



**Escuela Superior  
de Ingeniería y Tecnología**  
Universidad de La Laguna

# Trabajo de Fin de Grado

Grado en Ingeniería Informática

## Realidad aumentada en educación

*Augmented reality in education*

Juan José Pérez López

La Laguna, 7 de septiembre de 2021

D. **Evelio José González González**, con N.I.F. 43.372.115-A profesor Titular de Universidad adscrito al Departamento de Ingeniería Informática y de Sistemas de la Universidad de La Laguna, como tutor

## **C E R T I F I C A**

Que la presente memoria titulada:

*“Realidad aumentada en educación”*

ha sido realizada bajo su dirección por D. **Juan José Pérez López**,  
con N.I.F. 54.109.255-F.

Y para que así conste, en cumplimiento de la legislación vigente y a los efectos oportunos firman la presente en La Laguna a 7 de septiembre de 2021

## Agradecimientos

A Evelio José González González, por su ayuda y orientación durante la realización de este proyecto.  
También a mi familia y amigos, que me han apoyado a lo largo de esta etapa.

# Licencia



© Esta obra está bajo una licencia de Creative Commons Reconocimiento-NoComercial-CompartirIgual 4.0 Internacional.

## Resumen

*En los últimos años el desarrollo de aplicaciones de realidad aumentada ha crecido considerablemente. Su potencial es innegable, y multitud de campos se han beneficiado ya de las enormes posibilidades que ofrece. Esta tecnología en la educación presenta unas características muy viables, por lo que ya existen multitud de aplicaciones de realidad aumentada.*

*Este trabajo consiste en el análisis de los posibles beneficios de esta herramienta dentro de las aulas. También se ha desarrollado una aplicación que hace uso de esta tecnología y la implementa dentro de la propia aula, dando a los profesores la capacidad de elegir lo que ven los alumnos, y poder así adaptar el rumbo de la materia como mejor le convenga.*

*Para el desarrollo de este trabajo se han utilizado herramientas como Unity y ARCore. Estará disponible en dispositivos Android.*

**Palabras clave:** *Realidad aumentada, aplicación, Unity, ARCore, educación.*

## **Abstract**

*In recent years the development of augmented reality applications has increased significantly. Their potential is undeniable and multiple fields have already benefited from the possibilities that offers. In education, this technology presents viable features for that reason many applications of augmented reality have been created by now.*

*This assignment consists in the analysis of the potential benefits about this implement in the classroom. Also, has been developed an application which makes use of this kind of technology applying it on classes, giving the teachers the capability of choosing which contents the students are watching, with the aim of adapt the subject as needed.*

*Tools like Unity and ARCore have been used for the development of this project. It will be available in Android devices.*

**Key words:** *Augmented reality, application, Unity, ARCore, education.*

# Índice general

<b>Capítulo 1 - Introducción.....</b>	<b>1</b>
1.1 Problema .....	1
1.2 Motivación.....	2
1.3 Objetivos .....	2
1.4 Estructura de la memoria .....	2
<b>Capítulo 2 - Base teórica.....</b>	<b>3</b>
2.1 Historia.....	3
2.2 La realidad aumentada .....	3
2.3 Como funciona la realidad aumentada .....	4
2.4 Niveles de la realidad aumentada.....	4
<b>Capítulo 3 - Análisis.....</b>	<b>6</b>
3.1 La realidad aumentada actualmente .....	6
3.2 Realidad aumentada en educación .....	6
3.3 Aplicaciones de realidad aumentada para educación .....	7
3.4 Efectos sobre el aprendizaje .....	8
<b>Capítulo 4 - Herramientas de desarrollo.....</b>	<b>10</b>
4.1 Unity .....	10
4.2 ARCore.....	10
4.3 Visual Studio Community 2019 .....	11
4.4 Base de datos.....	11
4.5 FileZilla .....	11
<b>Capítulo 5 - Desarrollo.....</b>	<b>12</b>
5.1 Diseño de la base de datos .....	13
5.2 Conexión con la base de datos .....	14
5.3 Escenas de Unity .....	15
5.4 Autenticación.....	16
5.5 Selección de clases y cursos.....	17

5.6 Selección de asignatura y modelos.....	19
5.7 Cámara de realidad aumentada .....	20
5.8 Menú principal estudiantes .....	22
<b>Capítulo 6 - Conclusiones y líneas futuras.....</b>	<b>24</b>
<b>Capítulo 7 - Summary and Conclusions.....</b>	<b>25</b>
<b>Capítulo 8 - Presupuesto.....</b>	<b>26</b>
<b>Bibliografía .....</b>	<b>28</b>
<b>Anexos .....</b>	<b>30</b>

## Índice de figuras

Figura 1. Diagrama de Milgram [3].....	4
Figura 2. Interfaz Metaverse [15]. .....	8
Figura 3. Logo de unity [18].....	10
Figura 4. Logo ARCore [19].....	11
Figura 5. Esquema de uso de la aplicación.....	13
Figura 6. Tablas y columnas en phpMyAdmin.....	14
Figura 7. Pantalla de inicio de sesión.....	17
Figura 8. Pantalla de selección de clases. ....	19
Figura 9. Escena de selección de asignatura y modelo 3D .....	20
Figura 10. Colocación de un modelo 3D en RA.....	22
Figura 11. Escena principal de estudiantes.....	23



# Índice de tablas

Tabla 1. Tablas de la base de datos.....	14
Tabla 2. Presupuesto de la elaboración de la aplicación.....	27

# Capítulo 1

## Introducción

Aunque anteriormente ya se habían desarrollado máquinas de realidad aumentada (RA), no fue hasta la década de los 90's cuando empezó a desarrollarse tal como la conocemos hoy en día.

La RA aplicada a la educación es una tecnología realmente prometedora, pero con mucho por experimentar por delante, pues hay escasas referencias a su aplicación en la educación.

Es importante escoger bien los objetivos y su audiencia con estas aplicaciones pues se trata de pedagogías emergentes: “Conjunto de enfoques e ideas pedagógicas que surgen alrededor del uso de las TIC en educación y que intentan aprovechar todo su potencial comunicativo, informacional, colaborativo, interactivo, creativo e innovador en el marco de una nueva cultura de aprendizaje”. Muchos sostienen la idea de ser prudentes con estas tecnologías, sin embargo otros afirman que superan la limitación del tiempo y del espacio en los entornos tradicionales, y que no solo responden a este tipo de exigencia, si no que la amplían de manera cualitativamente significativa al ofrecer información situada, contextualizada, desde el lugar y en el momento que el consumidor la precisa [4].

De todas maneras sabemos que herramientas como la RA en un futuro, y dada la situación mundial actual de pandemia. Se hace indispensable contar con herramientas tecnológicas que faciliten las actividades educativas en sus distintos niveles. Hoy en día son muchos los estudiantes que cuentan con herramientas necesarias para adaptarse, como un teléfono móvil para poder ejecutar una aplicación de RA.

### 1.1 Problema

En los últimos años la realidad aumentada ha tenido un crecimiento considerable, sobretodo en el ámbito de los videojuegos, sin embargo es aplicable y útil en muchas otras ramas.

Por lo general una aplicación de RA tiene unos altos costos y un largo tiempo de desarrollo.

## 1.2 Motivación

Hoy en día existen una gran de dispositivos móviles y es muy fácil acceder a uno de ellos. Esto hace posible el desarrollo de herramientas de RA, ofreciendo así nuevas maneras de comunicación, entretenimiento y educación.

Como se ha comentado anteriormente, no existe una base sólida de RA en educación. Por lo que este trabajo pretende motivar el uso de esta tecnología en las aulas y crear una aplicación que tenga contacto directo con los centros y las aulas.

## 1.3 Objetivos

El objetivo principal es el desarrollo de una aplicación de realidad aumentada para usar en los centros educativos. Esta aplicación permitirá al profesor seleccionar un modelo 3D a una clase, y así permitir a los alumnos visualizar dicho modelo con su propio dispositivo. Para llevar a cabo este proyecto, se han realizado las siguientes tareas:

- Estudiar una base teórica sobre la situación actual de la RA, y sobre esta en la educación.
- Crear un prototipo inicial que permitirá desarrollar la versión final de la aplicación.
- Instalar y aprender sobre las herramientas usadas en el desarrollo.
- Desarrollar la aplicación.
- Hacer pruebas de la funcionalidad de la aplicación.

## 1.4 Estructura de la memoria

Este documento se encuentra dividido en ocho capítulos. El primer capítulo es la introducción al proyecto donde se introduce el trabajo y se explican los problemas, la motivación y los objetivos de este trabajo.

En el segundo capítulo se describe la base teórica de la RA.

En el tercer capítulo se analiza la RA en la educación, su trayectoria y necesidades, así como la presentación de otros proyectos de RA educativa.

En el cuarto capítulo se presentan las herramientas y tecnologías utilizadas para desarrollar este trabajo.

En el quinto capítulo se describe el proceso de desarrollo que se ha seguido para la elaboración de la aplicación final.

En los capítulos sexto y séptimo se presentan las conclusiones del trabajo y se comentan las posibles mejoras que pudiese tener la aplicación.

En el octavo y último capítulo se analiza el presupuesto que tendría este proyecto para su desarrollo.

# Capítulo 2

## Base teórica

### 2.1 Historia

La realidad aumentada aparece en la década de los 50's, cuando Morton Heilig, ideó un dispositivo en el cuál mezclar sentidos reales con una experiencia virtual. A esto lo llamó “El cine del futuro”. En 1968, Ivan Sutherland, creo el primer casco con el que conseguir una experiencia de RA. Y en 1975, Myron Krueger construyo “Videoplace”, una habitación con la que poder interactuar con elementos virtuales [1].

Sin embargo, no fue hasta la década de los 90's, cuando Thomas Caudell utilizara por primera vez el término de RA.

En 2014, Thomas Caudell, en una entrevista dijo: “La RA es una tecnología que aumenta la percepción sensorial humana con información auxiliar, que puede mejorar potencialmente el rendimiento en la realización de una tarea o experiencia” [6].

### 2.2 La realidad aumentada

La RA (RA), es el conjunto de tecnologías que permite al usuario visualizar elementos virtuales superpuestos en el mundo real. Por lo tanto, se trata de una combinación de elementos reales con virtuales en tiempo real [2].

Las diferencias entre RA y realidad virtual las explica Milgram y Kishino en su diagrama. En el cual muestran en un extremo un entorno puramente real y en el otro uno puramente virtual (realidad virtual). Consideran que cualquier entorno que mezcle objetos de ambas realidades es una realidad mixta, mientras que aquellos en la que la mayor parte del contenido es virtual pero existe cierta inclusión del mundo real, se denomina virtualidad aumentada.

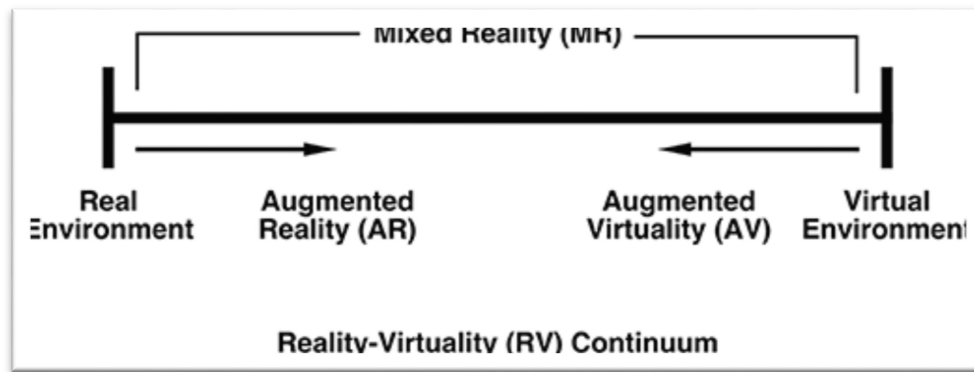


Figura 1. Diagrama de Milgram [3].

## 2.3 Como funciona la realidad aumentada

La RA debe estar formada por los siguientes elementos [2]:

- Una cámara para captar las imágenes del mundo real
- Un hardware que permita visualizar el resultado de la combinación de los elementos reales y virtuales.
- Un software que gestione el proceso de combinación.
- Un marcador, que es el elemento de la realidad física donde se proyectan los elementos virtuales. Por lo tanto este marcador es el cuál donde actúan los otros componentes, pues es el elemento que captura la cámara y bajo el cual el software genera los elementos virtuales para mostrarlos en la pantalla del hardware.

Hoy en día estos elementos los podemos tener fácilmente en un solo dispositivo, los teléfonos móviles.

## 2.4 Niveles de la realidad aumentada

Existe una clasificación de seis niveles para las aplicaciones de RA [2]:

- Nivel 0: Está directamente relacionado al mundo físico. Son enlaces representados con una simbología que permite al usuario a enlazar a otro tipo de información. Un ejemplo serían los códigos QR.
- Nivel 1: Basado en marcadores. Son imágenes en blanco y negro, que una vez escaneadas, muestran la información 3D a través de la pantalla del dispositivo.
- Nivel 2: Sin marcadores. Uso del GPS y la brújula para conseguir localizar los puntos de interés del mundo real.

- Nivel 3: Visión aumentada. La RA visualizada se convierte en visión aumentada gracias a los dispositivos tecnológicos.
- Nivel 4: Sistema de posicionamiento global. Según el GPS y las coordenadas se despliega distinta información virtual.
- Nivel 5: Se pretende que toda la superficie se convierta en táctil. Usando el calor de nuestros dedos se puede interactuar con el ambiente.

# Capítulo 3

## Análisis

### 3.1 La realidad aumentada actualmente

En la actualidad, existen multitud de aplicaciones de RA en muchos campos diferentes como: el entretenimiento, la educación, la ingeniería, la cultura, la publicidad o la medicina. Alguno de los proyectos más destacados son:

- Pokémon GO: A través de los mapas y de la función gps, mucho de los lugares emblemáticos de la realidad otorgan al jugador objetos y recompensas dentro del juego. También existen los “gimnasios” donde los jugadores compiten para hacerse con el control del lugar [7].
- Google glass: Es un dispositivo inventado por Google. Son unas gafas que llevan incorporado un proyector que, directamente al ojo proyecta una imagen virtual, superponiéndola a la realidad [8].
- Hololens: Se trata de un dispositivo diseñado por Microsoft que permite al usuario visualizar hologramas, además permite la interacción por voz, eyetracking o con sensores de movimiento [9].

### 3.2 Realidad aumentada en educación

El uso de la RA ofrece mejoras en el rendimiento de los estudiantes según el estudio “Engaging Computer Engineering Students with an Augmented Reality Software for Laboratory Exercises (2014)” [10].

Los estudiantes, sobretodo en su estancia de Primaria y Secundaria, obtienen una mejora en su aprendizaje [11]. Sin embargo, los estudiantes de infantil también se aprovechan de los beneficios de la RA debido a que esta tecnología ofrece la estimulación de las habilidades de exploración y motrices que tienen estos niños.

El uso de la RA fomenta un aprendizaje agradable, puesto que el uso de estas herramientas entran dentro del concepto de Gamificación. De este modo, este sistema de

aprendizaje ofrece interactividad, usando el sentido de la vista, el auditivo y el tacto. Además se trabaja la atención y los conceptos.

Hay que destacar que la usabilidad de estas herramientas en el aula deben ser muy sencillas para que sean experiencias positivas para el docente y los alumnos.

### **3.3 Aplicaciones de realidad aumentada para educación**

Algunas aplicaciones de RA que se utilizan o están destinadas a la educación y que se usan ya en varios centros educativos son:

- **Body Planet:** Se trata de una aplicación para aprender sobre el cuerpo humano. A través de unas tarjetas o unas camisetas la aplicación enseña a los estudiantes la anatomía. También ofrece la posibilidad de seguir el desarrollo de un embarazo a través de una camiseta [12].
- **ChromeVille:** Es una plataforma educativa que permite descargar láminas coloreables de contenido educativo. A través de su aplicación se puede observar el contenido de las láminas en 3D [13].
- **ZooKazam:** La aplicación utiliza láminas descargables para representar animales en 3D. Los usuarios pueden interactuar con los animales y estos emiten sonidos [14].
- **Metaverse:** Se trata de una plataforma que permite crear experiencias de RA personalizadas. Para ello permite enlazar varias escenas e incluir fotos, videos, modelos 3D en incluso cuestionarios [15].



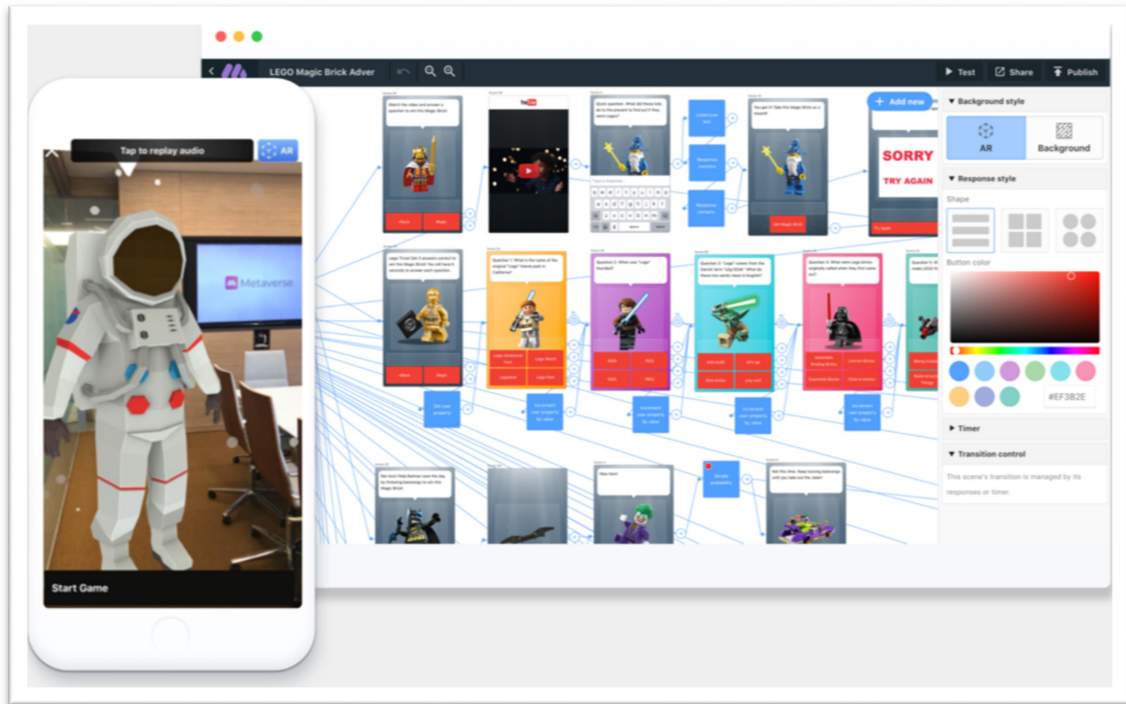


Figura 2. Interfaz Metaverse [15].

- Aumentaty: Es una comunidad de creación y visualización de contenido en RA. Se pueden crear proyectos propios de RA sin conocimientos técnicos.
- JigSpace: Es una aplicación que permite visualizar modelos en 3D e interactuar con ellos. Por ejemplo se puede visualizar el planeta Tierra y ver todas sus capas una a una.

### 3.4 Efectos sobre el aprendizaje

En el curso académico 2015-16 se realizó un estudio sobre los efectos que tiene la RA en la educación. Este estudio se llama “RA en Educación primaria: efectos sobre el aprendizaje” [5].

Este estudio consistió en la impartición de un tema con herramientas de RA. Se aplicó un pre-test y un pos-test a un grupo experimental y a uno de control, con el objetivo de averiguar si la RA influía en la adquisición de conocimientos. También se entrevistó a los alumnos y profesores sobre la experiencia.

Los resultados obtenidos indican una importante mejora en el pos-test después de haber utilizado aplicaciones de RA, el 41% de los alumnos obtuvieron una calificación de sobresaliente y no hubo ningún suspenso.

Aplicando un método tradicional se observó que los resultados obtenidos en el pos-test fueron mejores, pero las calificaciones fueron peores y en esta si hubo suspensos.

Podemos observar que la RA puede ser una buena opción a la hora de incluirla a la educación, pues ofrece un nuevo modelo de enseñanza más divertido e interactuable.

# Capítulo 4

## Herramientas de desarrollo

### 4.1 Unity

Unity [18] es un motor de videojuegos multiplataforma. Está disponible en Windows, Mac OS y Linux. Tiene dos tipos de licencias, libre y propietaria; la primera tiene algunas restricciones sobre la versión de pago limitando la calidad gráfica. Unity permite ser programado con lenguajes como C#, JavaScript o Boo.



*Figura 3. Logo de unity [18].*

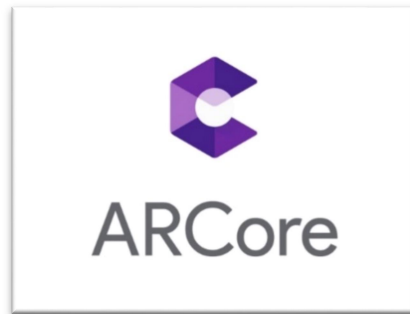
La facilidad que ofrece Unity con su licencia libre y su relativa facilidad de uso, son una de las causas por las que elegí esta herramienta para el desarrollo del trabajo. Además ofrece multitud de herramientas para el desarrollo de aplicaciones de realidad aumentada.

### 4.2 ARCore

ARCore [19], es un kit de desarrollo de software de Google para crear aplicaciones de realidad aumentada. Cabe destacar que no todos los móviles son compatibles con esta herramienta pero año a año se va extendiendo su compatibilidad. ARCore se basa en tres pilares:

- Tracking de movimiento usando para ello los sensores y el reconocimiento de la cámara.
- Comprensión ambiental: ARCore tiene la capacidad de detectar tamaños y distintas superficies y ángulos.
- Estimación de luz: Los objetos integrados permiten ser desplegados de forma más realista teniendo en cuenta la luz.

ARCore está disponible para desarrolladores en Unity, utilizando el framework AR Foundation [20] de Unity, específicamente desarrollado para aplicaciones de RA.



*Figura 4. Logo ARCore [19].*

Se ha optado por la utilización de esta herramienta debido a su fuerte potencial y a la gran capacidad que tiene de detección de planos para desplegar modelos 3D, que otros kits no disponen de estas cualidades. Además ARCore tiene una licencia completamente libre.

### **4.3 Visual Studio Community 2019**

Se trata de un entorno de desarrollo integrado para Windows y Mac OS. Es compatible con multitud de lenguajes de programación. Este IDE se integra perfectamente con Unity, por lo que se ha decidido utilizarlo para el desarrollo de este trabajo. Además la versión community es completamente gratuita.

### **4.4 Base de datos**

Para crear la base de datos se utilizó “FreeHostia” [22], un servicio que permite registrar un dominio y albergar una base de datos MySQL. Se ha contratado el dominio “classroomar.website” durante 1 año, que usaremos para alojar la base de datos y así tenerla en la nube.

### **4.5 FileZilla**

Se trata de una aplicación libre de código abierto que soporta los protocolos FTP y SFTP. Se ha utilizado para traspasar los archivos php necesarios para realizar las consultas sobre la base de datos [23].

# Capítulo 5

## Desarrollo

Como se ha explicado anteriormente, la realidad aumentada no tiene una base sólida en la educación. Existen multitud de aplicaciones destinadas a la educación y cada vez aparecen más y más. Pero no hay ninguna que se integre en las aulas. De esta idea nace este proyecto.

La aplicación consiste en una base de datos que almacena la información de alumnos y profesores. Los datos almacenados vienen a ser los integrantes de una clase, o las clases a las que un profesor imparte una asignatura.

Disponemos de una pantalla inicial de autenticación donde los usuarios podrán introducir su correo (cuenta de usuario) y su contraseña. Una vez introducida, la aplicación consultará en la nube la base de datos para comprobar si las credenciales son correctas. Si lo son, la aplicación redirigirá a una nueva pantalla dependiendo de si es alumno o profesor.

En el caso de los profesores, la aplicación llevará a un menú dinámico de selección de clases donde aparecerán todas las clases en las que imparte materia el profesor. Una vez seleccionada la clase la aplicación llevará a una pantalla donde elegir una asignatura y posteriormente un modelo 3D. Cuando el profesor haya escogido el modelo podrá acceder a la cámara que detectará superficies en las que se podrá desplegar el modelo 3D. Al pulsar sobre la pantalla se mostrará el modelo 3D en el lugar donde el usuario haya indicado, da igual que el usuario se mueva, el objeto se quedará estático en dicho lugar.

En el caso de los alumnos, después de autenticarse, la aplicación les redirigirá a una pantalla donde se muestra la clase a la que pertenecen, también indicará la información del modelo activo en el caso de que un profesor hubiese activado un modelo para esa clase. Si un modelo está activo el alumno podrá entrar en la cámara e interactuar como se detalló en el párrafo anterior.

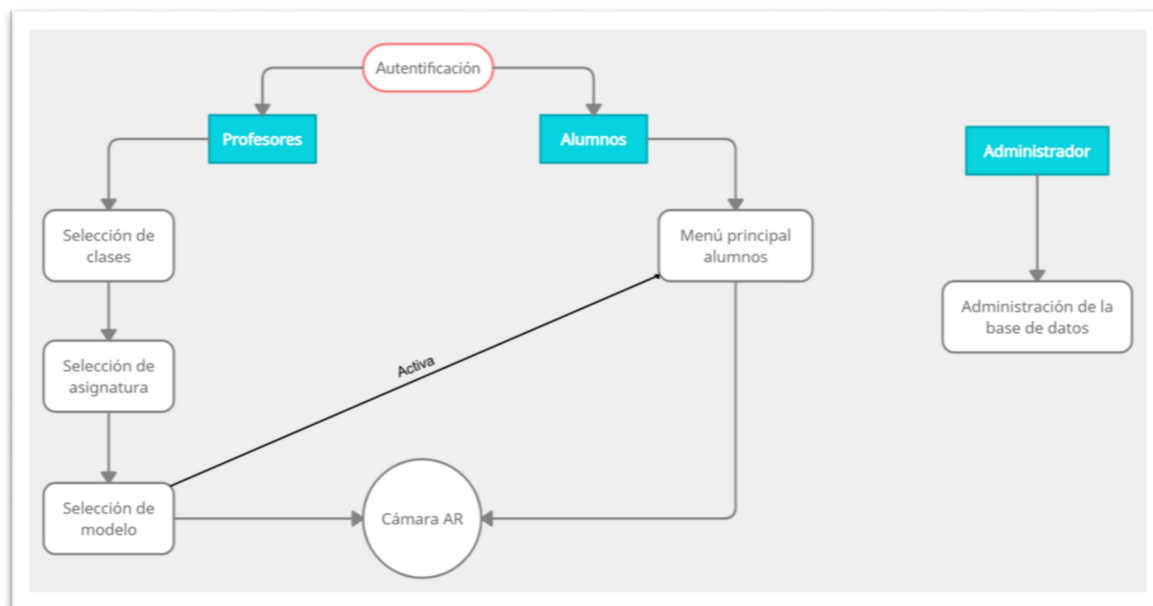


Figura 5. Esquema de uso de la aplicación.

Cabe destacar que los usuarios podrán cerrar la sesión en cualquier momento, pulsando un botón en la parte superior derecha de la pantalla. Naturalmente aparecerá un mensaje de confirmación para evitar errores y mejorar la usabilidad. También se implementó como medida de usabilidad la posibilidad de volver a una pantalla anterior pulsando el botón de retroceso del teléfono móvil.

Gracias a la aplicación el profesor podrá elegir que mostrar en cada instante lo que ven los alumnos en sus dispositivos y así organizar la materia adaptándola como lo desee.

Un centro escolar podría implementar su propia base de datos, así son los propios centros quienes tienen total control sobre los datos y uso de la aplicación.

## 5.1 Diseño de la base de datos

Para interactuar con la aplicación se ha desarrollado la siguiente base de datos:

Tabla	Función	Columnas
Usuarios	Recoge las credenciales de los usuarios	Correo, apellido1, apellido2, contraseña, esProfesor.
Profesores	Indica a que clases imparte materia un profesor	ID, correo, claseFK
Alumnos	Indica a que clases pertenece un alumno	correo, claseFK

Cursos	Indica los cursos académicos que disponibles	ID, nombre
Clases	Indica las clases de cada curso y el modelo activo que tiene la clase	ID, cursoFK, nombre, modelo

Tabla 1. Tablas de la base de datos.

Estos datos han sido alojados en un servidor en la nube. Para ello se ha contratado un dominio llamado “classroomar.website” en Freehostia, el cual ofrece la posibilidad de tener una base de datos MySQL.

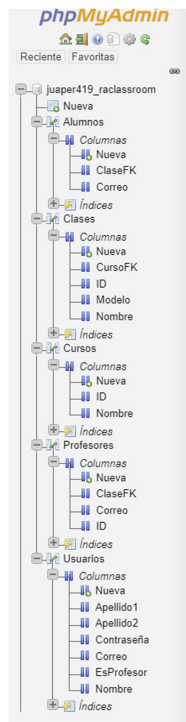


Figura 6. Tablas y columnas en phpMyAdmin.

## 5.2 Conexión con la base de datos

Para interactuar con la base de datos se ha creado un script llamado “DBQuery.cs”. Que nos permite recibir y enviar datos de la base de datos. Este script se instancia en el controlador de las escenas de Unity para poder acceder a ella.

Lo primero que se declara en la clase son las strings que contienen las URLs de los scripts php que se encuentran en el servidor. Estos scripts realizan una conexión a la base de datos y seleccionan o introducen información a través de consultas sql. También se declara la variable “requestedData” que contendrá el resultado de las consultas antes mencionadas.

La clase se divide en dos regiones, la primera contiene las corrutinas (Funciones que tienen la habilidad de pausar la ejecución y devolver el control a Unity para luego continuar donde lo dejo en el siguiente frame [25]), y la segunda (DB región tolos) se

corresponde a las funciones implementadas que permiten el correcto funcionamiento de la comunicación con la base de datos y con el servidor.

La región DB region tools tiene dos funciones:

- QuerySelectCo: Corrutina a la que llaman todas las corrutinas que necesitan operaciones con la base de datos. Después llamará a GetRequest.
- GetRequest: También se trata de una corrutina que se encarga de hacer un WebRequest con la URL que le indica QuerySelectCo.

## 5.3 Escenas de Unity

Se han implementado cinco escenas en total. Estas son:

- LoginScreen: Se trata de la pantalla principal de la aplicación. Donde el usuario insertará su correo electrónico y contraseña para autenticarse.
- SelectClass: Cuando un usuario inicia sesión, si es profesor es enviado a esta escena. Aquí se mostrará una lista de clases donde el profesor imparte asignaturas. El profesor seleccionará la clase en la que le toque estar.
- SelectModel: Una vez el profesor haya escogido la clase, puede elegir una asignatura y un modelo de esta. Cuando la active el modelo estará disponible para todos los alumnos de esa clase.
- MainStudent: Cuando un alumno inicia sesión aparecerá en esta escena. Aquí se muestran datos del alumno y si hay algún modelo activado.
- MainARCam: Se trata de la cámara de realidad aumentada. El usuario apuntará sobre una superficie y podrá colocar el modelo que el profesor ha elegido, así podrá visualizarlo.

Todas estas escenas, excepto MainARCam, tienen los siguientes componentes:

- Controlador de escena: Es el objeto que se encarga de gestionar lo que ocurre en la escena, tiene referencias al controlador de la interfaz de usuario, a la base de datos y al script SceneLogoutLoader que permite salir de la aplicación, desconectarse y volver hacia atrás.
- Controlador de la interfaz de usuario: Contiene una referencia a los componentes (botones, textos, ...) que aparecen en la escena. También se encarga de realizar cambios en la interfaz.
- EventSystem: Componente que se encarga que se pueda interactuar con los elementos de la escena y los canvas.



## 5.4 Autenticación

La primera pantalla es la de Inicio de sesión. Aquí los usuarios introducirán su correo electrónico y contraseña para acceder a la aplicación. La app envía respuestas al usuario en base a su interacción, esto lo hace con mensajes (Datos incorrectos, datos válidos y accediendo a la aplicación, ...). También se puede cerrar la aplicación pulsando el botón “atrás” del dispositivo móvil (escape).

Cuando el usuario pulsa el botón “Entrar”, se activa la función Enter() del controlador de escena. Esta comprueba que hay datos introducidos en los campos de texto, si es así desactiva el botón de “Entrar” para que no se realicen múltiples peticiones y llama a la función Login(), que se encarga de conectar a la base de datos.

La función Login(), llama a la función GetLogin(correo, contraseña) de DBQuery, y esta, a su vez, convoca a la corrutina Login(correo, contraseña) que hace una petición a la base de datos. Si la respuesta es negativa envía un mensaje en el que indica que los datos son incorrectos, pero si es afirmativo distingue entre profesor y alumno y le redigirá a una escena diferente. La estructura de la corrutina es la siguiente:

```
IEnumerator Login(string loginInput, string passwordInput)
{
    yield return StartCoroutine(QuerySelectCo(getUsuariosURL + "?loginUser=" + loginInput +
"&loginPass=" + passwordInput));
    print("Response: " + requestedData);

    if (requestedData == "error")
        yield break;

    if (requestedData == "-1")
    {
        LoginSceneController.instance.uiController.StateMessage.text = "Datos incorrectos";
        yield break;
    }

    if (requestedData == "1") // User is "profesor"
    {
        LoginSceneController.instance.uiController.StateMessage.text = "Entrando...";
        SceneManager.LoadScene("SelectClass");
        User.esProfesor = true;
    }

    if (requestedData == "0") //User is "alumno"
    {
        LoginSceneController.instance.uiController.StateMessage.text = "Entrando...";
        StartCoroutine(GetAlumno(loginInput));
        User.esProfesor = false;
    }
}
```

```
//Save User Mail
```

```
User.mail = loginInput;
```

User es una clase que no hace referencia a ningún objeto, pero se encarga de guardar datos del usuario, que serán consultados por la aplicación constantemente.



*Figura 7. Pantalla de inicio de sesión*

## 5.5 Selección de clases y cursos

Cuando un profesor entra en la aplicación aparece en esta pantalla. Aquí estarán todos los cursos y clases donde el profesor imparta una asignatura. Podrá hacer click en la clase deseada y la aplicación le llevará a la siguiente pantalla de selección de modelos.

Hay que destacar, que de esta escena, lo complicado ha sido realizar el menú dinámico. Donde los cursos despliegan las clases que contienen. Para realizar esto, se introdujo un panel contenedor de todos los botones de cursos y clases dentro de un scroll. Los botones de curso y clase se irán generando a medida que sea necesario.

Cuando se inicia la escena, lo primero que realiza es una consulta a la base de datos a través de la corrutina `GetClasesProf(mailProfesor)` de `DBQuery`. Una vez obtenidas las clases en las que imparte asignatura el profesor, el controlador de interfaz de usuario llama al método `Initialize(datos)` que parseará la entrada y clonará tantas veces como sea necesario los botones.

```

public void Initialize(string clasesCurses)
{
    // Split the content in lines
    string[] l_lines = clasesCurses.Split('\n');//Full Line

    string lastCurso = "";
    CursoElement currentCurso = null;
    //Create available test objects
    //split lines in Test Objects
    for (int i = 0; i < l_lines.Length - 1; i++)
    {

        string[] l_testSplit = l_lines[i].Split('*');

        //Curso
        if (lastCurso != l_testSplit[1])
        {
            lastCurso = l_testSplit[1];

            //Clone prefab curso
            currentCurso = Instantiate(prefabCurso, prefabCurso.transform.parent);
            currentCurso.cursoName.text = lastCurso;
            currentCurso.gameObject.SetActive(true);
        }

        //Clone prefab clase
        ClaseElement currentClase = Instantiate(prefabClase, prefabClase.transform.parent);
        currentClase.claseName.text = currentCurso.cursoName.text + "-" +
l_testSplit[2];//Clase
        currentCurso.clases.Add(currentClase);//Add to curso elemt class object

    }

}

```

Posteriormente, cada botón realizará su función correspondiente, los cursos guardan una lista de botones de clases, que se desplegará cuando se haga click encima. Y se encogerá cuando se vuelva a hacer click. Mientras que las clases redirigen a la siguiente escena, guardando la clase escogida en User.

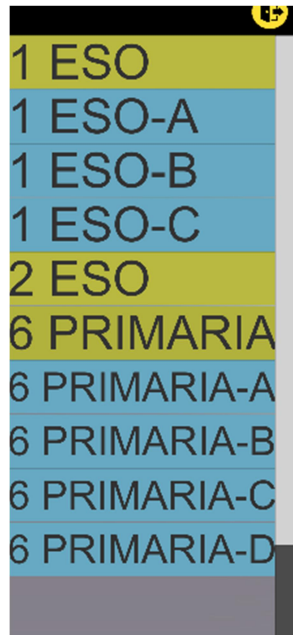


Figura 8. Pantalla de selección de clases.

## 5.6 Selección de asignatura y modelos

En esta escena los profesores elegirán una asignatura, la cual activará el selector de modelos en el que podrá escoger uno y activarlo. Cuando un modelo está activo, se guardará en la base de datos y estará disponible para todos los alumnos de esa clase. El profesor también podrá acceder a la cámara de RA desde esta pantalla.

Al principio, solo el selector de asignatura estará activado. Solamente se activará el selector de modelo cuando una materia se haya escogido. Tras elegir una asignatura y un modelo, se hará interactuable el botón de activar, el cuál enviará una petición a la base de datos a través de corrutinas. Cuando un modelo este activo, el botón de la cámara de realidad aumentada estará disponible y se podrá acceder a ella. El siguiente código muestra como funciona el botón activar:

```
public void Activate()
{
    string curso = smUIController.classText.text.Split('-')[0];
    string clase = smUIController.classText.text.Split('-')[1];

    if (smUIController.ddmodels.captionText.text == "Ninguno")
    {
        dbquery.SetClassModelNullDB(clase, curso);
        smUIController.btnAR.interactable = false;
    } else
    {
```

```

    dbquery.SetClassModelDB(smUIController.ddmodels.captionText.text, clase, curso);
    User.classmodel = smUIController.ddmodels.captionText.text;
    smUIController.btnAR.interactable = true;
}
}

```



*Figura 9. Escena de selección de asignatura y modelo 3D*

## 5.7 Cámara de realidad aumentada

Para que funcione la RA en la aplicación dependemos de ARCore que nos proporciona las herramientas necesarias para crearla. Para empezar, es necesario incluir los paquetes en “Window > Package Manager”. Se deben incluir los paquetes “AR Foundation”, “AR XR Plugin” y “ARKit XR Plugin”. Una vez se han añadido, se procede a insertar los componentes de la escena, que son:

- AR Session Origin: Es el padre de una aplicación de RA. Contiene la cámara y cualquier objeto creado a partir de entidades detectadas como pueden ser los planos y las nubes de puntos. Contiene los scripts ARPlaneManager, que se encarga de crear, actualizar y eliminar planos. Y ARRaycastManager que se encarga de manejar los Ray (lugar donde se hace click en la pantalla y calculado desde la cámara).
- AR Session: Controla el ciclo de vida y la configuración de opciones de AR. También contiene el script ARInputManager que permite obtener la información

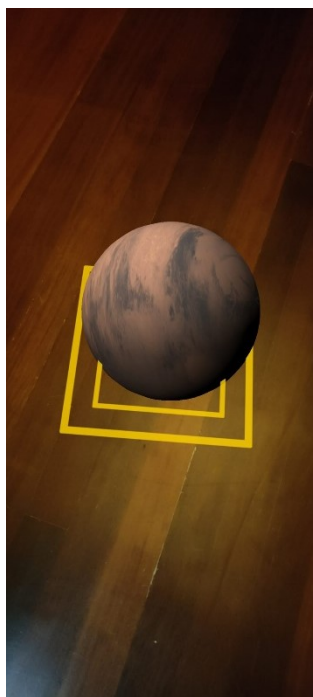
de la posición 3D del dispositivo.

- Directional Light: Simula la luz del sol, por lo tanto ilumina en totalidad la escena desde un punto.
- EventSystem: Componente que se encarga que se pueda interactuar con los elementos de la escena y los canvas.
- Indicator Controller: La detección de planos está oculta para presentar un diseño más limpio en la cámara. En su lugar se ha implementado un indicador, que como su nombre señala, indica en que lugar podrá ser colocado el modelo 3D.
- Spawn Controller: Se encarga de generar los objetos 3D. El script conoce cuando el usuario hace click en la pantalla. De esta manera puede poner y quitar objetos 3D.
- Scene Controller: La función de este componente es de controlar la escena. Escucha cuando el usuario pulsa escape para ir a la escena anterior.

Cuando se inicia la cámara, esta buscará superficies, cuando las encuentre mostrará el indicador. Al hacer click en la pantalla, aparecerá el modelo que haya elegido el profesor. Cuando se vuelve a hacer click, se destruye el objeto por si se quiere cambiar de lugar. Los objetos se crean de la siguiente manera:

```
void Start()
{
    pcm = FindObjectOfType<PlacementManager>();
    objectToSpawn = Resources.Load<GameObject>(User.classmodel);
}

// Update is called once per frame
void Update()
{
    if (Input.touchCount > 0 && Input.touches[0].phase == TouchPhase.Began)
    {
        if (created)
        {
            Destroy(obj);
            created = false;
        }
        else
        {
            obj = Instantiate(objectToSpawn, pcm.transform.position, pcm.transform.rotation);
            created = true;
        }
    }
}
```



*Figura 10. Colocación de un modelo 3D en RA.*

## 5.8 Menú principal estudiantes

Cuando un estudiante inicia sesión, será llevado a esta escena, donde se muestran los datos del estudiante, y si hay activo un modelo o no. Si lo hay, el botón de acceso a la cámara RA estará disponible. Esta pantalla se actualizará constantemente, así si un profesor activa un modelo nuevo, o lo desactiva, los alumnos lo recibirán, garantizando así que los alumnos visibilicen lo que el profesor desea.

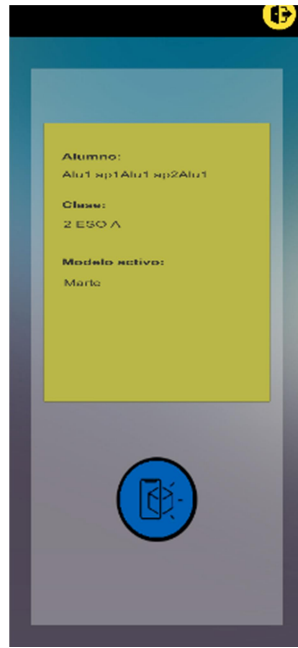
Para que el modelo activo se actualice constantemente, se utiliza la función `InvokeRepeating()` en el `Start()` del controlador de escena. Se le indica que llame a la base de datos cada 5 segundos. Así se mantendrá actualizado.

```
void Start()
{
    uiController.nameText.text = User.name + " " + User.lastname1 + " " + User.lastname2;
    uiController.classroomText.text = User.classroom;

    InvokeRepeating("GetState", 0 , 5);
}

void GetState()
{
    dbquery.GetModeloFromDB(User.mail);
    print(User.classmodel);
    if (User.classmodel == null || User.classmodel == "")
    {
```

```
    uiController.btnAR.interactable = false;
  }
  else
  {
    uiController.btnAR.interactable = true;
  }
}
```



*Figura 11. Escena principal de estudiantes.*



# Capítulo 6

## Conclusiones y líneas futuras

Como conclusión se ha de resaltar el importante aprendizaje que se ha tenido que hacer para la realización de este trabajo, pues las herramientas utilizadas no se conocían o se tenían escasos conocimientos sobre ellas. Aún con estas dificultades, se considera que se han logrado los objetivos deseados y con ello se ha conseguido un gran crecimiento personal y académico.

Durante el desarrollo del proyecto se hicieron algunos pequeños cambios en el planteamiento de la aplicación debido a un mal planteamiento inicial. Sin embargo se ha considerado que se han logrado cumplir los objetivos principales y completando de esta manera una aplicación funcional. De todas formas cabría mencionar que habría muchas funcionalidades que implementar a futuro para que esta aplicación pueda llegar a convertirse en una herramienta potente en el ámbito de la educación y la realidad aumentada.

Una de las mejoras que podría tener la aplicación es ampliar la base de datos, donde se puedan incluir diferentes asignaturas y sus modelos asociados. Esto implicaría que los modelos estuviesen en la nube y fuesen cargados cuando fuese necesario.

Otra mejora podría ser la implementación de funcionalidades que permitan a los profesores obtener mejores resultados al uso de la realidad aumentada e implicarla directamente en la clase. Como puede ser el poder visualizar datos de los alumnos, como notas obtenidas al crear cuestionarios dentro de la aplicación o poder visualizar el tiempo que pasan los alumnos dentro de la herramienta.

Con estas mejoras un centro educativo podría tener su propia base de datos y gestionar la herramienta ellos mismos sin terceros, sin ceder datos y adaptándola a sus necesidades. Como está la aplicación ahora puede servir para beneficiarse de la RA en la educación y obtener unas clases más dinámicas, además se le da el poder al profesor para que los alumnos puedan acceder solo a lo indicado.

# Capítulo 7

## Summary and Conclusions

As a conclusion, it is necessary to highlight the important learning that had to be done for the achievement of this assignment, due to the tools used were not known or the knowledge about them was limited. In spite of these difficulties, it is considered that the desired objectives have been accomplished and with that it has achieved a personal and an academic growth.

During the development of the project it had to be applied small changes in terms of the approach of the application because of an incorrectly initial proposal. Nevertheless, it has to be considered that the main objectives have been completed successfully, reaching in this way a functional application. In the same way, it is pertinent to mention that it would be necessary to implement several functionalities in the future for this application to become a powerful tool with regard to education and augmented reality.

One of the upgrades that could have the application is to expand the database, where different subjects and their associated models could be included. This would imply that the models would be in the cloud and would be charged when necessary.

Another improvement could be the implementation of functionalities that allows the teachers to obtain better results and to involve the class because of the augmented reality. As it can be visualizing the student's data, marks obtained in quizzes from the application, or even to view how much time the students are using the tool.

With these developments an educational centre could have their own databases and manage the device by themselves without third parties, without transferring data and adapting the tool to their own necessities. As how is the application created at this moment, can be beneficial for the augmented reality in an educational way and for getting dynamic lessons, furthermore gives the teacher the possibility to manage the content them want their students to see.

# Capítulo 8

## Presupuesto

En la tabla siguiente se muestran las horas invertidas en cada tarea y el precio de estas, teniendo como referencia que el coste de la mano de obra es de 18€ la hora. Cabe resaltar que para la creación de esta aplicación no es necesario el pago de licencias, puesto que podemos trabajar con la licencia gratuita de Unity, y ARCore no necesita licencia.

Tarea	Horas	Coste
Investigación	25	450 €
Creación de la base de datos	2	36 €
Conexión con la base de datos	20	360 €
Implementación de la aplicación	100	1.800 €
Diseñar la interfaz de usuario	10	180 €
Elaboración de modelos 3D	50	900 €
Corrección de errores	30	540 €
Total	237	4.266 €

*Tabla 2. Presupuesto de la elaboración de la aplicación.*

También hay que destacar que es necesario tener la base de datos en la nube. Si se puede configurar un servidor con una red interna el coste sería 0, pero si hay que contratar un hosting externo que aumentaría el coste mensualmente. Este tipo de hosting rondan los 15€ mensuales. Además hay que destacar que si se cogen modelos 3D gratuitos de la web el coste obviamente disminuiría, pero depende de la cantidad que necesiten elaborarse también podría aumentar.

# Bibliografía

1. Furht, B. Handbook of augmented reality. Springer Science and Business Media, 2011.
2. Realidad aumentada [https://es.wikipedia.org/wiki/Realidad\\_aumentada](https://es.wikipedia.org/wiki/Realidad_aumentada)
3. Richard Skarbez, Missie Smith y Mary C. Whitton. Revisiting Milgram and Kishino's Reality-Virtuality Continuum. Marzo 2021. <https://www.frontiersin.org/articles/10.3389/frvir.2021.647997/full>
4. Prendes, C. "Realidad aumentada y educación: Análisis de experiencias prácticas" Pixel-Bit. Revista de Medios y Educación. N°46. Enero 2015  
<https://idus.us.es/bitstream/handle/11441/45413/realidad%20aumentada%20y%20educacion.pdf?sequence=1>
5. Toledo Morales, P y Sánchez García, J. (2017) Revista Latinoamericana de Tecnología Educativa, 16. Junio 2017.
6. Reinoso, R. (2018) Realidad aumentada y virtual: Descubriendo sus posibilidades en educación y formación. <https://www.slideshare.net/tecnotic>
7. Pokémon Go. <https://pokemongolive.com/es/>
8. Google glass. <https://www.google.com/glass/start/>
9. Hololens, <https://www.microsoft.com/es-es/hololens>
10. J. R. L. Benito, E. A. González, M. Anastassova and F. Souvestre, "Engaging computer engineering students with an augmented reality software for laboratory exercises," *2014 IEEE Frontiers in Education Conference (FIE) Proceedings*, 2014, pp. 1-4, doi: 10.1109/FIE.2014.7044094.
11. Jorge Calvo. Lo que aporta la realidad aumentada en la educación. Educación 3.0. Abril 2021.  
<https://www.educacionrespuntocero.com/tecnologia/realidad-aumentada-educacion/>
12. Body Planet. <https://bodyplanet.es/>
13. ChromeVille. <https://chromville.com/>
14. ZooKazam. <http://www.zookazam.com/#home>
15. Metaverse. <https://studio.gometa.io/landing>
16. Aumentaty. <http://www.aumentaty.com/community/es/>
17. JigSpace. <https://jig.space/>
18. Unity. <https://unity.com/es>
19. Google ARCore. <https://developers.google.com/ar>

20. AR Foundation. <https://unity.com/unity/features/arfoundation>
21. Microsoft. Visual Studio Community 2019. <https://visualstudio.microsoft.com/es/vs/>
22. FreeHostia. <http://freehostia.com>
23. FileZilla. <https://filezilla-project.org/>
24. Modelos 3D usados para la elaboración de la aplicación. <https://free3d.com/es/>
25. Corrutinas de Unity. <https://docs.unity3d.com/es/2018.4/Manual/Coroutines.html>
26. Michael Soler Beatty. Crear Videojuego para ARCore realidad aumentada en Unity AR Tutorial #1. <https://www.youtube.com/watch?v=taWkX5t0v9E>

# Anexos

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

/// <summary>
/// Scene controller for Login
/// </summary>
public class LoginSceneController : MonoBehaviour
{
    public LoginUIController uiController;
    public DBQuery dbController;

    public static LoginSceneController instance;

    private void Awake()
    {
        instance = this;
    }

    /// <summary>
    /// Button Entrar function
    /// </summary>
    public void Enter()
    {
        uiController.StateMessage.text = "";

        if (string.IsNullOrEmpty(uiController.loginInput.text) ||
            string.IsNullOrEmpty(uiController.passwordInput.text))
        {
            uiController.StateMessage.text = "ERROR: Faltan datos de autenticación";
            return;
        }

        uiController.enterButton.interactable = false;
        Login();
        uiController.enterButton.interactable = true;
    }

    /// <summary>
    /// Call to the database
    /// </summary>
    private void Login()
    {
        dbController.GetLogin(uiController.loginInput.text, uiController.passwordInput.text);
    }

    private void Update()
    {
        if (Input.GetKey(KeyCode.Escape))
        {
            Application.Quit();
        }
    }
}
```

```

using System.Collections.Generic;
using UnityEngine;
using System.Net;
using System.IO;
using System;
using UnityEngine.Networking;
using UnityEngine.Rendering;
using System.Collections;
using UnityEngine.SceneManagement;

/// <summary>
/// This class is used to communicate with the database
/// </summary>
public class DBQuery : MonoBehaviour
{
    const string port = ":443";

    #if (UNITY_EDITOR)
        const string baseURL = "http://classroomar.website/php/";
    # else//AR APP
        const string baseURL = "http://classroomar.website/php/";
    #endif

    //Get data PHP: SELECT
    string getUsuariosURL = baseURL + "Login.php";
    string getAlumnoURL = baseURL + "CheckAlumnos.php";
    string getClaseURL = baseURL + "CheckCursoByClase.php";
    string getClaseProfURL = baseURL + "CheckClasesProf.php";
    string setModel = baseURL + "SetModel.php";
    string setModelNull = baseURL + "SetModelNull.php";
    string getModelURL = baseURL + "CheckModel.php";

    //Loaded Data
    static string requestedData;

    /// <summary>
    /// Function that is responsible for calling the Login coroutine
    /// </summary>
    /// <param name="loginInput"> Users input login </param>
    /// <param name="passwordInput"> Users input password </param>
    public void GetLogin(string loginInput, string passwordInput)
    {
        StartCoroutine(Login(loginInput, passwordInput));
    }

    /// <summary>
    /// Corroutine that makes the Login request to the database
    /// </summary>
    /// <param name="loginInput"> Users input login</param>
    /// <param name="passwordInput"> Users input password </param>
    /// <returns></returns>
    IEnumerator Login(string loginInput, string passwordInput)
    {
        yield return StartCoroutine(QuerySelectCo(getUsuariosURL + "?loginUser=" + loginInput +
"&loginPass=" + passwordInput));
        print("Response: " + requestedData);

        if (requestedData == "error")
            yield break;

        if (requestedData == "-1")
        {
            LoginSceneController.instance.uiController.StateMessage.text = "Datos incorrectos";
            yield break;
        }

        if (requestedData == "1" // User is "profesor"
        {
            LoginSceneController.instance.uiController.StateMessage.text = "Entrando...";
            SceneManager.LoadScene("SelectClass");
            User.esProfesor = true;
        }

        if (requestedData == "0" //User is "alumno"
        {
            LoginSceneController.instance.uiController.StateMessage.text = "Entrando...";
            StartCoroutine(GetAlumno(loginInput));
            User.esProfesor = false;
        }

        //Save User Mail
        User.mail = loginInput;
    }
}

```



```

/// <summary>
/// Corroutine that is responsible for obtaining student data
/// </summary>
/// <param name="loginInput"> Users login input </param>
/// <returns></returns>
IEnumerator GetAlumno (string loginInput)
{
    yield return StartCoroutine(QuerySelectCo(getAlumnoURL + "?loginUser=" + loginInput));
    print("Response: " + requestedData);

    if (requestedData == "error")
        yield break;

    string[] l_lines = requestedData.Split('*');//Full Line

    User.name = l_lines[0];
    User.lastname1 = l_lines[1];
    User.lastname2 = l_lines[2];
    StartCoroutine(GetClase(l_lines[3]));
}

/// <summary>
/// Corroutine called from GetAlumno to get the class where the student is
/// </summary>
/// <param name="claseID"> class identifier </param>
/// <returns></returns>
IEnumerator GetClase (string claseID)
{
    yield return StartCoroutine(QuerySelectCo(getClaseURL + "?idClase=" + claseID));
    print("Response: " + requestedData);

    if (requestedData == "error")
        yield break;

    string[] l_lines = requestedData.Split('*');//Full Line

    User.classroom = l_lines[0] + " " + l_lines[1];
    SceneManager.LoadScene("MainStudent");
}

```

```

/// <summary>
/// Function that calls the GetModelo coroutine
/// </summary>
/// <param name="userMail"> User mail </param>
public void GetModeloFromDB (string userMail)
{
    StartCoroutine(GetModelo(userMail));
}

/// <summary>
/// Corroutine that is responsible for returning the active model to the student
/// </summary>
/// <param name="userMail"> User mail </param>
/// <returns></returns>
IEnumerator GetModelo (string userMail)
{
    yield return StartCoroutine(QuerySelectCo(getModelURL + "?loginUser=" + userMail));
    print("Response: " + requestedData);

    if (requestedData == "error")
        yield break;

    User.classmodel = requestedData;
}

/// <summary>
/// Function that calls the GetClasesProf coroutine
/// </summary>
/// <param name="userMail"> User mail </param>
public void GetClasesProfFromDB(string mailProf)
{
    StartCoroutine(GetClasesProf(mailProf));
}

/// <summary>
/// Corroutine to obtain the courses where a teacher teaches
/// </summary>
/// <param name="mailProf"> User mail </param>
/// <returns></returns>
IEnumerator GetClasesProf(string mailProf)
{
    yield return StartCoroutine(QuerySelectCo(getClaseProfURL + "?mailProf=" + mailProf));
    print("Response: " + requestedData);

    if (requestedData == "error")
        yield break;

    SelectClassSceneController.instance.scUIController.Initialize(requestedData);
}

/// <summary>
/// Function that calls the SetClassModel coroutine
/// </summary>
/// <param name="model"> Active model </param>
/// <param name="clase"> class </param>
/// <param name="curso"> course </param>
public void SetClassModelDB(string model,string clase, string curso)
{
    StartCoroutine(SetClassModel(model, clase, curso));
}

/// <summary>
/// Corroutine that is responsible for saving the active model in the database
/// </summary>
/// <param name="model"> Active model </param>
/// <param name="clase"> class </param>
/// <param name="curso"> course </param>
/// <returns></returns>
IEnumerator SetClassModel(string model, string clase, string curso)
{
    yield return StartCoroutine(QuerySelectCo(setModel + "?newModel=" + model + "&classSelected=" +
clase + "&courseSelected=" + curso));
    print("Response: " + requestedData);

    if (requestedData == "error")
        yield break;
}

```

```

/// <summary>
/// Function that calls the SetClassModelNull coroutine
/// </summary>
/// <param name="clase"> class </param>
/// <param name="curso"> course </param>
public void SetClassModelNullDB(string clase, string curso)
{
    StartCoroutine(SetClassModelNull(clase, curso));
}

/// <summary>
/// Corroutine that is responsible for removing an active model from a class
/// </summary>
/// <param name="clase"> class </param>
/// <param name="curso"> course </param>
/// <returns></returns>
IEnumerator SetClassModelNull(string clase, string curso)
{
    yield return StartCoroutine(QuerySelectCo(setModelNull + "?classSelected=" + clase +
"&courseSelected=" + curso));
    print("Response: " + requestedData);

    if (requestedData == "error")
        yield break;
}

#region DB Tools

/// <summary>
/// Corroutine that connects and sends webrequest to the database
/// </summary>
/// <param name="uri"> database url </param>
/// <returns></returns>
static IEnumerator GetRequest(string uri) {
    print("Get Request : " + uri);
    UnityWebRequest uwr = UnityWebRequest.Get(uri);
    uwr.method = "POST";
    uwr.timeout = 1500;
    yield return uwr.SendWebRequest();

    if (uwr.isNetworkError) {
        Debug.Log("Error While Sending: " + uwr.error);
        requestedData = "error";
        print("NetworkError: " + uwr.error);
    } else {
        //Data Loaded
        requestedData = uwr.downloadHandler.text;
        print("NetworkOK! ");
    }
}

/// <summary>
/// WEB GL system for select Queries.
/// </summary>
/// <param name="p_pathPhp"> database url </param>
/// <returns></returns>
internal IEnumerator QuerySelectCo(string p_pathPhp)
{
    Debug.Log("Query:" + p_pathPhp);
    yield return StartCoroutine(GetRequest(p_pathPhp));
}

#endregion
}

```



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

/// <summary>
/// UI Controller for Login
/// </summary>
public class LoginUIController : MonoBehaviour
{
    public InputField loginInput;
    public InputField passwordInput;
    public Button enterButton;
    public Text StateMessage;
}
```

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

/// <summary>
/// Scene Controller for Selector class
/// </summary>
public class SelectClassSceneController : MonoBehaviour
{

    public DBQuery dbquery;
    public SelectClassUIController scUIController;
    public SceneLogoutLoader slogoutLoader;

    public static SelectClassSceneController instance;
    private void Awake()
    {
        instance = this;
    }

    // Start is called before the first frame update
    void Start()
    {
        dbquery.GetClassesProfFromDB(User.mail);
    }
}
```

```

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

/// <summary>
/// UI Controller for Selector Class
/// </summary>
public class SelectClassUIController : MonoBehaviour
{
    public CursorElement prefabCurso;
    public ClaseElement prefabClase;

    // Start is called before the first frame update
    void Start()
    {
        prefabCurso.gameObject.SetActive(false);
        prefabClase.gameObject.SetActive(false);
        //Initialize("1*1 ESO*A/1*1 ESO*B/");
    }

    /// <summary>
    /// Parse the response from the database and dynamically create buttons
    /// </summary>
    /// <param name="clasesCurses"> database response </param>
    public void Initialize(string clasesCurses)
    {
        //Recorrer los datos:
        // Split the content in lines
        string[] l_lines = clasesCurses.Split('/');//Full Line

        string lastCurso = "";
        CursorElement currentCurso = null;
        //Create available test objects
        //split lines in Test Objects
        for (int i = 0; i < l_lines.Length - 1; i++)
        {

            string[] l_testSplit = l_lines[i].Split('*');

            //Curso
            if (lastCurso != l_testSplit[1])
            {
                lastCurso = l_testSplit[1];

                //Clonamos el Prefab de curso
                currentCurso = Instantiate(prefabCurso, prefabCurso.transform.parent);
                currentCurso.cursoName.text = lastCurso;
                currentCurso.gameObject.SetActive(true);
            }

            //Clonamos el Prefab de clase
            ClaseElement currentClase = Instantiate(prefabClase, prefabClase.transform.parent);
            currentClase.claseName.text = currentCurso.cursoName.text + "-" + l_testSplit[2];//Clase
            currentCurso.clases.Add(currentClase);//Add to curso elemnt class object
        }
    }
}

```

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.SceneManagement;

/// <summary>
/// Scene controller for Selector model
/// </summary>
public class SelectModelSceneController : MonoBehaviour
{
    public DBQuery dbquery;
    public SelectModelUIController smUIController;

    /// <summary>
    /// Write the active model to the database
    /// </summary>
    public void Activate()
    {
        string curso = smUIController.classText.text.Split('-')[0];
        string clase = smUIController.classText.text.Split('-')[1];

        if (smUIController.ddmodels.captionText.text == "Ninguno")
        {
            dbquery.SetClassModelNullDB(clase, curso);
            smUIController.btnAR.interactable = false;
        } else
        {
            dbquery.SetClassModelDB(smUIController.ddmodels.captionText.text, clase, curso);
            User.classmodel = smUIController.ddmodels.captionText.text;
            smUIController.btnAR.interactable = true;
        }
    }

    /// <summary>
    /// Enter to AR Camera
    /// </summary>
    public void EnterARCam()
    {
        SceneManager.LoadScene("MainARCam");
    }
}

```

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.UI;

public class SelectModelUIController : MonoBehaviour
{
    public Dropdown ddcourses;
    public Dropdown ddmodels;

    public Button btnAR;
    public Button btnActivate;

    public Text classText;

    /// <summary>
    /// UI Controller for Selector Model
    /// </summary>
    private void Start()
    {
        ddmodels.interactable = false;
        classText.text = User.selectedClass;
        btnActivate.interactable = false;
        btnAR.interactable = false;
    }

    /// <summary>
    /// Initialize combo boxes
    /// </summary>
    /// <param name="val"></param>
    public void DropdownCourses (int val)
    {
        if (val == 0)
        {
            ddmodels.interactable = false;
            ddmodels.options.Clear();
            ddmodels.captionText.text = "";
            btnActivate.interactable = false;
        }
        else if (val == 1)
        {
            ddmodels.options.Clear();
            ddmodels.interactable = true;
            List<string> items = new List<string>();
            items.Add("Ninguno");
            items.Add("Cubo");
            items.Add("Cilindro");
            items.Add("Esfera");

            foreach(var item in items)
            {
                ddmodels.options.Add(new Dropdown.OptionData() { text = item });
            }
            ddmodels.captionText.text = "Ninguno";
            ddmodels.value = 0;
            btnActivate.interactable = true;
        }
        else if (val == 2)
        {
            ddmodels.options.Clear();
            ddmodels.interactable = true;
            List<string> items = new List<string>();
            items.Add("Ninguno");
            items.Add("Marte");
            items.Add("Procesador");
            items.Add("Celula");

            foreach (var item in items)
            {
                ddmodels.options.Add(new Dropdown.OptionData() { text = item });
            }
            ddmodels.captionText.text = "Ninguno";
            ddmodels.value = 0;
            btnActivate.interactable = true;
        }
    }
}

```



```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.SceneManagement;

/// <summary>
/// Scene Controller of students main screen
/// </summary>
public class MainStudentSceneController : MonoBehaviour
{
    public DBQuery dbquery;
    public MainStudentUIController uiController;

    // Start is called before the first frame update
    void Start()
    {
        uiController.nameText.text = User.name + " " + User.lastname1 + " " + User.lastname2;
        uiController.classroomText.text = User.classroom;

        InvokeRepeating("GetState", 0, 5);
    }

    /// <summary>
    /// Periodically update the active model
    /// </summary>
    void GetState()
    {
        dbquery.GetModeloFromDB(User.mail);
        print(User.classmodel);
        if (User.classmodel == null || User.classmodel == "")
        {
            uiController.btnAR.interactable = false;
        }
        else
        {
            uiController.btnAR.interactable = true;
        }
    }

    /// <summary>
    /// Enter to AR Camera
    /// </summary>
    public void EnterARCam()
    {
        SceneManager.LoadScene("MainARCam");
    }
}

```

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

/// <summary>
/// UI Controller of studens main screen
/// </summary>
public class MainStudentUIController : MonoBehaviour
{
    public Text nameText;
    public Text classroomText;
    public Text activeModelText;

    public Button btnLogout;
    public Button btnAR;

    private void Start()
    {
        if (User.classmodel == null || User.classmodel == "")
        {
            btnAR.interactable = false;
        }
        else
        {
            btnAR.interactable = true;
        }
    }

    // Update is called once per frame
    void Update()
    {
        activeModelText.text = User.classmodel;
    }
}
```

```

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

/// <summary>
/// Users data
/// </summary>
public static class User
{
    //For all
    public static string mail;
    public static bool esProfesor;
    public static string classmodel;

    //Only for proffesors
    public static string selectedClass;

    //Only for students
    public static string name;
    public static string lastname1;
    public static string lastname2;
    public static string classroom;

    /// <summary>
    /// Funtion for clear data from User
    /// </summary>
    public static void clearData()
    {
        mail = null;
        selectedClass = null;
        name = null;
        lastname1 = null;
        lastname2 = null;
        classroom = null;
        classmodel = null;
    }
}
}

```

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.SceneManagement;

/// <summary>
/// Class that controls the output of the scene
/// </summary>
public class SceneLogoutLoader : MonoBehaviour
{
    [SerializeField] GameObject exitPanel;
    public void Logout()
    {
        if (exitPanel)
        {
            exitPanel.SetActive(true);
        }
    }

    /// <summary>
    /// Logout fuction
    /// </summary>
    /// <param name="choice"></param>
    public void onUserClickYesNo(int choice) //choice==0 No,choice==1 yes
    {
        if (choice == 1)
        {
            User.clearData();
            SceneManager.LoadScene("LoginScreen");
        }
        exitPanel.SetActive(false);
    }

    private void Update()
    {
        if (Input.GetKey(KeyCode.Escape))
        {
            Scene currentScene = SceneManager.GetActiveScene();
            string sceneName = currentScene.name;
            if (sceneName == "MainStudent" || sceneName == "SelectClass")
            {
                Logout();
            }
            else if (sceneName == "SelectModel")
            {
                SceneManager.LoadScene("SelectClass");
            }
        }
    }
}

```

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.XR.ARFoundation;
using UnityEngine.XR.ARSubsystems;

/// <summary>
/// Place the indicator on the AR Camera
/// </summary>
public class PlacementManager : MonoBehaviour
{
    public ARRaycastManager raymanager;
    public GameObject indicatorObject;

    // Start is called before the first frame update
    void Start()
    {
        raymanager = FindObjectOfType<ARRaycastManager>();
        indicatorObject = this.transform.GetChild(0).gameObject;
    }

    // Update is called once per frame
    void Update()
    {
        List<ARRaycastHit> hitpoint = new List<ARRaycastHit>();
        raymanager.Raycast(new Vector2(Screen.width / 2, Screen.height / 2), hitpoint,
TrackableType.Planes);
        if (hitpoint.Count > 0)
        {
            transform.position = hitpoint[0].pose.position;
            transform.rotation = hitpoint[0].pose.rotation;
            if (!indicatorObject.activeInHierarchy)
            {
                indicatorObject.SetActive(true);
            }
        }
    }
}

```

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.EventSystems;

/// <summary>
/// Place 3D models
/// </summary>
public class SpawnManager : MonoBehaviour
{
    public GameObject objectToSpawn;
    public PlacementManager pcm;

    private bool created = false;

    private GameObject obj;

    // Start is called before the first frame update
    void Start()
    {
        pcm = FindObjectOfType<PlacementManager>();
        objectToSpawn = Resources.Load<GameObject>(User.classmodel);
    }

    // Update is called once per frame
    void Update()
    {
        if (Input.touchCount > 0 && Input.touches[0].phase == TouchPhase.Began)
        {
            if (created)
            {
                Destroy(obj);
                created = false;
            }
            else
            {
                obj = Instantiate(objectToSpawn, pcm.transform.position, pcm.transform.rotation);
                created = true;
            }
        }
    }
}

```



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

/// <summary>
/// Return to the previous scene
/// </summary>
public class SceneARController : MonoBehaviour
{
    // Update is called once per frame
    void Update()
    {
        if (Input.GetKey(KeyCode.Escape))
        {
            if (User.esProfesor)
            {
                SceneManager.LoadScene("SelectModel");
            }
            else
            {
                SceneManager.LoadScene("MainStudent");
            }
        }
    }
}
```