

**Diseño de un modelo arquitectónico interior en 3d de un apartamento, con  
elementos interactivos controlados por comandos de voz**

Jorge Enrique Monroy Correa

Asesor:

Francisco Javier Reina

Universidad Nacional Abierta y a Distancia - UNAD

Escuela de Ciencias Básicas, Tecnología e Ingeniería - ECBTI

Ingeniería Multimedia

2024

## Resumen

Con el pasar de los años se ha conseguido un mayor acercamiento a los usuarios a través de las tecnologías interactivas de todo tipo como, videojuegos, publicidad, entre otras. El uso de los motores gráficos que se utilizaba casi exclusivamente en los videojuegos ahora está siendo utilizado en otras industrias como en el desarrollo de proyectos arquitectónicos. Esta industria ha tenido un avance importante en cuanto a la presentación de proyectos y aunque aún se utiliza bastante la visualización de proyectos a través de imágenes y recorridos virtuales en formatos lineales, ahora se está apostando por las bondades que ofrecen los motores gráficos para el desarrollo de herramientas interactivas. El propósito de este proyecto es el diseño de un modelo arquitectónico interior en 3d de un apartamento, y la implementación de elementos interactivos para su visualización los cuales serán controlados por comandos de voz. Este material interactivo tiene como objetivo la comercialización de proyectos inmobiliarios y está dirigido a las constructoras y agencias de publicidad en Colombia que quieran implementar una nueva herramienta novedosa, y además incluir dentro de sus clientes y posibles compradores, personas con ciertas discapacidades que no les permiten el acceso a contenidos digitales. El proyecto está desarrollado en 3 etapas organizadas de la siguiente manera: En primer lugar, está la etapa de preproducción donde se realizó el análisis de los requerimientos funcionales y no funcionales, así como la planeación del proyecto partiendo de la elaboración del pitch multimedia, el diseño del guion, el prototipado del aplicativo además de las pruebas iniciales que determinaron la viabilidad y el boceto general del proyecto. En segundo lugar, se realizó la etapa de producción donde se creó el producto a partir de la creación de los modelos 3d, elaboración de las texturas, importación de los modelos en el motor de video juegos Unity, la creación de la iluminación, la creación y programación de los elementos interactivos y la programación de los comandos de voz. Además, se realizaron varios tipos de pruebas, para determinar el funcionamiento del

aplicativo y la percepción de usabilidad por parte de los usuarios. Finalmente, en la etapa de postproducción se realizaron los ajustes finales del producto para posteriormente realizar la creación del máster y su implementación.

***Palabras Clave:*** Modelos arquitectónicos en 3d, reconocimiento de voz, comandos de voz, 3d interactivo, modelos 3d interactivos, aplicaciones interactivas controladas por comandos de voz.

## **Abstract**

Over the years, a closer relationship with users has been achieved through interactive technologies of all kinds such as video games, advertising, among others. The use of graphics engines that was used almost exclusively in video games is now being used in other industries such as the development of architectural projects. This industry has advanced significantly in terms of the presentation of projects and although the visualization of projects through images and virtual tours in linear formats is still widely used, it is now committed to the benefits offered by graphic engines for the development of interactive tools. This project consists of designing a 3D interior architectural model of an apartment and implementing interactive elements for its visualization which are controlled using voice commands. This interactive material aims to market real estate projects and is aimed at construction companies and advertising agencies in Colombia that want to implement a new innovative tool, and also include among their clients and potential buyers people with certain disabilities that do not allow them to access to digital contents. The project has been developed in 3 stages organized as follows: First, the pre-production stage was carried out where the analysis of the functional and non-functional requirements, as well as the planning of the project based on the preparation of the multimedia pitch, script design, application prototyping as well as initial tests that have determined the feasibility and general outline of the project. Secondly, the production stage was carried out where the product was developed from the creation of the 3D models, elaboration of the textures, import of the models to the Unity video game engine, the creation of the lighting, the creation and programming of interactive elements, and voice command programming. In addition, various types of tests were carried out to determine the operation of the application and the perception of usability by users. Finally, in the post-production stage, the final adjustments of the product was made, to later create the master and its implementation.

**Keywords:** 3D architectural models, voice recognition, voice commands, interactive 3D, interactive 3D models, interactive applications controlled by voice commands.

## Tabla de Contenido

Introducción .....	16
Planteamiento del Problema .....	17
Justificación .....	19
Objetivos .....	20
Objetivo General .....	20
Objetivos Específicos .....	20
Marco Contextual y Teórico .....	21
Contexto Histórico .....	21
Marco Conceptual .....	22
Interactividad .....	22
Renderizado en Tiempo Real .....	22
Motor Grafico. ....	23
Reconocimiento de Voz.....	23
Proyecto Multimedia .....	23
En la Etapa de Preproducción.....	24
En la Etapa de Producción.....	24
Modelos Arquitectónicos 3d.....	24
Metodología .....	26
Etapa de Preproducción.....	26

Etapa de Producción.....	26
Etapa de Postproducción .....	27
Preproducción .....	28
Análisis de Requerimientos.....	28
Requerimientos Funcionales.....	28
Reconocimiento de Voz.....	28
Interactividad de Elementos.....	28
Desplazamiento y Rotación de Cámaras.....	28
Cambio de Materiales Piso. ....	28
Cambio de Materiales Mesón Cocina.....	28
Encendido y Apagado de Luces.....	29
Opción Tutorial.....	29
Opción Audio.....	29
Requerimientos No Funcionales.....	29
Interfaz del Usuario.....	29
Estabilidad.....	29
Documentación. ....	29
Optimización de Recursos. ....	29
Compatibilidad.....	30
Pitch Multimedia.....	30

Titulo .....	30
Requerimientos Mínimos .....	30
Sinopsis.....	31
Problema.....	32
Solución.....	32
Características Principales.....	33
Desplazamiento de Cámara Dinámico.....	33
Personalización de Materiales.....	33
Navegación Simplificada.....	33
Diseño Accesible.....	33
Ventajas Competitivas.....	33
Innovación Tecnológica.....	33
Conciencia de Accesibilidad.....	33
Mercado Objetivo .....	34
Profesionales de Diseño de Interiores y Arquitectos.....	34
Inmobiliarias Empresas de Bienes Raíces y Constructoras.....	34
Conclusión.....	34
Creación del Guion.....	34
Guion Literario .....	34
Público Objetivo.....	34

Plataformas. ....	34
Objetivo.....	34
Comando Tutorial. ....	35
Apagar Audio.....	35
Elementos Interactivos.....	35
Cámaras.....	35
Pisos. ....	35
Mesón Cocina. ....	36
Luces. ....	36
Guion Técnico .....	37
Generalidades del Proyecto.....	41
Programas Utilizados Para el Proyecto .....	42
Prototipado .....	45
Prototipado Desplazamiento de Cámaras Controladas por Comandos de Voz.....	45
Prototipado Texturas Interactivas Controladas por Comandos de Voz.....	48
Pruebas Funcionales Prototipo .....	52
Conclusión de las Pruebas de Prototipado.....	52
Boceto General del Proyecto.....	52
Producción .....	54
Modelado y Texturizado en el Software 3d .....	54

	10
Incorporación del Modelo 3d en Unity .....	55
Texturizado en Unity.....	56
Iluminación.....	57
Configuración de la iluminación Global .....	57
Configuración de Luces Artificiales.....	59
Configuración Elementos Interactivos .....	60
Cámaras .....	60
Materiales .....	65
Materiales Piso.....	65
Materiales Mesón.....	67
Luces.....	68
Reconocimiento de Voz: .....	71
Control de Audio.....	71
Interfaz UI (User Interface).....	74
Menú Cámaras.....	78
Menú Pisos .....	80
Menú Mesón.....	81
Menú Luces .....	82
Pruebas .....	83
Pruebas Funcionales .....	83

	11
Pruebas de Observación.....	85
Pruebas de Usabilidad .....	86
Criterios de Usabilidad Para Evaluar.....	87
Criterio de Contenido.....	87
Criterio de Navegabilidad.....	87
Graficas.....	87
Criterio de Accesibilidad .....	87
Utilidad .....	87
Preguntas Según Criterio de Evaluación.....	87
Diseño de Formulario de Evaluación.....	88
Publicación de Formulario de Evaluación y Aplicativo Preliminar. ....	88
Aplicativa Versión 2: .....	89
Realización de Pruebas de Usabilidad. ....	89
Documentación de Resultados .....	91
Análisis de Resultados .....	92
Documentación de Fallos .....	93
Plan de Mejora .....	94
Tabla 12.....	94
Pruebas de Sanidad (Sanity Testing).....	95
Postproducción.....	97

	12
Ajustes del Proyecto Para Renderizado .....	97
Generación del Máster e Implementación.....	101
Publicación del Máster .....	103
Conclusiones.....	104
Referencias Bibliográficas.....	106

## Lista de Tablas

<b>Tabla 1</b> ( <i>Guion Técnico</i> ) <i>Pantalla de Bienvenida</i> .....	38
<b>Tabla 2</b> ( <i>Guion Técnico</i> ) <i>Pantalla Inicio</i> .....	38
<b>Tabla 3</b> ( <i>Guion Técnico</i> ) <i>Cámaras</i> .....	39
<b>Tabla 4</b> ( <i>Guion Técnico</i> ) <i>Pisos</i> .....	40
<b>Tabla 5</b> ( <i>Guion Técnico</i> ) <i>Mesón Cocina</i> .....	40
<b>Tabla 6</b> ( <i>Guion Técnico</i> ) <i>Luces</i> .....	41
<b>Tabla 7</b> <i>Pruebas funcionales del prototipo</i> .....	52
<b>Tabla 8</b> <i>Pruebas funcionales del aplicativo desarrollado</i> .....	84
<b>Tabla 9</b> <i>Diseño de Preguntas Evaluativas por Criterio</i> .....	87
<b>Tabla 10</b> <i>Tabulación de Resultados Pruebas de Usabilidad</i> .....	91
<b>Tabla 11</b> <i>Escala de Medición</i> .....	91
<b>Tabla 12</b> <i>Plan de Mejora</i> .....	94
<b>Tabla 12</b> <i>Pruebas de Sanidad (Sanity Testing)</i> .....	95

## Lista de Figuras

<b>Figura 1</b> <i>Aplicación arquitectónica controlada por comandos de voz</i> .....	30
<b>Figura 2</b> <i>Plataformas compatibles</i> .....	31
<b>Figura 3</b> <i>Proyecto arquitectónico interactivo en Unity</i> .....	31
<b>Figura 4</b> <i>Árbol de navegación</i> .....	37
<b>Figura 5</b> <i>Programas utilizados en el proyecto</i> .....	44
<b>Figura 6</b> <i>Ubicación cámaras en el escenario</i> .....	45
<b>Figura 7</b> <i>Configuración del script de cámaras en el inspector</i> .....	47
<b>Figura 8</b> <i>Cámara principal toma ubicación del target1</i> .....	48
<b>Figura 9</b> <i>Creación de materiales en Unity</i> .....	49
<b>Figura 10</b> <i>Se añaden los materiales en el inspector</i> .....	51
<b>Figura 11</b> <i>Cambio de materiales al usar los comandos de voz</i> .....	51
<b>Figura 12</b> <i>Boceto General del Proyecto</i> .....	53
<b>Figura 13</b> <i>Modelo 3d Estructura del Apartamento</i> .....	54
<b>Figura 14</b> <i>Modelo 3d con Texturas y Muebles</i> .....	55
<b>Figura 15</b> <i>Modelo 3d incorporado en Unity</i> .....	56
<b>Figura 16</b> <i>Texturizado en Unity</i> .....	57
<b>Figura 17</b> <i>Iluminación ambiental objetos estáticos</i> .....	58
<b>Figura 18</b> <i>Configuración de la iluminación ambiental</i> .....	58
<b>Figura 19</b> <i>Configuración Skybox</i> .....	59
<b>Figura 20</b> <i>Configuración luces artificiales</i> .....	60
<b>Figura 21</b> <i>Cámara “planta” vista por defecto</i> .....	61
<b>Figura 22</b> <i>Distribución de cámaras</i> .....	61
<b>Figura 23</b> <i>Visualización de Cámara “sala1”</i> .....	62

<b>Figura 24</b> <i>Asignación de cámaras en el inspector</i> .....	64
<b>Figura 25</b> <i>Asignación de materiales de los pisos en el inspector</i> .....	67
<b>Figura 26</b> <i>Asignación de materiales del mesón en el inspector</i> .....	68
<b>Figura 27</b> <i>Distribución de luces en el apartamento</i> .....	68
<b>Figura 28</b> <i>Asignación de luces en el inspector</i> .....	71
<b>Figura 29</b> <i>Asignación audio de fondo en el inspector</i> .....	74
<b>Figura 30</b> <i>Pantalla de Bienvenida</i> .....	75
<b>Figura 31</b> <i>Pantalla Principal</i> .....	75
<b>Figura 32</b> <i>Tutorial</i> .....	76
<b>Figura 33</b> <i>Menú Cámaras</i> .....	78
<b>Figura 34</b> <i>Asignación de menú cámaras en el inspector</i> .....	79
<b>Figura 35</b> <i>Menú Pisos</i> .....	81
<b>Figura 36</b> <i>Menú Mesón</i> .....	82
<b>Figura 37</b> <i>Menú Luces</i> . .....	82
<b>Figura 38</b> <i>Usuarios Manipulando el Aplicativo</i> .....	85
<b>Figura 39</b> <i>Estructura del formulario</i> .....	88
<b>Figura 40</b> <i>Pruebas de Usabilidad</i> .....	90
<b>Figura 41</b> <i>Configuración Project Settings</i> .....	97
<b>Figura 42</b> <i>Project Settings Configuración Color Space</i> .....	98
<b>Figura 43</b> <i>Configuración Rendering Path</i> .....	99
<b>Figura 44</b> <i>Configuración resolución</i> .....	100
<b>Figura 45</b> <i>Calidad después del backing</i> .....	101
<b>Figura 46</b> <i>Configuración de las escenas</i> .....	102
<b>Figura 47</b> <i>Nombre del proyecto</i> .....	103

## Introducción

El presente trabajo se abordan aspectos relacionados a la comercialización de proyectos inmobiliarios y su desarrollo a través de herramientas multimedia no lineales para la creación de interactividad, por medio de la incorporación de un modelo arquitectónico de un apartamento en 3d, dentro del motor de video juegos Unity. Si bien este tipo de tecnología está siendo utilizada para la comercialización de proyectos inmobiliarios en la actualidad, este proyecto aborda aspectos novedosos en cuanto a la incorporación de interactividad y su manipulación y control a través de comandos de voz.

Por otro lado, los proyectos que se desarrollan en la actualidad están creados en su gran mayoría para su manipulación en forma digital, lo cual de alguna manera excluye a una gran cantidad de población que posee cierto tipo de discapacidades que no les impide acceder a este tipo de contenidos. Este proyecto se caracteriza por la incorporación de interactividad a través de comandos de voz con el fin de lograr que la población con algún tipo de discapacidad que les impida acceder a contenidos digitales pueda interactuar con este tipo de proyectos.

Además de procurar la inclusión de personas con ciertas discapacidades que les impiden acceder a contenidos digitales, con el siguiente aplicativo se pretende abrir un panorama para el desarrollo de aplicaciones controladas por comandos de voz, las cuales pueden ser implementadas en entornos educativos, comerciales, asociados a la salud y en muchos otros donde la interactividad sea más directa y accesible.

El desarrollo de este proyecto está basado en la tipología (MSP) Multimedia Smart Process la cual consiste en 3 etapas fundamentales que son la preproducción, donde se realizará la planificación, la producción y control de calidad que corresponde a la ejecución, así como las pruebas del aplicativo y la posproducción donde se realizará la optimización final del producto para la generación del master y su distribución.

## Planteamiento del Problema

Los avances tecnológicos en los últimos años han permitido que los usuarios de todo tipo de aplicaciones se involucren cada vez más por medio de la interactividad. Las aplicaciones interactivas han sido muy importantes y han logrado conectar con los usuarios no solamente en el área de entretenimiento a través de los videojuegos sino también se han utilizado para atraer clientes, vender proyectos, mostrar productos, etc., de una forma más dinámica.

Sin embargo, a pesar de los nuevos avances y de la implementación de motores de videojuegos para el desarrollo de aplicativos interactivos desarrollados para la visualización de proyectos inmobiliarios con fines comerciales, los proyectos están desarrollados para el uso netamente digital y aun no se encuentran en el mercado productos de este tipo cuya interactividad sea controlada por medio de comandos de voz, lo cual sería un medio interesante y novedoso de interactuar con los usuarios.

También es importante aclarar que las herramientas desarrolladas con motores de videojuegos para la visualización de proyectos inmobiliarios en forma interactiva están pensadas para usuarios que puedan interactuar digitalmente y no se está incluyendo a una población muy importante como son las personas con discapacidades en sus extremidades superiores las cuales no pueden acceder a los contenidos digitales.

De acuerdo con las cifras de la Organización Mundial de la Salud. (2023) Discapacidad. Al rededor de 1300 millones personas, es decir un 16% de la población mundial sufren de algún tipo de discapacidad, las cuales se ven limitadas al acceso a la información ya que los aplicativos no cumplen con los estandartes de accesibilidad.

Teniendo esto en mente se hace necesario crear aplicativos multimedia interactivos que ofrezcan nuevas alternativas que puedan ayudar a solucionar en parte esta problemática, además, de resaltar la importancia de la tecnología de comandos de voz la cual no se encuentra lo

suficientemente utilizada en los aplicativos interactivos en la actualidad pero que sin duda puede prestar un gran beneficio.

**Pregunta de Investigación:** ¿Cómo generar aplicaciones interactivas de espacios arquitectónicos interiores en 3d de proyectos inmobiliarios que puedan ser controladas a través de comandos de voz, que presenten una nueva alternativa para los usuarios comunes y una solución para usuarios con enfermedades o discapacidad en sus extremidades superiores?

## **Justificación**

Los avances tecnológicos han permitido el desarrollo de herramientas interactivas, las cuales están siendo orientadas a que los usuarios tengan un contacto más directo y dinámico con diferentes productos, es así como se están utilizando motores de videojuegos para la creación de diferentes contenidos interactivos multimedia. Sin embargo, la gran mayoría de aplicativos están siendo creados para su manipulación netamente digital, que, si bien representan una solución para una gran cantidad de usuarios, otra cantidad muy importante están quedando excluidos. Se trata de los usuarios con cierto tipo de discapacidades que les impide el acceso a contenidos digitales.

En este proyecto se desarrolla un aplicativo arquitectónico multimedia interactivo el cual es controlado por medio de comandos de voz, buscando que sea más universal y se pueda alcanzar mayor cantidad de usuarios, de manera que los usuarios comunes tengan otra alternativa de interacción y donde se genere más inclusión a los usuarios que no pueden acceder a contenidos digitales.

Si bien este aplicativo está dirigido a cubrir necesidades del sector inmobiliario para la comercialización de proyectos, también se busca marcar una pauta para el desarrollo de otras aplicativos en diferentes campos que puedan ser controlados por medio de comandos de voz generando una mayor accesibilidad e inclusión a personas con diferentes discapacidades y proporcionar de esta manera nuevos recursos orientados a tener una comunicación más directa y versátil que se acomode al desarrollo constante y a los avances tecnológicos.

## **Objetivos**

### **Objetivo General**

Realizar el diseño y el desarrollo de un modelo en 3d arquitectónico interior de un apartamento, el cual contenga elementos interactivos que puedan ser controlados a través de comandos de voz, para la comercialización de proyectos inmobiliarios en Colombia.

### **Objetivos Específicos**

Definir las características técnicas del aplicativo “VoiceExplorer” que corresponden al proceso de preproducción, incluyendo los requerimientos, pitch multimedia, bocetos, prototipado y pruebas, así como todos los recursos previos para su desarrollo.

Realizar los insumos necesarios para el aplicativo “VoiceExplorer” para su integración y desarrollo de la programación, interactividad, interfaz de usuario, además elementos visuales y ejecución de las pruebas necesarias para el correcto funcionamiento del aplicativo.

Generar la optimización final del aplicativo multimedia “VoiceExplorer”, así como los ajustes de calidad y preparativos para la generación del máster y su implementación.

## Marco Contextual y Teórico

### Contexto Histórico

La presentación de proyectos arquitectónicos ha tenido muchos cambios en los últimos años.

En el pasado se trabajaba con planos elaborados a mano por un arquitecto en una mesa de dibujo y se realizaban maquetas físicas para dar una idea general del proyecto. Si bien la computación gráfica se remonta a muchos años atrás, podemos decir que hacia la década de los 80 inicia la representación arquitectónica a través de Jim Blim quien empieza a experimentar conceptos de mapas de texturas. Balea, F. (2021).

De acuerdo con los avances tecnológicos en computación a inicios de los 90 se crearon softwares especiales para la elaboración de los planos como Autocad, y más adelante en esta misma década se empezaron a elaborar modelos en 3d de representación arquitectónica a través de programas como 3dStudio que daban una idea más clara del proyecto.

Estos eran programas de renderizado (offline) de imágenes que fueron avanzando y daban la posibilidad de crear materiales realistas, y con una iluminación adecuada obtener resultados bastante cercanos a la realidad del proyecto. También se empezaron a utilizar animaciones del proyecto en forma lineal lo cual resultaba bastante atractivo para los usuarios. A medida que fue avanzando la tecnología se empezaron a implementar nuevas herramientas para lograr una interacción con el usuario de forma más directa y se empezaron a utilizar los motores gráficos los cuales eran usados casi exclusivamente para los video juegos, posteriormente surgieron nuevas necesidades y se buscó la forma de crear aplicaciones interactivas renderizadas en tiempo real y se comenzaron a utilizar los motores gráficos para la elaboración de proyectos arquitectónicos buscando una interacción más directa con los usuarios.

Así mismo, si hablamos del reconocimiento de voz sus inicios datan de hace bastantes años donde alrededor de 1940 se empiezan a crear tecnologías que fueran capaces de crear una comunicación hombre-maquina a través de sistemas mecánicos que con el tiempo fueron evolucionando a dispositivos electrónicos, Buenaño, A., Veloz, M. (2003). Sin embargo, a pesar de los avances tecnológicos, aun en la actualidad no se han implementado como se debería ya que la mayoría de los recursos multimedia están desarrollados para uso digital, por eso la importancia de avanzar en el desarrollo de recursos multimedia interactivos con reconocimiento de voz, pensando en la población con discapacidades físicas.

## **Marco Conceptual**

### ***Interactividad***

La interactividad se puede definir como la comunicación que existe entre un usuario y un recurso multimedia no lineal donde según Minguell, M. (2007). el grado de interactividad depende de del proceso de participación del usuario con dichos recursos.

### ***Renderizado en Tiempo Real***

Es un sistema de renderizado donde la velocidad de cálculo es casi instantánea lo cual genera la sensación de movimiento. Este tipo de renderizado es usado en el mundo de los videojuegos, pero es cada vez más utilizado en la arquitectura por su nivel de flujo de trabajo en los procesos arquitectónicos Balea, F. (2021). La tecnología usada mayormente en video juegos está siendo utilizada en varios ámbitos incluyendo la visualización arquitectónica, por lo que se implementó para el desarrollo de este proyecto incluyendo comandos de voz para su interactividad.

### ***Motor Grafico.***

Para Romero, A. (2021) este término está relacionado con las rutinas de programación para el diseño, construcción y presentación de videojuegos, donde básicamente se provee una función de renderizado de gráficos 2d y 3d con varios elementos como el detector de colisiones, la animación, los scripts necesarios para creación de la interactividad, la inteligencia artificial etc. Aunque este término estuvo ligado inicialmente a la construcción de videojuegos, posteriormente se empezó a utilizar en otro tipo de proyectos aprovechando las bondades interactivas, y el renderizado en tiempo real. Para este proyecto se utiliza el motor grafico Unity en el cual se realiza la integración de los modelos 3d y se programará la interactividad.

### ***Reconocimiento de Voz***

Este término está relacionado con la comunicación que se establece entre un ser humano y una máquina a través del habla, Buenaño, A., Veloz, M. (2003). Donde un sistema informático detecta las palabras y las convierte a un formato legible. En este proyecto se realiza la interactividad a través del motor gráfico Unity y la interactividad es controlada por medio de comandos de voz, donde se utiliza el reconocimiento de voz de Windows.

### ***Proyecto Multimedia***

Según Belloch, C. (2012). Cuando se habla de multimedia se hace referencia al uso combinado de medios de comunicación como el texto, las imágenes, sonido, video y animación los cuales están encaminados a generar una interacción con los usuarios.

El desarrollo de proyectos multimedia se basa en 3 etapas fundamentales que son preproducción, producción y postproducción.

### ***En la Etapa de Preproducción***

se realiza el planeamiento o análisis del proyecto donde se estipulan las necesidades, el público objetivo, se realiza un análisis de los contenidos, las funcionalidades, así como de los recursos necesarios para su desarrollo. En esta etapa también se hace el diseño preliminar de las funcionalidades del producto multimedia y se crea un prototipo del producto que da una idea general y establece el alcance del proyecto.

### ***En la Etapa de Producción***

Es donde se desarrolla el producto multimedia con base en la planificación hecha en la etapa anterior y se desarrolla teniendo en cuenta los parámetros ya establecidos en cuanto a la estructura y el diseño del proyecto.

Una vez realizada la producción del producto multimedial es necesario verificar que todo esté funcionando correctamente por lo que se pasa a la **etapa de postproducción** donde hace una evaluación y se implementa un sistema de pruebas para determinar las fortalezas y debilidades y proceder con las correcciones y/o modificaciones o mejoras del producto multimedia. En esta etapa también es importante hacer la documentación respectiva del producto y posteriormente hacer la implementación del producto.

### ***Modelos Arquitectónicos 3d***

Se refiere a las representaciones tridimensionales de proyectos arquitectónicos, que pueden abarcar espacios exteriores, interiores o paisajes urbanos. Se crean con software especializado en estas áreas de modelado tridimensional utilizadas en el área de la arquitectura y el diseño. Confidencial (2022). Dentro de las herramientas que se utilizaron para el modelado en 3d se encuentra el 3dmax que es un programa para hacer modelado y animación 3d. Se crean los

modelos arquitectónicos y los elementos dentro de la escena, además de la creación de materiales y se exporta en formato FBX para ser incorporado en Unity.

## **Metodología**

Para llevar a cabo el desarrollo de esta aplicación multimedia interactiva se requiere una planeación y una serie de requerimientos que se desarrollan con base a tres etapas fundamentales en marcadas en el proceso (MSP) Multimedia Smart Process que son la etapa de preproducción, la etapa de producción y la etapa de posproducción las cuales se detallan a continuación:

### **Etapas de Preproducción**

En esta etapa se realiza el análisis de las características del proyecto, así como la identificación de los requerimientos funcionales y no funcionales que definen el proyecto, posteriormente se realiza un prototipado con sus respectivas pruebas para determinar la viabilidad del proyecto, un Pich multimedia con el fin de presentar la idea general del proyecto, se desarrolla el guion además de los bocetos por medio de vistas en planta y diagramas para definir las características y el alcance.

### **Etapas de Producción**

Con los bocetos realizados en la etapa anterior, se procede a la construcción de los assets necesarios como los modelos en 3d del apartamento, los modelos del mobiliario y sus elementos internos, así como el proceso de texturizado, para posteriormente incorporarlo en el motor Unity y hacer los ajustes correspondientes en cuanto a materiales e iluminación en la cual se realiza la configuración global y se agregarán y configurarán las luces artificiales. También se hace el diseño de la interfaz y los elementos visuales además de crear la programación de los elementos interactivos y los comandos de voz. Finalizada la construcción de la versión inicial completa del aplicativo se realizan varias pruebas orientadas a verificar el funcionamiento, además de la realización de pruebas de observación para verificar el impacto de los usuarios hacia el aplicativo y finalmente se efectúan pruebas con varios criterios para verificar que se cumplan los

requerimientos iniciales, donde se analizan los resultados obtenidos con el fin de documentar los fallos y realizar las respectivas correcciones.

### **Etapas de Postproducción**

Una vez verificado el aplicativo con base a la aplicación de las pruebas se procede a realizar los ajustes finales, donde se configuran aspectos de calidad de visualización, optimización de recursos y se prepara para la creación del master y su implementación.

## Preproducción

### Análisis de Requerimientos

En esta fase se realiza el análisis de requerimientos donde se identifican los requisitos del aplicativo para asegurar que el software se ajuste a las necesidades planteadas desde el inicio y se cumplan con los objetivos propuestos. A continuación, se plantean tanto los requerimientos funcionales como los requerimientos no funcionales.

### *Requerimientos Funcionales*

Los requerimientos funcionales definen las funciones que realizará el usuario cuando se está ejecutando el aplicativo, los cuales se describen a continuación:

**Reconocimiento de Voz.** El aplicativo debe reconocer comandos de voz, para que cuando el usuario pronuncie un comando específico se ejecute una acción. Esto puede implementarse utilizando servicios externos de voz como, Google Cloud Speech-to Text o Microsoft Azure Speech Service.

**Interactividad de Elementos.** El aplicativo debe tener elementos interactivos los cuales serán controlados por comandos de voz, dentro de los cuales se encuentran:

**Desplazamiento y Rotación de Cámaras.** Los comandos de voz para desplazamiento de cámaras son: sala1, sala 2, comedor, alcoba 1, alcoba 2, alcoba principal, baño y planta. Para la rotación de cámaras son: giro, derecha, izquierda.

**Cambio de Materiales Piso.** El aplicativo debe contar con comandos para cambio de materiales que son: piso laminado, piso baldosa y piso parquet.

**Cambio de Materiales Mesón Cocina.** El aplicativo debe tener comandos para cambiar el material del mesón de la cocina los cuales son: mesón granito, mesón mármol y mesón acero.

**Encendido y Apagado de Luces.** El aplicativo debe contar con los comandos de voz para apagar las luces los cuales son: apagar cocina-encender cocina, apagar sala-encender sala, apagar alcoba principal-encender alcoba principal, apagar alcoba 1-encender alcoba 1, apagar alcoba 2-encender alcoba 2, apagar baño-encender-baño.

**Opción Tutorial.** El aplicativo debe tener un comando comando de voz “tutorial” que active un video explicativo de cómo funciona el aplicativo, mostrando sus funciones principales y características que permiten su correcta manipulación, también debe tener un comando de voz para cerrar el tutorial “cerrar tutorial”.

**Opción Audio.** El aplicativo debe contar con una opción para “apagar audio” y para “encender audio”

### ***Requerimientos No Funcionales***

Para el desarrollo del aplicativo se deben tener en cuenta los requerimientos no funcionales que describen como debe ser el sistema, entre los que se encuentran:

**Interfaz del Usuario.** La interfaz del usuario debe ser lo suficientemente intuitiva para que el usuario tenga claro cuáles son los comandos de voz que se deben utilizar y qué acción van a realizar estos comandos de voz.

**Estabilidad.** El aplicativo debe ser estable con el fin de garantizar que funcione sin interrupciones o caídas abruptas.

**Documentación.** El aplicativo debe contar con la documentación necesaria para facilitar la resolución de problemas.

**Optimización de Recursos.** El aplicativo debe estar optimizado y garantizar un rendimiento eficiente.

**Compatibilidad.** El aplicativo debe ser compatible con las plataformas que han sido previstas para su desarrollo.

## **Pitch Multimedia**

### ***Titulo***

VoiceExplorer - Aplicación arquitectónica en Unity controlada con comandos de Voz.

### **Figura 1**

*Aplicación Arquitectónica controlada por comandos de voz*



*Fuente.* Autoría propia

### ***Requerimientos Mínimos***

El aplicativo se desarrolló para pc Sistema Operativo Windows 7 SP1 o posterior. Se recomienda que el equipo en el cual será ejecutado el aplicativo tenga tarjeta de video de 2GB o superior. Se utiliza el sistema de audio del computador y el reconocimiento de voz de Windows.

## Figura 2

### Plataformas compatibles



Fuente. 1000marcas.net. (2023). [Imágenes del logotipo de Windows]. En 1000marcas.net.

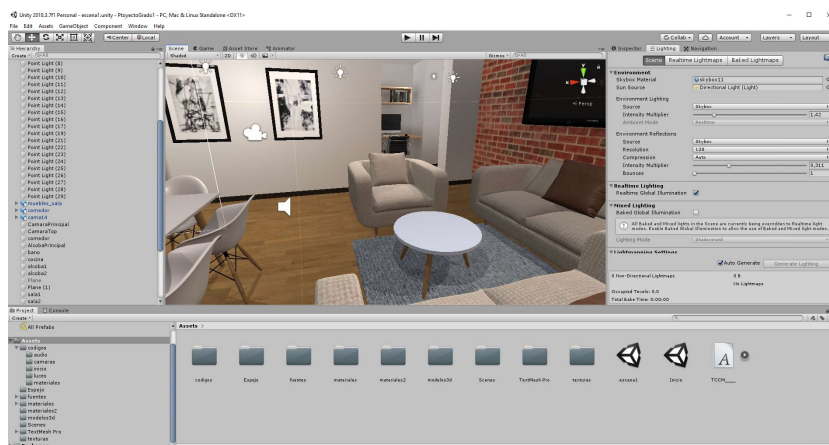
<https://1000marcas.net/windows-logo/>

## Sinopsis

“VoiceExplorer”, una revolucionaria aplicación arquitectónica desarrollada en Unity que transforma la experiencia de explorar y personalizar espacios virtuales mediante comandos de voz. Este proyecto pone el poder de la interactividad y la inmersión en manos del usuario y es una puerta abierta a la accesibilidad y la inclusión de personas que no pueden acceder a contenidos digitales debido a algún tipo de discapacidad, ofreciendo una manera intuitiva y emocionante de visualizar entornos arquitectónicos.

## Figura 3

### Proyecto arquitectónico interactivo en Unity.



Fuente. Autoría propia, entorno Unity [Captura de Pantalla]

El proyecto es realizado en el motor de video juegos Unity, donde a partir de la importación de modelos 3d de un apartamento amoblado previamente modelado en 3dmax, se realiza texturización, iluminación global, configuración de luces artificiales, además de la configuración de elementos interactivos, los cuales son controlados por medio de comandos de voz con el fin de que las personas con algún tipo de discapacidad en sus extremidades superiores o que por alguna razón no puedan acceder a contenidos digitales, puedan interactuar con este contenido y hacer la visualización interactiva del apartamento. El aplicativo cuenta con comandos de voz para controlar las cámaras, algunos materiales y las luces artificiales. El usuario tiene la opción de desplazarse virtualmente por los diferentes espacios del apartamento, así como hacer el cambio de algunos materiales y controlar la iluminación mediante comandos de voz. Este aplicativo se constituye en una excelente herramienta para la promoción y comercialización de productos inmobiliarios donde se pueden visualizar los espacios en calidad fotorrealista y en tiempo real por medio de la instalación de pantallas en las salas de ventas para que los usuarios puedan acceder en forma interactiva.

### ***Problema***

Para muchas personas con discapacidades, la experiencia de explorar entornos virtuales es limitada por la falta de opciones accesibles. La mayoría de las aplicaciones no consideran las necesidades de quienes enfrentan desafíos físicos.

### ***Solución***

“VoiceExplorer” utiliza tecnología de reconocimiento de voz para permitir a los usuarios interactuar con el entorno virtual de manera natural y sin esfuerzo. Desde cambiar el piso hasta desplazarse a una habitación específica, los comandos de voz hacen que la experiencia sea personalizada y única. “VoiceExplorer” abraza la diversidad y la inclusión de personas con

discapacidades que no les permiten acceder a contenidos digitales al integrar comandos de voz, ofreciendo una forma intuitiva y accesible para explorar y personalizar los espacios virtuales.

### *Características Principales*

**Desplazamiento de Cámara Dinámico.** Pronuncie un comando para desplazarse a un lugar específico del apartamento y el aplicativo responderá dirigiéndose al lugar seleccionado.

**Personalización de Materiales.** Cambie los materiales de algunos objetos con comandos de voz.

**Navegación Simplificada.** La aplicación tiene una interfaz sencilla donde los comandos se pueden encontrar fácilmente en la pantalla principal. En la parte superior derecha hay un tutorial donde se hace una breve explicación del aplicativo. Además, los comandos de los 4 menús interactivos también se encuentran en la pantalla principal. La aplicación es intuitiva y de fácil manejo.

**Diseño Accesible.** La interfaz de usuario ha sido diseñada considerando la accesibilidad, de manera que las personas con discapacidades físicas en sus extremidades superiores o que no puedan acceder a contenidos digitales, puedan interactuar con el aplicativo por medio de órdenes a través de comandos de voz, que son ejecutadas logrando que este tipo de usuarios tengan una buena experiencia de usuario.

### *Ventajas Competitivas*

**Innovación Tecnológica.** Este proyecto es pionero en el uso de comandos de voz para la personalización de entornos arquitectónicos en Unity.

**Conciencia de Accesibilidad.** Este proyecto busca aumentar la conciencia sobre la importancia de la accesibilidad en el desarrollo de aplicaciones interactivas.

### ***Mercado Objetivo***

**Profesionales de Diseño de Interiores y Arquitectos.** Simplifica la presentación de propuestas y visualización de proyectos.

**Inmobiliarias Empresas de Bienes Raíces y Constructoras.** Ofrece recorridos virtuales interactivos de propiedades, además da la opción de incluir nuevos clientes con algún tipo de discapacidad que no le permita acceder a contenidos digitales.

### ***Conclusión***

“VoiceExplorer” se compromete a hacer que la tecnología sea accesible para todos. Al unirse a esta revolución, se está construyendo un mundo virtual donde cada voz importa y cada persona tiene la capacidad de dar forma a su entorno. Descubre un espacio de grandes posibilidades con “VoiceExplorer”.

### **Creación del Guion**

#### ***Guion Literario***

**Público Objetivo.** Usuarios con discapacidades en sus extremidades superiores o que no puedan acceder a contenidos digitales y usuarios habituales.

**Plataformas.** El aplicativo está desarrollado para PC Windows 7 SP1 o superior.

**Objetivo.** Lograr que clientes de proyectos inmobiliarios con discapacidades que les impidan acceder a contenidos en forma digital y usuarios habituales puedan visualizar el proyecto “VoiceExplorer”, en una sala de ventas y puedan recorrer los espacios e interactuar con algunos elementos en tiempo real por medio de comandos de voz.

EL USUARIO accede a la aplicación “VoiceExplorer” en una sala de ventas.

EL USUARIO se encuentra con la pantalla principal que contiene un apartamento visto desde planta y con varios comandos de voz en la pantalla los cuales ejecutan una acción específica cuando son pronunciados.

**Comando Tutorial.** al lado izquierdo de la pantalla se encuentra el texto di,TUTORIAL, cuando el usuario diga la palabra “tutorial” se activa un video tutorial explicando el funcionamiento del aplicativo. Para cerrar el tutorial deben pronunciarse las palabras “CERRAR TUTORIAL”

**Apagar Audio.** Al lado derecho se encuentra el texto di,APAGAR AUDIO. Por defecto al iniciar el aplicativo el audio inicia automáticamente si EL USUARIO quiere apagar el audio debe pronunciar las palabras “APAGAR AUDIO”.

**Elementos Interactivos.** En la parte inferior se encuentra un texto que dice Elementos Interactivos con una flecha hacia la derecha señalando 3 textos:

**Cámaras.** En la parte inferior se encuentra el texto di,CÁMARAS. Al pronunciar la palabra” CÁMARAS” se muestra una pantalla de un plano en planta con la ubicación de las cámaras y el nombre de cada cámara, para que al pronunciarlo la cámara se desplace hacia el lugar deseado.

Los textos que se pueden elegir para el desplazamiento de las cámaras son:

*Sala 1- Sala 2 – Comedor - Alcoba Principal – Baño - Alcoba 1 - Alcoba 2 - Planta*

Al pronunciar cualquiera de las palabras anteriores la cámara se desplaza de acuerdo con el lugar que se quiere visualizar. Para cerrar este menú EL USUARIO debe pronunciar la palabra “CERRAR CÁMARAS”.

**Pisos.** En la parte inferior se encuentra el texto di,PISOS. Al pronunciar la palabra “PISOS” se muestran 3 círculos cada uno con un material diferente y el nombre en la parte de abajo, para que al pronunciarlo el piso cambie de material al seleccionado.

Los textos que se pueden elegir para cambiar el material del piso son:

*Laminado – Parquet - Baldosa*

Para cerrar este menú EL USUARIO debe pronunciar la palabra “CERRAR PISOS”.

**Mesón Cocina.** En la parte inferior se encuentra el texto di,MESÓN COCINA. Al pronunciar la palabra “MESÓN COCINA” se muestran 3 círculos cada uno con un material diferente y el nombre en la parte de abajo, para que al pronunciarlo el mesón cambie de material al seleccionado

Los textos que se pueden elegir para cambiar el material del piso son:

*Granito – Mármol - Acero*

Para cerrar este menú EL USUARIO debe pronunciar la palabra “CERRAR MESÓN”.

**Luces.** En la parte inferior se encuentra el texto di,LUCES. Al pronunciar la palabra “LUCES” se muestran 3 círculos cada uno con un material diferente y el nombre en la parte de abajo, para que al pronunciarlo el mesón cambie de material al seleccionado.

Los textos que se pueden elegir para cambiar el material de la cocina son:

*Encender cocina-apagar cocina*

*Encender sala-apagar sala*

*Encender alcoba principal-apagar alcoba principal*

*Encender baño-apagar baño*

*Encender alcoba 1-apagar alcoba 1*

*Encender alcoba 2-apagar alcoba 2*

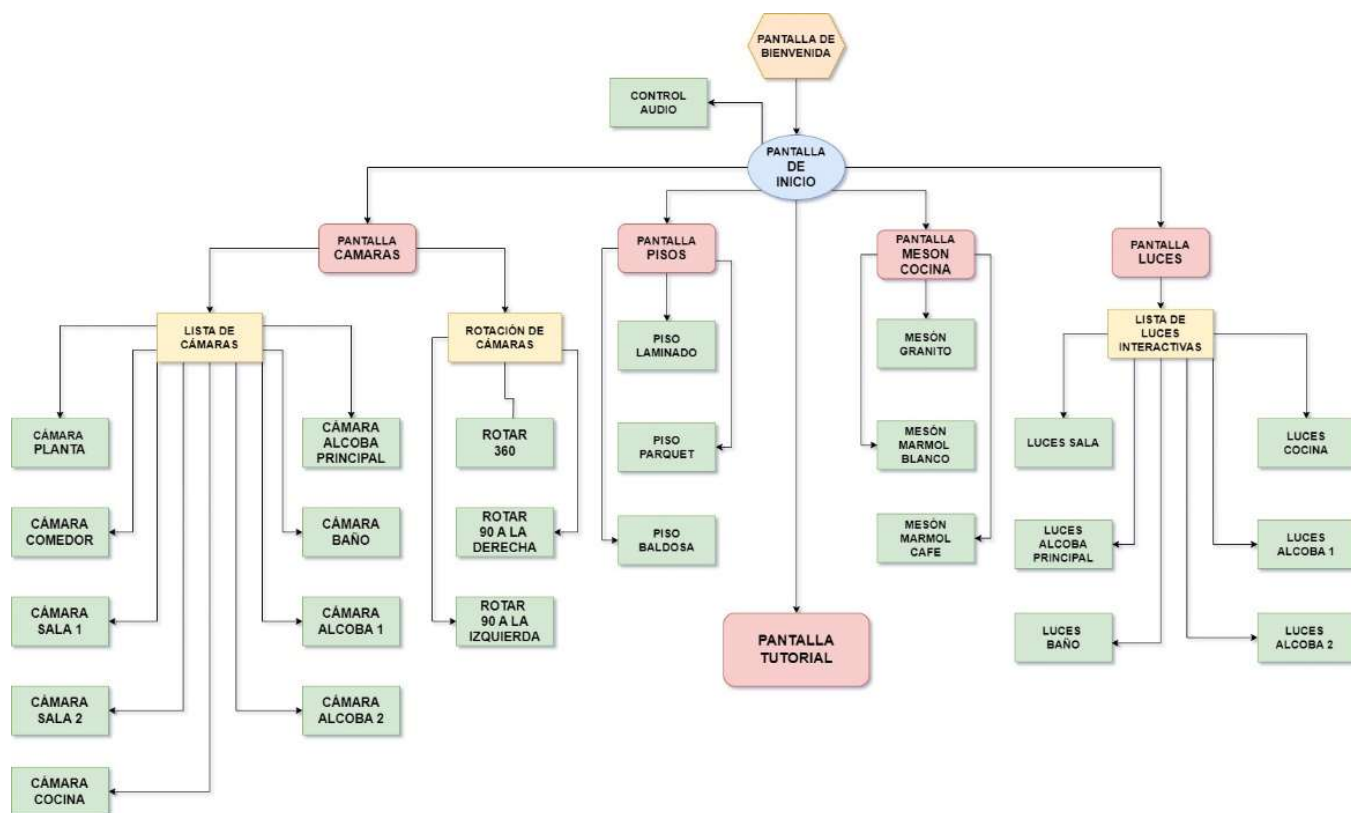
Para cerrar este menú EL USUARIO debe pronunciar la palabra “CERRAR LUCES”.

## Guion Técnico

Para la creación de guion técnico es necesario establecer como primera medida cual es el flujo de las pantallas y/o escenas por lo cual a continuación se muestra un **Árbol de Navegación** para determinar la estructura de los escenarios del aplicativo. (figura 4)

**Figura 4**

*Árbol de Navegación*



*Fuente.* Autoría propia.

**Tabla 1***(Guion Técnico) Pantalla de Bienvenida*

Escena y/o Pantalla	Plano	Descripción de la Pantalla y/o Escena	Angulo	Sonido	Duración Plano
1	General	Página de Bienvenida. Texto de bienvenida y contador	No Aplica	Sin Sonido	5 segundos

*Fuente.* Autoría propia.

**Tabla 2***(Guion Técnico) Pantalla Inicio*

Escena y/o Pantalla	Plano	Descripción de la Pantalla y/o Escena	Angulo	Sonido	Duración Plano
2	General	Página de Bienvenida. Título del aplicativo y menús de comandos interactivos	No Aplica	Música de fondo.	Definido por el usuario
3	General	Tutorial Se muestra un video tutorial cuando el usuario pronuncia la palabra "Tutorial". Para cerrar se pronuncian las palabras "cerrar tutorial".	No Aplica	Sonido creado en el video tutorial, voz en off y música de fondo.	5min.
4	General	Lista de comandos. Se muestra una lista de todos los comandos de voz, cuando el usuario pronuncia la palabra "lista", para cerrar se pronuncian las palabras " cerrar lista" Título del aplicativo y textos que sugieren comandos de voz generales.	No Aplica	Música de fondo.	Definido por el usuario
2	General	Apagar y encender audio... Al pronunciar las palabras "apagar audio" se desactivará el audio y al pronunciar las palabras "Encender Audio" se activará el audio.	No Aplica	Sonido de fondo activo por defecto cuando se inicie la escena.	1seg.
5	General	Cámaras Se muestra una lista de las cámaras, cuando el usuario pronuncia la palabra "cámaras". Para cerrar se pronuncian las palabras "cerrar cámaras".	No Aplica	Música de fondo.	Definido por el usuario
6	General	Pisos. Se muestra una lista de los pisos, cuando el usuario pronuncia la palabra "pisos". Para cerrar se pronuncian las palabras "cerrar pisos".	No Aplica	Música de fondo.	Definido por el usuario
7	General	Mesón cocina. Se muestra una lista de los materiales del mesón de la cocina,	No Aplica	Música de fondo.	Definido por el usuario

---

cuando el usuario pronuncia la palabra "mesón". Para cerrar se pronuncian las palabras "cerrar mesón".

---

*Fuente.* Autoría propia.

**Tabla 3**

*(Guion Técnico) Cámaras*

Escena y/o Pantalla	Plano	Descripción de la Pantalla y/o Escena	Angulo	Sonido	Duración Plano
9	General	La cámara se desplaza hacia la vista superior mostrando el apartamento en planta al pronunciar la palabra "Planta".	Normal	Música de fondo.	Definido por el usuario
10	Entero	La cámara se desplaza hacia la sala mostrando la sala desde la derecha al pronunciar la palabra "sala1"	Normal	Música de fondo.	Definido por el usuario
11	Entero	La cámara se desplaza hacia la sala mostrando la sala desde la Izquierda al pronunciar la palabra "sala2"	Contrapicado	Música de fondo.	Definido por el usuario
12	Americano	La cámara se desplaza hacia el comedor al pronunciar la palabra "comedor"	Normal	Música de fondo.	Definido por el usuario
13	Entero	La cámara se desplaza hacia la cocina al pronunciar la palabra "cocina"	Normal	Música de fondo.	Definido por el usuario
14	Entero	La cámara se desplaza hacia la alcoba principal al pronunciar las palabras "alcoba principal"	Normal	Música de fondo.	Definido por el usuario
15	Americano	La cámara se desplaza hacia el baño de la alcoba principal al pronunciar la palabra "baño"	Normal	Música de fondo.	Definido por el usuario
16	Entero	La cámara se desplaza hacia la alcoba1 al pronunciar la palabra "alcoba1"	Normal	Música de fondo.	Definido por el usuario
17	Entero	La cámara se desplaza hacia la alcoba2 al pronunciar la palabra "alcoba2"	Normal	Música de fondo.	Definido por el usuario

*Fuente.* Autoría propia.

**Tabla 4***(Guion Técnico) Pisos*

Escena y/o Pantalla	Plano	Descripción de la Pantalla y/o Escena	Angulo	Sonido	Duración Plano
18	General	Se cambia el material del piso a piso laminado, al pronunciar las palabras "piso laminado".	No Aplica	Música de fondo.	Definido por el usuario
19	General	Se cambia el material del piso a piso parquet, al pronunciar las palabras "piso parquet".	No Aplica	Música de fondo.	Definido por el usuario
20	General	Se cambia el material del piso a piso en baldosa, al pronunciar las palabras "piso baldosa".	No Aplica	Música de fondo.	Definido por el usuario

*Fuente.* Autoría propia.**Tabla 5***(Guion Técnico) Mesón Cocina*

Escena y/o Pantalla	Plano	Descripción de la Pantalla y/o Escena	Angulo	Sonido	Duración Plano
21	General	Se cambia el material del mesón a mesón en granito, al pronunciar las palabras "mesón granito".	No Aplica	Música de fondo.	Definido por el usuario
22	General	Se cambia el material del mesón a mesón en mármol, al pronunciar las palabras "mesón mármol".	No Aplica	Música de fondo.	Definido por el usuario
23	General	Se cambia el material del mesón a mesón en acero, al pronunciar las palabras "mesón acero".	No Aplica	Música de fondo.	Definido por el usuario

*Fuente.* Autoría propia.

**Tabla 6***(Guion Técnico) Luces*

Escena y/o Pantalla	Plano	Descripción de la Pantalla y/o Escena	Angulo	Sonido	Duración Plano
24	Depende de la Cámara en que se encuentre posicionado	Se apagan las luces de la sala al pronunciar las palabras "apagar sala"	Depende de la Cámara en que se encuentre posicionado	Música de fondo.	Definido por el usuario
25	Depende de la Cámara en que se encuentre posicionado	Se encienden las luces de la sala al pronunciar las palabras "encender sala"	Depende de la Cámara en que se encuentre posicionado	Música de fondo.	Definido por el usuario
26	Depende de la Cámara en que se encuentre posicionado	Se apagan las luces de la cocina al pronunciar las palabras "apagar cocina"	Depende de la Cámara en que se encuentre posicionado	Música de fondo.	Definido por el usuario
27	Depende de la Cámara en que se encuentre posicionado	Se encienden las luces de la cocina al pronunciar las palabras "encender cocina"	Depende de la Cámara en que se encuentre posicionado	Música de fondo.	Definido por el usuario
28	Depende de la Cámara en que se encuentre posicionado	Se apagan las luces de la alcoba principal al pronunciar las palabras "apagar alcoba principal"	Depende de la Cámara en que se encuentre posicionado	Música de fondo.	Definido por el usuario
29	Depende de la Cámara en que se encuentre posicionado	Se encienden las luces de la alcoba principal al pronunciar las palabras "encender alcoba principal"	Depende de la Cámara en que se encuentre posicionado	Música de fondo.	Definido por el usuario
30	Depende de la Cámara en que se encuentre posicionado	Se apagan las luces del baño de la alcoba principal al pronunciar las palabras "apagar baño"	Depende de la Cámara en que se encuentre posicionado	Música de fondo.	Definido por el usuario
31	Depende de la Cámara en que se encuentre posicionado	Se encienden las luces del baño de la alcoba principal al pronunciar las palabras "encender baño"	Depende de la Cámara en que se encuentre posicionado	Música de fondo.	Definido por el usuario

*Fuente.* Autoría propia.

### **Generalidades del Proyecto**

Para el desarrollo del proyecto se trabajó con el Sistema Operativo “WINDOWS 10” primero se realizó un diseño arquitectónico en AutoCad y posteriormente se importó el archivo 2d en 3dmax versión 2021, a partir del plano en planta se realizó el modelado en 3d de los espacios arquitectónicos y el amoblado. Una vez modelado el proyecto en 3d se asignaron unos materiales estándar en 3dmax junto con el mapeado, para posteriormente exportarlo como FBX e

importarlo al motor de videojuegos Unity, luego se creó la iluminación global, la iluminación artificial, y se configuraron los elementos interactivos, los cuales fueron programados en el lenguaje de programación c# mediante el gestor de código Visual Studio. Se utilizó Photoshop para la digitalización de los Sprites y la creación de la Interfaz.

### ***Programas Utilizados Para el Proyecto***

**AutoCAD.** Es un software de diseño asistido por computador desarrollado por la compañía Autodesk, es utilizado para la realización de diseños en 2d y 3d. En este proyecto específico se realizó el diseño arquitectónico en 2d del proyecto, en el cual se plasma la distribución de los espacios, así como del mobiliario y la ubicación de los elementos interactivos en planta.

**3dMax.** Es un software de diseño desarrollado por la compañía Autodesk especializado en la creación de modelos en 3d y animación digital. En este proyecto se realizó el modelado arquitectónico del apartamento, se crearon todos los espacios, puertas, ventanas, pisos, etc., además de la inclusión de los objetos que hacen parte del entorno como el amoblado y todos los accesorios.

**Unity.** Este software es desarrollado por la compañía Unity Technologies el cual fue creado primordialmente para el desarrollo de video juegos. Actualmente es uno de los motores de video juegos más utilizados en el mercado para el desarrollo de video juegos y aplicaciones interactivas. Su primera versión se lanzó en el año 2005 y a partir de entonces ha tenido una gran evolución en tecnología. Inicialmente se centró en el desarrollo de videojuegos para Mac, pero fueron cambiando su enfoque y lo desarrollaron hasta hacerlo más versátil. Esta herramienta permite no solo crear video juegos sino aplicativos interactivos con diferentes fines y con diversas plataformas como PC, móviles, consolas de video juegos, a través de un editor visual y

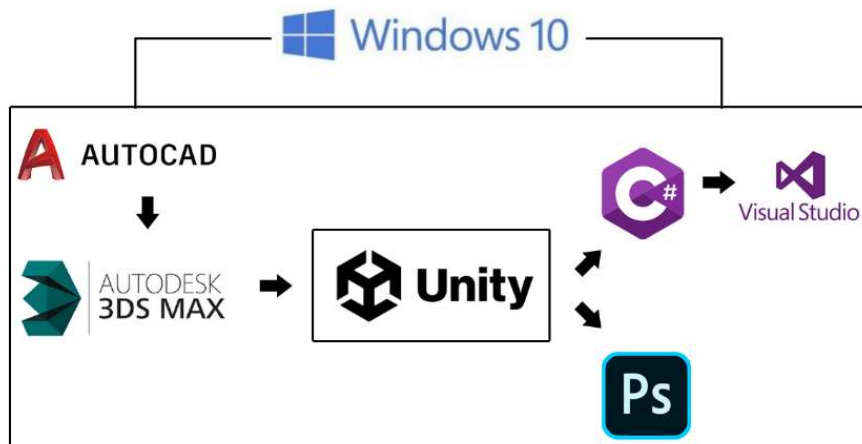
una programación en `c#`, por lo cual se pueden obtener resultados muy profesionales. Unity es una herramienta muy versátil y se caracteriza por ser multiplataforma ya que los proyectos pueden ser exportados para Windows, iOS, MacOS, Xbox, PlayStation, Android, etc.

Para este aplicativo se utilizó la versión 2018.3.7f1, donde se hizo la importación de los archivos 3d, se configuraron todas las interacciones en el editor y se programaron vía scripting en `c#`, luego el proyecto se exportó para ser reproducido en un computador de escritorio en Windows.

**C#.** Es un lenguaje de programación orientada a objetos el cual fue desarrollado por Microsoft y se deriva de C y C++, utilizando modelos de plataforma.NET. En este lenguaje de programación se escriben los scripts en Unity para controlar los comportamientos y la lógica de los objetos como, las colisiones, las animaciones, la interactividad, etc. Para este proyecto se desarrollaron varios scripts, para configurar la interactividad de las cámaras, los materiales, las luces y el control de los demás elementos.

**Visual Studio.** Es un entorno de desarrollo integrado para Windows y MacOS desarrollado por Microsoft y es compatible con varios lenguajes de programación. Se pueden desarrollar aplicaciones de escritorio, tipo web y para móviles. Para este proyecto se utilizó como editor de código para los scripts realizados con el lenguaje de programación `c#`.

**Photoshop.** Este es un programa para edición de imagen el cual fue desarrollado por la empresa Adobe Systems para Windows y MacOS. A partir de 1990 desde su lanzamiento ha ido evolucionando grandemente y hoy en día se ha convertido en la más completa herramienta preferida por diseñadores para retoque fotográfico. En este proyecto en particular se crearon los sprints necesarios para la interfaz del aplicativo.

**Figura 5***Programas Utilizados en el Proyecto*

*Fuente.* 1000marcas.net. (2023). [Imágenes logos]. En 1000marcas.net.

<https://1000marcas.net/logos>

## Prototipado

Se realizaron pruebas de prototipado como versión inicial con el fin de evaluar la idea y validar si realmente cumple con los objetivos iniciales del proyecto, además de confirmar la viabilidad para su desarrollo.

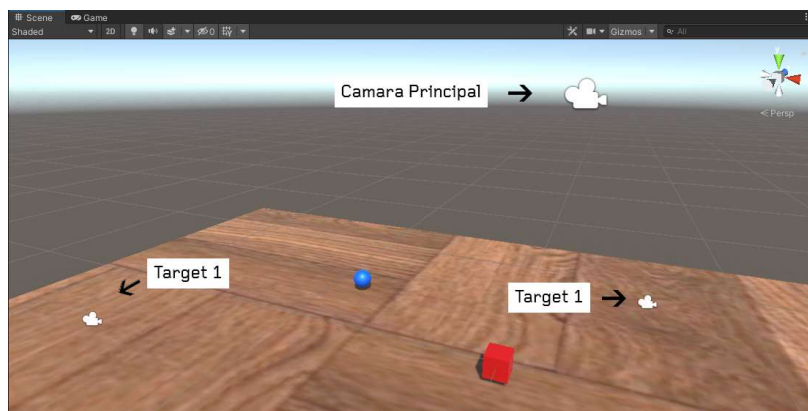
El objetivo de este prototipado fue comprobar el funcionamiento de los elementos interactivos controlados por comandos de voz. Inicialmente se probó el desplazamiento de las cámaras controladas por comandos de voz y posteriormente se probó el funcionamiento de los materiales interactivos controlados por comandos de voz.

### *Prototipado Desplazamiento de Cámaras Controladas por Comandos de Voz.*

Para esto se creó un escenario básico en Unity con un plano y dos objetos de referencia una esfera y un cubo. Ahora se crea una cámara principal donde lo que se quiere es que la cámara principal se desplace a la posición de otras cámaras que actuaran como Target.

### Figura 6

#### *Ubicación Cámaras en el escenario*



*Fuente.* Autoría propia, entorno Unity [Captura de Pantalla]

Creamos un script en Unity “ControlCamaraConVoz” para controlar el desplazamiento de la cámara hacia el target y lo editamos en c# VisualStudio.

En la primera línea se crea el primer Target

```
public Camera targetCamera;
```

Con la siguiente línea se controla el movimiento de la Cámara principal hacia el target

```
public float velocidadMovimiento = 5f;
```

Se declara la variable “KeywordRecognizer” con la cual se reconocen las palabras clave.

```
private KeywordRecognizer keywordRecognizer;
```

Declaramos la palabra clave:

```
public string palabraClaveMover = "Target 1";
```

Se declara la variable “moviendoHaciaTarget” donde se inicia su valor como “false”

```
private bool moviendoHaciaTarget = false;
```

Luego se invoca el siguiente método cuando se reconoce la palabra clave y si la

palabra coincide se establece la variable booleana “moviendoHaciaTarget” como “true”

```
private void OnPhraseRecognized(PhraseRecognizedEventArgs args)
```

```
{
    if (args.text == palabraClaveMover)
    {
        moviendoHaciaTarget = true;
    }
}
```

Finalmente, en el método “FixedUpdate”

```
private void FixedUpdate()
```

```
{
    if (moviendoHaciaTarget)
    {
```

// Se Calcula la dirección hacia la posición de la targetCamera

```
Vector3 direccionAlTarget = targetCamera.transform.position - transform.position;
```

// Se calcula la distancia entre la cámara y la targetCamera

```
float distanciaAlTarget = direccionAlTarget.magnitude;
```

// Si la distancia es mayor a un pequeño valor, seguir moviéndose hacia la targetCamera

```
if (distanciaAlTarget > 0.1f)
{
```

// Se normaliza la dirección para evitar cambios bruscos en la velocidad

```
Vector3 direccionNormalizada = direccionAlTarget.normalized;
```

// Se calcula la cantidad de movimiento para este frame

```
float movimientoStep = velocidadMovimiento * Time.fixedDeltaTime;
```

// Se limita el movimiento para no pasar más allá de la targetCamera

```
movimientoStep = Mathf.Min(movimientoStep, distanciaAlTarget);
```

// Se mueve la cámara hacia la targetCamera

```
transform.position += direccionNormalizada * movimientoStep;
```

// Se rota la cámara hacia la rotación de la targetCamera

```
float interpolacionRotacion = 0.01f;
```

// Se ajusta este valor para obtener una rotación más suave

```
transform.rotation = Quaternion.Slerp(transform.rotation,
```

```
targetCamera.transform.rotation, interpolacionRotacion);
```

```
}
else
```

```

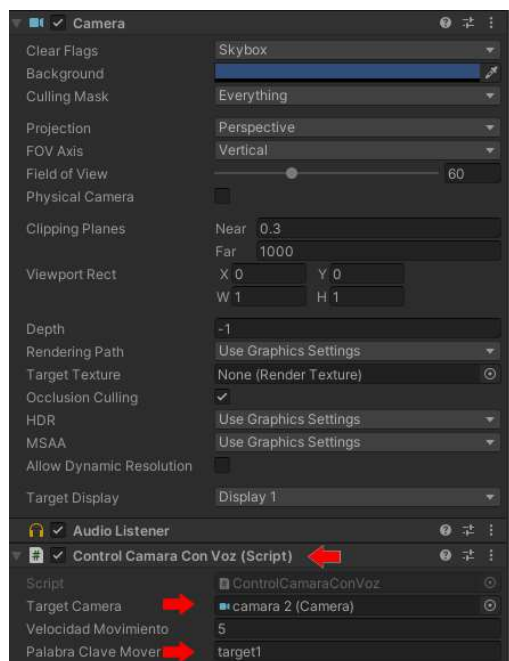
    {
    // Se detiene el movimiento si ya alcanzó la targetCamera
        moviendoHaciaTarget = false;
    }
}
}

```

Finalmente, en el inspector se agrega el script a la cámara principal la cual realiza los desplazamientos. Y se agrega la cámara que sirve como Target 1

## Figura 7

### Configuración del script de cámaras en el Inspector

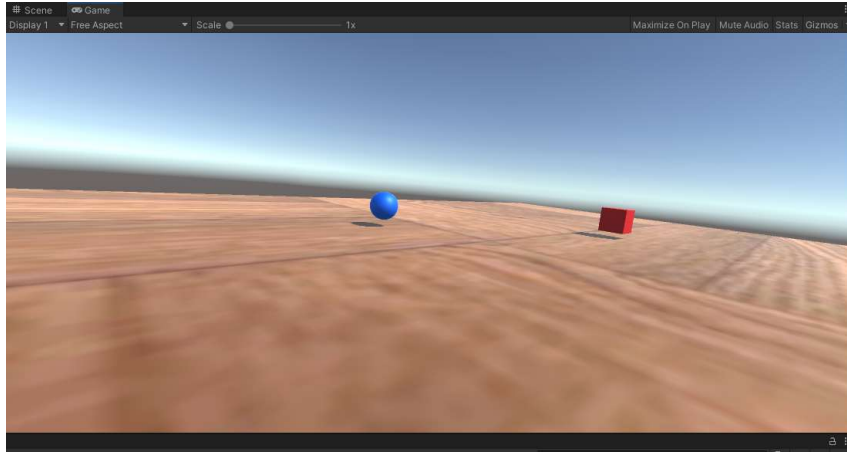


*Fuente.* Autoría propia, entorno Unity [Captura de Pantalla]

Ahora cuando ejecutamos el aplicativo en Unity al pronunciar la palabra “target1” la cámara principal se desplaza hacia la posición de la cámara2 a una velocidad 5f. Según los parámetros que se configuraron en el script.

## Figura 8

*Cámara principal toma ubicación del target1*



*Fuente.* Autoría propia, entorno Unity [Captura de Pantalla]

Se configura de igual manera el “target2” que va asociado a otra cámara y así sucesivamente dependiendo del número de posiciones de cámaras que se necesiten.

### ***Prototipado Texturas Interactivas Controladas por Comandos de Voz.***

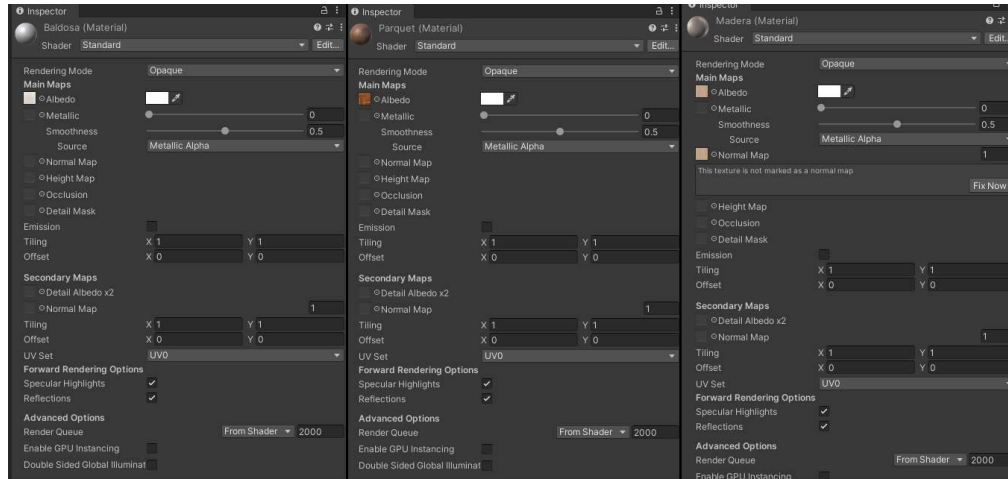
Para crear las texturas interactivas se hacen las pruebas sobre el plano que se creó que en este caso simulará el piso.

Creamos 3 materiales Unity con lo cual contaremos con 3 opciones para cambio de material. Los 3 materiales serán:

*Baldosa – Parquet - Madera*

## Figura 9

### Creación de materiales en Unity



*Fuente.* Autoría propia, entorno Unity [Captura de Pantalla]

Se crea un script desde Unity “CambiarTexturasPiso”. Lo que se quiere con este script es controlar la interactividad de los materiales sobre el plano creado “piso”, para que al pronunciar un comando de voz se cambie el material.

Primero se agregan las librerías necesarias para el funcionamiento del script incluyendo el reconocimiento de voz.

```
using System;
using System.Collections;
using System.Collections.Generic;
using System.Linq;
using UnityEngine;
using UnityEngine.Windows.Speech;

public class CambiarTexturaPiso : MonoBehaviour
{
    public List<Material> materials;
    // se crea la lista de materiales para después asignarlos desde el inspector
    KeywordRecognizer keywordRecognizer;
    // se crea un objeto “KeywordRecognizer” y un diccionario que asocia las palabras clave
    // con acciones (WordToAction)
    Dictionary<string, Action> wordToAction;
    // En el método “Start”, se inicializa el diccionario con palabras clave y las acciones
    // asociadas a ellas. Luego, se configura el reconocedor de palabras clave con esas
    // palabras y se inicia.
    void Start()
    {
```

```

wordToAction = new Dictionary<string, Action>();
wordToAction.Add("a", A);
wordToAction.Add("b", B);
wordToAction.Add("c", C);
keywordRecognizer = new KeywordRecognizer(wordToAction.Keys.ToArray());
keywordRecognizer.OnPhraseRecognized += WordRecognized;
keywordRecognizer.Start();
}

```

// Se llama el método “WordRecognized” cuando una palabra clave es reconocida y posteriormente se imprime la palabra reconocida en la consola y luego invoca la acción asociada a esa palabra clave.

```

private void WordRecognized(PhraseRecognizedEventArgs word)
{
    Debug.Log(word.text);
    wordToAction[word.text].Invoke();
}

```

// En respuesta a palabras clave específicas se llaman los métodos “A”, “B”, “C” Cada método invoca el método “ChangeMaterial” con un material específico de la lista “materials”.

```

private void C()
{
    ChangeMaterial(materials[0]); // Cambia al primer material en la lista
    Debug.Log("respuesta correcta");
}
private void B()
{
    ChangeMaterial(materials[1]); // Cambia al segundo material en la lista
}

private void A()
{
    ChangeMaterial(materials[2]); // Cambia al tercer material en la lista
}

```

// Finalmente, con el método “ChangeMaterial” se cambia el material del objeto al nuevo material proporcionado. Verifica que tanto el nuevo material como el componente “Renderer” del objeto no sean nulos antes de realizar el cambio.

```

private void ChangeMaterial(Material newMaterial)
{
    Renderer renderer = GetComponent<Renderer>();

    if (newMaterial != null && renderer != null)
    {
        renderer.material = newMaterial;
    }
}

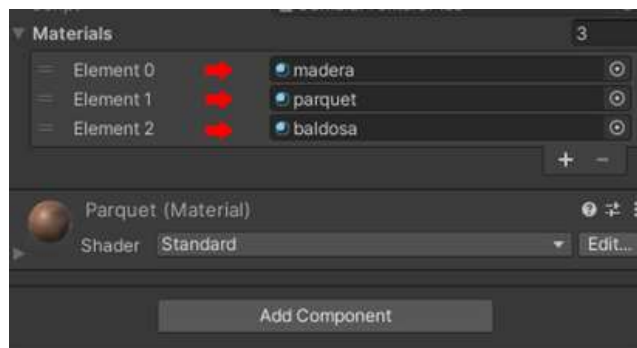
```

Ahora pasamos al Unity y en el inspector le asignamos al plano el script y en el script

agregamos los materiales que han sido creados previamente.

## Figura 10

*Se añaden los materiales en el inspector*

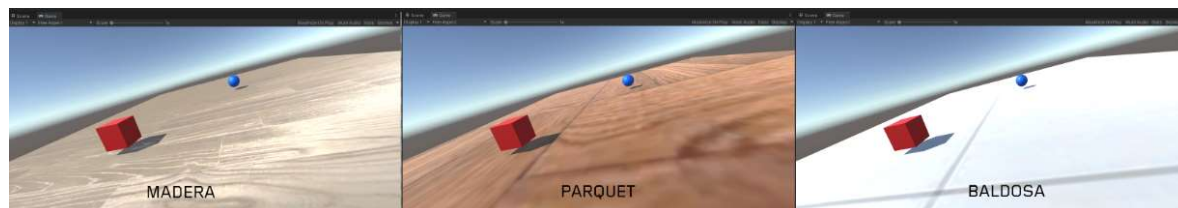


*Fuente.* Autoría propia, entorno Unity [Captura de Pantalla]

Al ejecutar el aplicativo al pronunciar el comando de voz “a” el material del plano cambia al material “madera” al pronunciar el comando “b” el material cambia al material “parquet” y al pronunciar el comando de voz “c” el material del plano cambia al material “baldosa”

## Figura 11

*Cambio de materiales al usar los comandos de voz*



*Fuente.* Autoría propia, entorno Unity [Captura de Pantalla]

## ***Pruebas Funcionales Prototipo***

**Tabla 7**

### *Pruebas funcionales del prototipo*

Identificador Prueba	Nombre de la Prueba	Descripción de la Prueba	Comportamiento Esperado	Comportamiento Obtenido
P.F.P.01	Desplazamiento de la cámara hacia el <b>target 1</b>	Pronunciar el comando de voz, " <b>target1</b> ", para que la cámara se desplace hacia la posición de la cámara " <b>target1</b> "	La cámara se desplaza a la posición del " <b>target1</b> "	La cámara se desplaza a la posición del " <b>target1</b> ", haciendo un movimiento continuo de rotación.
P.F.P.02	Desplazamiento de la cámara hacia el <b>target 2</b>	Pronunciar el comando de voz, " <b>target2</b> ", para que la cámara se desplace hacia la posición de la cámara " <b>target2</b> "	La cámara se desplaza a la posición del " <b>target2</b> "	La cámara se desplaza a la posición del " <b>target2</b> ", haciendo un movimiento continuo de rotación.
P.F.P.03	Cambio de material del plano a material " <b>A</b> "	Pronunciar el comando de voz, " <b>A</b> ", para que material del plano cambie al material " <b>Baldosa</b> "	El material del plano se cambia al nuevo material " <b>Baldosa</b> "	El material del plano se cambia al nuevo material " <b>Baldosa</b> ".
P.F.P.03	Cambio de material del plano a material " <b>B</b> "	Pronunciar el comando de voz, " <b>B</b> ", para que material del plano cambie al material "Laminado"	El material del plano se cambia al nuevo material " <b>laminado</b> "	El material del plano se cambia al nuevo material " <b>Laminado</b> ".
P.F.P.03	Cambio de material del plano a material " <b>C</b> "	Pronunciar el comando de voz, " <b>C</b> ", para que material del plano cambie al material " <b>Parquet</b> "	El material del plano se cambia al nuevo material " <b>Parquet</b> "	El material del plano se cambia al nuevo material " <b>Parquet</b> ".

*Fuente.* Autoría propia, entorno Unity [Captura de Pantalla]

### ***Conclusión de las Pruebas de Prototipado***

Al realizar los testeos del aplicativo según el prototipo creado se puede concluir que la realización del proyecto es viable técnicamente y se puede crear un aplicativo arquitectónico de un inmueble que contenga elementos interactivos como cámaras y materiales los cuales puedan ser controlados por medio de comandos de voz.

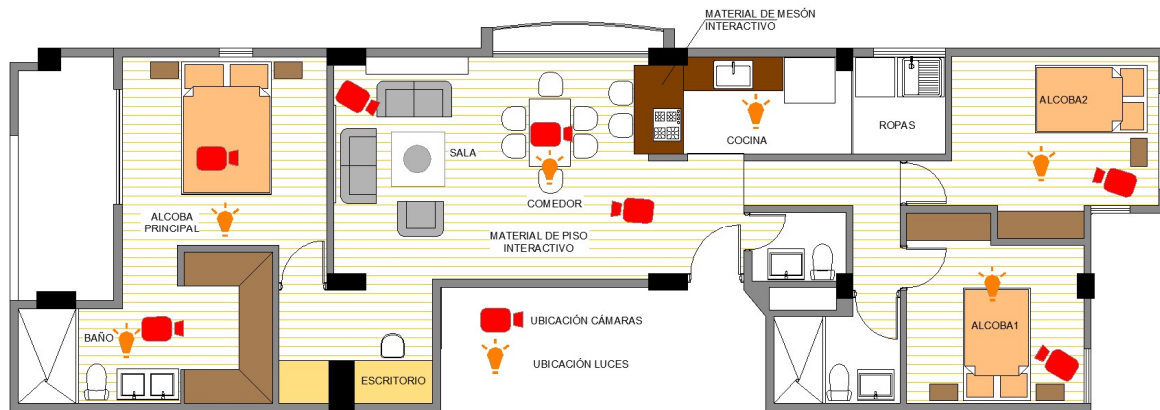
### **Boceto General del Proyecto**

En el boceto general del proyecto (Imagen 1) se puede apreciar la distribución arquitectónica del apartamento amoblado con la ubicación de los elementos interactivos como las

cámaras, las luces, el mesón de la cocina y el piso. Con base a este boceto se construyen los assets en la etapa de producción.

## Figura 12

### *Boceto General del Proyecto*



*Fuente.* Autoría propia, entorno Autocad [Captura de Pantalla]

## Producción

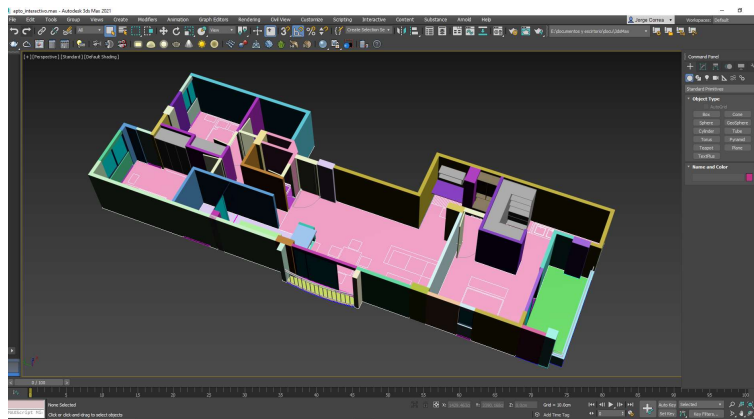
En esta etapa se desarrolló el aplicativo y se crearon todos los Assets necesarios, iniciando por el modelado en 3d, la creación de materiales iniciales en el programa 3d, y se importó el modelo en Unity, para cambiar los materiales a materiales Unity. Se configuró la iluminación tanto global como artificial y se creó la interactividad, a partir de la creación de códigos. Una vez creada la interactividad se diseñó la interfaz del usuario, para finalmente realizar las pruebas de funcionamiento observación y accesibilidad.

### Modelado y Texturizado en el Software 3d

Con base a un plano digitalizado en AutoCAD se levanta el modelo 3d del proyecto, modelando los muros, ventanas pisos y posteriormente se agregaron materiales estándar los que posteriormente fueron cambiados por materiales Unity. Para este proyecto el software utilizado para el modelado en 3d fue 3d Studio Max.

### Figura 13

*Modelo 3d Estructura del Apartamento*

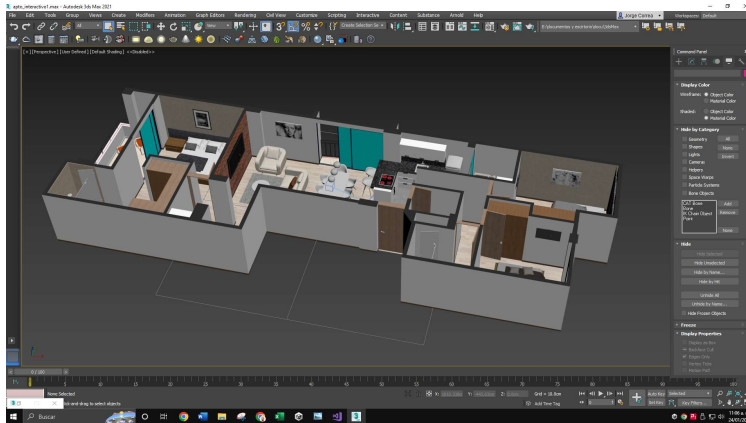


*Fuente.* Autoría propia, entorno 3dmax [Captura de Pantalla]

Una vez creada la estructura del apartamento se modelan y se agregan los muebles y los accesorios necesarios, que corresponden al entorno del apartamento y finalmente se exporta el modelo en formato FBX y después se importa en el motor de video juegos Unity.

## Figura 14

### *Modelo 3d con Texturas y Muebles*



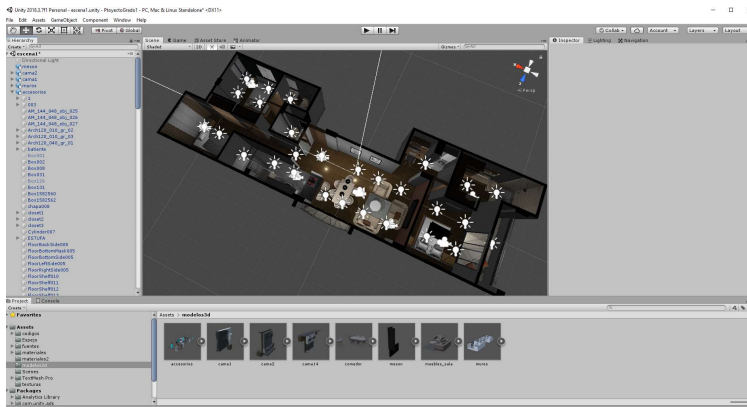
*Fuente.* Autoría propia, entorno 3dmax [Captura de Pantalla]

## **Incorporación del Modelo 3d en Unity**

Una vez el modelo 3d está listo y exportado desde 3d Studio Max a formato FBX, se crea un nuevo proyecto en Unity 2018.3.7f1 y se importa el modelo. Como el modelo es bastante pesado se importaron varios modelos de forma independiente, como los muros, accesorios y algunos muebles.

## Figura 15

### *Modelo 3d incorporado en Unity*



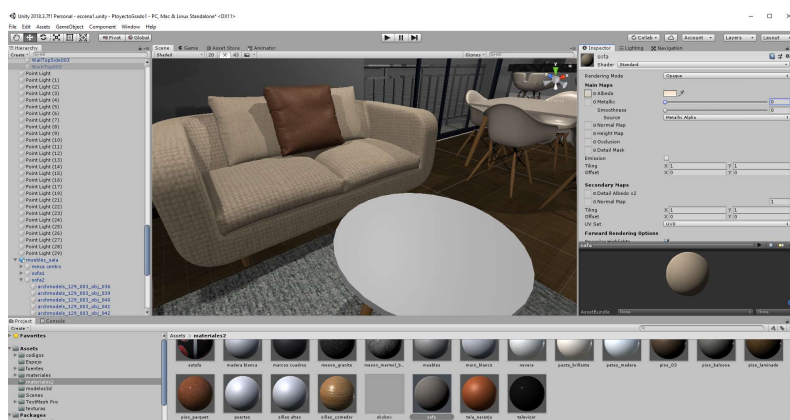
*Fuente. Autoría propia, entorno 3dmax [Captura de Pantalla]*

## Texturizado en Unity

Ya con el modelo incorporado se realiza el texturizado en Unity, se cambian los materiales estándar por materiales Unity, y posteriormente se genera la interactividad de materiales.

Se crea el material en Unity a través del menú Assets/create/material y en el material en Main Map Albedo se inserta la textura del material y se configura la suavidad y el brillo del material.

**Figura 16**  
*Texturizado en Unity*



*Fuente. Autoría propia, entorno 3dmax [Captura de Pantalla]*

## **Iluminación**

Una vez el modelo está listo con los materiales Unity se hizo la configuración de la iluminación, donde primero se configura la iluminación global y posteriormente se agregan luces artificiales.

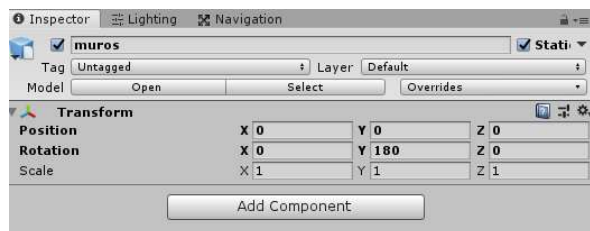
### ***Configuración de la iluminación Global***

Por defecto en Unity hay una luz (Directional Light) que simula la luz solar, esta luz se puede rotar y ubicar la dirección por donde se quiere que entre la luz del sol y se debe verificar que el tipo de sombras este activo en (Shadow Type).

Se deben iluminar las partes de la escena que no reciben iluminación directa, por medio la luz ambiental. Para realizar esto se deben seleccionar los modelos 3d que tenemos en la escena y activar la casilla Static (Imagen8) para que Unity haga el cálculo y se genere una iluminación ambiental sobre los objetos.

## Figura 17

### *Iluminación ambiental objetos estáticos*

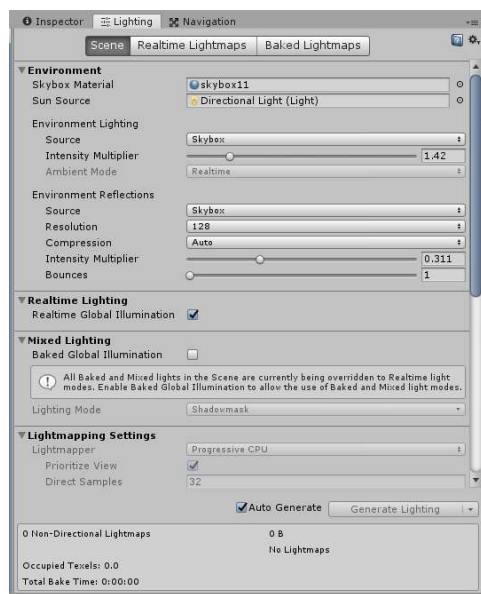


*Fuente.* Autoría propia, entorno Unity [Captura de Pantalla]

Posteriormente se hace la configuración de la iluminación ambiental, abriendo la ventana “Lighting” aquí se pueden modificar los parámetros como la intensidad de la luz y las reflexiones, además se puede agregar un material Skybox, para simular un cielo entorno más realista. Se puede utilizar un skybox de alguna librería de Unity, pero en este caso creamos el skybox con un mapa HDRI.

## Figura 18

### *Configuración de la iluminación Ambiental*

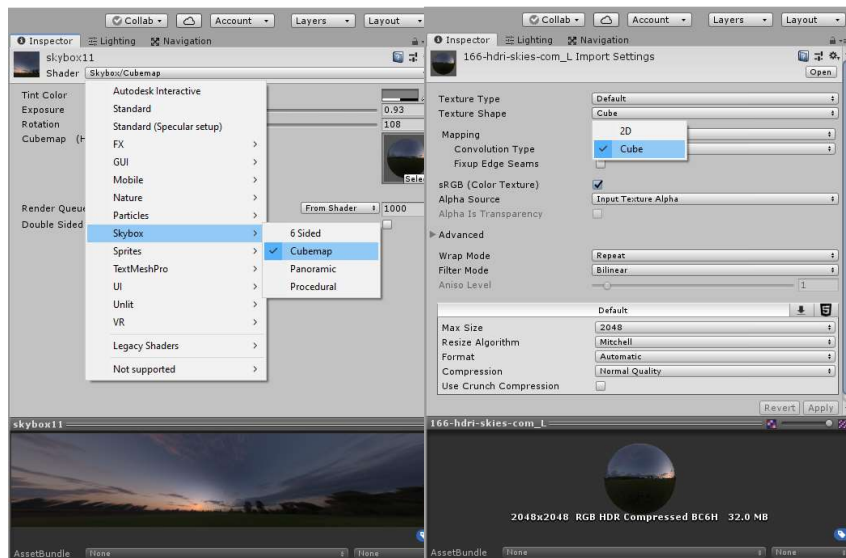


*Fuente.* Autoría propia, entorno Unity [Captura de Pantalla]

Para Crear el material Skybox se crea un material Unity y en la pestaña shader se activa Skybox/cubemap. Posteriormente se consigue un mapa HDRI de un cielo realista y se cambia de 2d a Cube y una vez creado el mapa se arrastra al inspector a la ventana Cubemap.

## Figura 19

### *Configuración Skybox*



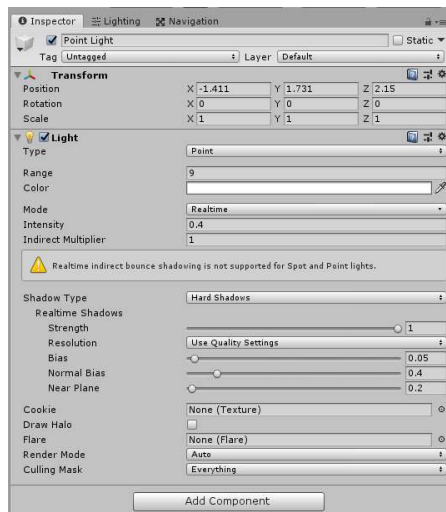
*Fuente.* Autoría propia, entorno Unity [Captura de Pantalla]

### *Configuración de Luces Artificiales*

Para agregar luces artificiales y simular los focos de luz, se utilizan luces “Point Light”, se configura el rango de luz, el color de la luz, la intensidad de la luz y el multiplicador indirecto. Además, en la pestaña modo utilizamos Realtime y se configura el tipo de sombra que pueden ser suaves o duras y finalmente se configura la fuerza y la resolución de las sombras

## Figura 20

### Configuración luces artificiales



*Fuente.* Autoría propia, entorno Unity [Captura de Pantalla]

## Configuración Elementos Interactivos

El aplicativo tiene varios elementos interactivos como cámaras, materiales y luces, que son controlados por medio de comandos de voz.

### Cámaras

El aplicativo inicia con la cámara “planta” por defecto (Imagen 21). En esta cámara se puede visualizar el proyecto desde arriba mostrando un plano general.

**Figura 21**

*Cámara “planta” vista por defecto.*



*Fuente.* Autoría propia, entorno Unity [Captura de Pantalla]

Existen 8 cámaras más que están ubicadas en diferentes espacios. Si el usuario quiere visualizar un espacio en específico, puede pronunciar la palabra asociada a la cámara que se encuentra en ese espacio y la cámara “planta” se desplaza hacia la posición de la cámara seleccionada.

**Figura 22**

*Distribución de cámaras*

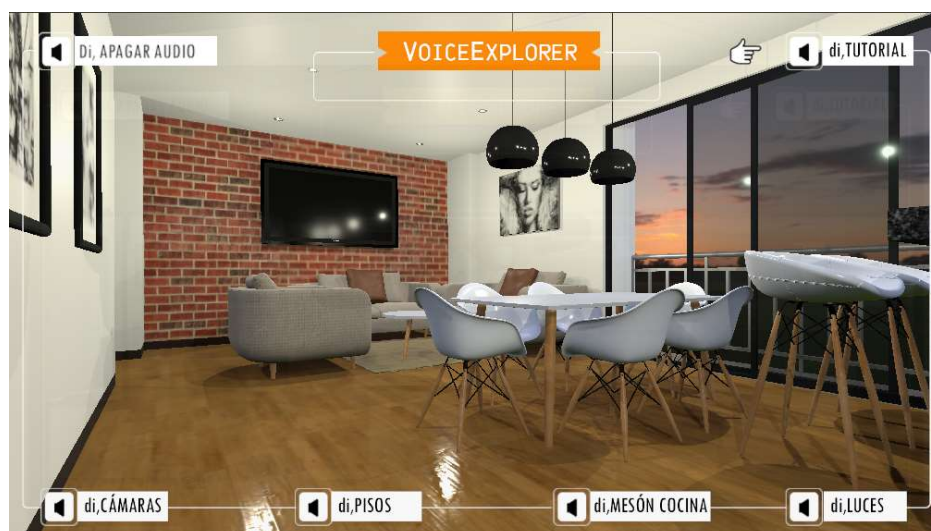


*Fuente.* Autoría propia, entorno Unity [Captura de Pantalla]

Por ejemplo, si el usuario quiere visualizar el espacio de la sala puede escoger entre las cámaras que se encuentran en ese espacio. Si el usuario escoge “sala1”, la cámara se desplaza hasta la posición de la cámara “sala1” (Imagen 23)

### Figura 23

*Visualización de Cámara “sala1”*



*Fuente.* Autoría propia, entorno Unity [Captura de Pantalla]

Para realizar la interactividad de las cámaras, se toma como base inicial la posición de la cámara “planta” la cual toma la posición de otra cámara según la orden dada por el usuario a través de un comando de voz.

Para lograr lo anterior se crea un script en *c#* que controle el movimiento de la cámara principal hacia un destino específico cuando se pronuncia la palabra clave.

Se crea una clase serializable para definir la palabra clave y la cámara destino

```
[System.Serializable]
public class DestinoPalabraClave
{
    public string palabraClave;
    public Camera targetCamera;
}
```

Después se crean las variables para la velocidad del movimiento, una lista para para los objetos de la clase y un objeto “keywordRecognizer” con el cual se reconocen las palabras clave.

```

Public float velocidadMovimiento = 5f;
public List<DestinoPalabraClave> destinosPalabrasClave;
private KeywordRecognizer keywordRecognizer;

```

En el “Start” se construye una lista de palabras clave (keywords) a partir de los destinos definidos en “destinosPalabrasClave”, se inicializa el “KeywordRecognizer” con las palabras clave y sus manejadores de eventos y se inicia el reconocimiento de palabras clave.

```

Private void Start()
{
    var keywords = new List<string>();
    foreach (var destino in destinosPalabrasClave)
    {
        keywords.Add(destino.palabraClave);
    }

    keywordRecognizer = new KeywordRecognizer(keywords.ToArray());
    keywordRecognizer.OnPhraseRecognized += OnPhraseRecognized;
    keywordRecognizer.Start();
}

```

El método OnPhraseRecognized, se ejecuta cuando se reconoce una palabra clave. Busca en la lista “destinosPalabrasClave” el destino correspondiente a la palabra clave reconocida. Si encuentra un destino, establece la cámara de destino actual y activa el movimiento hacia el destino.

```

Private void OnPhraseRecognized(PhraseRecognizedEventArgs args)
{
    DestinoPalabraClave destino = destinosPalabrasClave.Find(d =>
d.palabraClave == args.text);

    if (destino != null)
    {
        destinoActual = destino.targetCamera;
        moviendoHaciaTarget = true;
    }
}

```

Y finalmente en el Método FixedUpdate, se llama en cada frame fijo del aplicativo. Si “moviendoHaciaTarget” es verdadero y hay un “destinoActual”, la cámara se mueve hacia el destino, calcula la dirección y la distancia entre la posición actual de la cámara y la posición del destino, normaliza la dirección para evitar cambios bruscos en la velocidad, calcula la cantidad de

movimiento para el “frame” y limita el movimiento para no pasar más allá del destino. Luego mueve la cámara hacia el destino y rota la cámara para que mire en la dirección del destino y detiene el movimiento si la cámara alcanza el destino.

```

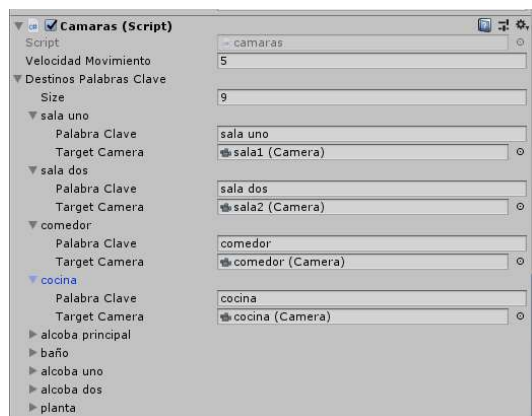
Private void FixedUpdate()
{
    if (moviendoHaciaTarget && destinoActual != null)
    {
        Vector3 direccionAlTarget = destinoActual.transform.position -
transform.position;
        float distanciaAlTarget = direccionAlTarget.magnitude;
        if (distanciaAlTarget > 0.1f)
        {
            Vector3 direccionNormalizada = direccionAlTarget.normalized;
            float movimientoStep = velocidadMovimiento * Time.fixedDeltaTime;
            movimientoStep = Mathf.Min(movimientoStep, distanciaAlTarget);
            transform.position += direccionNormalizada * movimientoStep;
            float interpolacionRotacion = 0.05f;
            transform.rotation = Quaternion.Slerp(transform.rotation,
destinoActual.transform.rotation, interpolacionRotacion);
        }
        else
        {
            moviendoHaciaTarget = false;
        }
    }
}

```

Por último, se selecciona la cámara principal se le asigna el código creado y se asignan las cámaras en el inspector.

## Figura 24

*Asignación de cámaras en el inspector*



*Fuente.* Autoría propia, entorno Unity [Captura de Pantalla]

## ***Materiales***

**Materiales Piso.** Para realizar el cambio de materiales del piso el usuario tiene 3 opciones que son: “piso laminado”, la cual carga en el modelo por defecto, “piso parquet” y “piso baldosa”. Para esto se crean los 3 materiales en Unity los cuales son asignados al inspector en el script que controla la interactividad. El reconocimiento de voz será manejado a través de la API de reconocimiento de voz de Windows.

Lo primero es importar las bibliotecas necesarias para trabajar con comandos de voz.

```
Using System;
using System.Collections;
using System.Collections.Generic;
using System.Linq;
using UnityEngine;
using UnityEngine.Windows.Speech;
```

Se define una clase llamada “ControlMaterialPiso” que hereda de “MonoBehaviour”, indicando que este script es un componente de Unity.

```
Public class ControlMaterialPiso : MonoBehaviour
```

Se declara una lista de materiales llamada “materials”. Estos materiales se asignarán desde el Inspector de Unity.

```
Public List<Material> materials;
```

Se declara un reconocedor de palabras clave (keywordRecognizer) y un diccionario que asocia palabras clave con acciones (wordToAction).

```
KeywordRecognizer keywordRecognizer;
Dictionary<string, Action> wordToAction;
```

En el método “Start”, se inicializa el diccionario con palabras clave y las acciones asociadas a ellas. Luego, se configura el reconocedor de palabras clave con esas palabras y se inicia.

```

Void Start()
{
    wordToAction = new Dictionary<string, Action>();
    wordToAction.Add("piso laminado", A);
    wordToAction.Add("piso parquet", B);
    wordToAction.Add("piso baldosa", C);
    keywordRecognizer = new KeywordRecognizer(wordToAction.Keys.ToArray());
    keywordRecognizer.OnPhraseRecognized += WordRecognized;
    keywordRecognizer.Start();
}

```

Se llama el método “WordRecognized” cuando una palabra clave es reconocida y posteriormente se imprime la palabra reconocida en la consola y luego invoca la acción asociada a esa palabra clave.

```

Private void WordRecognized(PhraseRecognizedEventArgs e)
{
    Debug.Log(e.Text);
    wordToAction[e.Text].Invoke();
}

```

En respuesta a palabras clave específicas se llaman los métodos “A”, “B”, “C”

Cada método invoca el método “ChangeMaterial” con un material específico de la lista “materials”.

```

Private void C()
{
    ChangeMaterial(materials[0]);
    Debug.Log("respuesta correcta");
}

private void B()
{
    ChangeMaterial(materials[1]);
}

private void A()
{
    ChangeMaterial(materials[2]);
}

```

Finalmente, con el método “ChangeMaterial” se cambia el material del objeto al nuevo material proporcionado. Verifica que tanto el nuevo material como el componente “Renderer” del objeto no sean nulos antes de realizar el cambio.

```

Private void ChangeMaterial(Material newMaterial)
{

```

```

    Renderer renderer = GetComponent<Renderer>();

    if (newMaterial != null && renderer != null)
    {
        renderer.material = newMaterial;
    }
}

```

Una vez creado el código que controla la interactividad de los materiales del piso a través de los comandos de voz, en el inspector se asignan los materiales previamente creados.

## Figura 25

*Asignación de materiales de los pisos en el inspector.*



*Fuente.* Autoría propia, entorno Unity [Captura de Pantalla]

**Materiales Mesón.** Para realizar el cambio de materiales del mesón el usuario tiene 3 opciones que son: “mesón granito”, la cual carga en el modelo por defecto, “mesón acero” y “mesón mármol”. Para esto se crean los 3 materiales en Unity los cuales son asignados al inspector cuando en el script que controla la interactividad. El reconocimiento de voz será manejado a través de la API de reconocimiento de voz de Windows.

La lógica de programación utilizada para la asignación de los materiales del mesón es exactamente la misma que se utilizó en la creación de los materiales del piso, pero en “Start” se cambian las palabras clave.

```

Void Start()
{
    wordToAction = new Dictionary<string, Action>();
    wordToAction.Add(“meson granito”, D);
    wordToAction.Add(“mesón mármol”, E);
    wordToAction.Add(“mesón acero”, F);
    keywordRecognizer = new KeywordRecognizer(wordToAction.Keys.ToArray());
}

```

```

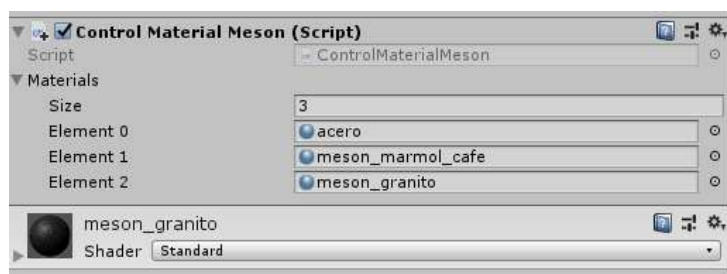
keywordRecognizer.OnPhraseRecognized += WordRecognized;
keywordRecognizer.Start();
}

```

Una vez creado el código que controla la interactividad de los materiales del mesón a través de los comandos de voz, en el inspector se asignan los materiales previamente creados.

## Figura 26

*Asignación de materiales del mesón en el inspector.*



*Fuente.* Autoría propia, entorno Unity [Captura de Pantalla]

## Luces

Existen 6 grupos de luces distribuidas en diferentes espacios (Imagen 27), las cuales son controladas por comandos de voz. Cuando el usuario pronuncia la palabra asociada a las luces que se encuentran en ese espacio las luces se apagan o encienden según el caso.

## Figura 27

*Distribución de luces en el apartamento.*



*Fuente.* Autoría propia, entorno Unity [Captura de Pantalla]

Cuando el usuario pronuncia una de las palabras claves asociadas al control de luces, por ejemplo, “apagar sala” las luces de la “sala” se apagan y cuando el usuario pronuncia las palabras “encender sala” las luces de la “sala” se encienden.

Para lograr lo anterior se creó un script en `c#` que controla el encendido y apagado de las luces cuando se pronuncia la palabra clave. Como son grupos se creó un script por cada grupo de luces. Para este caso se hará la explicación con el grupo de luces de la sala.

Lo primero es importar las bibliotecas necesarias para trabajar con comandos de voz.

```
Using System;
using System.Collections;
using System.Collections.Generic;
using System.Linq;
using UnityEngine;
using UnityEngine.Windows.Speech;
```

Se declara la clase “LucesSala” que hereda de “MonoBehaviour”. Se definen las variables públicas “lights”, que es una lista de luces que se asignarán desde el Inspector de Unity, y otras variables para el reconocimiento de voz.

```
Public List<Light> lights;

KeywordRecognizer keywordRecognizer;
Dictionary<string, Action> wordToAction;
```

En el método “Start”, se inicializa el diccionario “wordToAction” y se agregan dos comandos de voz (“encender sala” y “apagar sala”) asociados a los métodos “TurnOnLights” y “TurnOffLights”. Se configura el reconocedor de palabras clave “keywordRecognizer” con las palabras clave del diccionario y se inicializa.

```
Void Start()
{
    wordToAction = new Dictionary<string, Action>();
    wordToAction.Add(“encender sala”, TurnOnLights);
    wordToAction.Add(“apagar sala”, TurnOffLights);
    keywordRecognizer = new KeywordRecognizer(wordToAction.Keys.ToArray());
    keywordRecognizer.OnPhraseRecognized += WordRecognized;
    keywordRecognizer.Start(); } }
```

Se llama el método “WordRecognized” cuando una palabra clave es reconocida y se imprime la palabra reconocida en la consola y luego se invoca la acción asociada a esa palabra clave.

```
Private void WordRecognized(PhraseRecognizedEventArgs word)
{
    Debug.Log(word.text);
    wordToAction[word.text].Invoke();
}
```

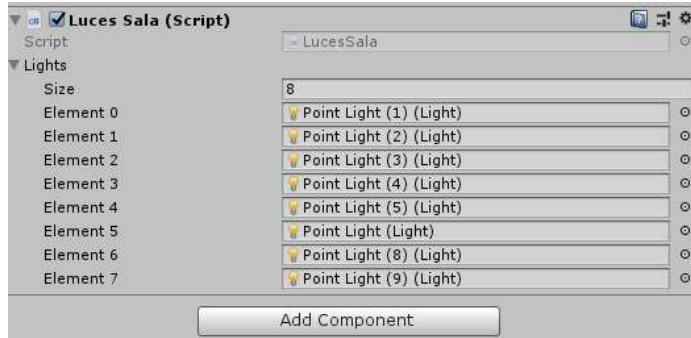
Se crean los métodos “TurnOnLights” y “TurnOffLights” que encienden y apagan las luces de la sala respectivamente. Estos métodos utilizan un bucle “foreach” para iterar sobre la lista de luces “lights” y ajustar su estado “enabled” según sea necesario.

```
Private void TurnOnLights()
{
    foreach (Light light in lights)
    {
        light.enabled = true;
    }
}
private void TurnOffLights()
{
    foreach (Light light in lights)
    {
        light.enabled = false;
    }
}
```

Una vez creado el código que controla la interactividad de las luces a través de los comandos de voz, en el inspector se asignan las que están ubicadas en la sala para que al pronunciar la palabra clave se apaguen o se enciendan según sea el caso.

## Figura 28

*Asignación de luces en el inspector.*



*Fuente.* Autoría propia, entorno Unity inspector [Captura de Pantalla]

### Reconocimiento de Voz:

El reconocimiento de voz en este código se lleva a cabo utilizando la clase `KeywordRecognizer` de la API de Windows en Unity. Para esto se debe acceder a las clases relacionadas con el reconocimiento de voz a través de:

```
using UnityEngine.Windows.Speech;
```

La clase “`KeywordRecognizer`” se utiliza para reconocer palabras clave o frases específicas pronunciadas por el usuario. Permite asociar acciones específicas con estas palabras clave, de modo que cuando una palabra clave se detecta, se ejecuta una determinada lógica en la aplicación.

### Control de Audio

El aplicativo inicia por defecto con el audio encendido o reproduciendo música de fondo, el cual se puede desactivar o apagar cuando el usuario pronuncia las palabras “apagar audio” y se puede encender nuevamente cuando el usuario pronuncia las palabras “encender audio.”

Para configurar el encendido y apagado del sonido de fondo, lo primero es importar las bibliotecas necesarias para trabajar con comandos de voz.

```

Using System;
using System.Collections;
using System.Collections.Generic;
using System.Linq;
using UnityEngine;
using UnityEngine.Windows.Speech;

```

Después dentro de la clase audio en la una lista creada se agregan las fuentes de audio en el inspector y una variable publica de tipo texto “audioStatusText” que se utiliza para mostrar el estado actual del audio. Además, se utiliza un objeto “KeywordRecognizer” para reconocer comandos de voz, un diccionario “wordToAction” que mapea palabras clave a acciones (métodos) que se deben ejecutar y un indicador “isAudioOn” que muestra el estado actual del audio iniciando con “true” para indicar que el audio inicia encendido.

```

Public List<AudioSource> audioSources;
public Text audioStatusText;
KeywordRecognizer keywordRecognizer;
Dictionary<string, Action> wordToAction;
bool isAudioOn = true;

```

En el método “Start” se asegura que el “tutorial” esté cerrado al inicio, se inicializa el diccionario “wordToAction”, se configura el reconocedor de palabras clave “KeywordRecognizer” y se llama “UpdateAudioStatusText” para actualizar el texto de estado del audio.

```

Void Start()
{
    isMenuOpen = false;
    menuCanvas.enabled = false;

    wordToAction = new Dictionary<string, Action>();
    wordToAction.Add("encender audio", TurnOnAudio);
    wordToAction.Add("apagar audio", TurnOffAudio);
    wordToAction.Add("tutorial", OpenMenu);
    wordToAction.Add("cerrar tutorial", CloseMenu);

    keywordRecognizer = new KeywordRecognizer(wordToAction.Keys.ToArray());
    keywordRecognizer.OnPhraseRecognized += WordRecognized;
    keywordRecognizer.Start();
}

```

Después se llama el método “WordRecognized” cuando una palabra es reconocida, registrando la palabra en la consola de Unity y ejecutando la acción.

```

Private void WordRecognized(PhraseRecognizedEventArgs e)
{
    wordToAction[e.Text].Invoke();
}

```

Posteriormente se agregan los métodos para el encendido y el apagado del audio, mediante la activación y desactivación de los componentes AudioSource en la lista “audioSource” y se llama a “UpdateAudioStatusText” para reflejar el cambio en el texto.

```

Private void ToggleAudio()
{
    isAudioOn = !isAudioOn;

    foreach (AudioSource audioSource in audioSources)
    {
        audioSource.enabled = isAudioOn;
    }
    UpdateAudioStatusText();
}

```

Y finalmente se actualiza el texto de estado “AudioStatusText” para indicar si el audio está encendido o apagado.

```

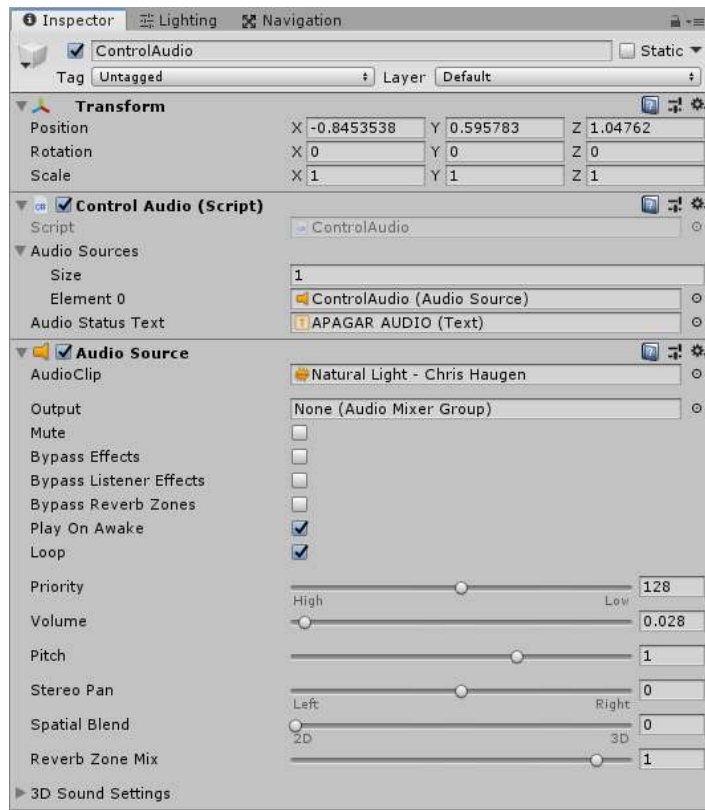
Private void UpdateAudioStatusText()
{
    if (audioStatusText != null)
    {
        audioStatusText.text = isAudioOn ? "Di, APAGAR AUDIO" : "Di, ENCENDER
AUDIO";
    }
}

```

Una vez creado el código que controla el audio, en el inspector se asigna el audio (imagen29).

**Figura 29**

*Asignación audio de fondo en el inspector.*



*Fuente.* Autoría propia, entorno Unity [Captura de Pantalla]

## **Interfaz UI (User Interface)**

Quando se inicia el aplicativo se muestra una pantalla de bienvenida con un video de fondo del aplicativo de una cámara rotando, un texto de bienvenida y un contador que al llegar a cero pasa a la pantalla principal.

## Figura 30

### *Pantalla de Bienvenida*



*Fuente.* Autoría propia, entorno Unity [Captura de Pantalla]

Quando se inicia el aplicativo se muestra la vista en planta de un apartamento de 3 habitaciones con varios textos y algunos iconos los cuales se describen a continuación.

(Imagen31).

## Figura 31

### *Pantalla Principal*



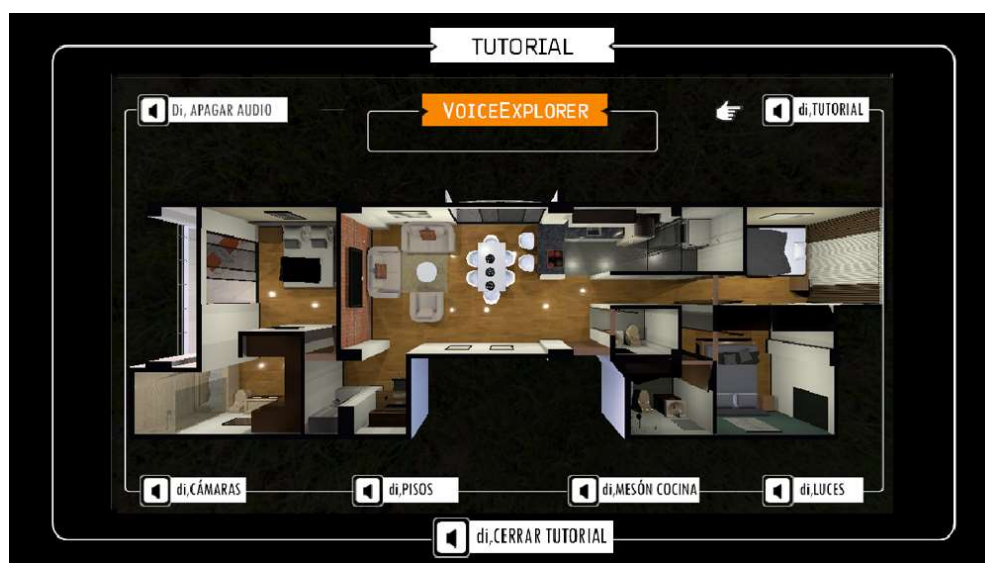
*Fuente.* Autoría propia, entorno Unity [Captura de Pantalla]

En la parte superior-central se encuentra el título del aplicativo. En la parte superior-derecha se encuentra un ícono de una mano que señala otro texto que dice “di,TUTORIAL” con un ícono de audio.

Se pretende que el usuario antes de empezar a interactuar con el aplicativo le dé un vistazo al tutorial para saber cómo funciona el aplicativo. Cuando el usuario pronuncia la palabra “tutorial” se abre una ventana de video con un tutorial que muestra el funcionamiento del aplicativo (Imagen 32). En la parte de debajo de la ventana se encuentran las palabras “cerrar tutorial” para cerrar la ventana y continuar en la pantalla principal.

### Figura 32

#### *Tutorial*



*Fuente.* Autoría propia, entorno Unity [Captura de Pantalla]

Se creó un script para controlar la apertura y cierre del “tutorial” por medio de comandos de voz, además de la detección y reproducción del video dentro del menú y la manipulación del audio.

Dentro de los aspectos más relevantes del script están los campos públicos

Donde se agrega la referencia del objeto “Canvas” y las referencias de los componentes “VideoPlayer” y “AudioSource”

```
public Canvas menuCanvas;
public VideoPlayer videoPlayer;
public AudioSource audioSource;
```

En el “Start” se asegura que el menú esté cerrado al inicio

```
isMenuOpen = false;
menuCanvas.enabled = false;
```

En el método “OpenMenú” se abre el menú si no está abierto, se habilita el canvas, se inicia la reproducción del video con “videoPlayer.Play()” y se detiene la reproducción del audio de fondo del aplicativo “audioSource.Stop”

```
private void OpenMenu()
{
    if (!isMenuOpen)
    {
        menuCanvas.enabled = true;
        isMenuOpen = true;
        videoPlayer.Play();
        audioSource.Stop();
    }
}
```

En el método “CloseMenu” se cierra el menú si está abierto, se deshabilita el canvas, se detiene la reproducción del video con “videoPlayer.Stop()” y se inicia la reproducción del audio de fondo del aplicativo “audioSource.Play”

```
private void CloseMenu()
{
    if (isMenuOpen)
    {
        menuCanvas.enabled = false;
        isMenuOpen = false;
        videoPlayer.Stop();
        audioSource.Play();
    }
}
```

En la parte superior-izquierda se encuentran los textos “di, APAGAR AUDIO”. Aquí el usuario tiene la opción de apagar el audio o encender el audio como se explica en el numeral 7.7 control de audio.

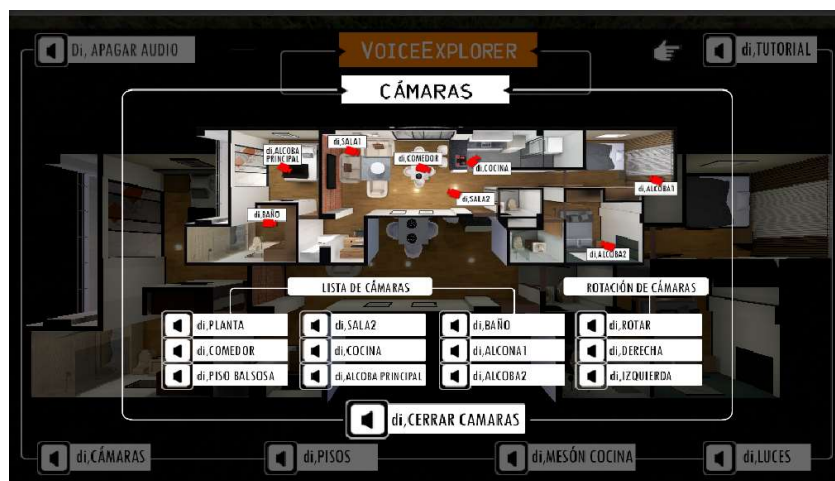
En la parte inferior izquierda se encuentra el texto “Elementos Interactivos” seguido de una flecha, donde se encuentran 4 comandos que abren 4 menús diferentes correspondientes a los elementos interactivos.

### *Menú Cámaras*

Al pronunciar la palabra “cámaras”, se abre el menú cámaras (imagen 33) en el cual se muestra una vista en planta con la ubicación de las cámaras disponibles y el comando de voz asociado.

**Figura 33**

### *Menú Cámaras*



*Fuente.* Autoría propia, entorno Unity [Captura de Pantalla]

Para que el menú de cámaras se abra al pronunciar la palabra “cámaras” se creó un código en c# que es añadido en el inspector a un “canvas” que contiene los elementos del menú de

cámaras como la imagen de la vista aérea del apartamento y los textos e iconos propios de este menú.

Este código controla la visibilidad del menú “cámaras” cuando el comando de voz es pronunciado.

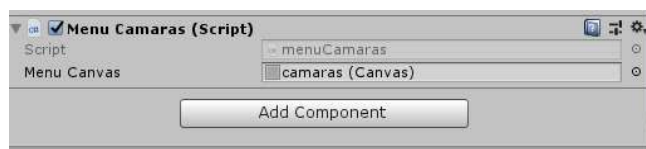
Como en los anteriores códigos explicados lo primero es añadir las bibliotecas necesarias para trabajar con las funciones específicas y el reconocimiento de voz de Windows.

Se crea el “menú canvas” para que en el inspector pueda ser añadido el “canvas” de cámaras.

```
Public Canvas menuCanvas;
```

### Figura 34

*Asignación de menú cámaras en el inspector*



*Fuente.* Autoría propia, entorno Unity [Captura de Pantalla]

Se crea un objeto “KeywordRecognizer” utilizado para reconocer comandos de voz, un diccionario “wordToAction” que mapea palabras clave a acciones (métodos) que se deben ejecutar y un indicador “isMenuOpen” booleano que rastrea si el menú está abierto o cerrado.

```
KeywordRecognizer keywordRecognizer;
Dictionary<string, Action> wordToAction;
bool isMenuOpen = false;
```

En el método “start” se asegura que el menú está cerrado al iniciar el aplicativo, se inicializa el diccionario “wordToAction” y se configura el reconocedor de palabras clave.

```
Void Start()
{
    isMenuOpen = false;
    menuCanvas.enabled = false;
```

```

wordToAction = new Dictionary<string, Action>();
wordToAction.Add("cámaras", OpenMenu);
wordToAction.Add("cerrar cámaras", CloseMenu);

keywordRecognizer = new KeywordRecognizer(wordToAction.Keys.ToArray());
keywordRecognizer.OnPhraseRecognized += WordRecognized;
keywordRecognizer.Start();
}

```

Después se llama el método “wordRecognized” cuando la palabra clave es reconocida y se invoca la acción asociada a dicha palabra.

```

Private void WordRecognized(PhraseRecognizedEventArgs e)
{
    wordToAction[e.Text].Invoke();
}

```

Finalmente, con el método “OpenMenu” si el menú no está abierto se activa la visibilidad del “menuCanvas” y actualiza el estado de “isMenuOpen” a verdadero.

```

Private void OpenMenu()
{
    if (!isMenuOpen)
    {
        // Lógica para mostrar el menú
        menuCanvas.enabled = true;
        isMenuOpen = true;
    }
}

```

Si el menú está abierto, el método “CloseMenu” desactiva la visibilidad del “menuCanvas” y actualiza el estado de “isMenuOpen” a falso. En la parte inferior del menú se encuentran los comandos de voz para cerrar el menú. “CERRAR CAMARAS”.

### ***Menú Pisos***

Al pronunciar la palabra “pisos”, se abre el menú pisos (imagen 34) en el cual se muestran 3 materiales con sus respectivos comandos de voz y en la parte de abajo se encuentra la instrucción “cerrar pisos” para cerrar este menú.

La configuración y programación funciona de la misma manera que como se hizo el menú de cámaras controlando la visibilidad del “canvas” En la parte inferior del menú se encuentran los comandos de voz para cerrar el menú. “CERRAR PISOS”.

### Figura 35

#### *Menú Pisos*



*Fuente.* Autoría propia, entorno Unity [Captura de Pantalla]

#### *Menú Mesón*

Al pronunciar las palabras “menú mesón”, se abre el menú con los materiales del mesón (imagen 35) en el cual se muestran 3 materiales con sus respectivos comandos de voz y en la parte de abajo se encuentra la instrucción “CERRAR MESÓN” para cerrar este menú.

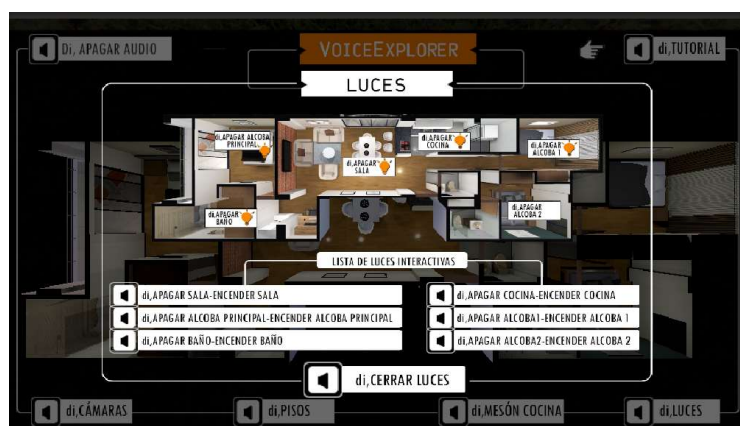
La configuración y programación funciona de la misma manera que como se hizo el menú de cámaras controlando la visibilidad del “canvas” En la parte inferior del menú se encuentran los comandos de voz para cerrar el menú. “CERRAR MESÓN”.

**Figura 36***Menú Mesón*

*Fuente.* Autoría propia, entorno Unity [Captura de Pantalla]

*Menú Luces*

Al pronunciar la palabra “luces”, se abre el menú de luces (imagen 37) en el cual se muestra una vista en planta con la ubicación de las luces distribuidas en los diferentes espacios y el comando de voz asociado.

**Figura 37***Menú Luces*

*Fuente.* Autoría propia, entorno Unity [Captura de Pantalla]

La configuración y programación funciona de la misma manera que como se hizo el menú de cámaras controlando la visibilidad del “canvas” En la parte inferior del menú se encuentran los comandos de voz para cerrar el menú. “CERRAR LUCES”.

## **Pruebas**

En primer lugar, se realizaron pruebas funcionales para determinar si el aplicativo cumple con el propósito para el cual fue creado y si el aplicativo funciona como se esperaba, y en segundo lugar se realizaron pruebas de usabilidad, basadas en la experiencia de usuario (UX), con las cuales con el objetivo de percibir la reacción de los usuarios hacia el aplicativo teniendo en cuenta varios criterios de evaluación.

### ***Pruebas Funcionales***

Con estas pruebas se busca determinar que el software funciona correctamente de acuerdo con los objetivos planteados inicialmente, así como la verificación de que no falten aspectos importantes establecidos dentro de los requerimientos funcionales en el desarrollo del aplicativo multimedial.

**Tabla 8***Pruebas funcionales del aplicativo desarrollado*

Identificador Prueba	Nombre de la Prueba	Descripción de la Prueba	Comportamiento Esperado	Comportamiento Obtenido	Observaciones
P.F.A.01	Reconocimiento de voz	Ejecutar el comando de voz "tutorial", para que se abra la pantalla y/o escena TUTORIAL y se ejecute el video.	Se debe abrir la pantalla y/o escena TUTORIAL y se debe ejecutar automáticamente el video	Se abre la pantalla y/o escena TUTORIAL y se ejecuta automáticamente el video.	Se documenta prueba realizada aun comando de voz, pero todos funcionan de la misma manera.
P.F.A.02	Desplazamiento de cámaras	Ejecutar el comando de voz de cámaras "sala1" para que la cámara principal se desplace a la posición de la cámara "sala1"	La cámara se debe desplazar a la posición de la cámara "sala1", realizando un movimiento de rotación uniforme hasta llegar al objetivo.	La cámara se desplaza a la posición de la cámara "sala1", haciendo un movimiento de rotación uniforme hasta llegar al objetivo.	Se documenta prueba realizada a una cámara, pero todas de la misma manera
P.F.A.03	Cambio de materiales del piso	Ejecutar el comando de voz de materiales del piso "piso parquet" para que el material del piso cambie al nuevo material "piso parquet"	El material actual del piso debe cambiar al nuevo seleccionado "piso parquet"	El material actual del piso se cambia al nuevo seleccionado "piso parquet".	Se documenta prueba realizada a un comando de pisos, pero funciona con todos de la misma manera.
P.F.A.03	Cambio de materiales del mesón de la cocina	Ejecutar el comando de voz de materiales del mesón de la cocina "mesón mármol" para el material del mesón cambie al nuevo material "mesón mármol"	El material actual del mesón de la cocina debe cambiar al nuevo seleccionado "mesón mármol"	El material actual del mesón de la cocina se cambia al nuevo seleccionado "mesón mármol".	Se documenta prueba realizada a un comando del mesón de la cocina, pero funciona con todos de la misma manera.
P.F.A.03	Encendido y apagado de luces	Ejecutar el comando de voz "apagar sala" para que se apaguen las luces de la sala	Las luces de la sala se deben apagar al pronunciar el comando de voz "apagar sala"	Las luces de la sala se deben apagar al pronunciar el comando de voz "apagar sala"	Se documenta prueba realizada aun comando de voz de las luces, pero funciona con todos de la misma manera.
P.F.A.03	Encendido y apagado de audio	Ejecutar el comando de voz "apagar audio" para que el audio de fondo quede silenciado. Para prueba de encendido se ejecuta el comando de voz "encender audio" para que el audio se reproduzca nuevamente.	El sonido de fondo debe quedar silenciado al ejecutar el comando "apagar audio". De la misma manera al ejecutar el comando de voz "encender audio" el audio de fondo se debe reproducir nuevamente.	El sonido de fondo queda silenciado al ejecutar el comando "apagar audio". De la misma manera al ejecutar el comando de voz "encender audio" el audio de fondo se reproduce nuevamente.	

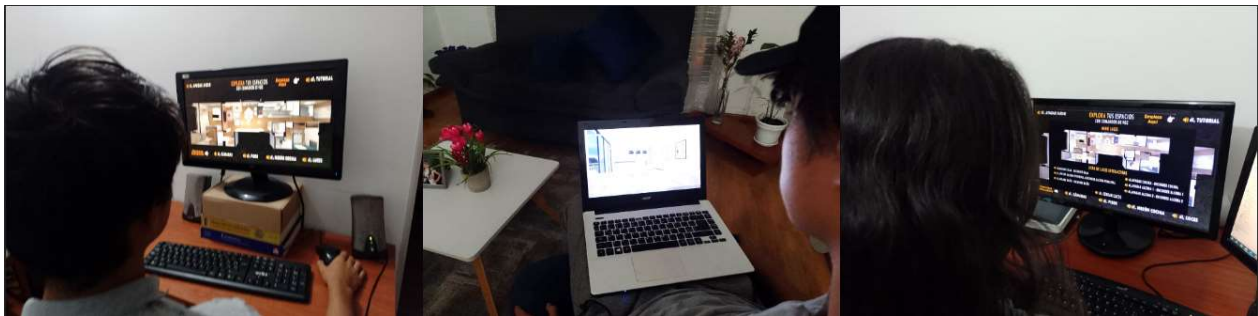
*Fuente.* Autoría propia.

### ***Pruebas de Observación***

Para lograr una mejor experiencia de usuario (UX) se realizaron unas pruebas preliminares donde se observa a algunos usuarios manipulando el aplicativo, para determinar aspectos importantes en cuanto al primer impacto, así como la reacción frente a la navegabilidad y la interacción con el aplicativo.

### **Figura 38**

*Usuarios Manipulando el Aplicativo*



*Fuente.* Foto autoría propia

Se observa que la mayoría de los usuarios abren primero el tutorial, y de acuerdo con la información dada, continúan con la manipulación del aplicativo.

Algunos usuarios intentaron manipular el aplicativo sin entrar al tutorial y tuvieron éxito, y otros tuvieron que devolverse al tutorial para mirar las instrucciones.

Algunos usuarios tuvieron inconvenientes con algunos comandos de voz, pero se detectó que son errores de programación que se deben corregir.

Se pregunta a los usuarios si la palabra “DI” es la adecuada para pedir al usuario que pronuncie un comando de voz, a lo que la gran mayoría de los usuarios respondió positivamente, ya que dicen que es la palabra más corta.

Se observa que algunos usuarios tienen inconvenientes con la interfaz ya que dicen que no se ve lo suficientemente clara en determinados escenarios, por lo que se toma nota para realizar modificaciones posteriormente.

Algunos usuarios no entendieron inicialmente que es un aplicativo controlado por comando de voz y trataron de manipularlo digitalmente. Algunos dicen que sería bueno tener una pantalla de inicio que explique que es un aplicativo controlado con comandos de voz y que es necesario conectar el micrófono.

Se pudo observar que una vez que los usuarios entienden el fin del aplicativo se conectan con facilidad, además de que a la mayoría de los usuarios les parece un aplicativo llamativo y novedoso.

### ***Pruebas de Usabilidad***

Para realizar el plan de pruebas se evalúa la usabilidad del aplicativo en base a la experiencia del usuario (UX) con el objetivo de emitir juicios de valor cualitativos y cuantitativos y así determinar la reacción de los usuarios a las diferentes pantallas y/o escenas del aplicativo. Teniendo en cuenta lo anterior se diseñó un formato con varias preguntas orientadas a entender la percepción del usuario frente al aplicativo y se evaluaron los criterios de contenido, navegabilidad, las gráficas, la accesibilidad y la utilidad además de recoger las observaciones de los usuarios con el fin de realizar la documentación de los fallos y plasmarlos en un plan de mejora y así lograr una mejor experiencia de usuario (UX). La prueba se realizó a 15 usuarios con el objetivo de evaluar los aspectos antes mencionados y captar la percepción hacia el aplicativo.

## Criterios de Usabilidad Para Evaluar.

**Criterio de Contenido.** Impacto de los usuarios frente al contenido

**Criterio de Navegabilidad:** Experiencia de usuario frente la navegabilidad del aplicativo.

**Graficas.** Se evalúa la apariencia y distribución de los elementos gráficos en las diferentes escenas y/o pantallas.

**Criterio de Accesibilidad:** Se evalúa el nivel de dificultad para acceder al aplicativo.

**Utilidad:** Se evalúa si el aplicativo es útil y cumple con los objetivos planteados para el cual fue creado.

## Preguntas Según Criterio de Evaluación.

**Tabla 9**

*Diseño de Preguntas Evaluativas por Criterio*

	Formulario de Evaluación
Criterios de Navegabilidad	¿Los contenidos presentados en el aplicativo son pertinentes de acuerdo con los objetivos del aplicativo? ¿Los contenidos son claros y bien estructurados? ¿Las Instrucciones para el desarrollo del aplicativo son suficientemente y claras? ¿La distribución del contenido es la adecuada?
Criterios de Navegabilidad	¿La forma como esta implementada de la navegabilidad del aplicativo es la adecuada? ¿Los comandos de voz funcionan correctamente?
Gráficas	¿Las gráficas son llamativas? ¿Las gráficas cargan correctamente? ¿La combinación de colores es la adecuada? ¿Los tipos de fuente y tamaños son los adecuados?
Criterio de Accesibilidad	¿Se puede acceder a los contenidos fácilmente? ¿La información proporcionada es suficiente para navegar por el aplicativo?
Utilidad	¿Después de ejecutar el aplicativo le parece que se cumple con el objetivo para el cual fue creado? ¿Los contenidos del aplicativo son útiles para la visualización de proyectos inmobiliarios por parte de personas con discapacidades en sus extremidades superiores? ¿La forma secuencial como está programado el aplicativo es adecuada? ¿La forma como se muestran los contenidos es la adecuada?
Observaciones	

*Fuente.* Autoría propia

## Diseño de Formulario de Evaluación.

**Figura 39**

*Estructura del Formulario*

The image shows a Google Forms interface for an evaluation. The title is "Pruebas del Aplicativo 'ExploreSpaces'". Below the title is a subtitle: "Evaluación de criterios de usabilidad del aplicativo interactivo controlado por comandos de voz 'ExploreSpaces'". The form consists of several sections:

- Personal Information:** Two short-answer questions: "Escriba su Nombre \*" and "Profesión: \*".
- Content Criteria:** A section titled "Criterios de Contenido" with the subtitle "Impacto de los usuarios frente al contenido".
- Rating Scale 1:** A question "¿Le parece pertinente la selección de contenidos en el aplicativo? \*" with a 5-point Likert scale. The scale is labeled "Totalmente en Desacuerdo" on the left and "Totalmente de Acuerdo" on the right.
- Rating Scale 2:** A question "¿Los contenidos son claros y bien estructurados? \*" with a 5-point Likert scale, also labeled "Totalmente en Desacuerdo" and "Totalmente de Acuerdo".

*Fuente.* Autoría propia, entorno Google Forms [Captura de Pantalla]

## Publicación de Formulario de Evaluación y Aplicativo Preliminar.

Formulario: <https://forms.gle/T1Ldiomyx3dLzW8B8>

**Aplicativa Versión 2:**

<https://drive.google.com/drive/folders/180HNdz3V7SHfg7RDTzAZzM4OtoYQNOYz?usp=sharing>

**Realización de Pruebas de Usabilidad.** Se envió el aplicativo y el formulario de pruebas el cual fue diligenciado por 15 usuarios con diferentes perfiles, contestando las preguntas diseñadas de acuerdo con cada criterio de evaluación y recogiendo las observaciones de los usuarios para con esta información realizar el análisis y la documentación de resultados.

**Figura 40**  
*Pruebas de Usabilidad*



Fuente. Autoría propia, Google Forms [Captura de Pantalla]

## Documentación de Resultados

Una vez realizadas las pruebas se continuó con la documentación de resultados. De acuerdo con las tabulaciones se hizo un análisis de los puntos tanto fuertes como débiles del aplicativo.

**Tabla 10**

*Tabulación de Resultados Pruebas de Usabilidad*

Fecha	Escriba su Nombre	Profesión:	¿Le parece pertinente la selección de contenidos en el aplicativo?	¿Los contenidos son claros y bien estructurados?	¿Las instrucciones para el desarrollo del aplicativo son suficientemente y claras?	¿La distribución del contenido es la adecuada?	¿La forma como esta implementada la navegabilidad del aplicativo es la adecuada?	¿Se puede identificar fácilmente la forma de navegar del aplicativo?	¿Puede regresar a la pantalla principal fácilmente?	¿Los comandos de voz funcionan correctamente?	¿La pantalla llamativa los elementos gráficos del aplicativo?	¿Las gráficas cargan correctamente?	¿La combinación de colores es la adecuada?	¿Los tipos de fuente y tamaños son los adecuados?	¿Le parece que el aplicativo es equilibrado gráficamente?	¿Se puede acceder a los contenidos fácilmente?	¿Encuentra el aplicativo de fácil acceso para personas con algún tipo de discapacidad que no les permite acceder a contenidos?	¿La información proporcionada es suficiente para navegar por el aplicativo?	¿Después de ejecutar el aplicativo la navegación cumple con el objetivo para el cual fue creado?	¿Los contenidos del aplicativo son útiles para la visualización de proyectos inmobiliarios por parte de personas con algún tipo de discapacidad que no les permite acceder?	¿La forma secuencial como esta programado el aplicativo es adecuada?	Observaciones
Criterios de Evaluación			Criterios de Contenido				Criterios de Navegabilidad				Gráficas				Criterios de Accesibilidad				Criterios de Utilidad			
3/02/2024	15:48:15	Carlos Erendo Barea	4	3	3	3	4	3	2	3	2	3	2	2	3	3	4	4	5	4	3	Los textos de los menús no se ven en determinados espacios.
3/02/2024	15:57:22	Adriana Gutierrez	4	3	5	5	3	3	4	4	2	3	3	2	3	3	4	4	4	3	4	
3/02/2024	18:23:17	Maria Paula Morroy	4	4	4	3	4	5	4	4	3	3	3	2	4	4	5	4	5	5	4	Los comandos de rotación no funcionan
3/5/2024	18:27:03	Jairo Valdez	4	4	5	4	4	4	3	4	3	4	3	1	2	4	4	4	4	5	4	Le falta un poco de diseño a la interfaz
3/6/2024	16:30:22	Juan Pablo Morroy	3	3	4	3	4	4	2	4	4	4	4	3	2	4	5	4	4	4	3	El texto del menú de audio debería cambiar de estado cuando se apaga o se enciende el audio.
3/6/2024	18:34:08	Adriana Salazar	5	5	4	4	3	4	2	4	4	3	3	4	3	4	4	4	4	5	4	
3/8/2024	10:40:55	Miriam Morroy	4	4	4	4	4	4	3	4	3	4	3	2	3	4	4	3	3	5	3	Se puede mejorar la interfaz de las pantallas, se ve un poco plano.
3/9/2024	22:46:50	Valentina Rojas	3	4	4	4	4	4	5	3	5	4	3	3	3	4	5	4	4	5	4	Los textos se pierden cuando el fondo es plano, sería mejor ponerles un recuadro para que se roten más. La fuente del título debe ser más llamativa.
3/12/2024	20:50:42	Andres Cortés	4	4	4	4	4	4	5	4	3	5	4	4	4	4	4	4	4	4	4	No me funcionan unos comandos o voz
3/12/2024	18:54:15	Fior Morroy	5	5	5	5	4	5	3	4	4	5	4	4	4	4	5	5	5	5	4	
3/15/2024	9:01:36	Fredy Quiroga	4	5	5	3	3	4	4	4	3	3	2	2	2	4	3	3	4	4	3	Cuando se abre el tutorial suena audífono el audio de fondo.
3/16/2024	23:04:59	Guacima Moreno	5	5	5	5	5	5	4	5	4	4	4	4	4	5	5	5	5	5	5	Buen Trabajo
3/20/2024	19:08:30	Alfonso Birto	4	5	4	4	3	3	2	3	3	4	3	3	3	4	4	3	4	5	4	Existen algunos errores en los menús y algunos comandos no funcionan como los de rotación.
3/22/2024	15:14:01	Jorge Mendez	4	3	3	3	3	4	3	4	3	3	3	3	3	4	4	4	4	4	4	Los menús quedan abiertos cuando se abre otro menú. Cada vez que se abre un menú se deberían cerrar los demás.
3/22/2024	11:18:19	Andrea Roa	4	4	4	5	5	5	4	3	4	4	4	2	4	4	5	5	5	4	4	
PROMEDIO POR PREGUNTA			4,87	4,87	4,28	3,93	3,86	4,26	3,13	3,87	3,33	3,80	3,27	2,87	3,15	3,95	4,33	4,00	4,27	4,47	3,80	
PROMEDIO POR CRITERIO			4,07				3,75				3,20				4,09				4,16			

Fuente. Autoría propia.

**Tabla 11**

*Escala de Medición*

COLOR	VALOR	ESCALA DE MEDICIÓN
	5	TOTALMENTE DE ACUERDO
	4	
	3	
	2	
	1	TOTALMENTE EN DESACUERDO

Fuente. Autoría propia

## **Análisis de Resultados**

De acuerdo con los resultados obtenidos se pudieron identificar varios factores:

En términos generales el aplicativo tuvo un buen nivel de aceptación ya que el promedio general del aplicativo de acuerdo los resultados son de 3.86/5.

Observando los promedios por criterio de evaluación vemos que los criterios de contenido, accesibilidad y utilidad tuvieron un promedio por encima de 4 y los criterios de navegabilidad y gráficas tuvieron un promedio superior a 3 pero inferior a 4. De lo anterior se puede deducir que los criterios de evaluación que pueden presentar más inconvenientes son los de navegabilidad y gráficas.

Si analizamos el criterio de gráficas los usuarios tuvieron inconvenientes a la hora de evaluar los contenidos gráficos, presentando el nivel más bajo en lo concerniente a las fuentes utilizadas y los tamaños de fuentes, ya que en la pregunta relacionada con este aspecto el promedio fue de 2.67, que constituye el más bajo en las pruebas. Además de esto se pudo apreciar que los usuarios tienen problemas con los elementos gráficos del aplicativo.

También es importante mencionar que según los promedios obtenidos puede existir elementos que estén fallando en cuanto a navegabilidad del aplicativo ya que hay unos promedios por debajo de 4 que indican que se puede está presentando algún tipo de dificultad en cuanto al funcionamiento de algunos comandos de voz o de respuesta a los mismos.

En cuanto a los criterios de contenido se deduce que se deben revisar los aspectos concernientes a la distribución ya que según la evaluación presenta el punto más bajo fue de 3.93, que si bien no es un promedio muy bajo es importante revisar que puede estar fallando.

En los criterios de accesibilidad se concluye que si bien la evaluación fue satisfactoria en términos generales hay que revisar la facilidad de acceso a los contenidos para mejorar la experiencia de usuario.

El criterio de utilidad fue el mejor evaluado con un promedio de 4.18, pero sería importante revisar las observaciones para verificar si hay algún aspecto que esté fallando en cuanto a la forma secuencial como está diseñado el aplicativo.

Si contrastamos estos valores con las observaciones presentadas por los usuarios podemos sacar una conclusión de fallos y elementos que deben ser corregidos en el aplicativo con el fin de realizar un plan de mejoras que corrija todos estos errores y se logre obtener un aplicativo más optimizado.

### **Documentación de Fallos**

Con los resultados obtenidos se identificaron los fallos errores y puntos débiles del aplicativo los cuales necesitan ser corregidos.

De acuerdo con análisis realizado y contrastándolo con las observaciones de los usuarios se lograron detectar los siguientes fallos en el aplicativo.

Existen algunos errores en el menú de “cámaras” ya que los comandos de rotación no coinciden en el menú.

En el menú tutorial tampoco se encuentra actualizado el menú de cámaras y no coinciden los comandos de rotación con los comandos pronunciados en el video.

En el menú tutorial, cuando esté encendido el audio de fondo, debería quedar apagado ya que cuando inicia el audio del tutorial se escucha duplicado.

En el menú principal cuando el audio se encuentre apagado, el texto “DI, APAGAR AUDIO” debería cambiar a “DI, ENCENDER AUDIO” para que el usuario entienda que el audio esta apagado y tiene la opción de encenderlo, lo mismo cuando el audio este encendido la opción que se debe mostrar es “DI, APAGAR AUDIO”.

Los menús se sobreponen unos con otros al pronunciarlos de manera continua. El aplicativo debería cerrar el menú que se encuentra activo al abrir otro menú.

En la interfaz se confunde el título con los otros textos, debería haber un factor diferencial del título con respecto a los otros textos.

En Interfaz los textos de los menús en la pantalla principal no se ven con claridad en determinadas escenas o pantallas.

Es recomendable poner una pantalla de bienvenida, en vez de que entre directamente al aplicativo.

Es recomendable revisar el diseño de la interfaz del aplicativo para corregir aquellos aspectos que presentan inconvenientes y lograr un mejor impacto visual

## Plan de Mejora

**Tabla 12**

### *Plan de Mejora*

Criterio	Problema Detectado	Solución	Responsable	Fecha de Entrega	Verificación	Responsable	Fecha Verificación
Contenido	El aplicativo no responde a los comandos de voz de rotación de cámaras	Cambiar los comandos rotar cámaras por rotar, rotar a la derecha por derecha y rotar a la izquierda por izquierda.	Jorge Monroy	14/02/2024	Se corre el aplicativo y se verifica que el problema fue corregido	Javier Reyna	15/02/2024
Contenido	No se encuentra actualizado el menú de cámaras dentro del tutorial, con los comandos de voz de rotación	Corregir el tutorial y reemplazar el archivo de video.	Jorge Monroy	14/02/2024	Se corre el aplicativo y se verifica que el problema fue corregido	Javier Reyna	15/02/2024
Contenido	En el menú tutorial, cuando esté encendido el audio de fondo, debería quedar apagado ya que cuando inicia el audio del tutorial se escucha duplicado.	Corregir el script del menú tutorial para que cuando esté activo, se desactive	Jorge Monroy	14/02/2024	Se corre el aplicativo y se verifica que el problema fue corregido	Javier Reyna	15/02/2024
Gráficos y Contenido	- En el menú principal cuando el audio se encuentre apagado, el texto "DI, APAGAR AUDIO" debería cambiar a "DI, ENCENDER AUDIO" para que el usuario entienda que el audio esta apagado y tiene la opción de encenderlo, lo mismo cuando el audio este encendido la opción que se mostrará será "DI, APAGAR AUDIO".	Actualizar el Script "Control audio" creando un método que actualice el estado de un texto dinámico en la pantalla principal.	Jorge Monroy	14/02/2024	Se corre el aplicativo y se verifica que el problema fue corregido	Javier Reyna	15/02/2024
Gráficos	- En la interfaz se confunde el título con los otros textos, debería haber un factor diferencial del título con respecto a los otros textos.	Corregir el diseño de la interfaz de la pantalla principal de manera que se entienda perfectamente que es el título del aplicativo.	Jorge Monroy	14/02/2024	Se corre el aplicativo y se verifica que el problema fue corregido	Javier Reyna	15/02/2024
Gráficos	- En Interfaz los textos de los menús en la pantalla principal no se ven con claridad en determinadas escenas o pantallas.	Corregir la interfaz de la pantalla principal agregando un fondo a las letras de los menús para que se puedan apreciar con claridad sin importar en que escena se encuentre.	Jorge Monroy	14/02/2024	Se corre el aplicativo y se verifica que el problema fue corregido	Javier Reyna	15/02/2024

Gráficos y contenido	- Sería recomendable poner una pantalla de bienvenida, en vez de que entre directamente al aplicativo.	Diseñar una nueva escena, donde se creará un cava que mostrará un video de fondo, un panel oscuro con un texto de bienvenida, además de un contador de 5 hasta cero para que automáticamente pase a la siguiente escena.	Jorge Monroy	14/02/2024	Se corre el aplicativo y se verifica que el problema fue corregido	Javier Reyna	15/02/2024
----------------------	--	--	--------------	------------	--	--------------	------------

*Fuente.* Autoría propia

## Pruebas de Sanidad (Sanity Testing)

Una vez realizadas las correcciones planteadas en el plan de mejora de los errores detectados, se procede a verificar que los cambios realizados quedaron de acuerdo con lo esperado y el aplicativo funciona correctamente, por medio de la realización de pruebas de sanidad (Sanity Testing).

**Tabla 13**

### *Pruebas de Sanidad (Sanity Testing)*

Identificador Prueba	Nombre de la Prueba	Descripción de la Prueba	Comportamiento Esperado	Comportamiento Obtenido	Observaciones
P.S.A. 01	Reconocimiento de voz	Ejecutar el comando de rotación de cámaras "rotar", para verificar que esté funcionando correctamente	Al decir el comando "rotar" la cámara se debe rotar 360 grados a la derecha.	Al decir el comando "rotar" la cámara rota 360 grados a la derecha.	Se documenta prueba realizada aun comando de voz, pero funciona correctamente con los comandos "derecha" e "izquierda".
P.S.A. 02	Verificar que la pantalla y/o escena se encuentra actualizada con los comandos de rotación.	Ejecutar el comando de voz de voz "tutorial" para para verificar que los comandos de rotación están actualizados	Se debe observar que los comandos de rotación están actualizados dentro del video "tutorial"	Se observa que los comandos de rotación están actualizados dentro del video "tutorial"	
P.S.A. 03	Verificar que el audio se apague cuando se ejecuta el comando de voz "Tutorial".	Decir el comando de voz "tutorial" para que se abra el menú "tutorial" se apague el audio del aplicativo.	Al decir el comando de voz "tutorial" se debe apagar el audio de fondo del aplicativo.	Al decir el comando de voz "tutorial" se apaga el audio de fondo del aplicativo.	El audio ya no se duplica al ejecutar el comando "tutorial"
P.S.A. 04	Cambio de texto a "DI, APAGAR AUDIO" Y "DI,	Ejecutar los comandos "apagar audio" y "encender audio" para que el	Al decir "apagar audio" el texto debe cambiar a "DI, ENECENDER AUDIO" y al decir "encender audio" el	Al decir "apagar audio" el texto cambia a "DI, ENECENDER AUDIO" y al	

	ENCENDER AUDIO"	texto cambie según sea al caso.	texto debería cambiar a "DI, APAGAR AUDIO"	decir "encender audio" el texto cambia a "DI, APAGAR AUDIO"
P.S.A. 05	Título del aplicativo en la interfaz	Verificar que el título del aplicativo ha sido cambiado y se ajusta a los cambios sugeridos por los usuarios	Al ejecutar el aplicativo y entrar a la pantalla principal el título del aplicativo debe haber sido cambiado, para que no se confunda con los textos de los otros menús.	Al ejecutar el aplicativo y entrar a la pantalla principal el título del aplicativo ha sido cambiado, y se aprecia una diferencia notable con los otros textos.
P.S.A. 06	Texto de los menús	Verificar que los menús en la pantalla principal se ven bien en otras pantallas y/o escenas	Al ejecutar el aplicativo los textos de los menús deberían verse perfectamente en cualquier pantalla y/o escena.	Al ejecutar el aplicativo los textos de los menús se ven perfectamente en cualquier pantalla y/o escena.
P.S.A. 07	Pantalla de Bienvenida	Verificar que se creó una pantalla de bienvenida de acuerdo con las sugerencias de los usuarios.	Al ejecutar el aplicativo debería cargar una página de inicio mostrando un texto de bienvenida y un contador.	Al ejecutar el aplicativo carga una página de inicio mostrando un texto de bienvenida y un contador.
P.S.A. 08	Superposición de menús	Verificar que los menús no se superponen unos con otros al ser ejecutados.	Al ejecutar cualquier menú del aplicativo, si hay uno abierto debería cerrarse para abrir el nuevo seleccionado.	Al ejecutar cualquier menú del aplicativo, se cierra el menú que este abierto.

*Fuente.* Autoría propia

## Postproducción

Ya con el aplicativo desarrollado es momento de hacer la optimización final. En esta etapa se implementan los ajustes finales de iluminación y/o efectos finales y después se hace el renderizado y la generación del máster.

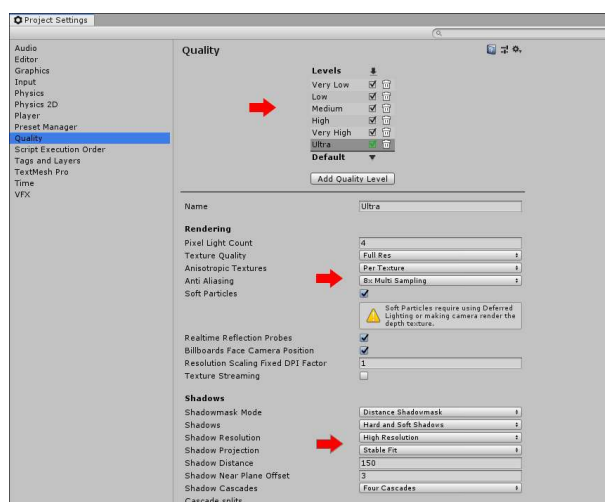
### Ajustes del Proyecto Para Renderizado

Ya se han hecho todos los ajustes del proyecto incluyendo la iluminación global y artificial, ahora antes de exportar el proyecto es importante hacer unas configuraciones para garantizar una buena calidad de visualización.

Para mejorar la calidad del proyecto se revisan varios aspectos como la pestaña “Project Settings” donde se pueden personalizar las opciones de salida en diferentes calidades para dar al usuario la opción de elegir, además de que el aplicativo pueda correr en computadores con características reducidas. Dependiendo de la calidad se configura la resolución, el Anti Aliasing y la calidad de las sombras.

### Figura 41

#### Configuración Project Settings

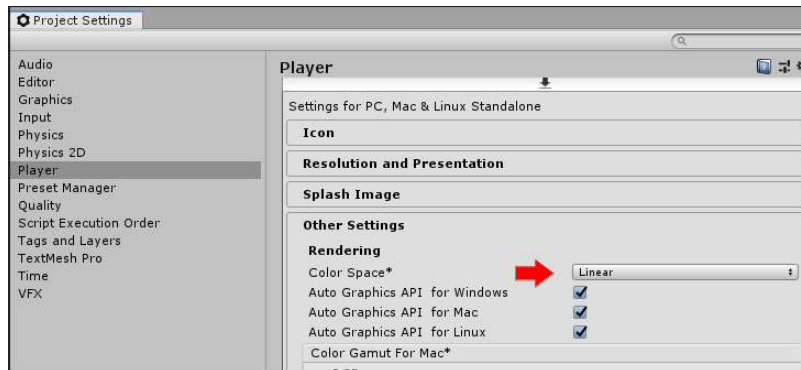


Fuente. Autoría propia, entorno Unity [Captura de Pantalla]

En esta misma pestaña de “Project Settings” en “Other Settings” la opción “Color Space”, por defecto se encuentra en “gamma”, podemos cambiar a la opción “linear”. Esta nos proporciona mayor calidad y más realismo a la escena y una mejor mezcla de los colores.

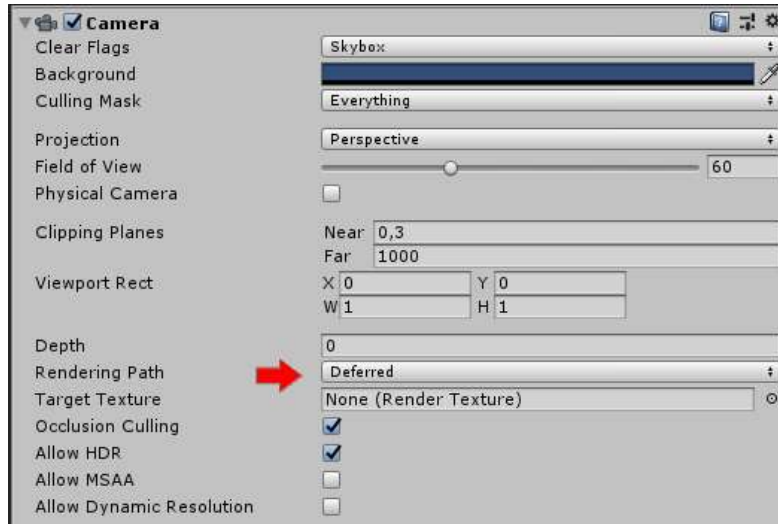
## Figura 42

### *Project Settings Configuración Color Space*



*Fuente.* Autoría propia, entorno Unity [Captura de Pantalla]

Para mejorar la calidad de visualización también se ajusta un parámetro en la cámara principal que es la que se está desplazando por todo el escenario. Dentro de la cámara principal en el inspector en la pestaña de “Rendering Path” se selecciona la opción “Deferred”. Con este parámetro se puede conseguir una mejor iluminación.

**Figura 43***Configuración Rendering Path*

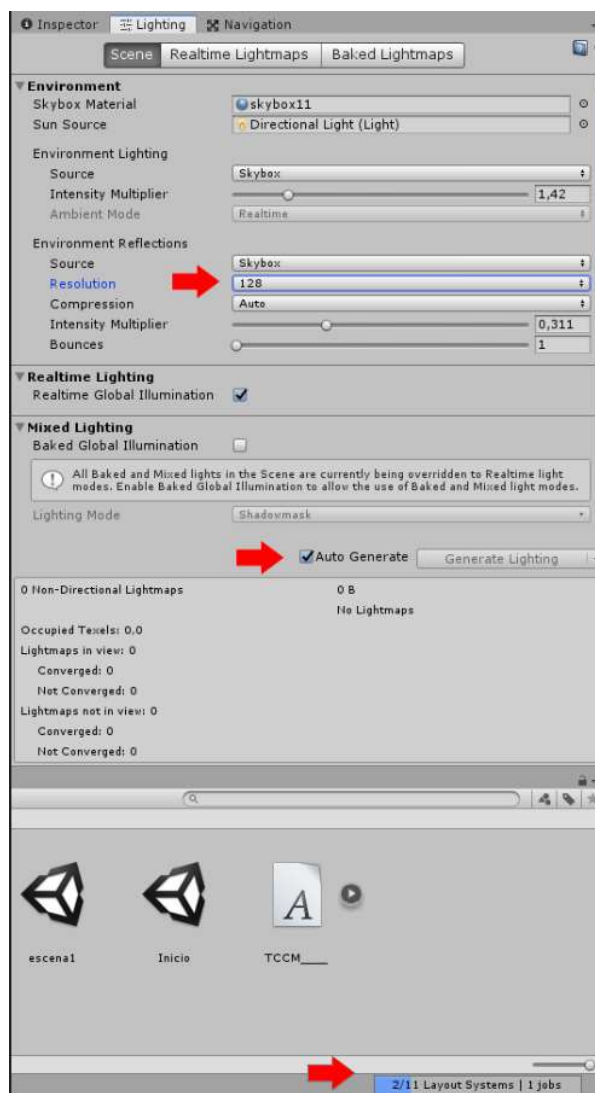
*Fuente.* Autoría propia, entorno Unity [Captura de Pantalla]

En la pestaña de lighting en el inspector se configuran unos parámetros para establecer la calidad del proyecto final.

Se puede modificar la resolución para mejorar la calidad de las reflexiones, lo cual se calculará de acuerdo con el skybox que ha sido previamente creado. Si presionamos la casilla Auto Generate, Unity realiza los cálculos y el baking cada vez que hagamos un cambio en la escena o cuando iniciemos la escena. Podemos presionar Generate Lighting y los cálculos son realizados solo en el momento que lo indiquemos.

## Figura 44

### Configuración Resolución

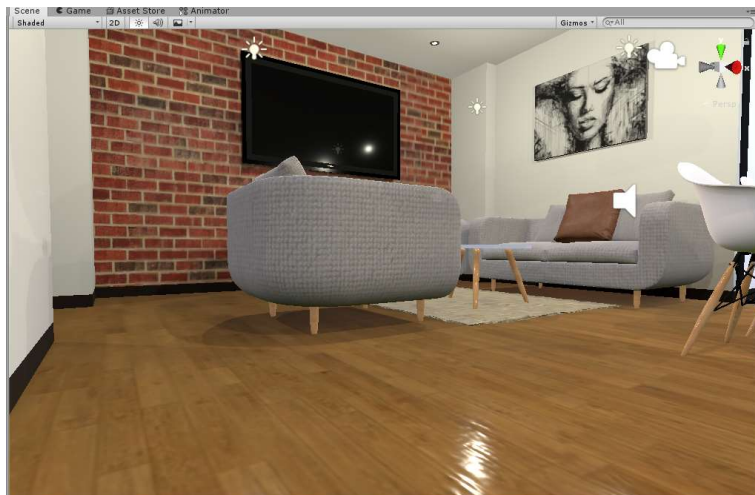


*Fuente.* Autoría propia, entorno Unity [Captura de Pantalla]

Una vez hechos los cálculos y el baking podemos apreciar en la (Ilustración 44) que la calidad mejoró considerablemente, ahora tenemos una escena más realista con reflexiones y sombras de mejor calidad.

## Figura 45

*Calidad después del Baking*



*Fuente.* Autoría propia, entorno Unity [Captura de Pantalla]

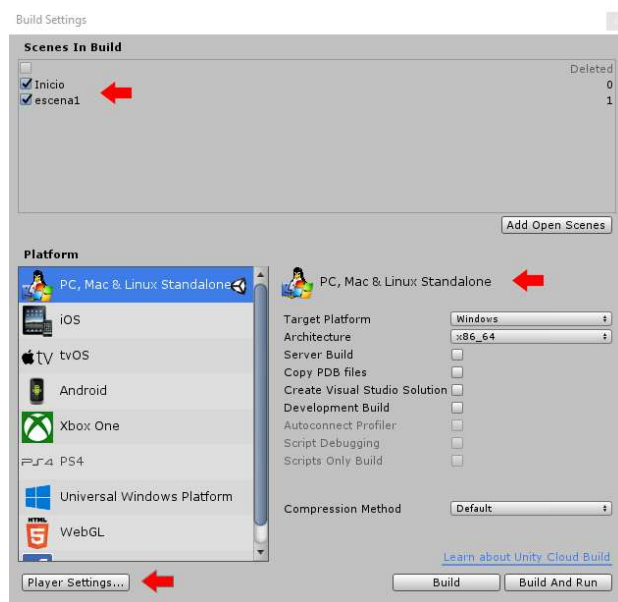
## Generación del Máster e Implementación

Ahora que se han configurado todos los parámetros en una buena calidad procedemos a hacer la generación del máster o archivo ejecutable. Este es el archivo que se utiliza para abrir e interactuar con el aplicativo.

Para exportar el aplicativo nos dirigimos a la pestaña “Building Settings” seleccionamos PC, Mac y Linux Standalone, nos aseguramos de que las dos escenas de nuestro proyecto están agregadas y confirmamos que la escena inicio está de primera.

## Figura 46

### Configuración de las Escenas

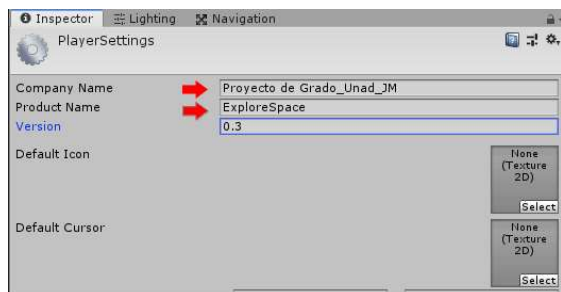


*Fuente.* Autoría propia, entorno Unity [Captura de Pantalla]

Antes de crear el archivo ejecutable presionamos la opción “Player Settings” y aquí podemos poner el nombre de la compañía y el nombre del aplicativo que en este caso es “ExploreSpace”. Finalmente presionamos el botón “Build” y se crea el master listo para su distribución.

## Figura 47

### *Nombre del Proyecto*



*Fuente.* Autoría propia, entorno Unity [Captura de Pantalla]

### ***Publicación del Máster***

<https://drive.google.com/file/d/1->

[Na6srW2kuqZFmmDp4AanUqeTmyH2SRh/view?usp=sharing9](https://drive.google.com/file/d/1-Na6srW2kuqZFmmDp4AanUqeTmyH2SRh/view?usp=sharing9)

## Conclusiones

El proyecto se desarrolló con el propósito de crear una herramienta interactiva novedosa, para la visualización de un proyecto arquitectónico de un apartamento, cuya interactividad fuera controlada a través de comandos de voz, con el fin de llegar a aquellos usuarios que, por algún tipo de discapacidad, no pueden acceder a contenidos digitales, además de crear una nueva alternativa para usuarios comunes. A lo largo del desarrollo de este proyecto se lograron superar algunas dificultades presentadas y conseguir resultados satisfactorios que condujeron al cumplimiento de los objetivos presentados desde el inicio. De acuerdo con lo anterior se presentan a continuación las siguientes conclusiones.

Se logró crear un proyecto arquitectónico interactivo en 3d de un inmueble completo amoblado con materiales aplicados, así como la configuración de una iluminación global y artificial para dar una apariencia realista de todos los espacios donde por medio de comandos de voz se puede controlar la interactividad de varios elementos como lo son las cámaras, materiales, luces y audio.

De acuerdo con el proceso realizado en la fase de preproducción se definieron las características técnicas y con base a la realización del prototipado y las pruebas de prototipado se logró establecer la viabilidad del proyecto.

A partir de la elaboración de los insumos se desarrolló la producción del aplicativo, realizando la construcción de todos los elementos, la creación de la interactividad, la interfaz del usuario además con base en las pruebas de funcionamiento, observación y accesibilidad se pudo determinar que el aplicativo funciona de acuerdo con los objetivos planteados desde el inicio.

Se hizo la configuración de los elementos finales de calidad del aplicativo además de los preparativos finales para la exportación y creación del máster en la etapa de postproducción.

Se logró crear un aplicativo más universal y útil para el acceso a contenidos interactivos controlado por medio de comandos de voz, que representa nuevas formas de interacción a todo tipo de usuarios además de generar inclusión a personas con cierto tipo de discapacidades que les imposibilita el acceso a contenidos digitales.

Se pudo determinar que esta herramienta puede ser muy útil como base para la creación de aplicativos de diferentes ramas como la educación, comercial, salud etc. Donde se pueden implementar los comandos de voz para generar mayor inclusión a personas con discapacidades en sus extremidades superiores.

En resumen, se logró cumplir con los objetivos planteados inicialmente proporcionando un producto novedoso y funcional que presenta una nueva alternativa para la visualización de proyectos arquitectónicos interactivos controlados por comandos de voz.

### Referencias Bibliográficas

- Balea Domínguez, F. (2021). *Renderizado en tiempo real en la visualización arquitectónica*.  
<http://hdl.handle.net/2183/28201>
- Banchoff Tzancoff, C. M., Fava, L. A., Martin, E. S., Diaz Gira, F., & Aguirre, A. (2022).  
*Explorando con realidad virtual interactiva*. In *XXIV Workshop de Investigadores en Ciencias de la Computación (WICC 2022, Mendoza)*.
- Berecibar Gorostiza, L. (2018). *Reconocimiento de voz para interacciones con aplicación de Realidad Virtual o Aumentada* (Bachelor's thesis, Universitat Politècnica de Catalunya).
- Binti Azizo, A. S., bin Mohamed, F., Siang, C. V., & Isham, M. I. M. (2020, December). *Virtual reality 360 UTM campus tour with voice commands*. In *2020 6th International Conference on Interactive Digital Media (ICIDM)* (pp. 1-6). IEEE.
- Devesa Marco, F. (2019). *Estudio y experimentación de dispositivos de interacción avanzados sobre Unity 3D: PSMove* (Doctoral dissertation, Universitat Politècnica de València).
- EuroInnova. (2023) *Etapas del Proceso de Creación Multimedia*. EuroInnova.  
<https://www.euroinnova.com/blog/etapas-del-proceso-de-creacion-multimedia>
- Ferrone, H. (2020). *Learning C# by Developing Games with Unity 2020: An enjoyable and intuitive approach to getting started with C# programming and Unity*. Packt Publishing Ltd.
- García, J. J., & Ramírez Salazar, D. L. (2023). *Realidad Virtual para Proyectos de Vivienda* (Master's thesis, Maestría en Gerencia de Sistemas de Información y Proyectos Tecnológicos-Virtual).
- García Roig, P. (2022). *Visualización arquitectónica VR*. <http://hdl.handle.net/10459.1/85311>

- Li, C., & Tang, B. (2019, November). Research on voice interaction technology in vr environment. In *2019 International Conference on Electronic Engineering and Informatics (EEI)* (pp. 213-216). IEEE.
- Linowes, J. (2020). *Unity 2020 virtual reality projects: Learn VR development by building immersive applications and games with Unity 2019.4 and later versions*. Packt Publishing Ltd.
- López Vargas, J. M., García Quintero, P. A., & Cuteño Ramos, D. A. (2020). *Laieer SCV: Sistema de comando de voz*.
- Martines, S. (2021) *Motores Gráficos en Tiempo Real aplicados a la Arquitectura*.  
<http://hdl.handle.net/10045/117031>
- OVI - Sasa, I.D. (2022) *OVI Aplicaciones Multimedia Diseño y Desarrollo*  
[Objeto\_virtual\_de\_Información] Repositorio Institucional.
- Proaño, r., díaz, c., & montenegro, r. (2023) *Preproducción Multimedia: Del Concepto al Pitch*.  
Instituto Superior Tecnológico Quito
- Reina, J. (2022). *Diseño de Metáforas para interfaz de usuario*.  
[Objeto\_virtual\_de\_Informacion\_OVI]. Repositorio Institucional UNAD.  
<https://repository.unad.edu.co/handle/10596/49597>.
- Reina, J. (2022). *Pruebas de usuario*. [Objeto\_virtual\_de\_Informacion\_OVI]. Repositorio Institucional UNAD. <https://repository.unad.edu.co/handle/10596/49592>
- Rodríguez González, J. L. (2016). *Manual básico de AutoCAD v. 2016 para 2D: técnicas y herramientas para ingeniería, expresión gráfica*.
- Saza, Ilber, D. (2019). *Interfaz gráfica en Videojuegos*. [Objeto\_virtual\_de\_Informacion\_OVI].  
<https://repository.unad.edu.co/handle/10596/30198>
- Tickoo, S. (2020). *Autodesk 3Ds Max 2021: A comprehensive guide*. Cadcim Technologies.

Valcárcel, D.R, Toro, C.H.F., & Ojea, F.J. de la P. (2011). *Multimedia Smarth Process (MSP)*.

CISTI (Iberian Conference on Information System & Technologies / Conferencia Ibérica de Tecnologías de Información) Proceedings, 320-325

Zamacola Crespo, J. F. (2021). *Técnicas de videojuegos para visualización arquitectónica: diseño computacional, renderizado en tiempo real y realidad virtual*.

<https://oa.upm.es/view/institution/Arquitectura>