

# ◆ VR-Based Product Evaluation System ◆

## Instrucciones (ES)

Introducir publicación asociada al proyecto (INGEGRAF)

## **ÍNDICE**

ÍNDICE .....	2
INTRODUCCIÓN AL PROYECTO .....	3
PRIMEROS PASOS .....	4
1. Instalación de Unity Hub y Unity .....	4
2. Descargar proyecto .....	4
RECURSOS .....	5
1. Escenas .....	5
2. Modelos .....	5
3. Prefabs .....	5
4. StreamingAssets .....	5
5. Otros .....	6
5.1. Materiales .....	6
5.2. Scripts .....	6
5.3. Sprites .....	6
5.4. Skyboxes .....	6
CONFIGURACIÓN .....	7
1. Añadir códigos de usuario .....	7
2. Introducir nuevo producto .....	7
2.1. Importar FBX .....	7
2.2. Añadir modelo a la jerarquía .....	8
2.3. Configurar Volúmenes de Interés .....	8
2.4. Configuración del Diferencial Semántico .....	9
FUNCIONAMIENTO .....	11
1. En el editor .....	11
2. Mediante APK .....	11
ANÁLISIS DE DATOS .....	12
1. Instalación de Python .....	12
2. Ejecutar interface.exe .....	12
3. Revisión de datos .....	12

## ◆ INTRODUCCIÓN AL PROYECTO

Este proyecto, titulado *VR-Based Product Evaluation System*, es una herramienta diseñada para investigadores que deseen realizar evaluaciones de productos en entornos de Realidad Virtual (RV) sin necesidad de contar con conocimientos especializados en desarrollo de entornos virtuales o gestión de los datos brutos obtenidos.

La plataforma permite recopilar y analizar la percepción de distintos usuarios sobre un prototipo virtual de un objeto específico, mientras se registran datos de Eye-Tracking (ET) y la posición del usuario dentro de un entorno virtual inmersivo. Además, incluye una aplicación complementaria para convertir los datos obtenidos en formatos comprensibles y fácilmente interpretables por el investigador.

Para utilizar esta herramienta de manera óptima, es necesario contar con el Head Mounted Display (HMD) Meta Quest Pro, un avanzado dispositivo de RV equipado con tecnología de ET.

Este manual se ha organizado en cinco secciones clave:

- ✓ **Primeros pasos:** Explicación de los softwares esenciales para el uso de la herramienta, junto con instrucciones detalladas sobre la descarga e instalación del proyecto.
- ✓ **Recursos:** Descripción de la estructura del proyecto y la ubicación de sus distintos recursos para facilitar su localización y uso.
- ✓ **Configuración:** Aunque el proyecto incluye recursos prediseñados, algunos deben ser adaptados a los casos específicos de estudio. Esta sección guía al investigador en la modificación de los elementos configurables.
- ✓ **Funcionamiento:** Explicación detallada del uso del sistema tras su configuración, incluyendo instrucciones para su correcta operación.
- ✓ **Análisis de datos:** Los datos obtenidos en formato CSV deben ser procesados mediante la aplicación incluida en el proyecto. En esta sección se describe cómo transformar la información en resultados útiles para la investigación.

## ◆ PRIMEROS PASOS

### 1. Instalación de Unity Hub y Unity

Descarga e instala [Unity](#) (versión mínima 2022.3.46f1) para poder abrir el proyecto. Aunque [Unity Hub](#) es opcional, se recomienda su instalación.

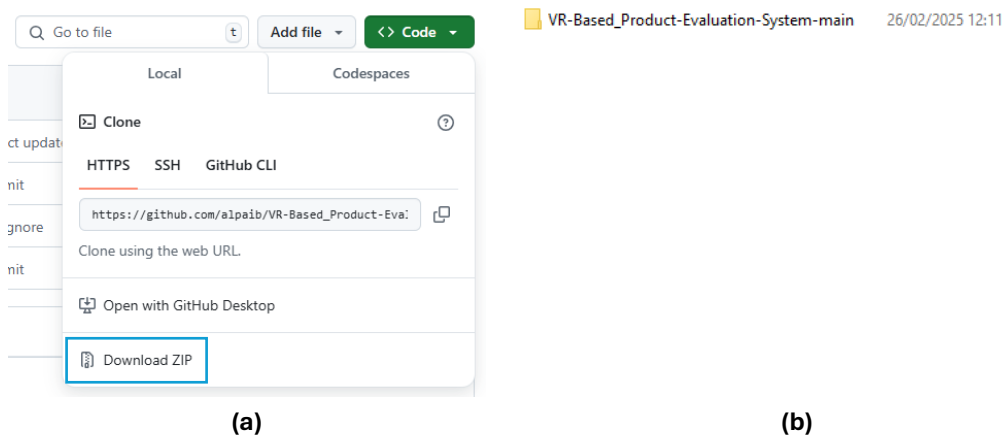
Si nunca has usado Unity, puedes encontrar información adicional y tutoriales introductorios [aquí](#). También se aconseja aprender de forma global en qué consiste su [interfaz](#), pues se hará referencia a algunas de las partes a la hora de localizar opciones, llevar a cabo acciones específicas, etc.

### 2. Descargar proyecto

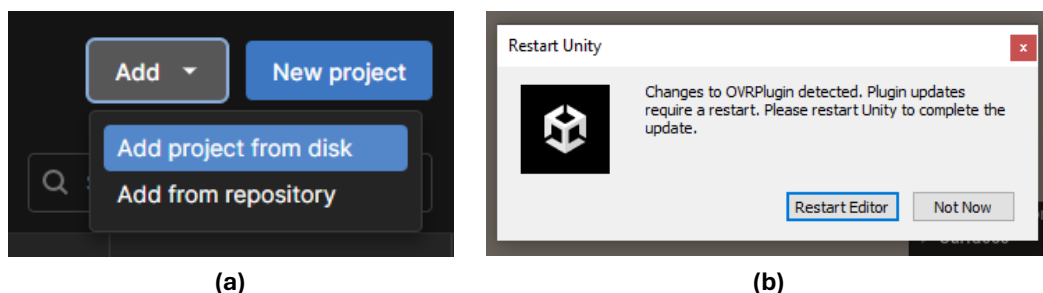
Descarga el proyecto desde [aquí](#) en formato ZIP (Figura 1.a), guárdalo en tu PC y extrae los ficheros (Figura 1.b).

A continuación, abre Unity Hub y haz clic sobre *Add Project from disk* (Figura 2.a). Busca el proyecto descargado y haz clic sobre la carpeta de descarga hasta ver subcarpetas como *Assets*, *Logs* y *Obj*. En ese momento, haz clic en *Add project*.

El proyecto aparecerá en la interfaz de Unity Hub y podrás abrirlo. Es posible que Unity te solicite reiniciar el programa después de cargar los recursos, así que haz clic en *Restart* como se muestra en la Figura 2.b.



**Figura 1.** Descarga del proyecto en formato ZIP desde GitHub.



**Figura 2.** En (a) se muestra cómo añadir el proyecto a Unity Hub, y en (b) la ventana emergente que pide reiniciar Unity tras cargar los recursos.

## RECURSOS

Este proyecto incluye una serie de recursos organizados en carpetas que garantizan su funcionamiento. En este contexto, la Figura 3 muestra todas estas carpetas, siendo el objetivo de este apartado describir cada una de ellas, lo que te permitirá localizarlas fácilmente una vez que se haya explicado el contenido de cada una.

### 1. Escenas

Las escenas del proyecto se encuentran en la carpeta “01 Scenes” y son las siguientes:

- Intro-scene (S-1). Es la primera escena en la que se emplaza al usuario al comenzar la experiencia. Se trata de una sala localizada en un entorno neutro de baja luminosidad. En ella, durante la experiencia virtual el usuario selecciona su código de sujeto (asignado por el investigador previamente) en un panel con ayuda de los controladores de las Meta Quest Pro.
- Evaluation-scene (S-2). Esta escena retrata un entorno neutro de luminosidad más alta. En ella el sujeto puede observar un producto específico mientras se recopilan datos de ET y posición, y a través de los controladores de las Meta Quest Pro, puede evaluar los diferentes atributos de este mediante un Diferencial Semántico.

### 2. Modelos

En la carpeta “02 Models” se guardan los modelos 3D que posteriormente se pueden añadir a la S-2. Se recomienda que el formato sea FBX, aunque también se pueden introducir otros [formatos](#).

### 3. Prefabs

En la carpeta “03 Prefabs” se encuentran dos elementos importantes:

- Snapshot. Este elemento se agrega a la escena S-2 para tomar una captura en plano cenital de la escena. Luego, se utiliza para pintar la trayectoria del sujeto y generar el mapa de calor de su recorrido.
- Vol-Prefab. Esto representa un Volumen de Interés (Vol). Este objeto define, mediante su reconfiguración, distintas partes del producto que se está evaluando y será detectado por el *gaze* de las Meta Quest Pro para recopilar datos de ET.

### 4. StreamingAssets

En la carpeta denominada “streamingAssets” se guarda la captura tomada con el prefab Snapshot. Adicionalmente, aquí se guardan los archivos CSV que contienen los datos de evaluación, ET y posición en caso de que la experiencia virtual se realice con el HMD conectado al ordenador una vez se pare la aplicación. Cada archivo se guarda con una nomenclatura única para cada sujeto, siguiendo el siguiente formato:

- Code\_Eye-tracking (datos de ET).
- Code\_Position (datos de posición).
- Code\_Semantic-Differential (datos de evaluación).

## 5. Otros

### 5.1. Materiales

Para garantizar una buena organización del proyecto, se aconseja que todos los nuevos materiales que sean creados se guarden en la carpeta “04 Materials”.

### 5.2. Scripts

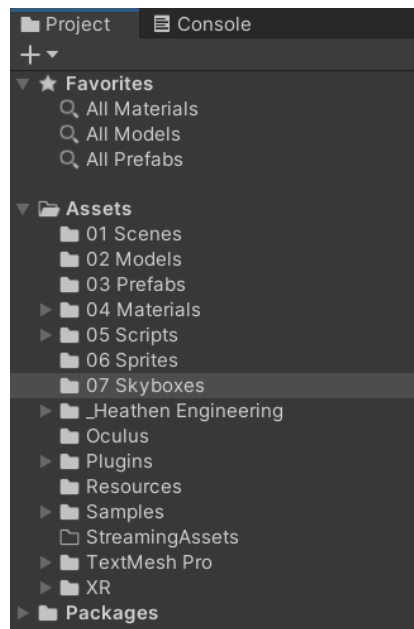
Los scripts que se han utilizado para desarrollar este proyecto están localizados en la carpeta “05 Scripts”. Además, dentro de “Data analysis – dist” se encuentra la aplicación “interface.exe”, la cual se utilizará para procesar los datos obtenidos durante la experiencia virtual.

### 5.3. Sprites

Para garantizar una buena organización del proyecto, se aconseja que todos los nuevos sprites que sean creados y/o importados se guarden en la carpeta “06 Sprites”.

### 5.4. Skyboxes

De forma similar, se aconseja que todos los nuevos skyboxes que sean creados se guarden en la carpeta “07 Skyboxes”. Además, aquí se encuentran los materiales “Dark-Skybox” y “Soft-Skybox”, utilizados en S-1 y S-2 respectivamente, por lo que pueden ser modificados sin que afecte al funcionamiento de la aplicación.



**Figura 3.** Carpetas principales que componen el proyecto de Unity.

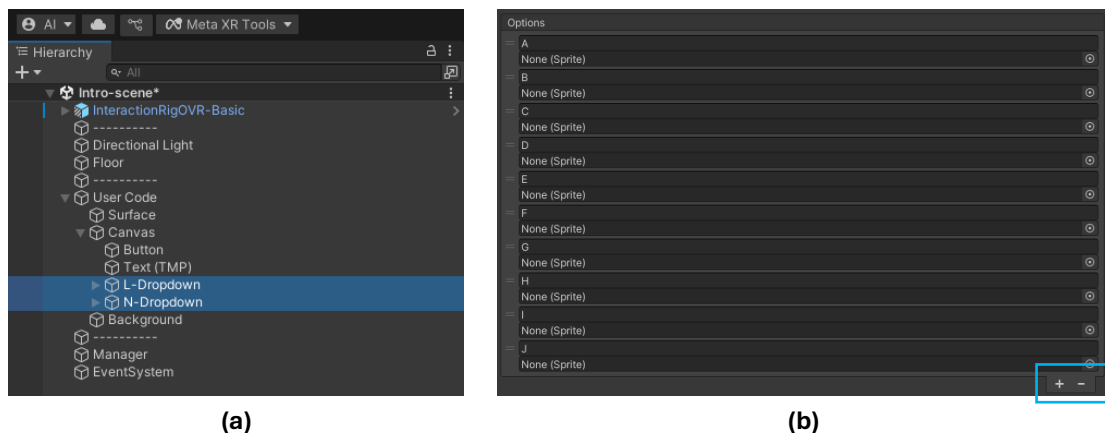
## ◆ CONFIGURACIÓN

### 1. Añadir códigos de usuario

Actualmente, existen códigos alfanuméricos disponibles para 100 personas, generados mediante la combinación de una letra (A-J) y un número (1-10). Si el investigador necesita más combinaciones, puede añadir más letras o números utilizando los componentes “L-Dropdown” (letras) y “N-Dropdown” (números), que se encuentran dentro del componente User Code/Canvas en la jerarquía (Figura 4.a).

Al seleccionar uno de estos componentes, podrás añadir más opciones dentro del campo Dropdown – TextMeshPro, ubicado en el inspector (lateral derecho de la interfaz de Unity). En la parte inferior de este componente, verás los símbolos “+” (para añadir) y “-” (para quitar) que te permitirán modificar las opciones disponibles (Figura 4.b).

Tras realizar estas modificaciones, es recomendable guardar la escena y el proyecto (File – Save o Ctrl + S; File – Save Project o Ctrl + Shift + S).

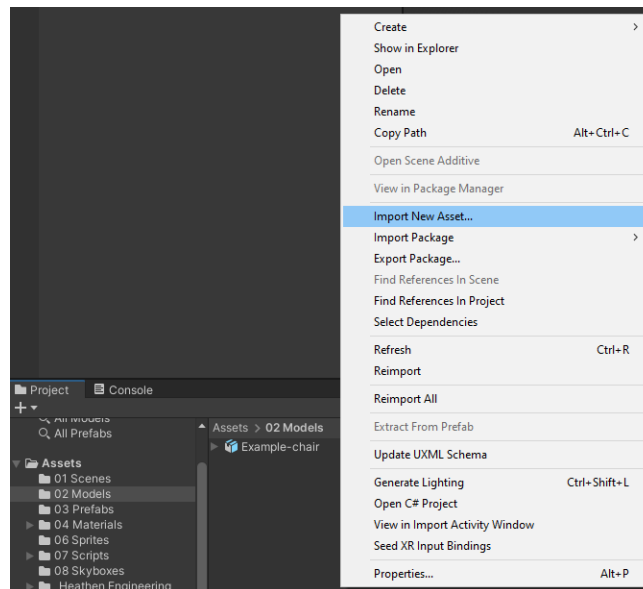


**Figura 4.** En (a) se muestran los componentes dentro de la jerarquía de Unity, y en (b) se muestra cómo añadir más elementos.

### 2. Introducir nuevo producto

#### 2.1. Importar FBX

En primer lugar, necesitamos un modelo 3D en formato FBX. En la ventana “Project” (abajo en la interfaz de Unity) nos dirigimos a la ventana “02 Models”, y haciendo clic derecho en ella podremos seleccionar la opción *Import New Asset...* (Figura 5). Buscamos nuestro modelo en el ordenador y lo importamos.

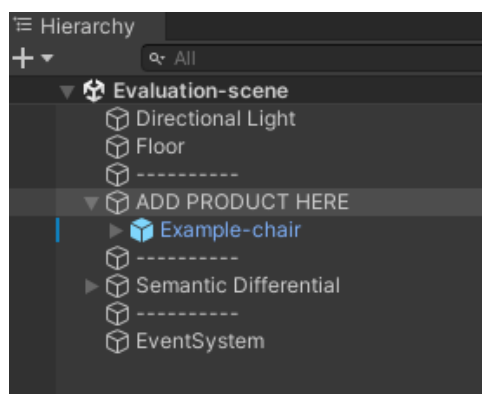


**Figura 5.** Importar FBX al proyecto.

## 2.2. Añadir modelo a la jerarquía

En segundo lugar, es necesario introducir nuestro modelo en la escena “Evaluation-scene” (S-2). Para ello, hacemos clic en el FBX recién importado, y manteniendo el clic, arrastramos a la jerarquía sobre “ADD PRODUCT HERE”. Esto hace que tu modelo quede como hijo de ese objeto, como se ve en la Figura 6. Para saber más sobre este proceso, usar este [enlace](#).

En este punto, los valores de posición, rotación y escala en el objeto pueden modificarse al gusto del investigador (Inspector).



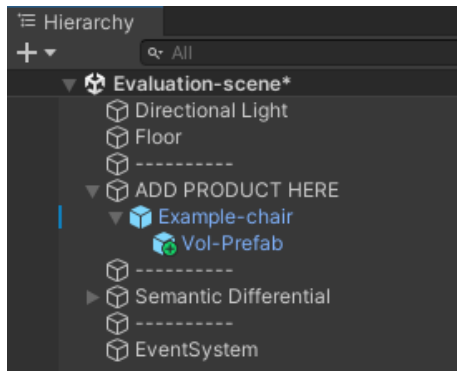
**Figura 6.** Añadir modelo 3D a la jerarquía en la escena “Evaluation-scene”.

## 2.3. Configurar Volúmenes de Interés

Para que se puedan recoger datos sobre la visualización del producto, es necesario definir Volúmenes de Interés (Vols) concretos. Para ello, dentro de la carpeta “03 Prefabs” existe un objeto llamado “Vol-Prefab” previamente configurado para funcionar como Vol. Este objeto debe arrastrarse a la jerarquía como hijo del producto, y duplicarse (Ctrl + D) tantas veces como volúmenes de interés existan (por ejemplo, si tengo una silla con cuatro patas, asiento y respaldo, tendré seis Vol-Prefabs en la escena).



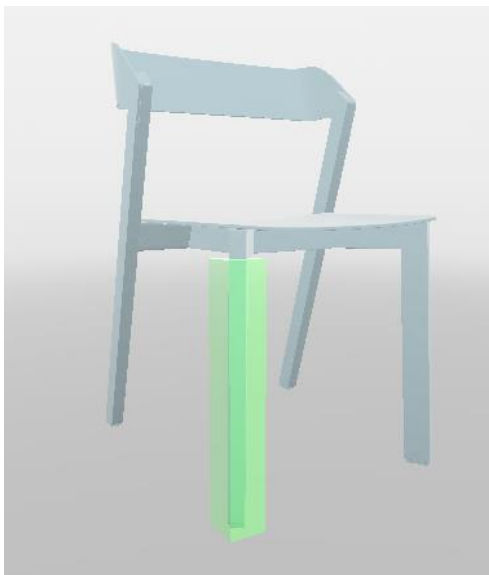
En la Figura 7.a y Figura 7.b puede verse un Vol-Prefab recién añadido a la escena. Si lo seleccionamos, podemos ajustar sus valores de escala, posición y rotación hasta que englobe una de las partes que nos interese (Figura 7.c). Por último, en la jerarquía hay que actualizar el nombre, ya que es el nombre con el que aparecerá en el CSV final de datos y apagar el componente “Mesh renderer” (Figura 7.d).



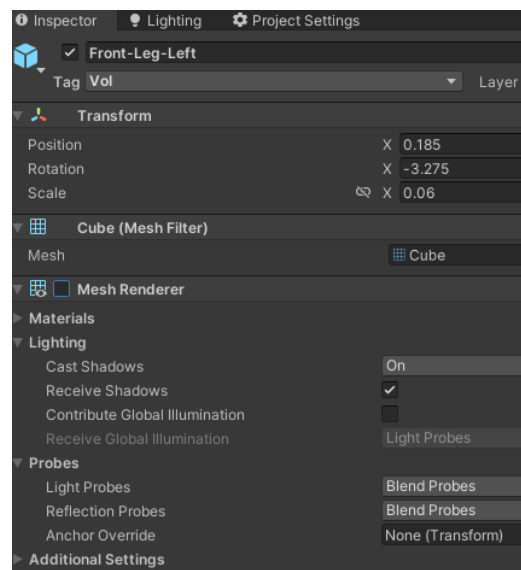
(a)



(b)



(c)



(d)

Figura 7. Vol en la escena.

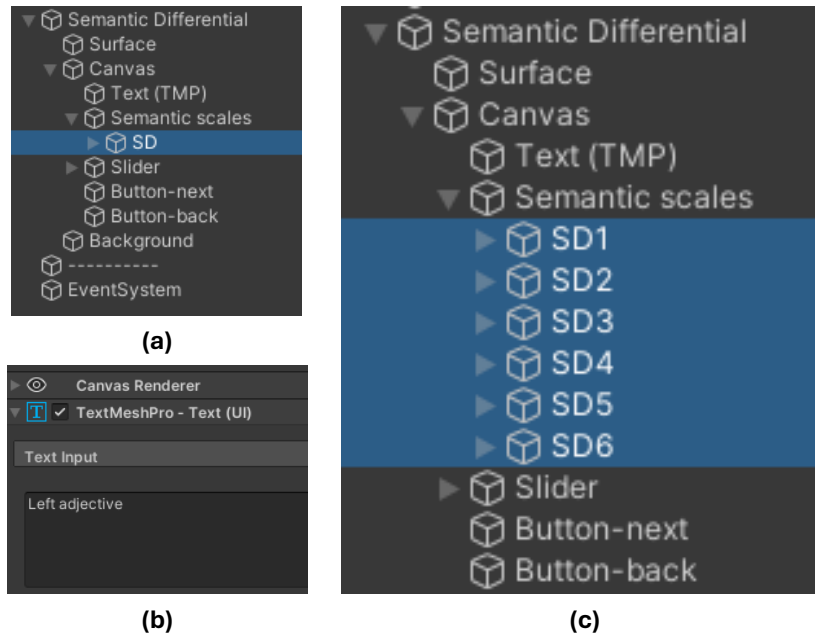
## 2.4. Configuración del Diferencial Semántico

Para poder conocer la opinión del usuario en cuanto al objeto mostrado, se dispone de un panel que muestra una serie de escalas semánticas usando una escala tipo Likert del -3 al 3, siendo 0 un valor neutral. Este componente se llama “Semantic Differential” en la jerarquía de la escena “Evaluation-scene”.

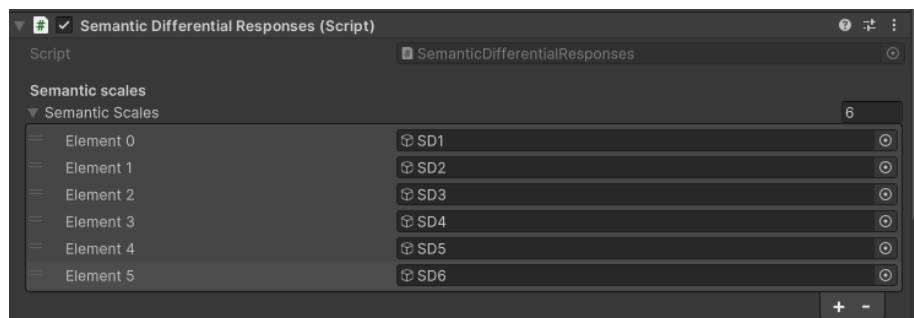
Actualmente hay únicamente una escala de ejemplo. Sin embargo, este componente puede duplicarse (Ctrl + D) en la jerarquía y configurar ateniendo a atributos específicos. Si desplegamos el componente Semantic Differential – Canvas – Semantic scales, encontraremos el componente “SD” (Figura 8.a). Si queremos conocer la opinión sobre seis atributos (por ejemplo), debemos duplicar el componente seis veces, y podemos renombrarlo de alguna forma específica para tenerlos mejor identificados (Figura 8.b). Por otro lado, si desplegamos cada uno de estos componentes, veremos uso llamados “L-Adj” (que corresponde al adjetivo de la izquierda) y “R-

Adj”, que corresponde al de la derecha. Si lo seleccionamos, podemos cambiar en el componente TextMeshPro – Text (UI) del inspector el nombre del adjetivo (Figura 8.c).

Por último, hay que seleccionar nuevamente el componente “Semantic Differential” en la jerarquía. En el inspector, buscar el componente “Semantic Differential Responses (Script)”, y añadir cada una de las escalas semánticas creadas con anterioridad (Figura 9) en la parte que pone “Semantic Scales”. Si hay que añadir más elementos, pulsar “+”.



**Figura 8.** Vol en la escena.



**Figura 9.** Añadir escalas semánticas al Diferencial Semántico.

## ◆ FUNCIONAMIENTO

### 1. En el editor

Si vas a utilizar el casco **conectado al ordenador**, esta es tu opción.

En primer lugar, deberás seguir los pasos indicados [aquí](#) para poder utilizar tu HMD.

Antes de hacer clic en *Play*, deberás buscar los siguientes objetos en la jerarquía de Unity para aplicar unos cambios:

- Manager (S-1): Desmarcar en “Eye Tracking Recorder (Script)” y “Position Recorder (Script)” (dentro del Inspector) la casilla “To HMD” en *Export to HMD?* (Figura 10).
- Semantic Differential (S-2): Desmarcar en “ Semantic Differential Responses (Script)” (dentro del Inspector) la casilla “To HMD” en *Export to HMD?* (Figura 10).



**Figura 10.** Componente sin marcar para poder exportar los datos a la carpeta streamingAssets del proyecto de Unity.

### 2. Mediante APK

Si quieres utilizar el casco de forma **autónoma**, esta es tu opción.

En primer lugar, debes construir una APK (para más información, visitar este [enlace](#)), pero de forma previa, debes marcar la opción de la Figura 10 en los mismos componentes comentados en el apartado anterior. Esto hará que los datos se exporten dentro del casco. Debe quedar como en la Figura 11.

Una vez cambiada la configuración anterior, ve a *Project settings – Meta*. Haz clic sobre *Fix* y *Apply all* en caso de que existan errores o cambios recomendados. A continuación, ve a *File – Build Setting*, y en el apartado *Platform*, haz clic en *Android* y en *Switch Platform*. Esto hará que puedas hacer una build.

Cuando hagas clic en *Build*, deberás darle nombre a la aplicación (nombre con el que identificas la APK en tu PC, pues en el casco saldrá con “VR-Based\_Product-Evaluation-System” una vez se instale).

Cuando tengas la APK, instálala en el casco. Deberás [descargar](#) el software Meta Quest Developer Hub (MQDH) y seguir [este](#) procedimiento. Una vez instalada, podrás usarla.



**Figura 11.** Componente marcado para poder exportar los datos al casco.

## ◆ ANÁLISIS DE DATOS

Si has utilizado la aplicación a través del editor, los CSV se han guardado en una carpeta llamada “streamingAssets” del proyecto de Unity. Si no es así, deberás buscar los archivos en la carpeta de instalación de la aplicación dentro del casco (la ruta es similar a “QuestPro/Android/Data/AppName/”).

Solo en este último caso, copia todos los CSV y guárdalos de forma local en el PC. Puedes guardarlo en cualquier carpeta, pero ten localizada la ruta, pues tendrás que usarla más adelante.

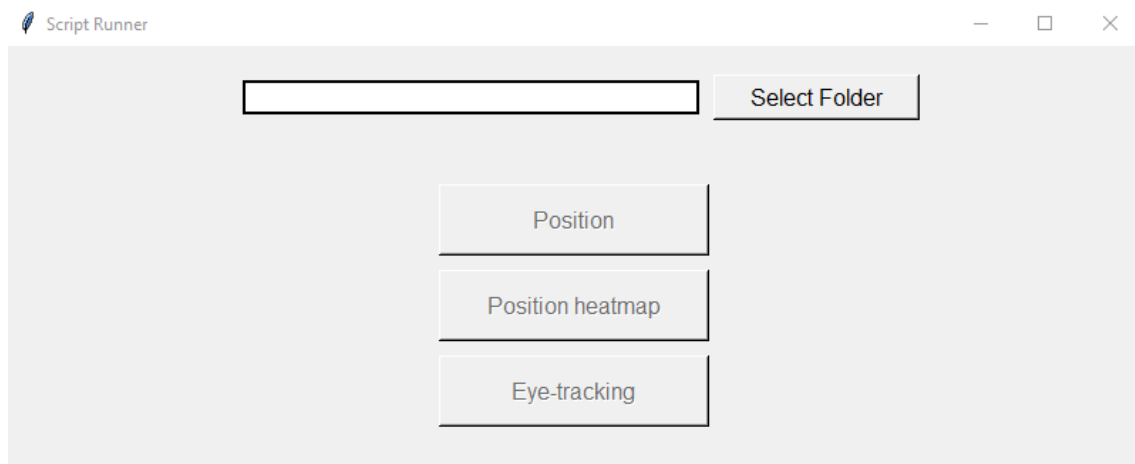
### 1. Instalación de Python

Dirígete a la carpeta “05 Scripts – Data analysis” y ejecuta “python-3.13.2.exe”. En el caso de que quieras una versión más actual, dirígete a la [web oficial](#) de Python para descargarla.

### 2. Ejecutar interface.exe

En “05 Scripts – Data analysis – dist” existe un .exe llamado “interface.exe” que debes ejecutar (Figura 12). En primer lugar, deberás especificar la ruta en la que se encuentran todos los CSV generados mediante el botón “Select Folder”.

Una vez especificada, haz clic en cada uno de los botones que se encuentran en la parte inferior para gestionar los datos obtenidos (Position, Position heatmap y Eye-tracking).



**Figura 12.** Componente sin marcar para poder exportar los datos a la carpeta streamingAssets del proyecto de Unity.

### 3. Revisión de datos

Todas las imágenes, así como el CSV finales con los datos de ET (“final-data.csv”) se almacenan dentro de la carpeta en la que se encuentran los CSV originales (aquella que introdujiste en el apartado anterior).