Test > EditMode > C# Test2Managers.cs
You, 2 weeks ago | 1 author (You)
using System.Collections;
using System.Collections.Generic;
using NUnit.Framework;
using UnityEngine;
using UnityEngine.TestTools;

public class Test2Managers {

    AppManager manager = new AppManager ();

    // A Test behaves as an ordinary method
    [Test]
    public void Test2AppManagerSetTimeScale () {

        manager.SetTimeScale (0.75f);

    }

    [Test]
    public void Test2AppManagerResetTimeScale () {

        manager.ResetTimeScale ();

    }

}
```

Ilustración 12. Fragmento de la prueba unitaria para la clase *AppManager* (Modo Edición).

Las pruebas realizadas en el modo **Edición**, permitieron verificar la calidad del código cuyo manejo de la información no precisaba la ejecución de las escenas en el Editor. Este proceso fue aplicado a clases como: **AppManager**, **SoundManager** y **AppTools**, las cuales no se relacionaban con **MonoBehaviour**²⁷ y podrían comprobarse sin necesidad de ejecutar la aplicación. Como resultado, se comprobó que no existían errores en tiempo de ejecución, con lo que todos los casos de prueba resultaron satisfactorios.

²⁷ MonoBehaviour es la clase base de Unity de la que derivan todos los scripts. Contiene un grupo de funciones que se llaman automáticamente durante la ejecución de la aplicación.

Modo Juego

```
You, 2 weeks ago | 1 author (You)
using System.Collections;
using System.Collections.Generic;
using NUnit.Framework;
using UnityEngine;
using UnityEngine.TestTools;
using UnityEngine.UI;

0 references | You, 2 weeks ago | 1 author (You)
public class PlayModeBookSelector {

    // A UnityTest behaves like a coroutine in Play Mode. In Edit Mode you can use
    // `yield return null;` to skip a frame.
    [UnityTest]
    0 references
    public IEnumerator PlayM_BookSelector_SelectedTitle () {
        var menuItem = new GameObject ();
        menuItem.transform.name = "Title";
        menuItem.AddComponent<Text> ().text = "Unit Testing";
        var bookSelector = menuItem.AddComponent<BookSelector> ();
        You, 2 weeks ago • Test 2.0 ...
        bookSelector.GetComponent<Text> ().text = menuItem.GetComponent<Text> ().text

        // Use yield to skip a frame.
        yield return null;
    }
}
```

Ilustración 13. Fragmento de prueba unitaria para la clase BookSelector (Modo Juego)

Las pruebas realizadas en el modo **Juego**, facilitaron la comprobación del código que funcionaba durante la ejecución de las escenas de la aplicación. La mayoría de las clases del sistema fueron sometidas a pruebas de este tipo, debido a su relación directa con **MonoBehaviour** y, de acuerdo a los resultados obtenidos, fue necesario cambiar, así como refactorizar algunas funciones, principalmente aquellas asociadas al trabajo con el cálculo de las posiciones y las animaciones de los objetos en el espacio tridimensional. El resto de los casos de prueba resultaron satisfactorios.

3.2.2 Pruebas de aceptación

Las pruebas de aceptación son especificadas por el cliente, por lo que se centran en las características y funcionalidades generales del sistema que son visibles. Estas pruebas derivan de las **HU** que se han implementado como parte de la liberación del software. Una prueba de aceptación es como una caja negra, donde cada una de ellas representa una salida esperada del sistema [25]. Es responsabilidad del cliente verificar la corrección y la toma de decisiones acerca de estas pruebas [26]. A continuación, se muestra una representación de las pruebas de aceptación que se realizan en cada iteración.

Pruebas de aceptación para la Iteración I

Para la primera iteración, se definieron un total de 10 casos de pruebas de aceptación. Todas enfocadas a evaluar la implementación de las **HU: Mostrar vista principal, Mostrar vista de contenidos y Mostrar información de contenidos**. A continuación, se describen algunas de las pruebas realizadas, permitiéndole al resto ser consultado en los [anexos](#) de la investigación.

Tabla 32. Prueba de aceptación #1

Caso de prueba de aceptación	
Código: HU1_P1	Historia de Usuario: 1
Nombre: Mostrar vista principal de la aplicación.	
Descripción: Prueba para la funcionalidad: Desarrollar la vista principal de la aplicación.	
Condiciones de ejecución:	
<ul style="list-style-type: none"> ➤ El usuario debe estar disponible para la interacción con el sistema. ➤ El sistema debe estar en ejecución para cargar la vista principal de la aplicación. 	
Pasos de ejecución:	
<ol style="list-style-type: none"> 1. El usuario ejecuta la aplicación. 2. Se carga la aplicación. 3. Se muestra la vista principal de la aplicación. 	
Resultado: Satisfactorio.	

Tabla 33. Prueba de aceptación #2

Caso de prueba de aceptación	
Código: HU1_P2	Historia de Usuario: 1
Nombre: Mostrar menú principal desplegable.	
Descripción: Prueba para la funcionalidad: Desarrollar menú principal desplegable.	
Condiciones de ejecución:	
<ul style="list-style-type: none"> ➤ El usuario debe estar disponible para la interacción con el sistema. ➤ El sistema debe haber cargado la vista principal de la aplicación. 	
Pasos de ejecución:	
<ol style="list-style-type: none"> 1. El usuario selecciona la pestaña de despliegue para el menú 2. Se muestra el menú desplegable para acceder al resto de secciones de la aplicación. 	

Resultado: No satisfactorio.

Tabla 34. Prueba de aceptación #3

Caso de prueba de aceptación	
Código: HU1_P3	Historia de Usuario: 1
Nombre: Comprobar enlaces al resto de secciones de la aplicación.	
Descripción: Prueba para la funcionalidad: Desarrollar enlaces al resto de secciones de la aplicación.	
Condiciones de ejecución:	
<ul style="list-style-type: none"> ➤ El usuario debe estar disponible para la interacción con el sistema. ➤ El sistema debe haber cargado la vista principal de la aplicación. ➤ El sistema debe haber mostrado el menú principal desplegable. 	
Pasos de ejecución:	
<ol style="list-style-type: none"> 1. El usuario selecciona una de las secciones del menú desplegable. 2. Se carga la sección seleccionada. 3. Se muestra la sección seleccionada. 	
Resultado: No satisfactorio.	

Tabla 35. Prueba de aceptación #4

Caso de prueba de aceptación	
Código: HU1_P4	Historia de Usuario: 1
Nombre: Salir del sistema	
Descripción: Prueba para la funcionalidad: Implementar salida del sistema.	
Condiciones de ejecución:	
<ul style="list-style-type: none"> ➤ El usuario debe estar disponible para la interacción con el sistema. ➤ El sistema debe haber cargado la vista principal de la aplicación. ➤ El sistema debe haber mostrado el menú principal desplegable. 	
Pasos de ejecución:	
<ol style="list-style-type: none"> 1. El usuario selecciona la opción “Salir” del menú principal. 2. Se cierra la aplicación. 	
Resultado: Satisfactorio.	

Pruebas de aceptación para la Iteración II

En una segunda iteración, se diseñaron un total de 2 casos de pruebas de aceptación. Todas enfocadas a evaluar la implementación de las **HU: Controlar opciones** y **Consultar ayuda**. A continuación, se describen las pruebas realizadas. Véase el resto en los [anexos](#) de la investigación.

Tabla 36. Prueba de aceptación #11

Caso de prueba de aceptación	
Código:HU4_P11	Historia de Usuario:4
Nombre: Mostrar vista para controlar las opciones de configuración de la aplicación.	
Descripción: Prueba para la funcionalidad: Desarrollar vista para controlar las opciones de configuración de la aplicación.	
Condiciones de ejecución:	
<ul style="list-style-type: none">➤ El usuario debe estar disponible para la interacción con el sistema.➤ El sistema debe haber mostrado el menú principal desplegable.➤ El usuario debe haber seleccionado la opción “Opciones” en el menú principal.	
Pasos de ejecución:	
<ol style="list-style-type: none">1. El usuario selecciona la opción “Opciones” del menú desplegable.2. Se carga la interfaz correspondiente.3. Se muestra la interfaz seleccionada, en forma de panel; permitiéndole al usuario configurar las opciones de interacción o sonido presentes en la aplicación.	
Resultado: No satisfactorio.	

Pruebas de aceptación para la Iteración III

Para una tercera y última iteración, se confeccionó un 1 caso de prueba de aceptación. Este último, enfocado en evaluar la implementación de la **HU: Visualizar PDF**. Se describe a continuación.

Tabla 37. Prueba de aceptación #13

Caso de prueba de aceptación	
Código:HU7_P13	Historia de Usuario:4
Nombre: Mostrar vista para acceder al material digital del contenido seleccionado en formato PDF.	
Descripción: Prueba para la funcionalidad: Desarrollar interfaz para acceder al material digital del contenido seleccionado en formato PDF.	
Condiciones de ejecución: <ul style="list-style-type: none">➤ El usuario debe estar disponible para la interacción con el sistema.➤ El sistema debe haber mostrado el menú principal desplegable.➤ El usuario debe haber seleccionado la opción “Catálogo” en el menú principal para acceder a la vista que muestra los contenidos por área de interés.➤ El sistema debe haber mostrado la vista de contenidos.➤ El usuario debe haber seleccionado la portada del material correspondiente en la vista de contenidos.➤ El sistema debe haber mostrado el panel con la información asociada al material correspondiente en la vista de contenidos.	
Pasos de ejecución: <ol style="list-style-type: none">1. El usuario selecciona el botón con el icono correspondiente a la vista en formato PDF.2. Se carga el documento en el navegador del sistema predeterminado por el usuario.3. Se muestra la información correspondiente al material digital del contenido seleccionado en formato PDF.	
Resultado: Satisfactorio.	

3.2.3 Análisis de las pruebas de aceptación

Hasta el momento se han desarrollado un total de 13 casos de pruebas de aceptación. Las mismas se llevaron a cabo de forma organizada, por cada iteración definida. En las en gráficas mostradas a continuación, se recogen los porcentos de satisfacción alcanzados en cada iteración.

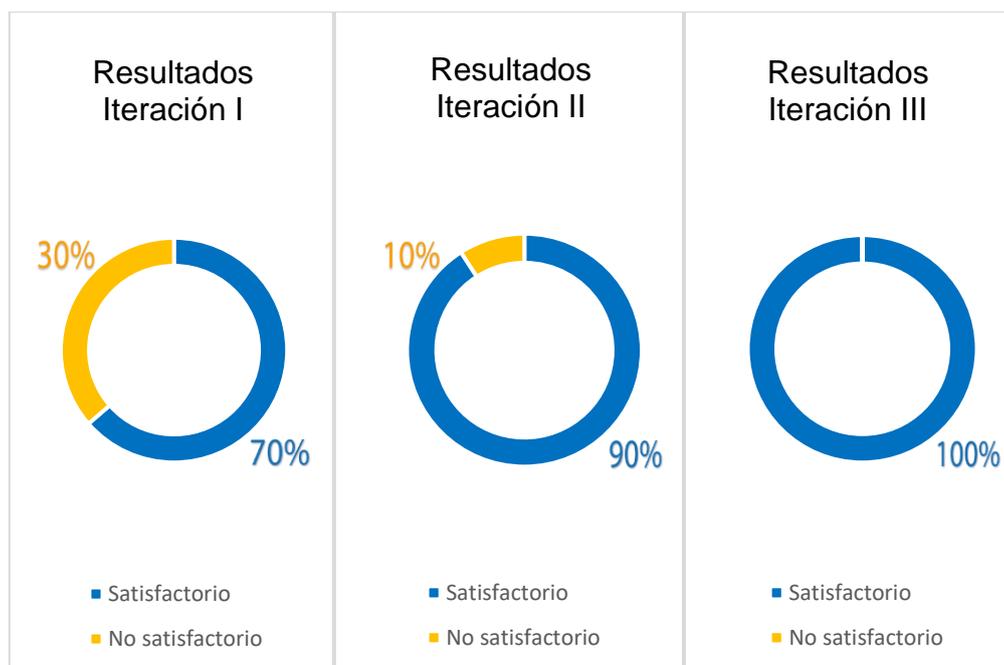


Ilustración 14. Resultados obtenidos en las pruebas de aceptación.

Como se puede observar en la representación anterior, en la primera iteración se realizaron un total de 10 pruebas, de ellas 7 alcanzaron el nivel de satisfacción esperado con un 70 %, mientras que en los 3 casos de prueba restantes: **Mostrar menú desplegable**, **Comprobar enlaces al resto de secciones de la aplicación**, y **Mostrar contenidos del catálogo por área de interés** existieron no conformidades. Lo anterior, permitió corregir las fallas, y dar tratamiento a futuros hallazgos negativos en el sistema. A pesar de esto, en la iteración II, en la que se realizaron 2 casos de pruebas, se identificó una deficiencia en el **Control de las opciones de configuración** del sistema, por lo que fue necesario revisar el controlador y darle solución a este hallazgo en el sistema. Finalmente, en la iteración III, se determinó que el caso de prueba evaluado era capaz de calificar como satisfactorio, pues no existían errores en el proceso de carga y visualización de los archivos en formato **PDF**. Lo anterior permitió obtener un índice del 100 % de satisfacción en la iteración final del producto, quedando comprobando el correcto funcionamiento de todas las funcionalidades implementadas en la solución.

Conclusiones parciales del capítulo

En este capítulo se especificó el proceso de implementación del catálogo virtual a partir del desglose de las **HU** en tareas de ingeniería. Esto permitió:

- Especificar los procedimientos necesarios para dar cumplimiento a cada **HU**.
- Definir y aplicar las pruebas de aceptación a las funcionalidades del sistema desarrollado.

Estas pruebas permitieron detectar las fallas en la solución y corregirlas, contribuyendo a mejorar de forma significativa el funcionamiento y la aceptación del mismo por el cliente final.

Conclusiones generales

Con el desarrollo de la presente investigación se arriba a las siguientes conclusiones:

- El empleo del patrón creacional **Singleton** permitió la existencia de un objeto de su tipo proporcionando un único punto de acceso global a él.
- El catálogo virtual interactivo incorpora nuevas funcionalidades asociadas al control de gestos con **Kinect 2**.
- La aplicación integra interfaces que favorecen la representación de la información, así como su visualización por parte de los usuarios, sin afectar su atención.
- La solución contribuye a enriquecer el proceso de interacción de los usuarios con la información para lograr un mejor aprovechamiento del fondo bibliográfico existente en la universidad.

Recomendaciones

A partir de los resultados obtenidos se recomienda:

- Aumentar el número de contenidos bibliográficos en el catálogo virtual interactivo, a fin de ofrecer a los usuarios un mayor volumen de información para las consultas.
- Integrar gestos de voz en el reconocedor que aporten variedad en la interacción de los usuarios con la información de los contenidos del catálogo.

Referencias bibliográficas

- [1] «Concept Glossary». <https://ine.es/DEFIne/en/concepto.htm?c=4904&op=30403&p=2&n=20> (accedido 7 de octubre de 2022).
- [2] «Ciberoteca | Biblioteca Pública Digital», *bibliodigitalfree*. <https://bibliodigitalfree.blogia.com/2005/061301-ciberoteca.php> (accedido 7 de octubre de 2022).
- [3] «9780123822314: Brave NUI World: Designing Natural User Interfaces for Touch and Gesture - AbeBooks - Wigdor, Daniel; Wixon, Dennis: 0123822319». <https://www.abebooks.com/9780123822314/Brave-NUI-World-Designing-Natural-0123822319/plp> (accedido 7 de octubre de 2022).
- [4] B. A. Cárdenas-Sainz, M. L. Barrón-Estrada, R. Zatarain-Cabada, y J. M. Ríos-Félix, «Integration and acceptance of Natural User Interfaces for interactive learning environments», *International Journal of Child-Computer Interaction*, vol. 31, p. 100381, mar. 2022, doi: 10.1016/j.ijcci.2021.100381.
- [5] «Expression Lab» interfaz natural de usuario - Artículos, tutoriales y reviews sobre User Experience, Usabilidad, Diseño, Expression Suite, Silverlight y Windows Presentation Foundation», 17 de octubre de 2010. <https://web.archive.org/web/20101017072904/http://expressionlab.net/tag/interfaz-natural-de-usuario> (accedido 7 de octubre de 2022).
- [6] O. Erazo, R. Pico, O. Erazo, y R. Pico, «Interfaces de usuario basadas en gestos manuales sin contacto para la sala de clases: una revisión bibliográfica», *Enfoque UTE*, vol. 5, n.º 4, pp. 34-53, dic. 2014, doi: 10.29019/enfoqueute.v5n4.46.
- [7] «Interfaces Naturales de Usuario - La Experiencia de la Universidad de Colima | SG Buzz». <https://sg.com.mx/revista/43/interfaces-naturales-usuario-la-experiencia-la-universidad-colima> (accedido 7 de octubre de 2022).
- [8] M. Rocchetti, G. Marfia, y A. Semeraro, «Playing into the wild: A gesture-based interface for gaming in public spaces», *Journal of Visual Communication and Image Representation*, vol. 23, n.º 3, pp. 426-440, abr. 2012, doi: 10.1016/j.jvcir.2011.12.006.
- [9] C. Hurrell y J. Baker, «Immersive learning: Applications of virtual reality for undergraduate education», *College & Undergraduate Libraries*, vol. 27, n.º 2-4, pp. 197-209, oct. 2020, doi: 10.1080/10691316.2020.1796879.
- [10] S. Amaya, «El rol de las bibliotecas nacionales en la era digital y los entornos virtuales», 2018.
- [11] A. A. D. V. Candell y S. D. F. Adum, «Importancia del uso de las bibliotecas virtuales en el desarrollo del conocimiento y actividades investigativas.», *REVISTA*

CIENTÍFICA ECOCIENCIA, vol. 6, n.º 1, Art. n.º 1, feb. 2019, doi: 10.21855/ecociencia.61.148.

[12] J. A. L. Cárdenas, «Importancia de la implementación de bibliotecas virtuales en la Universidad para complementar la educación virtual», *Big Bang Faustiniiano*, vol. 9, n.º 3, Art. n.º 3, sep. 2020, doi: 10.51431/bbf.v9i3.631.

[13] «Información general sobre la Red | Red de Bibliotecas y Archivos del CSIC». <http://bibliotecas.csic.es/es/informacion-general> (accedido 7 de octubre de 2022).

[14] «Biblioteca de Literatura Infantil y Juvenil», *Biblioteca Virtual Miguel de Cervantes*.

https://www.cervantesvirtual.com/portales/biblioteca_literatura_infantil_juvenil/ (accedido 7 de octubre de 2022).

[15] «INTERACTIVE AR CATALOGUE - Augmented and Virtual Reality platform for employee training and education». <https://mojoapps.co/portfolio/interactive-ar-catalogue/> (accedido 7 de octubre de 2022).

[16] «Aplicaciones para Android de Zumoko en Google Play». <https://play.google.com/store/apps/developer?id=Zumoko&hl=es&gl=US> (accedido 7 de octubre de 2022).

[17] L. Cortés-Rico y G. Piedrahita-Solórzano, «Interacciones basadas en gestos: revisión crítica», *Tecnológicas*, vol. 22, pp. 119-132, dic. 2019, doi: 10.22430/22565337.1512.

[18] O. Erazo, N. Baloián, J. A. Pino, y S. F. Ochoa, «Designing hand gesture interfaces for easing students participation from their spot», en *2017 IEEE 21st International Conference on Computer Supported Cooperative Work in Design (CSCWD)*, abr. 2017, pp. 133-138. doi: 10.1109/CSCWD.2017.8066683.

[19] H.-M. J. Hsu, «The Potential of Kinect in Education», *International Journal of Information and Education Technology*, vol. 1, n.º 5, p. 6, 2011.

[20] A. Mehler, A. Lücking, y G. Abrami, «WikiNect: image schemata as a basis of gestural writing for kinetic museum wikis», *Univ Access Inf Soc*, vol. 14, n.º 3, pp. 333-349, ago. 2015, doi: 10.1007/s10209-014-0386-8.

[21] «GitHub - farisNassif/FourthYear_GestureBasedUIDevelopment: A Unity Application developed using the Kinect V2 that incorporates the unique use of various Gestures and Speech Patterns into two separate games linked together by a gesture-controlled UI». https://github.com/farisNassif/FourthYear_GestureBasedUIDevelopment (accedido 7 de octubre de 2022).

[22] C. Radhakrishna, «gesture-recognition-app». 26 de septiembre de 2022. Accedido: 7 de octubre de 2022. [En línea]. Disponible en: <https://github.com/chethanMysore/gesture-recognition-app>

[23] «Kinect v2 Examples with MS-SDK», *RF Solutions - Technology, Health and More*, 1 de Agosto de 2014. <https://rfilekov.com/2014/08/01/kinect-v2-with-ms-sdk/> (accedido 7 de octubre de 2022).

[24] «Kinect status and setup the Kinect for interaction - CodeProject». <https://www.codeproject.com/Tips/701338/Kinect-status-and-setup-the-Kinect-for-interaction> (accedido 7 de octubre de 2022).

[25] J. Joskowicz, «Reglas y Prácticas en eXtreme Programming», p. 22.

[26] G. F. Escribano, «Introducción a Extreme Programming», p. 14.

[27] «Index of /~penzes/soft/Visual Paradigm UML 2013-2014». <https://ics.upjs.sk/~penzes/soft/Visual%20Paradigm%20UML%202013-2014/> (accedido 7 de octubre de 2022).

[28] «UML_y_Patrones_-_Larman_2_Edicion.pdf». Accedido: 7 de octubre de 2022. [En línea]. Disponible en: https://upload.wikimedia.org/wikipedia/commons/a/a8/UML_y_Patrones_-_Larman_2_Edicion.pdf

[29] R. S. Pressman, «Ingeniería del Software. Un Enfoque Práctico», p. 810.

[30] U. T. SL, «Unity (Windows)», *Uptodown.com*. <https://unity.uptodown.com/windows> (accedido 7 de octubre de 2022).

[31] «Unity User Manual (2019.4 LTS) - Unity Manual». <https://docs.unity3d.com/es/current/Manual/index.html> (accedido 7 de octubre de 2022).

[32] «ReSharper: The Visual Studio Extension for .NET Developers by JetBrains», *JetBrains*. <https://www.jetbrains.com/resharper/> (accedido 7 de octubre de 2022).

[33] «Rider: El IDE .NET multiplataforma de JetBrains». <https://www.jetbrains.com/es-es/rider/> (accedido 7 de octubre de 2022).

[34] «> Todo sobre el lenguaje de programación C# 【2022】», *Lenguajes de programación*. <https://lenguajesdeprogramacion.net/c-sharp/> (accedido 7 de octubre de 2022).

[35] M. Katrib, L. Paneque, L. Leonard, A. Piad, L. Sánchez, y A. Tamayo, *Empiece a Programar. Un enfoque multiparadigma con C#*. 2019.

[36] «Requerimientos NO funcionales | OVA2_IngenieriaDeSoftware». https://repositorio.konradlorenz.edu.co/micrositios/001-1527/requerimientos_no_funcionales.html (accedido 9 de noviembre de 2022).

[37] T. Gangavarapu, «TushaarGVS/Design-Patterns-Mentorship». 1 de octubre de 2022. Accedido: 7 de octubre de 2022. [En línea]. Disponible en: <https://github.com/TushaarGVS/Design-Patterns-Mentorship/blob/58a59d6666d7a7303cc77ad90a5c3639bef4a115/Erich%20Gamma%2C%20Richard%20Helm%2C%20Ralph%20Johnson%2C%20John%20M.%20Vlisside>

s-Design%20Patterns_%20Elements%20of%20Reusable%20Object-
Oriented%20Software%20%20-Addison-Wesley%20Professional%20(1994).pdf

[38] U. Technologies, «Unity - Manual: Unit Testing». <https://docs.unity3d.com/Manual/testing-editor-test-runner.html> (accedido 9 de noviembre de 2022).

Anexos

Análisis del Catálogo Virtual Interactivo

El catálogo virtual interactivo para la visualización de contenidos bibliográficos con **Kinect 2** consiste en un producto elaborado bajo la guía de la metodología de desarrollo **XP**, empleando para el modelado de los artefactos el Lenguaje de Modelado Unificado (**UML**, por sus siglas en inglés). Para desarrollar la aplicación se empleó el lenguaje de programación **C#**, asistido por el editor de código **Visual Studio Code**. Se utilizó el motor de gráficos **Unity 3D (versión 2020.2.7f1)** producto a su potencia como motor **cross-platform** (multiplataforma) para realizar proyectos tanto 2D como 3D. Para la integración del sensor de gestos **Kinect 2** en la solución fue requerida la **SDK** en su versión 2.0, además del complemento **KinectForWindows_UnityPro_2.0.1410** para el trabajo desde **Unity** respectivamente. A continuación, se realiza un análisis completo de la herramienta, donde se evalúan sus características principales, encargadas de enmarcar el objeto de estudio de la presente investigación.

Principales funcionalidades y comportamiento actual

A continuación, se describe el comportamiento básico de las principales funcionalidades agrupadas por paquetes para obtener un mejor entendimiento de acuerdo al enfoque lógico y al propósito cumplido dentro de la aplicación.

Catálogo de Contenidos: Permite el acceso de los usuarios a un entorno virtual donde podrán visualizar variedad de contenidos (libros, revistas, artículos científicos y tesis de grado) pertenecientes al área de la informática. Brinda una mejor representación de la información que puede ser utilizada para apoyar el proceso de enseñanza y aprendizaje en la universidad.

Módulo de Gestión de Contenidos: Responde a las necesidades del catálogo de mantenerse actualizado con los materiales más recientes por área de interés, garantizando la disponibilidad de estos en todo momento para las consultas de información por parte de los usuarios del sistema. Su funcionamiento está basado en la recolección de datos almacenados en ficheros de tipo **JSON** que se sincronizan con la aplicación para generar los materiales de forma de paneles 3D, alistando la información necesaria para el momento en que se realice la consulta.

Módulo de Reconocimiento de Gestos: Permite la integración de un reconocedor que asistido por el empleo del sensor **Kinect 2**, facilita la interacción a través de gestos entre los usuarios y la aplicación. Para ello, se hace uso de un complemento especializado con todos los recursos necesarios para el trabajo con este dispositivo.

Los gestos que se emplean obtienen el control del cursor sobre la interfaz, simulando los eventos del ratón (**click, drag, drop**) mediante la detección e interpretación de los gestos con las manos: **abrir** y **cerrar**. Para facilitar la comprensión del lector sobre el tema, se muestra a continuación una representación de los gestos empleados en la solución.



Ilustración 15. Representación de los gestos incluidos en la solución.

Finalmente se integra el reconocedor de los gestos a emplear por el usuario, permitiéndole procesar sus acciones e interactuar con la aplicación.



Ilustración 16. Flujo de trabajo en el catálogo virtual interactivo.

Tarjetas CRC

Tabla 38. Tarjeta CRC # 10

Clase: SoundVolumeUpdater (Singleton)	
Responsabilidad	Colaboradores
<p>Maneja la reproducción de sonido de la aplicación, ya sea el sonido de fondo o los efectos. También maneja la actualización del volumen después de un cambio de estos en la configuración.</p> <ul style="list-style-type: none"> ➤ Reset(): Se llama cuando se añade el componente por primera vez. ➤ Awake(): Para inicializar variables o estados antes de que la aplicación comience. Unity llama a Awake sólo una vez durante el tiempo de vida de la instancia del script. En este caso se emplea para verificar que existe el componente deseado y registrar los cambios de sonido correspondientes. ➤ OnDestroy(): Ocurre cuando la escena termina. En este caso se emplea para llevar la configuración de sonido a un estado predeterminado una vez que termina la ejecución de la aplicación. ➤ Update(): Mantiene actualizado los valores de volumen del catálogo en todo momento. 	<p>AppSettingsManager</p>

Tabla 39. Tarjeta CRC # 11

Clase: SoundVolumeSlider	
Responsabilidad	Colaboradores
<p>Regula el volumen de un tipo de sonido especificado.</p> <ul style="list-style-type: none"> ➤ Reset(): Llamado cuando añade el componente por primera vez. ➤ Start(): Busca el componente de tipo Slider, y obtiene su volumen a la espera de algún cambio para ajustar. 	<p>AppSettingsManager</p>

<ul style="list-style-type: none"> ➤ OnValueChanged(value): Maneja los cambios de volumen al desplazar el Slider. 	
--	--

Tabla 40. Tarjeta CRC # 12

Clase: Kinect2Toggle	
Responsabilidad	Colaboradores
<p>Maneja el comportamiento del ToggleButton y por ende, la interacción en las opciones de configuración.</p> <ul style="list-style-type: none"> ➤ Reset(): Llamado cuando añade el componente por primera vez. ➤ Awake(): Para inicializar variables o estados antes de que la aplicación comience. Unity llama a Awake sólo una vez durante el tiempo de vida de la instancia del script. En este caso se emplea para gestionar el valor inicial del ToggleButton y controlar con ello si hay interacción o no con Kinect 2 desde un inicio. ➤ OnValueChanged(isOn): Maneja los cambios de valor del ToggleButton habilitando o no la interacción con el Kinect 2. 	<p>AppSettingsManager InteractionInputModule StandaloneInputModule</p>

Tabla 41. Tarjeta CRC # 13

Clase: AppSettingsManager	
Responsabilidad	Colaboradores
<p>Maneja los elementos generales en la configuración de las opciones de interacción y sonido en la aplicación.</p> <ul style="list-style-type: none"> ➤ RestoreSettings(): Restaura la configuración guardada. Se llama tan pronto como la ejecución de la aplicación es iniciada. ➤ ApplyPreset(): Aplica la configuración predeterminada. ➤ SaveBool(_key,_value): Guarda el valor de preferencia escogido por el usuario para la interacción. 	<p>UIDataType<T> BoolToggle</p>

<ul style="list-style-type: none"> ➤ LoadBool(_key,_defaultvalue): Carga al inicio el valor de preferencia escogido por el usuario para la interacción . ➤ SaveToDisk(): Guarda todas los cambios de la reciente sesión en el disco. ➤ RegisterVolumeUpdater(type, target): Actualiza con un valor determinado el volumen de un objeto. ➤ RemoveVolumeUpdater(type, target): Elimina el sonido del objeto. ➤ SetMainSoundVolume(value): Actualiza con un valor determinado el volumen del sonido principal del catálogo. ➤ GetMainSoundVolume(): Obtiene el volumen de sonido principal del catálogo. ➤ SetSoundVolume(type, value): Establece el volumen de sonido del tipo de sonido especificado. ➤ GetSoundVolume(type): Obtiene el volumen del sonido del tipo de sonido especificado. ➤ InitializeVolumeTypes(): Inicializa todos los tipos de sonido. 	
--	--

Tabla 42. Tarjeta CRC # 14

Clase: <i>AppManager (PersistentSingleton)</i>	
Responsabilidad	Colaboradores
Maneja todo lo relacionado con el funcionamiento general de la aplicación y los tiempos. <ul style="list-style-type: none"> ➤ SetTimeScale(): Establece la escala de tiempo. ➤ ResetTimeScale(): Restablece la escala de tiempo a la última escala de tiempo guardada. 	<i>GUIManager</i>

Tabla 43. Tarjeta CRC # 15

Clase: <i>KinectManager</i>	
Responsabilidad	Colaboradores

<p>Representa el principal componente en la relación entre la aplicación y el dispositivo Kinect 2.</p> <ul style="list-style-type: none"> ➤ Instance(): Devuelve una sola instancia de Kinect. ➤ IsKinectInitialized(): Determina si el sensor y el propio componente del KinectManager están inicializados. ➤ GetSensorData(): Obtiene los datos de la estructura del sensor. Estructura que no debe ser modificada y solo debe emplearse de forma interna. ➤ GetBodyCount(): Devuelve el número de cuerpos que están siendo identificados por el sensor. ➤ IsUserDetected(): Verifica que el Kinect se haya inicializado y que el número de usuarios disponibles sea positivo. ➤ IsUserTracked(userId): Determina cuando un usuario con un id específico pertenece a la lista de usuarios identificados o no. ➤ GetUsersCount(): Retorna el número actual de usuarios que están siendo identificados. ➤ IsLeftHandConfidenceHigh(userId): Determina si el nivel de confianza con el que se está identificando la mano izquierda de un usuario es alta o no. ➤ IsRightHandConfidenceHigh(userId): Determina si el nivel de confianza con el que se está identificando la mano derecha de un usuario es alta o no. ➤ GetLeftHandState(userId): Devuelve el estado en el que se encuentra la mano izquierda de un usuario. Si se está identificando o no. ➤ GetRightHandState(userId): Devuelve el estado en el que se encuentra la mano derecha de un usuario. Si se está identificando o no. ➤ GetLeftHandInteractionBox(userId, leftBotBack, rightTopFront, bValidBox): Define 	<p>No aplica</p>
---	-------------------------

<p>un área de interacción alrededor de la mano izquierda de un usuario determinado.</p> <p>➤ <i>GetRighthandInteractionBox(userId, rightbotBack, righthTopFront, bValidBok):</i> Define un área de interacción alrededor de la mano derecha de un usuario determinado.</p>	
---	--

Tabla 44. Tarjeta CRC # 16

Clase: <i>InteractionInputModule</i>	
Responsabilidad	Colaboradores
<p>Consiste en un módulo para el procesamiento de determinadas acciones que permiten la interacción entre el usuario y la aplicación, mediante el empleo del Kinect 2.</p> <p>➤ <i>ShouldActivateModule():</i> Determina si el módulo de interacción debe ser activado o no.</p> <p>➤ <i>IsModuleSupported():</i> Determina si el módulo de interacción es soportado o no.</p> <p>➤ <i>ActivateModule():</i> Activa el módulo.</p> <p>➤ <i>DeactivateModule():</i> Desactiva el módulo.</p> <p>➤ <i>ProcessInteractionEvent(id):</i> Emula los datos del ratón y procesa los datos de interacción.</p> <p>➤ <i>GetMousePointerEvent(id):</i> Obtiene la posición del mouse.</p> <p>➤ <i>ProcessHandPressRelease(data):</i> Procesa el evento de apertura y cierre de las manos.</p> <p>➤ <i>HandGripDetected(userId, userIndex, isRighthand, isHandInteracting, handScreenPos):</i> Determina si la mano que interactúa realiza una acción de cierre.</p> <p>➤ <i>HandReleaseDetected(userId, userIndex, isRighthand, isHandInteracting, handScreenPos):</i> Determina si la mano que interactúa realiza una acción de apertura.</p> <p>➤ <i>HandClickDetected(userId, userIndex, isRighthand, HandScreenPos):</i> Iden</p>	<p><i>InteractionListenerInterface</i></p> <p><i>InteractionManager</i></p>

tifica si ha ocurrido un clic luego de la apertura y cierre de la mano.	
---	--

Tabla 45. Tarjeta CRC # 17

Clase: <i>InteractionManager</i>	
Responsabilidad	Colaboradores
<p>El gestor de interacciones es el componente que se encarga de las interacciones manuales.</p> <ul style="list-style-type: none"> ➤ Instance(): La instancia del InteractionManager. ➤ IsInteractionInited(): Determina si el Interaction Manager fue inicializado con éxito. ➤ GetUserID(): Obtiene el ID del usuario actual, o 0 si no hay ningún usuario rastreado. ➤ GetLeftHandEvent(): Obtiene el evento actual de la mano izquierda (ninguno, agarre o liberación). ➤ GetLastLeftHandEvent(): Obtiene el último evento detectado de la mano izquierda (agarre o liberación). ➤ GetLeftHandScreenPos(): Obtiene la posición actual normalizada de la vista de la mano izquierda, en el rango [0, 1]. ➤ IsLeftHandPress(): Determina si la mano izquierda está presionada. ➤ IsLeftHandClickDetected(): Determina si se detecta un clic con la mano izquierda o no en caso contrario. ➤ GetLeftHandClickProgress(): Obtiene el progreso del clic realizado con la mano izquierda, en el rango [0,1]. ➤ GetRightHandEvent(): Obtiene el evento actual de la mano derecha (ninguno, agarre o liberación). ➤ GetLastRightHandEvent(): Obtiene el último evento detectado de la mano derecha (agarre o liberación). 	<p>No aplica</p>

<ul style="list-style-type: none"> ➤ GetRightHandScreenPos(): Obtiene la posición actual normalizada de la vista de la mano derecha, en el rango [0, 1]. ➤ IsRightHandPress(): Determina si la mano derecha está presionada. ➤ IsRightHandClickDetected(): Determina si se detecta un clic con la mano derecha o no en caso contrario. ➤ GetRightHandClickProgress(): Obtiene el progreso del clic realizado con la mano derecha, en el rango [0,1]. ➤ GetCursorPosition(): Obtiene la posición actual del cursor normalizado de la ventana gráfica. 	
--	--

Tareas de ingeniería asociadas a Historias de Usuario

Tareas de ingeniería para la Iteración I

HU Mostrar vista principal

Tabla 46. Tarea de Ingeniería #3

Tarea	
Número de Tarea:3	Número de Historia de Usuario:1
Nombre de la tarea: Desarrollar enlaces al resto de secciones de la aplicación.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.5
Fecha de inicio: 2 de mayo del 2022	Fecha de fin: 5 de mayo del 2022
Programador responsable: Leudis Rafael Estrada González	
Descripción: Se implementa una funcionalidad que permite la navegación por todas las secciones de la aplicación, mediante la transición de escenas, a través del menú y los botones correspondientes.	

Tabla 47. Tarea de Ingeniería #4

Tarea	
Número de Tarea:4	Número de Historia de Usuario:1
Nombre de la tarea: Implementar salida del sistema.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.3
Fecha de inicio: 5 de mayo del 2022	Fecha de fin: 6 de mayo del 2022
Programador responsable: Leudis Rafael Estrada González	

Descripción: Se implementa una funcionalidad que permite la salida del sistema al usuario en el momento en que éste lo desee, a través del menú y su botón correspondiente.

HU Mostrar vista de contenidos

Tabla 48. Tarea de Ingeniería #7

Tarea	
Número de Tarea:7	Número de Historia de Usuario:2
Nombre de la tarea: Desarrollar funcionalidad para mostrar las secciones de contenidos del catálogo por área de interés.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.1
Fecha de inicio: 11 de mayo del 2022	Fecha de fin: 12 de mayo del 2022
Programador responsable: Leudis Rafael Estrada González	
Descripción: Se desarrolla una funcionalidad que permite al usuario acceder a las diferentes secciones de contenidos que posee el catálogo, por área de interés.	

Tabla 49. Tarea de Ingeniería #8

Tarea	
Número de Tarea:8	Número de Historia de Usuario:2
Nombre de la tarea: Desarrollar menú desplegable para acceder a las secciones de contenidos del catálogo por área de interés.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.2
Fecha de inicio: 13 de mayo del 2022	Fecha de fin: 16 de mayo del 2022
Programador responsable: Leudis Rafael Estrada González	
Descripción: Se desarrolla un panel que permite mostrar al usuario el menú en forma de desplegable en la vista de contenidos de la aplicación. Permite el acceso a las secciones de contenidos: libros, revistas, artículos científicos, tesis y eventos . Integra un componente de animación para el despliegue.	

Tabla 50. Tarea de Ingeniería #9

Tarea	
Número de Tarea:9	Número de Historia de Usuario:2
Nombre de la tarea: Desarrollar enlaces al resto de áreas de interés en la vista de contenidos.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.2
Fecha de inicio: 16 de mayo del 2022	Fecha de fin: 17 de mayo del 2022
Programador responsable: Leudis Rafael Estrada González	
Descripción: Se implementa una funcionalidad que permite la navegación por todas las áreas de interés presentes en la vista de contenidos, a través del menú y los botones correspondientes.	

Tabla 51. Tarea de Ingeniería #10

Tarea	
Número de Tarea:10	Número de Historia de Usuario:2
Nombre de la tarea: Implementar funcionalidad para generar paneles 3D a partir de los datos recogidos del archivo en formato JSON.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.5
Fecha de inicio: 17 de mayo del 2022	Fecha de fin: 20 de mayo del 2022
Programador responsable: Leudis Rafael Estrada González	
Descripción: Se implementa una funcionalidad que transforma los datos del archivo en formato JSON, a paneles 3D en la vista de contenidos, correspondientes a cada uno de los materiales del catálogo.	

Tabla 52. Tarea de Ingeniería #11

Tarea	
Número de Tarea:11	Número de Historia de Usuario:2
Nombre de la tarea: Implementar funcionalidades para aplicar movimiento, rotación y animación a los paneles 3D generados.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.5
Fecha de inicio: 23 de mayo del 2022	Fecha de fin: 26 de mayo del 2022
Programador responsable: Leudis Rafael Estrada González	
Descripción: Se implementan varias funcionalidades que convergen en una función principal capaz de aplicar físicas y animar los paneles 3D generados por la lectura del archivo en formato JSON, mostrándolos en la vista de contenidos.	

HU Mostrar información de contenidos

Tabla 53. Tarea de Ingeniería #13

Tarea	
Número de Tarea:13	Número de Historia de Usuario:3
Nombre de la tarea: Desarrollar panel informativo desplegable para mostrar la información correspondiente a cada uno de los materiales del catálogo.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.5
Fecha de inicio: 30 de mayo del 2022	Fecha de fin: 2 de junio del 2022
Programador responsable: Leudis Rafael Estrada González	
Descripción: Se desarrolla un panel que muestra al usuario la información necesaria en la vista de contenidos de la aplicación. Permite observar parámetros tales como: título , descripción , autor o autores , idioma , número de páginas y código ISBN entre otros. Integra un componente de animación para el despliegue.	

Tareas de ingeniería para la Iteración II

HU Controlar opciones

Tabla 54. Tarea de Ingeniería #15

Tarea	
Número de Tarea:15	Número de Historia de Usuario:4
Nombre de la tarea: Implementar funcionalidad para controlar las opciones de configuración en la aplicación.	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha de inicio: 9 de junio del 2022	Fecha de fin: 15 de junio del 2022
Programador responsable: Leudis Rafael Estrada González	
Descripción: Se implementa una funcionalidad que es capaz de controlar las opciones de configuración de interacción o sonido presentes en la aplicación.	

HU Consultar ayuda

Tabla 55. Tarea de Ingeniería #17

Tarea	
Número de Tarea:17	Número de Historia de Usuario:5
Nombre de la tarea: Implementar funcionalidad para mostrar la información en la sección de “Ayuda” de la aplicación.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.5
Fecha de inicio: 21 de junio del 2022	Fecha de fin: 23 de junio del 2022
Programador responsable: Leudis Rafael Estrada González	
Descripción: Se implementa una funcionalidad que muestra la información en la sección de “Ayuda” de la aplicación, directamente desde el menú o a través de su botón correspondiente	

Tareas de ingeniería para la Iteración III

HU Generar reconocedor de gestos

Tabla 56. Tarea de Ingeniería #19

Tarea	
Número de Tarea:19	Número de Historia de Usuario:6
Nombre de la tarea: Implementar funcionalidades para la integración del sensor de gestos Kinect 2 en la aplicación.	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha de inicio: 29 de junio del 2022	Fecha de fin: 5 de julio del 2022
Programador responsable: Leudis Rafael Estrada González	

Descripción: Se implementan un conjunto de funcionalidades que permiten reconocer el sensor de gestos Kinect 2 en la aplicación.

Tabla 57. Tarea de Ingeniería #20

Tarea	
Número de Tarea:20	Número de Historia de Usuario:6
Nombre de la tarea: Implementar funcionalidad para el reconocimiento de gestos, mediante el empleo del sensor Kinect 2 en la aplicación.	
Tipo de tarea: Desarrollo	Puntos estimados: 1.5
Fecha de inicio: 6 de julio del 2022	Fecha de fin: 13 de julio del 2022
Programador responsable: Leudis Rafael Estrada González	
Descripción: Se implementa una funcionalidad que permite integrar el reconocedor de los gestos empleados por el usuario para interactuar con la aplicación, asistido por el sensor Kinect 2.	

[HU Visualizar PDF](#)

Tabla 58. Tarea de Ingeniería #22

Tarea	
Número de Tarea:22	Número de Historia de Usuario:7
Nombre de la tarea: Implementar funcionalidad para visualizar los materiales digitales de los contenidos bibliográficos en formato PDF.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.6
Fecha de inicio: 18 de julio del 2022	Fecha de fin: 20 de julio del 2022
Programador responsable: Leudis Rafael Estrada González	
Descripción: Se implementa una funcionalidad que permite visualizar los materiales digitales en formato PDF correspondientes a los contenidos bibliográficos seleccionados, directamente desde el navegador del sistema, si este dispone de acceso a la red.	

Pruebas unitarias

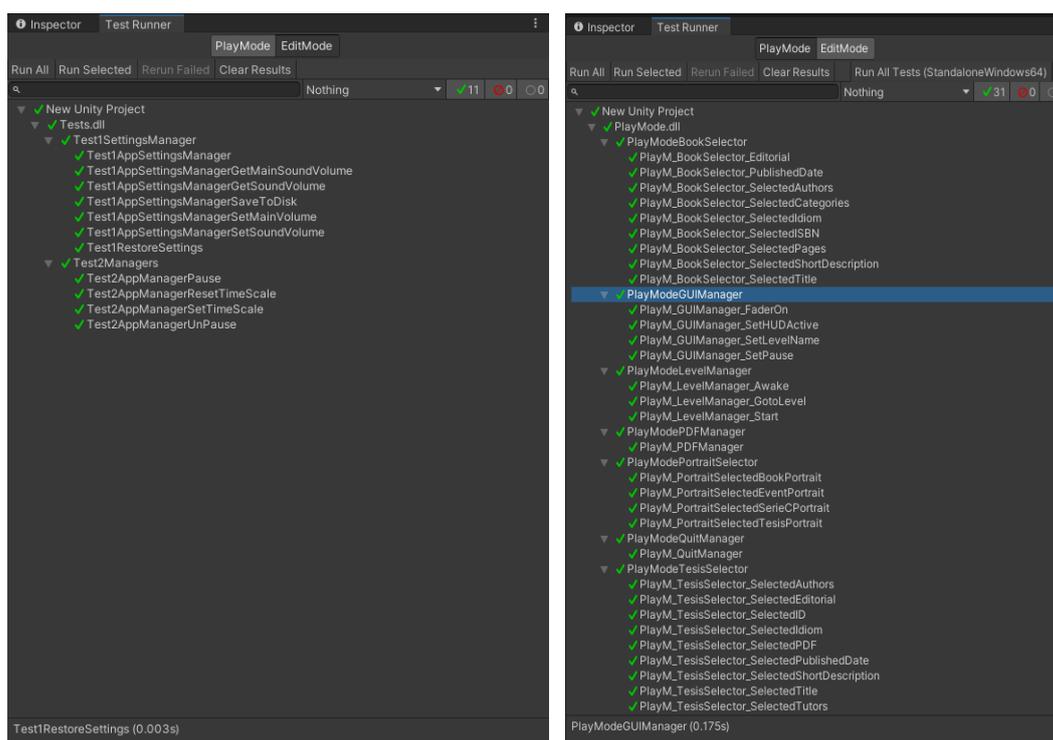


Ilustración 17. Representación de las pruebas unitarias en Modo Editor y en Modo Juego

Pruebas de aceptación

Pruebas de aceptación para la Iteración I

Tabla 59. Prueba de aceptación #5

Caso de prueba de aceptación	
Código:HU2_P5	Historia de Usuario:2
Nombre: Mostrar vista de contenidos bibliográficos del sistema.	
Descripción: Prueba para la funcionalidad: Desarrollar vista para mostrar los contenidos bibliográficos.	
Condiciones de ejecución:	
<ul style="list-style-type: none"> ➤ El usuario debe estar disponible para la interacción con el sistema. ➤ El sistema debe haber cargado la vista principal de la aplicación. ➤ El sistema debe haber mostrado el menú principal desplegable. 	
Pasos de ejecución:	
<ol style="list-style-type: none"> 1. El usuario selecciona la opción “Catálogo” del menú principal. 2. Se carga la vista correspondiente. 	

3. Se muestra la vista de contenidos bibliográficos del sistema.
Resultado: No satisfactorio.

Tabla 60. Prueba de aceptación #6

Caso de prueba de aceptación	
Código: HU2_P6	Historia de Usuario: 3
Nombre: Mostrar menú desplegable para acceder a las secciones de contenidos del catálogo por área de interés.	
Descripción: Prueba para la funcionalidad: Desarrollar menú desplegable para acceder a las secciones de contenidos del catálogo por área de interés.	
Condiciones de ejecución:	
<ul style="list-style-type: none"> ➤ El usuario debe estar disponible para la interacción con el sistema. ➤ El sistema debe haber mostrado el menú principal desplegable. ➤ El usuario debe haber seleccionado la opción “Catálogo” en el menú principal para acceder a la vista que muestra los contenidos bibliográficos. ➤ El sistema debe haber mostrado la vista de contenidos bibliográficos. ➤ El usuario debe haber seleccionado el desplegable del menú que muestra las diferentes secciones de contenidos del catálogo por área de interés. 	
Pasos de ejecución:	
<ol style="list-style-type: none"> 1. El usuario selecciona el desplegable correspondiente al menú que muestra las diferentes secciones de contenidos del catálogo por área de interés. 2. Se muestra el menú desplegable para acceder al resto de las secciones de contenidos del catálogo. 	
Resultado: Satisfactorio.	

Tabla 61. Prueba de aceptación #7

Caso de prueba de aceptación	
Código: HU2_P7	Historia de Usuario: 3
Nombre: Comprobar enlaces al resto de áreas de interés disponibles en la vista de contenidos.	
Descripción: Prueba para la funcionalidad: Desarrollar enlaces al resto de áreas de interés en la vista de contenidos.	
Condiciones de ejecución:	
<ul style="list-style-type: none"> ➤ El usuario debe estar disponible para la interacción con el sistema. 	

➤ El sistema debe haber mostrado el menú con las diferentes secciones de contenidos.
Pasos de ejecución:
<ol style="list-style-type: none"> 1. El usuario selecciona una de las secciones del menú desplegable en la vista de contenidos. 2. Se carga la sección de contenidos seleccionada. 3. Se muestra la sección seleccionada.
Resultado: Satisfactorio.

Tabla 62. Prueba de aceptación #8

Caso de prueba de aceptación	
Código: HU2_P8	Historia de Usuario: 3
Nombre: Generar paneles 3D a partir de los datos del archivo en formato JSON.	
Descripción: Prueba para la funcionalidad: Implementar funcionalidad para generar paneles 3D a partir de los datos recogidos del archivo en formato JSON.	
Condiciones de ejecución:	
➤ El archivo en formato JSON debe estar adjunto a la clase “ Menú ” que se encarga de transformar los datos en paneles 3D en la vista de contenidos, correspondientes a cada uno de los materiales del catálogo.	
Pasos de ejecución:	
<ol style="list-style-type: none"> 1. El archivo en formato JSON está adjunto a la clase “Menú”. 2. La clase “Menú” convierte los datos del archivo en formato JSON a objetos en 3D que representan los paneles en la vista de contenidos. 3. Se muestran los paneles 3D, correspondientes a cada uno de los contenidos del catálogo. 	
Resultado: Satisfactorio.	

Tabla 63. Prueba de aceptación #9

Caso de prueba de aceptación	
Código: HU2_P9	Historia de Usuario: 3
Nombre: Aplicar movimiento, rotación y animación a los paneles 3D generados.	
Descripción: Prueba para la funcionalidad: Implementar funcionalidades para aplicar movimiento, rotación y animación a los paneles 3D generados.	
Condiciones de ejecución:	
➤ El sistema debe haber cargado la vista de contenidos.	

➤ Los paneles 3D generados deben mostrarse en la vista de contenidos.
Pasos de ejecución:
<ol style="list-style-type: none"> 1. Los paneles 3D generados se muestran en la vista de contenidos. 2. Las clases “Menú” y “Swipe Handler” Se encargan de aplicar físicas y animar los paneles 3D generados por la carga dinámica de información desde el archivo en formato JSON, mostrándolos en la vista de contenidos. 3. Los paneles 3D generados, se animan de manera uniforme en la vista de contenidos.
Resultado: Satisfactorio.

Tabla 64. Prueba de aceptación #10

Caso de prueba de aceptación	
Código: HU3_P10	Historia de Usuario: 3
Nombre: Mostrar panel informativo desplegable para con la información correspondiente a los materiales del catálogo.	
Descripción: Prueba para la funcionalidad: Desarrollar panel informativo desplegable para mostrar a la información correspondiente a los materiales del catálogo.	
Condiciones de ejecución:	
<ul style="list-style-type: none"> ➤ El usuario debe estar disponible para la interacción con el sistema. ➤ El sistema debe haber cargado la vista principal. ➤ El sistema debe haber mostrado el menú principal desplegable. ➤ El usuario debe haber seleccionado la opción “Catálogo” en el menú principal para acceder a la vista que muestra los contenidos por área de interés. ➤ El sistema debe haber cargado la vista de contenidos. ➤ El usuario debe haber seleccionado la portada del material correspondiente. 	
Pasos de ejecución:	
<ol style="list-style-type: none"> 1. El usuario selecciona la portada del material correspondiente en la vista de contenidos. 2. Se muestra el panel con la información asociada al material correspondiente. 	
Resultado: Satisfactorio.	

Pruebas de aceptación para la Iteración II

Tabla 65. Prueba de aceptación #12

Caso de prueba de aceptación	
Código: HU5_P12	Historia de Usuario: 4
Nombre: Mostrar vista para acceder a la sección de “ Ayuda ” de la aplicación.	
Descripción: Prueba para la funcionalidad: Desarrollar la vista para acceder a la sección de “Ayuda” de la aplicación.	
Condiciones de ejecución: <ul style="list-style-type: none">➤ El usuario debe estar disponible para la interacción con el sistema.➤ El sistema debe haber mostrado el menú principal desplegable.➤ El usuario debe haber seleccionado la opción “Ayuda” en el menú principal.	
Pasos de ejecución: <ol style="list-style-type: none">1. El usuario selecciona la opción “Ayuda” del menú desplegable.2. Se carga la interfaz correspondiente.3. Se muestra la interfaz seleccionada, en forma de panel; permitiéndole al usuario acceder a información de apoyo sobre la aplicación.	
Resultado: Satisfactorio.	