

ESCUELA POLITÉCNICA NACIONAL

FACULTAD DE INGENIERÍA DE SISTEMAS

**DISEÑO Y APLICACIÓN DE UN MODELO DE RECONOCIMIENTO
DE 11 GESTOS DE LA MANO USANDO SEÑALES EMG, DTW Y
KNN**

**DESARROLLO DE UNA INTERFAZ HUMANO-MÁQUINA USANDO
GESTOS DE LA MANO**

**TRABAJO DE INTEGRACIÓN CURRICULAR PRESENTADO COMO
REQUISITO PARA LA OBTENCIÓN DEL TÍTULO DE INGENIERO DE
SOFTWARE**

BRYAN NORBERTO FLORES SARANGO

bf97930@gmail.com

DIRECTOR: ANGEL LEONARDO VALDIVIESO CARAGUAY

angel.valdivieso@epn.edu.ec

DMQ, agosto 2022

CERTIFICACIONES

Yo, BRYAN NORBERTO FLORES SARANGO declaro que el trabajo de integración curricular aquí descrito es de mi autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.



BRYAN NORBERTO FLORES SARANGO

Certifico que el presente trabajo de integración curricular fue desarrollado por BRYAN FLORES, bajo mi supervisión.



ÁNGEL LEONARDO VALDIVIESO
DIRECTOR

DECLARACIÓN DE AUTORÍA

A través de la presente declaración, afirmamos que el trabajo de integración curricular aquí descrito, así como el (los) producto(s) resultante(s) del mismo, son públicos y estarán a disposición de la comunidad a través del repositorio institucional de la Escuela Politécnica Nacional; sin embargo, la titularidad de los derechos patrimoniales nos corresponde a los autores que hemos contribuido en el desarrollo del presente trabajo; observando para el efecto las disposiciones establecidas por el órgano competente en propiedad intelectual, la normativa interna y demás normas.

BRYAN NORBERTO FLORES SARANGO

ANGEL LEONARDO VALDIVIESO CARAGUAY

DEDICATORIA

Quiero dedicar este Proyecto de Titulación a mi Madre, Padre, Hermanos y a mi Tía Isabel que han aportado con su apoyo a través de estos 5 años de vida universitaria, con sus consejos, su apoyo físico y verbal en los momentos difíciles

AGRADECIMIENTO

Así mismo, agradezco a mis profesores durante toda esta etapa que han sabido cuales son los conocimientos necesarios para el actual mundo laboral, y finalmente a la Escuela Politécnica Nacional que me alberga desde 2017 por todo el conocimiento en el cual fui sumergido.

ÍNDICE DE CONTENIDO

CERTIFICACIONES.....	I
DECLARACIÓN DE AUTORÍA.....	II
DEDICATORIA.....	III
AGRADECIMIENTO.....	IV
LISTA DE FIGURAS	VII
LISTA DE TABLAS.....	VIII
LISTA DE ANEXOS	IX
RESUMEN	X
ABSTRACT	XI
1. DESCRIPCIÓN DEL COMPONENTE DESARROLLADO.....	1
1.1 Objetivo general	2
1.2 Objetivos específicos	2
1.3 Alcance	2
2. MARCO TEÓRICO.....	5
2.1 Realidad virtual	5
2.1.1 Características de la realidad virtual.....	5
2.1.2 Tipos de realidad virtual.....	6
2.1.3 Contextos donde es trascendental la aplicación de la realidad virtual.....	6
2.2 Modelos 3D.....	6
3. METODOLOGÍA.....	8
3.1 Especificación de requisitos y restricciones	8
3.1.1 Requisitos funcionales.....	8
3.1.2 Mapeo de gestos con tareas en la aplicación	8
3.1.3 Requisitos no funcionales	9
3.2 Diseño.....	9
3.2.1 Diagrama de paquetes	9
3.2.2 Diagrama de clases.....	10
3.3 Desarrollo.....	12
3.4 Verificación y validación de requisitos.....	13
3.4.1 Pruebas de Integración	13
3.4.2 Pruebas de Aceptación.....	14

3.5 Operación de la aplicación.....	15
3.6 Integración con el modelo de reconocimiento de gestos	17
3.7 Validación de la integración con el modelo de reconocimiento de gestos ..	18
4. RESULTADOS, CONCLUSIONES Y RECOMENDACIONES	19
4.1 Resultados	19
4.2 Conclusiones.....	21
4.3 Recomendaciones	22
5. REFERENCIAS BIBLIOGRÁFICAS	23
6. ANEXOS	24
ANEXO I Diagrama de casos de uso de la aplicación.....	25
ANEXO II Diagrama de clases de la aplicación.....	26
ANEXO III Descripción de casos de uso	27
ANEXO IV Manual de usuario de la aplicación	32
ANEXO V Código fuente de la aplicación	32
ANEXO VI Código fuente script Matlab ejecución de tareas mediante el uso del teclado.....	32
ANEXO VII Manual de uso de la aplicación integrada con el modelo de reconocimiento de gestos.....	32
ANEXO VIII Código fuente de la integración con la aplicación y el modelo de reconocimiento de gestos.....	32
ANEXO IX Video del funcionamiento de la aplicación integrada con el modelo de reconocimiento de gestos.....	32

LISTA DE FIGURAS

Figura 1 Clases de ListeningCommands.....	10
Figura 2 Clases de Commands.....	11
Figura 3 Clases de LoadParts.....	11
Figura 4 Clases de Tools.....	12
Figura 5 Imágenes para la construcción del ejecutable de la aplicación.....	16
Figura 6 Escena Galery y Model.....	17

LISTA DE TABLAS

Tabla 1 Gestos y acciones en la aplicación.....	3
Tabla 2 Flujo CU1	8
Tabla 3 Especificaciones técnicas del ambiente de desarrollo de la aplicación ...	13
Tabla 4 Plataformas para descarga de modelos 3D	13
Tabla 5 Pruebas de integración de los diferentes módulos de la aplicación	14
Tabla 6 Pruebas de aceptación de los requisitos de la aplicación	15
Tabla 7 Mapeo de teclas y gestos.....	16
Tabla 8 Pruebas de aceptación de la Integración de la aplicación y el modelo de reconocimiento de gestos.....	18
Tabla 9 Resultados de las Pruebas de Integración del Visor de Modelos 3D	19
Tabla 10 Resultados de las Pruebas de Aceptación del Visor de Modelos 3D	20
Tabla 11 Resultados de las Pruebas de Aceptación en la integración del modelo de reconocimiento de gestos.....	21
Tabla 12 Flujo CU1	27
Tabla 13 Flujo CU2	27
Tabla 14 Flujo CU3 primera parte	27
Tabla 15 CU3 segunda parte	28
Tabla 16 Flujo CU4	28
Tabla 17 Flujo CU5	28
Tabla 18 Flujo CU6	29
Tabla 19 Flujo CU7	29
Tabla 20 Flujo CU8	30
Tabla 21 Flujo CU9	30
Tabla 22 Flujo CU10	30
Tabla 23 Flujo CU11	31
Tabla 24 Flujo CU12	31
Tabla 25 Flujo CU13	31

LISTA DE ANEXOS

ANEXO I Diagrama de casos de uso de la aplicación.....	25
ANEXO II Diagrama de clases de la aplicación.....	26
ANEXO III Descripción de casos de uso	27
ANEXO IV Manual de usuario de la aplicación	32
ANEXO V Código fuente de la aplicación	32
ANEXO VI Código fuente script Matlab ejecución de tareas mediante el uso del teclado.....	32
ANEXO VII Manual de uso de la aplicación integrada con el modelo de reconocimiento de gestos.....	32
ANEXO VIII Código fuente de la integración con la aplicación y el modelo de reconocimiento de gestos.....	32
ANEXO IX Video del funcionamiento de la aplicación integrada con el modelo de reconocimiento de gestos.....	32

RESUMEN

En el presente trabajo se desarrolló un Visor de modelos 3D en realidad virtual guiado por 11 gestos de la mano. La aplicación permite una interacción más natural en la visión de modelos tridimensionales. Además, posibilita el uso de los gestos en una herramienta que mejora la convivencia en un entorno simulado. En el presente desarrollo se utilizó el enfoque metodológico en cascada, el cual divide el proyecto por fases. Esta metodología, al ser secuencial, transmite de mejor forma la información referente a los requisitos. El resultado del proceso es una aplicación funcional e integrada con el modelo de reconocimiento de gestos. La aplicación aprovecha las propiedades de los modelos tridimensionales y la naturalidad de la interacción con la aplicación mediante gestos de la mano. Además, la aplicación cuenta con ventajas técnicas tales como no necesitar de una instalación previa, o la autodetección de las gafas para la realidad virtual.

PALABRAS CLAVE: Aplicación, Cascada, Gestos de la mano, Integración, Modelo de reconocimiento, Visor de modelos 3D

ABSTRACT

In the present work, a 3D model viewer in virtual reality guided by eleven hand gestures was developed. The application allows a more natural interaction in the vision of three-dimensional models. In addition, it enables the use of gestures in a tool that improves coexistence in a simulated environment. In the present development, the cascade methodological approach was used, which divides the project by phases. This methodology, being sequential, better transmits the information about the requirements. The result of the process is a functional application integrated with the gesture recognition model. The application takes advantage of the properties of three-dimensional models and the naturalness of interaction with the application through gestures. In addition, the application has technical advantages such as not requiring prior installation, or the autodetection of glasses for virtual reality.

KEYWORDS: Application, Hand Gestures, Integration, Model Recognition, Waterfall, 3D Model Viewer.

1. DESCRIPCIÓN DEL COMPONENTE DESARROLLADO

En el presente componente se desarrolló una aplicación en realidad virtual que es controlada mediante 11 gestos de la mano. La aplicación es un Visor de modelos tridimensionales en realidad virtual que funciona en tiempo real; es decir, la aplicación dará una respuesta en un intervalo de tiempo de 58 milisegundos a 70 milisegundos [1]. El componente fue desarrollado con el fin de facilitar la interacción humano-maquina con la realidad virtual y brindar un aporte a la comunidad.

Con el fin de construir esta aplicación se empleó el enfoque metodológico del Modelo en Cascada [2], en el cual están contempladas las siguientes fases: especificación de requisitos y restricciones, diseño, desarrollo, verificación y validación de los requisitos y operación de la aplicación. Las fases anteriormente mencionadas contribuyen secuencialmente al desarrollo de la aplicación.

En el producto de software desarrollado, los 11 gestos de la mano sirven para ejecutar acciones de interacción con modelos dentro de la aplicación como, por ejemplo: moverse entre modelos, mover el modelo, aceptar una acción, entre otros. Los gestos son detectados gracias al uso de un modelo de reconocimiento, el cual fue desarrollado como primer componente del actual proyecto. El componente de reconocimiento de gestos se basa en Machine Learning, el cual usa principalmente los algoritmos DTW [3] y KNN [4], lleva a cabo la tarea de reconocimiento de gestos brindando unas tasas de exactitud de clasificación y reconocimiento adecuadas para la aplicación.

La interacción natural de acciones mediante gestos es una ventaja de la aplicación. Además, permite una examinación a detalle de los modelos 3D. También facilita la visualización en 360 grados de los modelos. La realidad virtual entrega control total del ambiente al usuario a través de las gafas y gestos. La aplicación a futuro puede ser utilizada en mercados como marketing, medicina, automotriz, educación, entre otros. La aplicación podría utilizarse para la visualización de modelos anatómicos del cuerpo humano. En el marketing ayudaría a la recreación y vivencia cercana a la realidad con el producto o servicio ofrecidos. La aplicación en la educación podría servir para visualizar modelos parecidos a los reales donde se ejecutarían acciones sin consecuencias graves.

1.1 Objetivo general

Desarrollar una aplicación en realidad virtual para visualizar modelos 3D usando 11 gestos de la mano.

1.2 Objetivos específicos

1. Revisar el estado del arte y establecer las características que tendrá la aplicación de realidad virtual para visualizar modelos 3D
2. Desarrollar una aplicación en realidad virtual para visualizar modelos 3D controlado por 11 gestos de la mano utilizando el enfoque metodológico del Modelo en Cascada
3. Integrar la aplicación desarrollada con el componente de modelo de reconocimiento de 11 gestos de la mano

1.3 Alcance

La aplicación se desarrolló de forma secuencial utilizando el enfoque metodológico del Modelo en Cascada [2]; el mismo que cuenta con las siguientes fases:

- **Especificación de requisitos y restricciones**

El propósito de esta fase es establecer los requisitos y las restricciones para la aplicación. Con este fin, se mantuvo sesiones con el usuario final para entender sus necesidades y traducirlas en los correspondientes requisitos y restricciones.

- **Diseño de la aplicación software.**

Con el fin de cumplir con esta fase se diseñó la arquitectura de software de la aplicación y se describió las relaciones entre las partes de la arquitectura. La arquitectura de software se diseñó a partir de los requisitos definidos anteriormente, convirtiéndolos en una representación del software con su detalle procedimental detallado, anterior a la fase de Desarrollo.

- **Desarrollo de la aplicación de software.**

La fase tiene como objetivo implementar el diseño de la aplicación de software. El objetivo se consiguió integrando cada elemento del diseño dentro de la codificación y programación de la aplicación. Con este propósito, se programó y desarrolló la aplicación con la ayuda de un entorno de desarrollo integrado (IDE).

- **Verificación y validación de los requisitos.**

La fase tiene el objetivo de probar y ejecutar el código desarrollado con el fin de determinar si se cumple o no con los requisitos establecidos en la fase de requisitos y restricciones. Con este fin, se realizó pruebas de integración y pruebas de aceptación.

- **Operación de la aplicación.**

La fase tiene como propósito otorgar una versión operativa de la aplicación al usuario. Por lo tanto, al finalizar esta fase la aplicación se encuentra en pleno funcionamiento y lista para la integración con el modelo de reconocimiento.

Para la integración con el modelo de reconocimiento se contó con dos fases:

- **Integración de la aplicación con el modelo de reconocimiento.**









El propósito perseguido por esta fase es la integración de la aplicación con el modelo de reconocimiento de 11 gestos de la mano que sirven para ejecutar acciones dentro de la aplicación. A fin de cumplir con este propósito se estableció comunicación entre el entorno de desarrollo de la aplicación y el entorno de desarrollo del modelo de reconocimiento.

- **Validación de la integración.**

El objetivo principal de esta fase es la validación de la integración entre la aplicación y el modelo de reconocimiento de gestos. Con el objetivo de validar la integración se realizarán pruebas de aceptación. Las acciones que se realizan por cada gesto del modelo de reconocimiento se describen en la Tabla 1; donde se especifica el nombre del gesto, su imagen y la acción que se ejecuta dentro de la aplicación.

Tabla 1 Gestos y acciones en la aplicación

Nombre del Gesto	Gesto de la Mano		Acción dentro de la aplicación
Mano a la izquierda Wave In			Siguiete modelo de la galería a la izquierda
Mano a la derecha Wave Out			Siguiete imagen de la galería a la derecha

<p>Puño Fist</p>			<p>Cerrar modelo</p>
<p>Mano abierta Open</p>			<p>Abrir modelo</p>
<p>Doble golpe de los dedos Pinch</p>			<p>Eliminar</p>
<p>Arriba Up</p>			<p>Cámara hacia arriba del modelo</p>
<p>Abajo Down</p>			<p>Cámara hacia abajo del modelo</p>
<p>Izquierda Left</p>			<p>Cámara hacia izquierda del modelo</p>
<p>Derecha Right</p>			<p>Cámara hacia derecha del modelo</p>
<p>Apuntar al frente Forward</p>			<p>Aceptar</p>
<p>Apuntar hacia atrás Backward</p>			<p>Cancelar</p>

2. MARCO TEÓRICO

2.1 Realidad virtual

La Realidad virtual es una forma de simulación interactiva donde los usuarios exploran ambientes tridimensionales con los sentidos propios de un humano como la visión, audición y tacto.

2.1.1 Características de la realidad virtual

Dentro de las características principales se destacan la Inmersión, Navegación, Interacción, Presencia, Manipulación, Percepción, simulación. La inmersión es la apreciación por parte del usuario de estar situado en un mundo tridimensional ficticio. De acuerdo con Chirinos [5] se considera un conjunto de dos factores a la inmersión, según el autor este está compuesto de un factor tecnológico y otro psicológico generado en el usuario. Así mismo la inmersión es considerada una característica técnica de una sistema de realidad virtual por Slater y Wilbur [6].

La Navegación se define como el desplazamiento del lugar desde donde el usuario está observando el mundo virtual. Por su parte, la Interacción es la característica que permite al usuario realizar acciones en el entorno virtual y sus elementos. Las acciones se realizan mediante controles, trajes adecuados, etcétera. El no permitir la interacción en la realidad virtual simplemente podría considerarse un video en 3D.

Asimismo, la Presencia es la experiencia de estar en el entorno virtual estando físicamente en otro lugar. La presencia como tal se considera cuando un usuario de la realidad virtual es incapaz de diferenciar mediante sus sentidos el mundo real de la realidad del entorno virtual.[7]De la misma forma la Manipulación permite que la interacción en el entorno virtual y su transformación sean posibles.[5]

Respectivamente la Percepción hace referencia a que hay sistemas que ven enfocados sus esfuerzos dirigiéndolos hacia los sentidos visuales, tácticos, auditivos. Existen otros sistemas que no necesitan de interfaces sensibles a los sentidos humanos, más bien únicamente a la mente, incluso recurriendo a la imaginación del usuario para la experiencia en realidad virtual. Por otro lado la Simulación es la realización de tareas o actividades dentro de un mundo remoto o creado por una computadora.[5]

2.1.2 Tipos de realidad virtual

Existen diversos tipos de realidad virtual entre los más conocidos están los **sistemas no inmersivos**, sistemas inmersivos, sistemas semi-inmersivos. Por su parte, en los sistemas no inmersivos, el usuario interactúa con el entorno mostrado en el monitor de un computador. Esta interacción es mediante el uso de periféricos conocidos tales como el teclado, pantalla táctil, mandos, etcétera [8] .

Asimismo los **sistemas inmersivos** sumergen al usuario en la realidad virtual con la asistencia de instrumentos de hardware o software, siendo un ejemplo de cada una un HMD[9], y un sistema de sonido. Esto hará que el usuario pueda ser estimulado digitalmente como si estuviera en el mundo real[10]. Cabe recalcar que para que un sistema sea totalmente inmersivo los 5 sentidos humanos deben estar involucrados.

En los **sistemas semi-inmersivos**, el usuario se encuentra en un espacio específico como una habitación donde toda pared, piso, techo son pantallas de proyección. Además, dando paso el uso de gafas 3D por parte del usuario con las cuales se siente sumergido en esta realidad pudiendo interactuar con diferentes objetos en este entorno[11].

2.1.3 Contextos donde es trascendental la aplicación de la realidad virtual

De acuerdo con los autores [12][13][14], se han propuesto circunstancias donde la realidad virtual es adecuada a la hora de aplicarla, por ejemplo cuando se realiza simulaciones. De igual forma, cuando el contexto real resulta de difícil acceso, porque es peligroso, no es de fácil conseguir ese contexto, o no es muy real. Además, esto evitará errores que serían desastrosos en el mundo real tales como contaminación, envenenamiento, explosiones, etc. Asimismo, brinda soporte para transmitir conocimiento cuando se quiere dar a conocer algo del mundo real. En igual forma cuando se cuenta con un presupuesto limitado, muchas veces la realidad virtual será menos costosa que la creación del contexto en el mundo real. Por ejemplo, cuando la construcción de contextos para mejorar destrezas que en la realidad no se consiguen fácilmente. También, a la hora de la inclusión de personas con capacidades especiales para que realicen experimentos que no pueden realizarse de otra forma.

2.2 Modelos 3D

Un modelo 3D es el resultado del despliegue de una interpretación matemática de alguna superficie tridimensional de un objeto a través de un software especializado. Existen dos tipos principales de modelos 3D: B-spline racional no uniforme y Poligonal [15]. B-spline es un modelo tridimensional de superficie suave creado mediante curvas Bézier y la ayuda de

una aplicación de software. B-spline tiene el nivel más alto de precisión matemática siendo usados para el modelado en ingeniería y diseño automotriz.

Los modelos Poligonales también llamados “mallas” son ampliamente utilizados en industrias como el cine, animación y juegos. Estos modelos están compuestos por Caras, Bordes, Vértices, Texturas y Shaders. Los modelos poligonales tienen facetas, lo que significa que la superficie del modelo está compuesta por cientos o miles de caras geométricas.

- Bordes: Es cualquier punto en la superficie de un modelo 3D donde se encuentran dos caras poligonales.
- Vértices: Es el punto de intersección entre tres o más aristas.
- Shaders: Son un conjunto de instrucciones aplicadas a un modelo 3D que le permite a la computadora saber cómo debe mostrarse.
- Texturas: Son archivos de imágenes bidimensional que se asigna a la superficie 3D de los modelos a través de un proceso de asignación de texturas.

Los modelos 3D utilizados en este Trabajo de Integración Curricular son poligonales en el formato FBX. El formato FBX es un formato de intercambio de activos 3D que facilita la transacción de datos de mayor fidelidad entre software propietarios de Autodesk y de terceros [16]. FBX posee características como interoperabilidad, flujo de trabajo más eficiente y el intercambio de datos más fácil.

3. METODOLOGÍA

3.1 Especificación de requisitos y restricciones

Se realizó una serie de reuniones con el director del Plan de diseño de trabajo de integración curricular, en las cuales se fijaron los requisitos y restricciones imprescindibles para la realización de la aplicación.

3.1.1 Requisitos funcionales

Los Requisitos funcionales para expresar la intención que tiene el usuario en la aplicación fueron modelados en casos de uso contenidos en la Anexo I. Cada caso de uso consta de nombre, descripción y un actor para brindar trazabilidad de requisitos durante todas las fases de la metodología.

Existen 13 casos de uso que representan los requisitos del usuario. El diagrama de casos de uso expresa las funciones y la forma que el usuario interactúa con ellas. Existen ampliaciones entre casos de uso conocida como relación “extend”. También existe la relación de inclusión entre dos casos expresada como relación “include”. El caso de uso 1 (CU1) expuesto a continuación sirve de ejemplificación de la estructura de estos. El detalle de los 13 casos de uso se encuentra en el Anexo III.

- CU1. Ingresar al visor de modelos 3D

Descripción: Se podrá ingresar a la galería de modelos 3D.

Actor: Usuario.

Flujo: Véase Tabla 2.

Tabla 2 Flujo CU1

Paso	Actor	Paso	Sistema	Excepción
1	Inicia la aplicación	2	Presenta escena inicial de la aplicación	

3.1.2 Mapeo de gestos con tareas en la aplicación

Los gestos están destinados a ejecutar tareas en la aplicación. Las tareas fueron elegidas de acuerdo con la facilidad de relacionar el gesto con la actividad en la vida cotidiana. Esto con la finalidad de permitir al usuario una memorización de gestos más rápida. De acuerdo con el usuario, el mapeo óptimo es el observado en la Tabla 1.

3.1.3 Requisitos no funcionales

La programación de la aplicación se realizó con C# como lenguaje de programación. La versión de Unity elegida para la construcción de la aplicación fue 2021.2.3f1. El IDE de desarrollo utilizado fue Visual Studio 2022. El formato de modelo 3D de tipo polinomial elegido fue FBX.

3.2 Diseño

En este proceso se describió los nombres de espacio "SpaceName", las escenas, el diagrama de clases y los complementos. Cabe destacar que el diagrama de clases es fundamental ya que modela los requisitos expresados por el usuario.

3.2.1 Diagrama de paquetes

Los nombres de espacios se utilizaron para declarar un conjunto de clases relacionadas con el fin de organizar elementos de código. Para el Visor de modelos 3D se definieron cuatro nombres de espacio. Los nombres de espacios son "GestureCommands", "ListeningCommands", "LoadParts" y "Tools".

Para la aplicación se construyeron cuatro escenas, cada una con un objetivo propio y funcionalidades distintas. La escena "Start" es la que se muestra al ejecutar la aplicación y tiene por objetivo presentar 3 opciones de navegación a elegir por el Usuario. La escena "Instructions" muestra las diferentes acciones que se pueden ejecutar en cada escena. La escena "Galery" muestra todos los modelos 3D de exhibición con los que cuenta la aplicación. La escena "Model" permite la visualización del modelo seleccionado y la ejecución de acciones que mejoren el ángulo de visión del usuario.

Desde la escena Start se puede navegar hacia las escenas "Instructions" y "Galery". Asimismo, desde la escena "Galery" se puede navegar hacia la escena "Start" y "Model". Por otro lado, desde la escena "Model" se puede navegar hacia la escena "Galery". Por su lado la escena Instructions solo permite navegar hacia la escena "Start".

Se modificó y mejoró el aspecto visual en las escenas de Unity creando los complementos "SkyBox", "Blocks", "Images", "T-M-Models" y "Models". El complemento "T-M-Models" contiene las texturas y materiales respectivas por cada modelo. El complemento "Model" contiene los modelos 3D en formato FBX. El complemento "SkyBox" contiene elementos para el entorno de realidad virtual de cada escena. El complemento "Images" y "Blocks" contienen imágenes 2D para las texturas de los bloques donde se muestra información.

3.2.2 Diagrama de clases

El diagrama de clases visualiza la relación que existe entre las clases de la aplicación. Al utilizar esta herramienta se posibilita el acoplamiento de los diferentes objetos en la aplicación. Adicionalmente, el diagrama permite entregar una parte de la arquitectura en un lenguaje de modelado Unificado UML. El diagrama se observa en el Anexo II. El diagrama de clases está conformado por cuatro nombres de espacios, los cuales se describen a continuación:

- ListeningCommands

El Nombre de espacio “ListeningCommands” cuenta con las clases observadas en la Figura 1, las mismas que son necesarias para la recepción y gestión de los gestos por la aplicación.

GesCGallery	GesCStart	GesCInstructions	GesCModel
+ command: ICommand + listener: TcpListener + msg: String + visible: Bool	+ command: ICommand + listener: TcpListener + msg: String + visible: Bool	+ command: ICommand + listener: TcpListener + msg: String	+ command: ICommand + listener: TcpListener + msg: String
- Start(): Void - Update(): Void + setCommand(ICommand):Void + waveInGesture():Void + waveOutGesture():Void + openGesture():Void + pinchGesture():Void + forwardGesture():Void + backwardGesture():Void + fistGesture():Void	- Start(): Void - Update(): Void + setCommand(ICommand):Void + openGesture():Void + pinchgesture():Void + fistGesture():Void + forwardGesture(): Void + backwardGesture(): Void	- Start(): Void - Update(): Void + setCommand(ICommand):Void + fistGesture():Void	- Start(): Void - Update(): Void + setCommand(ICommand):Void + leftGesture():Void + rightGesture():Void + upGesture():Void + downGesture():Void + fistGesture():Void

Figura 1 Clases de ListeningCommands

- Commands

El Nombre de espacio “Commands” utilizó el patrón de comportamiento Comando. El mismo permite brindar mayor facilidad al desarrollador para agregar nuevos comandos sin cambiar el código existente. Las clases que conforman este nombre de espacio se muestran en la Figura 2.

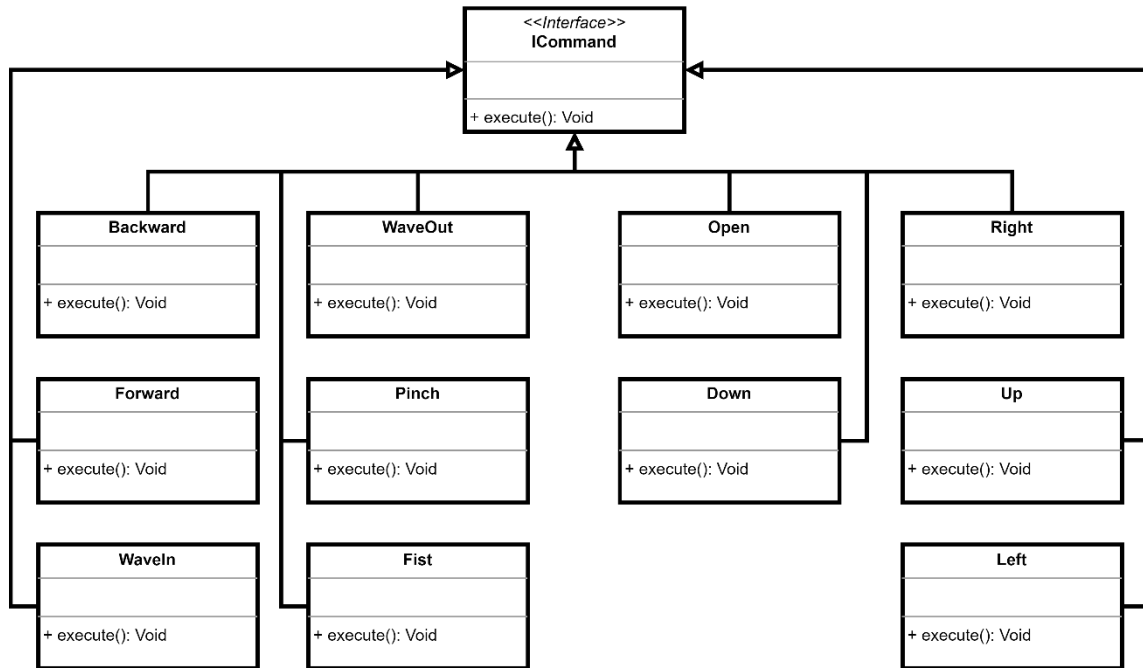


Figura 2 Clases de Commands

- LoadParts

El Nombre de espacio “LoadParts” se conformó con clases que se visualizan en la Figura 3. Estas clases permiten cargar los elementos y otorgan la lógica a los mismos.

LoadModels	LoadApplication	LoadInstructions	LoadModel
+ instance: LoadModels + models[]: Model + modelToOpen: Model + answerToDelete: bool +options: GameObject -visualizer: Visualizer -instantiatedObject: GameObject -cursorObject: Model -modelNumber: Int	+ instance: LoadApplication +answerToClose: bool +options: gameObject	+ instance: LoadModels	+ instance: LoadModel +instantiateObject: GameObject -modelSelected: Model -degrees: Int
- Start(): Void -cUpdate(): IEnumerator +showModels():Void +loadModelObjects(ArrayList):Void +loadModelFromResource(string):Model +moveRight():Void +moveLeft():Void +goToModel():Void +goToStart():Void +optionsDelete():Void +delete():Void +resetScene():Void +noDelete():Void +accept():Void +cancel():Void +cursor(int):Void	- Start(): Void +goToGalery():Void +closeApp():Void +goToInstructions():Void +noClose(): Void +optionsClose(): Void +accept(): Void +cancel(): Void	- Start(): Void -goToStart(): Void	- Start(): Void +turnRight():Void +turnLeft():Void +turnUp():Void +turnDown():Void +goToGalery():Void

Figura 3 Clases de LoadParts

- Tools

El Nombre de espacio “Tools” se compuso por clases observadas en la Figura 4. Las clases de este nombre de espacio permiten el tratamiento de los modelos. En este punto, se colocan las escalas adecuadas por cada modelo, se define los modelos que se cargarán en el nombre de espacio “LoadParts”, entre otros.

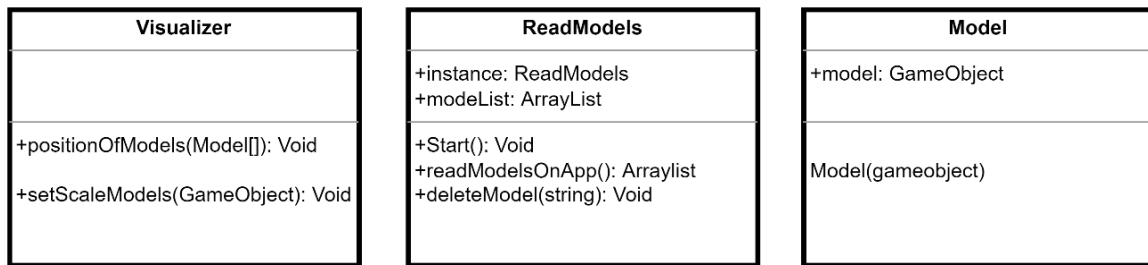


Figura 4 Clases de Tools

Cabe recalcar que se utilizó el patrón de diseño “Singleton” en diversas clases. Las clases de los nombres de espacio “ListeningCommands”, “LoadParts” y la clase “ReadModels” utilizan este patrón. Por su parte, “Singleton” permite acceder fácilmente a cada parte de la aplicación donde se use este tipo de diseño.

3.3 Desarrollo

En esta fase, la arquitectura prevista por la fase de Diseño se implementó mediante programación del código fuente de la aplicación. Para la codificación se utilizó el IDE de desarrollo Visual Studio 2022 en su versión 17.2. Los estándares de codificación utilizados para clases e interfaces fue Pascal Case y para métodos, argumentos y variables el estándar Camel Case.

La aplicación se desarrolló en el sistema operativo Windows 11 Pro [17], con los cascos de realidad virtual Oculus Rift[18] y el motor de desarrollo 3D Unity [19]. La Tabla 3 muestra mayores especificaciones técnicas del ambiente de desarrollo.

Tabla 3 Especificaciones técnicas del ambiente de desarrollo de la aplicación

Versión Unity	2021.2.3f1
GPU	NVIDIA GeForce GTX 1650
CPU	Intel Core i7-10700F
RAM	16 GB
Disco HDD	932 GB
Disco SSD	233 GB
Motherboard	Prime H410M-E

Los modelos para la exhibición utilizados fueron descargados de diferentes plataformas mostradas en la Tabla 4, donde se permite el uso y descarga gratuita de los mismos. Las unidades y dimensiones originales exactas del modelo dependen totalmente del autor. El formato de los modelos es FBX. Por otro lado, se construyó interfaces simples para mejorar la experiencia en realidad virtual y se buscó la claridad del objetivo perseguido por el usuario en cada escena. Además, se optimizó la utilización de recursos para la fluidez en la navegación entre escenas.

Tabla 4 Plataformas para descarga de modelos 3D

Nombre	Enlace
Sketchfab	https://sketchfab.com/
Free3D	https://free3d.com/
cgtrader	https://www.cgtrader.com/free-3d-models
Turbosquid	https://www.turbosquid.com/es/

Se utilizaron tres paquetes dentro de Unity a fin de desarrollar las escenas para realidad virtual. El primer paquete es “Oculus” que fue utilizada para la integración con las gafas de realidad virtual. “TextMesh Pro” es utilizada para el uso de textos en las escenas mostradas al usuario. Finalmente, “XR” es utilizada para simular la realidad virtual.

3.4 Verificación y validación de requisitos

En esta fase se determinó que la aplicación cumple con los requisitos definidos en la fase de Requisitos y restricciones.

3.4.1 Pruebas de Integración

Con el objetivo de verificar la integración entre módulos, se desarrolló pruebas de integración ascendentes. En este tipo de pruebas, se realizan evaluaciones desde módulos de nivel inferior hacia los de nivel superior. El conjunto de pruebas de integración para los módulos de la aplicación se observa en la Tabla 5. Dicha tabla relaciona el módulo de la

aplicación intervenido, el caso de prueba con la descripción y el resultado que se espera de realizar la prueba.

Tabla 5 Pruebas de integración de los diferentes módulos de la aplicación

Id	Módulos intervenidos	Caso de Prueba	Descripción de caso de prueba	Resultado Esperado
1	<ul style="list-style-type: none"> Nombre de espacios "Tools" y "LoadParts" 	Verifique el enlace del módulo de "Tools" con "LoadParts"	Cargar modelos 3D en escena "Galery" y "Model"	La aplicación debe mostrar al menos un modelo 3D cargando en ambas escenas.
2	<ul style="list-style-type: none"> Nombre de espacios "Commands" y "ListeningCommands" 	Verifique el enlace del módulo "Commands" con "ListeningCommands"	Ejecutar al menos una tarea que el gesto ejecute.	La aplicación debe ejecutar la acción designada para cada gesto en cada escena.
3	<ul style="list-style-type: none"> Nombre de espacios "LoadParts" y "Commands" 	Valide el enlace del módulo "LoadParts" con "Commands"	Ejecutar tareas en la aplicación.	La aplicación debe ejecutar acciones mediante el ingreso de comandos del usuario.
4	<ul style="list-style-type: none"> Escenas "Star", "Star", "Instructions", "Galery" y "Model" 	Verifique el enlace entra las escenas "Start" y "Instructions"	Navegar entre las escenas "Start", "Instructions", "Galery", "Model".	La aplicación debe permitir la navegación entre escenas.
5	<ul style="list-style-type: none"> Complementos "SkyBox", "Blocks", "T-M-Model", "Images" y "Models" Escenas "Star", "Instructions", "Galery" y "Model" 	Verifique el enlace entre los complementos y las escenas de la aplicación	Observar los componentes visuales a través de todas las escenas de la Aplicación	La aplicación debe colocar cada complemento en la escena adecuada, otorgando retroalimentación visual al usuario.

3.4.2 Pruebas de Aceptación

Con el objetivo de entregar un producto totalmente completo respecto a los requisitos inicialmente planteados se realizó pruebas de aceptación. Estas pruebas están basadas en los requisitos realizados por el usuario en la primera fase de esta metodología. Las pruebas de aceptación se observan en la Tabla 6.

Tabla 6 Pruebas de aceptación de los requisitos de la aplicación

Id	Caso de Uso	Resultado esperado
1	Ingresar al visor de modelos 3D	La aplicación debe ejecutarse
2	Visualización de instrucciones-gestos	La aplicación debe mostrar la escena de Instrucciones
3	Salir de la Escena/Aplicación	La aplicación debe cerrarse si se encuentra en la escena "Start"
		La aplicación debe salir de la escena desde cualquier escena excepto "Start"
4	Visualización de modelos 3D	La aplicación debe mostrar la escena "Galery"
5	Escoger siguiente modelo hacia la derecha	La aplicación debe permitir la selección del modelo que se encuentra a la derecha del actualmente seleccionado
6	Escoger siguiente modelo hacia la izquierda	La aplicación debe permitir la selección del modelo que se encuentra a la izquierda del actualmente seleccionado
7	Elegir un modelo 3D	La aplicación debe permitir la navegación entre modelos hacia la derecha o izquierda del modelo actualmente seleccionado
8	Eliminar un modelo 3D	La aplicación debe permitir la eliminación de un modelo seleccionado
9	Abrir un modelo 3D	La aplicación debe permitir abrir un modelo seleccionado
10	Girar modelo hacia la izquierda	La aplicación debe permitir girar el modelo 3D hacia la izquierda
11	Girar modelo hacia la derecha	La aplicación debe permitir girar el modelo 3D hacia la derecha
12	Girar modelo hacia arriba	La aplicación debe permitir girar el modelo 3D hacia arriba
13	Girar modelo hacia abajo	La aplicación debe permitir girar el modelo 3D hacia abajo

3.5 Operación de la aplicación

La compilación de la aplicación se realizó para el sistema operativo Windows en una arquitectura de 64 bits la cual soporta la realidad virtual. Asimismo, lleva el nombre de "VisorDeModelos3D". Además, la aplicación tiene un icono, una imagen de fondo para el inicio en 2D y una imagen inicial para la aplicación en realidad virtual. Las imágenes se observan en la Figura 5 de izquierda a derecha.

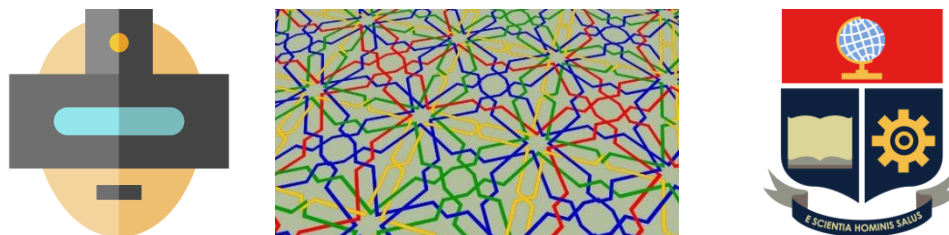


Figura 5 Imágenes para la construcción del ejecutable de la aplicación

Inicialmente, para simular el uso de gestos y como verificación de funcionamiento, se utilizó teclas como entrada. Posteriormente, la aplicación se integró con el modelo de reconocimiento de gestos el cual devuelve una etiqueta de un gesto como entrada en lugar de una tecla. La información de la tecla presionada es enviada desde Matlab mediante sockets. Las teclas y sus respectivos gestos se indican en la Tabla 7.

Tabla 7 Mapeo de teclas y gestos

Tecla	Gesto
A	Left
D	Right
W	Up
S	Down
→	Wave Out
←	Wave In
X	Fist
Suprimir	Pinch
Enter	Forward
Z	Backward
Espacio	Open

Para la operación de la aplicación es necesario la instalación del software de las gafas para realidad virtual “Oculus Rift”. La misma que se debe estar operativa previa a la ejecución de la aplicación. La escena “Galery” y “Model” se muestran en la Figura 6. Todas las interfaces gráficas del visor se encuentran en el Anexo IV. La aplicación se encuentra disponible para la descarga en el Anexo V.

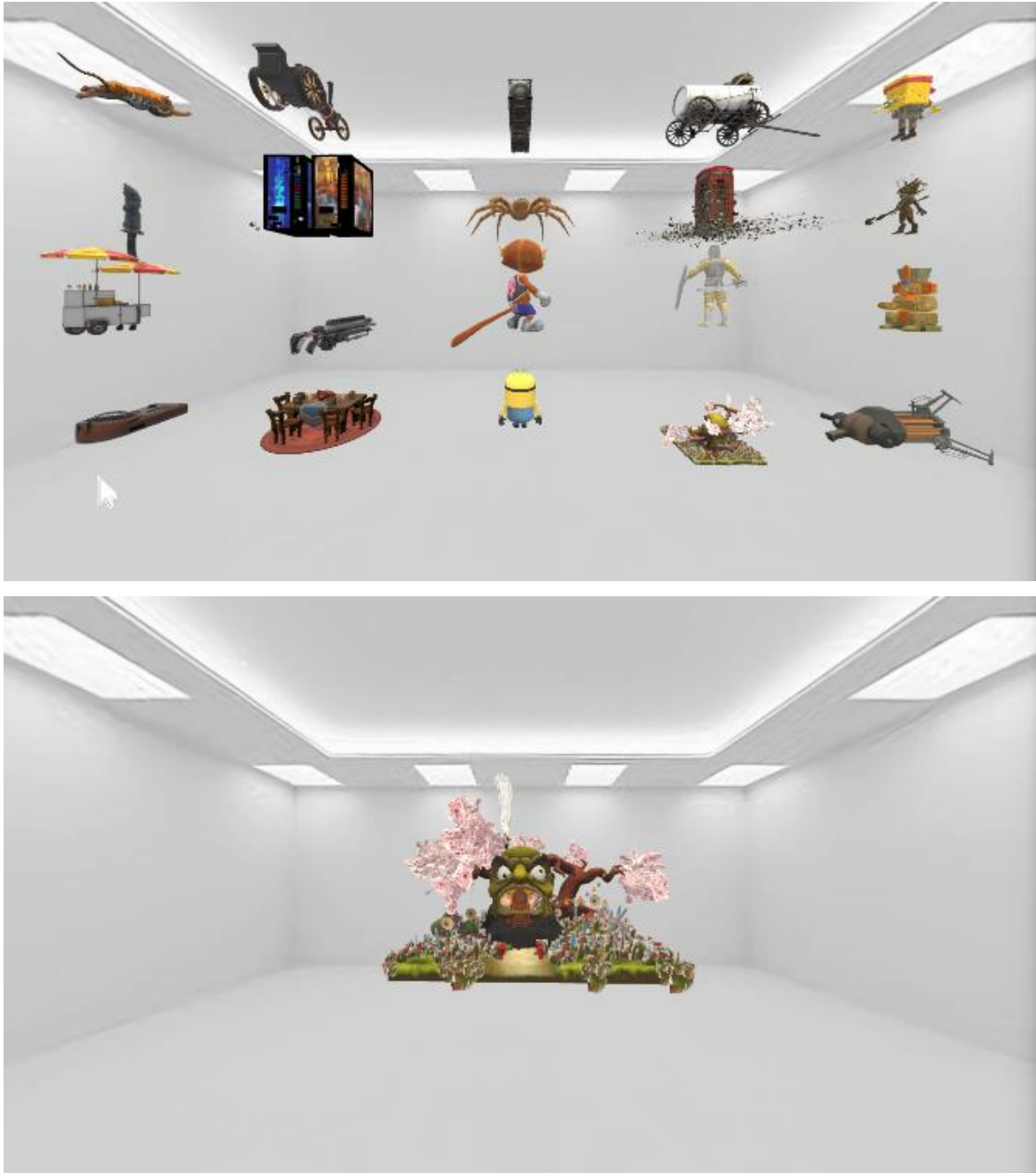


Figura 6 Escena Galery y Model

3.6 Integración con el modelo de reconocimiento de gestos

Con el objetivo de integrar ambos subcomponentes se estableció comunicación entre el entorno de desarrollo de la aplicación y del modelo de reconocimiento. Los entornos de desarrollo son Unity y Matlab, respectivamente. La integración utilizó la tecnología de sockets. Los roles fueron que Unity sea el servidor y Matlab el cliente. La utilización de la librería System.Net [20] y System.Net.Sockets [21] fueron necesarias para Unity. La

inclusión del tipo de variable adecuada fue la única necesidad para Matlab. La dirección IP utilizada fue "127.0.0.1", el puerto fue 55001 y como tiempo de espera (Time out) diez segundos.

3.7 Validación de la integración con el modelo de reconocimiento de gestos

Con el fin de desarrollar esta etapa se realizó pruebas de aceptación, tal como se observa en la Tabla 8. Cada prueba de aceptación consta de un Identificador, la acción que se debe realizar, el identificar del caso de Uso, y el resultado esperado. Además de realizar la integración con el modelo de reconocimiento de gestos, se realizó un script en Matlab para que se pueda ingresar el comando por teclado. Este script se halla en el Anexo VI.

Tabla 8 Pruebas de aceptación de la Integración de la aplicación y el modelo de reconocimiento de gestos

Id	Acción	CU	Resultado esperado
1	Ejecución Manual	1	La aplicación debe ejecutarse
2	Pinch	2	La aplicación debe mostrar la escena de Instrucciones
3	Fist	3	La aplicación debe cerrarse si se encuentra en la escena "Start"
			La aplicación debe salir de la escena desde cualquier escena excepto "Start"
4	Open	4	La aplicación debe mostrar la escena "Galery"
5	Wave Out	8	La aplicación debe permitir la selección del modelo que se encuentra a la derecha del actualmente seleccionado
6	Wave In	9	La aplicación debe permitir la selección del modelo que se encuentra a la izquierda del actualmente seleccionado
7	Pinch	6	La aplicación debe permitir la eliminación de un modelo seleccionado
8	Open	7	La aplicación debe permitir abrir un modelo seleccionado
9	Left	10	La aplicación debe permitir girar el modelo 3D hacia la izquierda
10	Right	11	La aplicación debe permitir girar el modelo 3D hacia la derecha
11	Up	12	La aplicación debe permitir girar el modelo 3D hacia arriba
12	Down	13	La aplicación debe permitir girar el modelo 3D hacia abajo
13	Forward	3	La aplicación debe cerrarse.
	Backward		La aplicación no debe cerrarse.
14	Forward	6	La aplicación debe eliminar el modelo seleccionado
	Backward		La aplicación no debe eliminar el modelo seleccionado

4. RESULTADOS, CONCLUSIONES Y RECOMENDACIONES

4.1 Resultados

Se evaluó la integración entre módulos y la aceptación del usuario en diferentes pruebas expuestas respectivamente en las Secciones 3.4.1 “Pruebas de Integración” y 3.4.2. “Pruebas de Aceptación”. Los resultados de la sección 3.4.1 “Pruebas de Integración” son los observados en la Tabla 9. Cada resultado está relacionado con el identificador de la Prueba de Integración (PI).

Tabla 9 Resultados de las Pruebas de Integración del Visor de Modelos 3D

Id-PI	Resultado Obtenido
1	La aplicación muestra veinte modelos 3D en la escena de “Galery” y un modelo 3D en la escena “Model”
2	La aplicación ejecuta la acción designada para cada gesto en cada escena.
3	La aplicación ejecuta acciones mediante el ingreso de comandos del usuario.
4	La aplicación permite la navegación entre escenas.
5	La aplicación coloca cada complemento en la escena adecuada, otorgando retroalimentación visual al usuario.

La valoración a los resultados de la Tabla 9 son satisfactorios. Este éxito en las pruebas indica que los módulos de la aplicación se encuentran fusionados de una forma adecuada. Además, se demuestra que el acoplamiento entre módulos está dando solución a los requisitos del usuario eficazmente.

Continuando en la sección 3.4.2 “Pruebas de Aceptación” se obtuvo los resultados de la Tabla 10. Estos resultados aseguran que el usuario experimentará una solución totalmente acoplada a sus necesidades. Además, el resultado es un indicador que la metodología fue aplicada de acuerdo con su respaldo teórico científico. Cada resultado está relacionado con el identificador de la Prueba de aceptación (Pa).

Tabla 10 Resultados de las Pruebas de Aceptación del Visor de Modelos 3D

Id-Pa	Resultado Obtenido
1	La aplicación se ejecuta
2	La aplicación muestra la escena de Instrucciones
3	La aplicación se cierra
	La aplicación sale de la escena
4	La aplicación muestra la escena "Galery"
5	La aplicación permite la selección del modelo que se encuentra a la derecha del actualmente seleccionado
6	La aplicación permite la selección del modelo que se encuentra a la izquierda del actualmente seleccionado
7	La aplicación permite la navegación entre modelos hacia la derecha o izquierda del modelo actualmente seleccionado
8	La aplicación elimina el modelo seleccionado
9	La aplicación permite abrir un modelo seleccionado
10	La aplicación permite girar el modelo 3D hacia la izquierda
11	La aplicación permite girar el modelo 3D hacia la derecha
12	La aplicación permite girar el modelo 3D hacia arriba
13	La aplicación permite girar el modelo 3D hacia abajo

A fin de comprobar la integración con el modelo de reconocimiento de gestos se realizaron pruebas de aceptación. Dichas pruebas fueron planteadas en la sección 3.7 “Validación de la integración con el modelo de reconocimiento de gestos”, los resultados de estas pruebas son los mostrados en la Tabla 11. Los resultados dejan en evidencia que los gestos realizan las acciones mapeadas en la aplicación. Además, respaldan los beneficios de utilizar un enfoque en Cascada.

Tabla 11 Resultados de las Pruebas de Aceptación en la integración del modelo de reconocimiento de gestos

Id-Pa	Acción	Resultado Obtenido
1	Ejecución Manual	La aplicación se ejecuta
2	Pinch	La aplicación muestra la escena de Instrucciones
3	Fist	La aplicación se cierra
		La aplicación sale de la escena
4	Open	La aplicación muestra la escena "Galery"
5	Wave Out	La aplicación permite la selección del modelo que se encuentra a la derecha del actualmente seleccionado
6	Wave In	La aplicación permite la selección del modelo que se encuentra a la izquierda del actualmente seleccionado
7	Pinch	La aplicación elimina el modelo seleccionado
8	Open	La aplicación permite abrir un modelo seleccionado
9	Left	La aplicación permite girar el modelo 3D hacia la izquierda
10	Right	La aplicación permite girar el modelo 3D hacia la derecha
11	Up	La aplicación permite girar el modelo 3D hacia arriba
12	Down	La aplicación permite girar el modelo 3D hacia abajo
13	Forward	La aplicación se cierra
	Backward	La aplicación no se cierra, y permanece en la escena actual
14	Forward	La aplicación elimina el modelo seleccionado
	Backward	La aplicación no elimina el modelo seleccionado

Al finalizar todas las fases del proyecto se obtuvo la aplicación integrada con el modelo de reconocimiento de gestos. La forma de utilizar la aplicación integrada modelo de reconocimiento se encuentra en el Anexo VII. El código fuente de la integración se encuentra en el Anexo VIII. El video del funcionamiento de la aplicación se encuentra en el Anexo IX.

4.2 Conclusiones

El diseño e implementación de la aplicación de Visor de Modelos 3D controlado mediante 11 gestos de la mano en realidad virtual se desarrolló de forma satisfactoria. Se cumplieron con todos los requisitos expuestos en la fase de requisitos y restricciones. De igual manera, se demostró que el enfoque metodológico de cascada fue la más apropiada para este proyecto. Este enfoque metodológico permite tener una estructura clara y al ser un modelo secuencial permite una implementación más simple. En este documento se encuentra evidenciado el proceso secuencial característico de esta metodología.

La etapa de descubrimiento de las necesidades del usuario y la revisión del estado del arte fue de importancia clave en este proyecto, ya que permitió fijar las principales características que tiene la aplicación. Estas características, así como los requisitos no funcionales, se obtuvieron mediante sesiones con el director del Trabajo de Integración.

La aplicación de visión de modelos se encuentra satisfactoriamente integrada con el modelo de reconocimiento de gestos. La integración de las aplicaciones ha sido afectada por efectos externos tales como recursos del computador, temperatura y batería del Myo Arm Band y la forma en que se tomó las muestras. Ante esa problemática se planteó soluciones tales como tener al menos a la mitad la batería del Myo Arm Band, desconectar el dispositivo en cada ejecución de la aplicación y evitar consumir recursos del sistema de forma innecesaria. La solución a nivel de software es colocar un filtro en la integración para que solo se envíe un gesto y no una seguidilla de gestos del mismo tipo. Además, se tomaron muestras a el usuario que pretende utilizar la aplicación para que el reconocimiento del gesto sea mucho más acertado.

4.3 Recomendaciones

Se recomienda, de ser posible, el uso de Visor de Modelos 3D inalámbricos, ya que permite mejorar la movilidad del usuario. Adicionalmente, el desarrollo propuesto puede mejorarse mediante la opción de agregar modelos a la galería. Es decir, brindar tareas de gestión de los modelos desde su interfaz gráfica. Asimismo, aumentar el soporte para distintos formatos de modelos tridimensionales es una funcionalidad que brindará escalabilidad al usuario.

Para mejorar la interfaz gráfica se podría utilizar componentes creados exclusivamente para el Visor de modelos 3D. Estos componentes pueden ser escenas con imágenes en formatos flexibles como “svg”, o “png”. Así mismo se podría seguir estándares para interfaces gráficas de realidad virtual. Además, con el objetivo de brindar una aplicación accesible se podría tomar en cuenta conceptos ya realizado en la WCAG para páginas web.

Se recomienda que el equipo donde se ejecute la aplicación junto con el modelo de reconocimiento de gestos tenga al menos 16 GB de RAM, el procesador debe contar con al menos 8 núcleos y una tarjeta gráfica dedicada de 4 GB o superior. Al ejecutar la aplicación en cualquier máquina el usuario verificará que tenga recursos suficientes para ejecutar la aplicación junto con el modelo de reconocimiento de gestos.

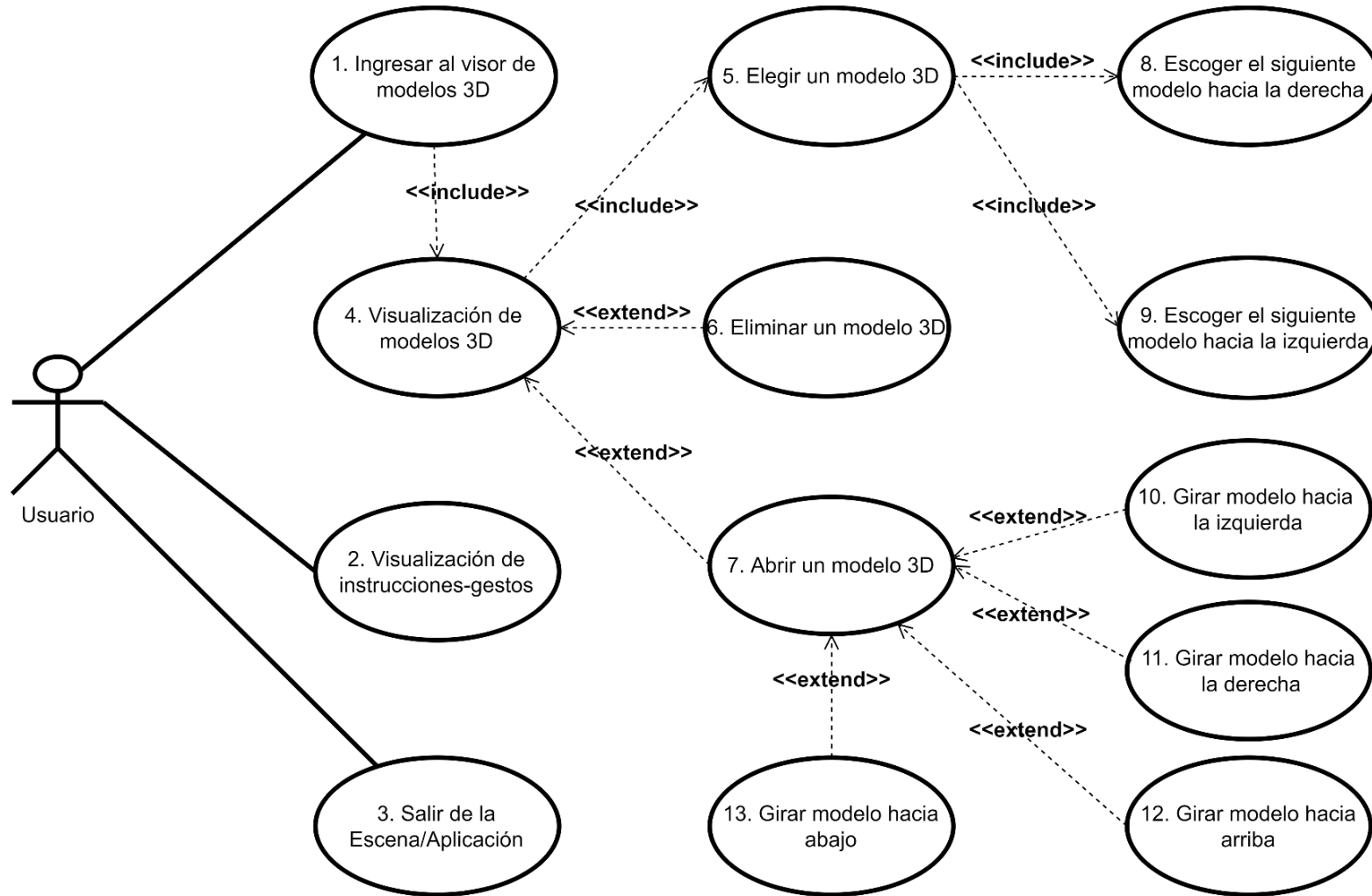
5. REFERENCIAS BIBLIOGRÁFICAS

- [1] K. Raaen and I. Kjellmo, "Measuring latency in virtual reality systems," in Entertainment Computing - ICEC 2015, K. Chorianopoulos, M. Divitini, J. Baalsrud Hauge, L. Jaccheri, and R. Malaka, Eds. Cham: Springer International Publishing, 2015, pp. 457–462.
- [2] I. Sommerville, Software Engineering, ser. International Computer Science Series. Pearson, 2011. [En línea]. Disponible: <https://books.google.com.ec/books?id=l0egcQAACAAJ>
- [3] P. Senin, "Dynamic time warping algorithm review," Information and Computer Science Department University of Hawaii at Manoa Honolulu, USA, vol. 855, no. 1-23, p. 40, 2008.
- [4] K. Taunk, S. De, S. Verma, and A. Swetapadma, "A brief review of nearest neighbor algorithm for learning and classification," in 2019 International Conference on Intelligent Computing and Control Systems (ICCS), 2019, pp. 1255–1260.
- [5] Y. Chirinos Delfino, "La realidad virtual como mediadora de aprendizajes: Desarrollo de una aplicación móvil de realidad virtual orientada a niños," Revista Iberoamericana de Tecnología en educación y Educación en Tecnología, no. 27, pp. 98–99, 2020.
- [6] M. Slater and S. Wilbur, "A framework for immersive virtual environments five: Speculations on the role of presence in virtual environments," Presencia: Teleoper. Virtual Environ., vol. 6, no. 6, p. 603–616, 12 1997.
- [7] D. B. Chertoff, S. L. Schatz, R. McDaniel, and C. A. Bowers, "Improving Presence Theory Through Experiential Design," Presencia: Teleoperators and Virtual Environments, vol. 17, no. 4, pp. 405–413, 08 2008.
- [8] E. Ai-Lim Lee, K. W. Wong, and C. C. Fung, "How does desktop virtual reality enhance learning outcomes? a structural equation modeling approach," Computers Education, vol. 55, no. 4, pp. 1424–1442, 2010.
- [9] G. Inc, "Definition of head mounted display gartner information technology glossary." [En línea]. Disponible: <https://www.gartner.com/en/information technology/glossary/headmounted-displays-hmd>
- [10] B. Sousa Santos, P. Dias, A. Pimentel, J.-W. Baggerman, C. Ferreira, S. Silva, and J. Madeira, "Head-mounted display versus desktop for 3d navigation in virtual reality: A user study," Multimedia Tools Appl., vol. 41, no. 1, p. 161–181, 01 2009.
- [11] L. Freina and M. Ott, "A literature review on immersive virtual reality in education: state of the art and perspectives," in The international scientific conference elearning and software for education, vol. 1, no. 133, 2015, pp. 10–1007.
- [12] R. Ghanbarzadeh and A. Ghapanchi, Applied Areas of Three Dimensional Virtual Worlds in Learning and Teaching, 01 2019, pp. 172–192.
- [13] B. Dalgarno, J. Hedberg, and B. Harper, "The contribution of 3D environments to conceptual understanding," 01 2002, pp. 149–158.
- [14] F. Mantovani, "Vr learning: Potential and challenges for the use of 3d environments in education and training," 01 2001.
- [15] J. Slick, "3d model components-vertices, edges, polygons amp; more," Mar 2021. [En línea]. Disponible: <https://www.lifewire.com/3d-modelcomponents-1952>
- [16] "Fbx: Adaptable file formats for 3d animation software," Jun 2021. [En línea]. Disponible: <https://www.autodesk.com/products/fbx/overviewintro>
- [17] M. Halsey, "Top tips for getting the very best from windows 11," in Windows 11 Made Easy. Springer, 2022, pp. 215–237
- [18] P. R. Desai, P. N. Desai, K. D. Ajmera, and K. Mehta, "A review paper on oculus rift-a virtual reality headset," 2014.
- [19] U. Technologies. [En línea]. Disponible: <https://unity.com/>
- [20] M. Namespaces, "System.net namespace." [En línea]. Disponible: <https://docs.microsoft.com/en-us/dotnet/api/system.net?view=net-6.0>
- [21] M. Sockets, "System.net.sockets namespace." [En línea]. Disponible: <https://docs.microsoft.com/en-us/dotnet/api/system.net.sockets?view=net-6.0>

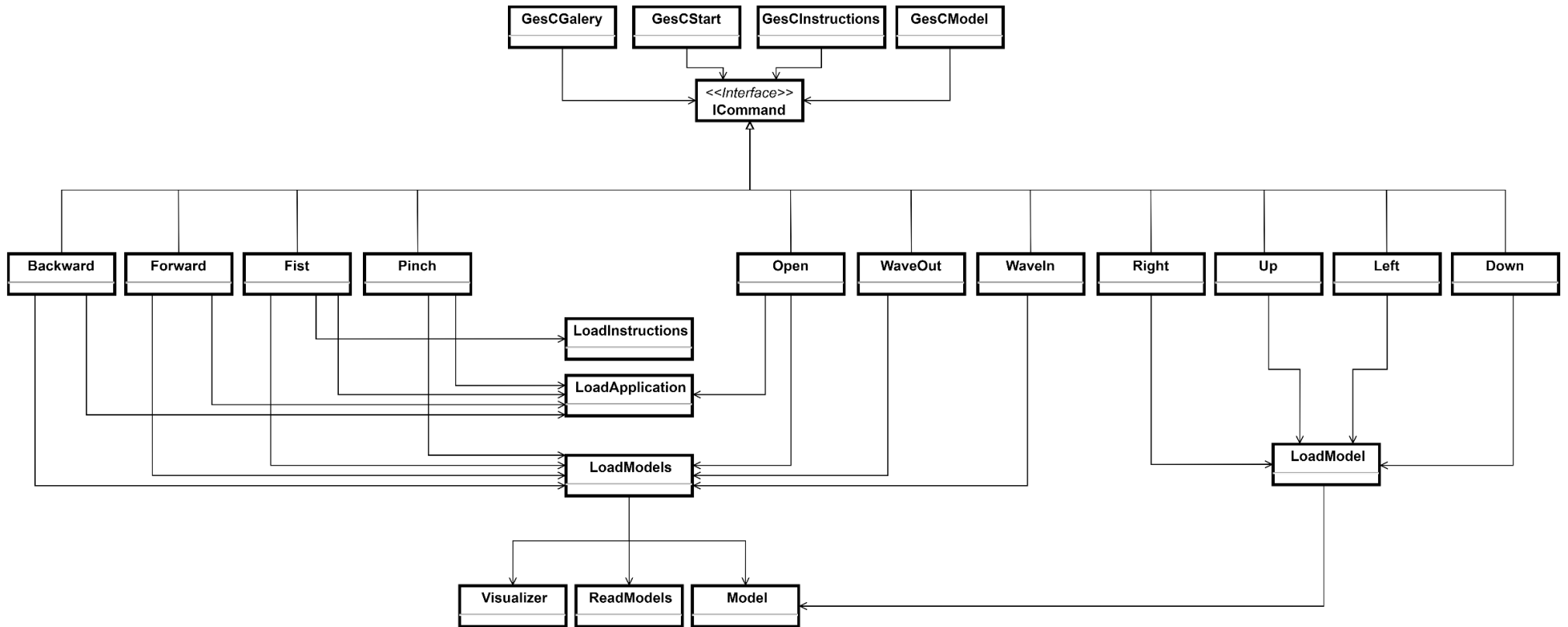
6. ANEXOS

ANEXO I Diagrama de casos de uso de la aplicación.....	25
ANEXO II Diagrama de clases de la aplicación.....	26
ANEXO III Descripción de casos de uso	27
ANEXO IV Manual de usuario de la aplicación	32
ANEXO V Código fuente de la aplicación	32
ANEXO VI Código fuente script Matlab ejecución de tareas mediante el uso del teclado.....	32
ANEXO VII Manual de uso de la aplicación integrada con el modelo de reconocimiento de gestos.....	32
ANEXO VIII Código fuente de la integración con la aplicación y el modelo de reconocimiento de gestos.....	32
ANEXO IX Video del funcionamiento de la aplicación integrada con el modelo de reconocimiento de gestos.....	32

ANEXO I Diagrama de casos de uso de la aplicación



ANEXO II Diagrama de clases de la aplicación



ANEXO III Descripción de casos de uso

Se presenta la descripción detallada de los 13 casos de uso.

- CU1. Ingresar al visor de modelos 3D

Descripción: Se podrá ingresar a la galería de modelos 3D.

Actor: Usuario.

Flujo: Véase Tabla 12.

Tabla 12 Flujo CU1

Paso	Actor	Paso	Sistema	Excepción
1	Inicia la aplicación	2	Presenta escena inicial de la aplicación	

- CU2. Visualización de instrucciones-gestos

Descripción: Se podrá ingresar a las instrucciones de la aplicación.

Actor: Usuario.

Flujo: Véase Tabla 13.

Tabla 13 Flujo CU2

Paso	Actor	Paso	Sistema	Excepción
1	Inicia la aplicación	2	Presenta escena inicial de la aplicación	
3	Realiza gesto "Pinch"	4	Ingresa a escena de instrucciones de la aplicación	

- CU3. Salir de la Escena/Aplicación

Descripción: Se podrá retroceder entre escenas.

Actor: Usuario.

Flujo: Véase Tabla 14.

Tabla 14 Flujo CU3 primera parte

Paso	Actor	Paso	Sistema	Excepción
		1	Presenta alguna escena de la aplicación	
2	Realiza gesto "Fist"	3	Ingresa a escena predecesora	En escena inicial de la aplicación, el realizar gesto "Fist" finalizará la aplicación

Descripción: Se podrá cerrar la aplicación

Actor: Usuario

Flujo: Véase Tabla 15.

Tabla 15 CU3 segunda parte

Paso	Actor	Paso	Sistema	Excepción
		1	Presenta la escena "Start"	
2	Realiza gesto "Fist"	3	Presenta pregunta de seguir con el deseo de salir de la aplicación	
4	Realizará gesto "Forward" si acepta salir de la aplicación	5	Saldrá de la aplicación	
4	Realizará el gesto "Backward" si no acepta salir de la aplicación	5	No saldrá de la aplicación	
		6	Sale de la aplicación	

- CU4. Visualización de modelos 3D

Descripción: Se podrá ingresar a la visualización de todos los modelos 3D de la aplicación.

Actor: Usuario.

Flujo: Véase Tabla 16.

Tabla 16 Flujo CU4

Paso	Actor	Paso	Sistema	Excepción
		1	Presenta escena de galería de modelos 3D	
2	Visualiza modelos 3D de la aplicación			

- CU5. Elegir un modelo 3D

Descripción: Se podrá elegir un modelo entre todos los modelos 3D de la aplicación.

Actor: Usuario.

Flujo: Véase Tabla 17.

Tabla 17 Flujo CU5

Paso	Actor	Paso	Sistema	Excepción
1	Trasladarse al modelo que desee ingresar	2	Presentará la escena de modelos 3D seleccionado el primer modelo.	

- CU6. Eliminar modelo 3D

Descripción: Se podrá eliminar el modelo seleccionado.

Actor: Usuario.

Flujo: Véase Tabla 18.

Tabla 18 Flujo CU6

Paso	Actor	Paso	Sistema	Excepción
1	Selecciona una modelo entre los modelos 3D de la aplicación			
2	Realiza gesto "Pinch"	3	Presenta pregunta de seguir con el deseo de eliminar el modelo	
4	Realizará gesto "Forward" si acepta eliminar el modelo	5	Eliminará el modelo	
4	Realizará el gesto "Backward" si no acepta eliminar el modelo	5	No eliminará el modelo	
		6	Presenta escena de modelos 3D	

- CU7. Abrir modelo 3D

Descripción: Se podrá abrir el modelo seleccionado.

Actor: Usuario.

Flujo: Véase Tabla 19.

Tabla 19 Flujo CU7

Paso	Actor	Paso	Sistema	Excepción
1	Selecciona una modelo entre los modelos 3D de la aplicación			
2	Realiza gesto "Open"	3	Ingresa a la escena "Model" de la aplicación	

- CU8. Seleccionar modelo hacia la derecha

Descripción: Se podrá seleccionar el modelo que se encuentra a la derecha del actualmente seleccionado.

Actor: Usuario.

Flujo: Véase Tabla 20.

Tabla 20 Flujo CU8

Paso	Actor	Paso	Sistema	Excepción
1	Visualiza modelos 3D de la aplicación			
2	Realiza gesto "Wave Out"	3	Presenta escena de galería de modelos 3D señalando el siguiente modelo hacia la derecha	Cuando se encuentre en el último modelo de la galería el sistema no podrá seleccionar el siguiente modelo a la derecha.

- CU9. Seleccionar modelo hacia la izquierda

Descripción: Se podrá seleccionar el modelo que se encuentra a la izquierda del actualmente seleccionado.

Actor: Usuario.

Flujo: Véase Tabla 21.

Tabla 21 Flujo CU9

Paso	Actor	Paso	Sistema	Excepción
1	Visualiza modelos 3D de la aplicación			
2	Realiza gesto "Wave In"	3	Presenta escena de galería de modelos 3D señalando el siguiente modelo hacia la izquierda	Cuando se encuentre en el primer modelo de la galería el sistema no podrá seleccionar el siguiente modelo a la izquierda.

- CU10. Girar modelo 3D hacia la izquierda

Descripción: Se podrá girar el modelo 3D hacia la izquierda.

Actor: Usuario.

Flujo: Véase Tabla 22.

Tabla 22 Flujo CU10

Paso	Actor	Paso	Sistema	Excepción
		1	Presenta la escena del modelo 3D	
2	Realiza gesto "Left"	3	Presenta la escena del modelo 3D girado 45 grados hacia la izquierda	

- CU11. Girar modelo 3D hacia la derecha

Descripción: Se podrá girar el modelo 3D hacia la derecha.

Actor: Usuario.

Flujo: Véase Tabla 23.

Tabla 23 Flujo CU11

Paso	Actor	Paso	Sistema	Excepción
		1	Presenta la escena del modelo 3D	
2	Realiza gesto "Right"	3	Presenta la escena del modelo 3D girado 45 grados hacia la derecha	

- CU12. Girar modelo 3D hacia arriba

Descripción: Se podrá girar el modelo 3D hacia arriba.

Actor: Usuario.

Flujo: Véase Tabla 24.

Tabla 24 Flujo CU12

Paso	Actor	Paso	Sistema	Excepción
		1	Presenta la escena del modelo 3D	
2	Realiza gesto "Up"	3	Presenta la escena del modelo 3D girado 45 grados hacia arriba	

- CU13. Girar modelo 3D hacia abajo

Descripción: Se podrá girar el modelo 3D hacia abajo.

Actor: Usuario.

Flujo: Véase Tabla 25.

Tabla 25 Flujo CU13

Paso	Actor	Paso	Sistema	Excepción
		1	Presenta la escena del modelo 3D	
2	Realiza gesto "Down"	3	Presenta la escena del modelo 3D girado 45 grados hacia abajo	

ANEXO IV Manual de usuario de la aplicación

[Manual Usuario Visor Modelos 3D.pdf](#)

ANEXO V Código fuente de la aplicación

[Código Fuente Visor Modelos 3D](#)

ANEXO VI Código fuente script Matlab ejecución de tareas mediante el uso del teclado

[Comunicacion Mat Unity.m](#)

ANEXO VII Manual de uso de la aplicación integrada con el modelo de reconocimiento de gestos

[Manual Uso Integracion Visor Modelos 3D.pdf](#)

ANEXO VIII Código fuente de la integración con la aplicación y el modelo de reconocimiento de gestos

[Código Fuente Integración](#)

ANEXO IX Video del funcionamiento de la aplicación integrada con el modelo de reconocimiento de gestos.

[Video Aplicación Integrada](#)