



UNIVERSITAT  
POLITÈCNICA  
DE VALÈNCIA



UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Escuela Técnica Superior de Ingeniería Industrial

Desarrollo de una aplicación de realidad aumentada para  
teléfono móvil para la localización de la arteria femoral  
mediante secuencias de ecografía

Trabajo Fin de Grado

Grado en Ingeniería Biomédica

AUTOR/A: Blanquer Piña, Coral

Tutor/a: Moratal Pérez, David

Cotutor/a: Vallés Lluch, Ana

Cotutor/a externo: FERNANDEZ CISNAL, AGUSTIN

CURSO ACADÉMICO: 2021/2022



UNIVERSITAT  
POLITÈCNICA  
DE VALÈNCIA



ESCUELA TÉCNICA  
SUPERIOR INGENIERÍA  
INDUSTRIAL VALENCIA

**TRABAJO FIN DE GRADO EN INGENIERÍA BIOMÉDICA**

# **DESARROLLO DE UNA APLICACIÓN DE REALIDAD AUMENTADA PARA TELÉFONO MÓVIL PARA LA LOCALIZACIÓN DE LA ARTERIA FEMORAL MEDIANTE SECUENCIAS DE ECOGRAFÍA**

AUTORA: CORAL BLANQUER PIÑA

TUTOR: DAVID MORATAL PÉREZ

COTUTOR: AGUSTÍN FERNÁNDEZ CISNAL

**Curso Académico: 2021-22**



# AGRADECIMIENTOS

En primer lugar, quiero agradecer a mi tutor David y a mi cotutor Agustín por permitirme realizar el Trabajo de Fin de Grado junto al Hospital Clínico Universitario de Valencia que me ha dado mi primer acercamiento real a la realidad aumentada y a sus aplicaciones en la cirugía. Me ha abierto los ojos a mi futuro profesional y me ha acercado más a la ingeniera que quiero ser.

En segundo lugar, me gustaría agradecer a Pepe Roda del LabLENI del Instituto de Investigación e Innovación en Bioingeniería (I3B) por dejarme parte del material necesario y por prestarme su conocimiento para poder realizar parte de esta investigación. Gracias por soportar mis eternos correos y mis constantes preguntas.

También quiero agradecer a mis amigas Laura Almendros, Elena Bernabé, Ekaterina Panova, Sara Pérez, Shana Jin, Raúl Alós y Carla Farnham por estar conmigo durante estos últimos cuatro años, darme ánimos cuando lo he necesitado y hacerme mejor persona.

Por último, pero no menos importante, a mi madre, por siempre estar ahí y apoyarme en mis decisiones. Sin ti no sería lo que soy ahora ni estaría donde estoy. Me faltan palabras para agradecerte todo lo que has hecho por mí. Gracias por todo y por fin puedes decir que tu hija es ingeniera.



## RESUMEN

La punción de la arteria femoral es una técnica que se lleva practicando desde hace años para la intervención por cateterismos de vasos sanguíneos o del corazón. Tradicionalmente su localización se realiza a través de referencias anatómicas como la cabeza del fémur y tras la introducción de la ecografía, esta localización se efectúa con el ecógrafo. Es esencial que esta sea realizada de forma correcta puesto que de lo contrario el paciente puede sufrir diversas complicaciones que pueden derivar en otra operación.

La realidad aumentada es una tecnología en auge que combina la realidad con elementos virtuales por lo que combinando la imagen médica con la anatomía de la persona se obtiene una mejor visualización de la arteria en todo momento y en consecuencia el cirujano tiene más probabilidades de acertar la posición, puesto que puede seguir viendo la imagen aun cuando deje la sonda y coja la aguja.

Con *Unity*, *Vuforia*, *OBS*, *ffmpeg*, una *raspberry pi* y un *router* entre otros materiales, se construye una aplicación de realidad aumentada para que funcione en teléfonos móviles con sistema operativo tipo *Android*. Esta *app* debe ser capaz de reproducir en tiempo real la señal de vídeo proporcionada por el ecógrafo además de dejar de seguir al *target* cuando se aprieta un botón.

Se prueban distintos métodos, desde un *multitarget* hasta de un a varios *targets*, igualmente se prueban dos formas diferentes de reproducir la ecografía, una de ellas con *OBS* y la otra con una *raspberry pi*. La primera es una codificación por *software* y la segunda es por *hardware* por lo que es más rápida y por tanto introducirá menor latencia.

Los resultados obtenidos no se acercan a los esperados puesto que el retraso es considerable y surgieron otros problemas inesperados. Sin embargo, fueron bastante buenos y se consiguieron algunos de los objetivos propuestos como seguir al objetivo una vez es reconocido y poder visualizarlo. Por tanto, puede decirse que el presente Trabajo Fin de Grado es una buena primera aproximación para lograr la solución óptima.

**Palabras clave:** realidad aumentada; arteria femoral; *target*; ecografía; vídeo en directo; *Unity*.

## RESUM

La punció de l'artèria femoral és un tècnica que es porta practicant des de fa anys per a la intervenció per cateterismes de vasos sanguinis o del cor. Tradicionalment la seua localització es realitza a través de referències anatòmiques com el cap del fèmur i després de de la introducció de l'ecografia, aquesta localització s'efectua amb l'ecògraf. És essencial que aquesta siga efectuada de una forma correcta perquè en cas contrari el pacient pot patir diverses complicacions que poden derivar en una altra operació.

La realitat augmentada és una tecnologia en auge que combina la realitat amb elements virtual pel que combinant la imatge mèdica amb l'anatomia de la persona s'obté una millor visualització de l'artèria en tot moment i en conseqüència el cirurgià té més probabilitats d'encertar la posició, ja que pot continuar veient la imatge encara que deixe la sonda i agafe l'agulla.

Amb *Unity*, *Vuforia*, *OBS*, *ffmpeg*, una *raspberry pi* i un *router* entre altres materials, es construeix una aplicació de realitat augmentada perquè funcione en telèfons mòbils amb sistema operatiu tipus *Android*. Aquesta *app* ha de ser capaç de reproduir en temps real el senyal de vídeo proporcionada per l'ecògraf a més de deixar de seguir al *target* quan s'aprete un botó.

Es proven diferents mètodes, des d'un *multitarget* fins d'un a diversos *targets*, igualment es proven dues maneres distintes de reproduir l'ecografia, una d'elles amb *OBS* i l'altra amb una *raspberry pi*. La primera és una codificació per *software* y la segona és per *hardware* que és més ràpida i per tant introduirà una menor latència.

Els resultats obtinguts no s'acosten als esperats perquè el retard és considerable i van sorgir altres problemes inesperats. No obstant això, van ser bastants bons i es van aconseguir alguns dels objectius proposats con seguir a l'objectiu una vegada es reconegut i poder visualitzar-lo. Per tant, pot dir-se que el present Treball Fi de Grau és una bona primera aproximació per a aconseguir la solució òptima.

**Paraules clau:** realitat augmentada; *target*; ecografia; vídeo en directe; *Unity*.

## ABSTRACT

Femoral artery puncture is a technique that has been practiced for years for catheterization of blood vessels or the heart. Traditionally its localization is done through anatomical references such as the femoral head and after the introduction of the ultrasound, this localization is performed with the ultrasound machine. It is essential that this is done correctly, otherwise the patient may suffer various complications that might lead to another operation.

Augmented reality is a growing technology that combines reality with virtual elements so that by combining the medical image with the person's anatomy, a better visualization of the artery is always obtained and consequently the surgeon is more likely to get the position right, since he or she can continue to see the image even when he or she leaves the probe and picks up the needle.

With Unity, Vuforia, OBS, *ffmpeg*, a raspberry pi and a router among other materials, an augmented reality application is built to run on cell phones with Android type operating system. This app must be able to reproduce in real time the video signal provided by the ultrasound scanner as well as stop following the target when a button is pressed.

Different methods are tested, from a multitarget to a single or multiple targets, also two different ways of reproducing the ultrasound are tested, one of them with OBS and the other with a raspberry pi. The first one is a software encoding and the second one is hardware encoding so it is faster and therefore will introduce less latency.

The results obtained are not close to the expected ones since the delay is considerable and other unexpected problems arose. However, they were quite good and some of the proposed objectives were achieved, such as tracking the target once it is recognized and being able to visualize it. Therefore, it can be said that the present Final Degree Project is a good first approximation to achieve the optimal solution.

**Keywords:** augmented reality; femoral artery; target; ultrasound; stream video; Unity.





# ÍNDICE GENERAL

## Documentos contenidos en el TFG

- Memoria
- Presupuesto

### Documento 1: Memoria

1. MOTIVACIÓN, OBJETIVOS Y JUSTIFICACIÓN
2. INTRODUCCIÓN
3. MATERIALES
4. MÉTODOS
5. RESULTADOS Y DISCUSIÓN
6. CONCLUSIONES
7. LÍNEAS FUTURAS
8. BIBLIOGRAFÍA

### Documento 2: Presupuesto

1. INTRODUCCIÓN
2. CUADRO DE MANO DE OBRA
3. CUADRO DE MATERIALES
4. CUADRO DE MAQUINARIA
5. CUADRO DE PRECIOS UNITARIOS
6. CUADRO DE PRECIOS DESCOMPUESTOS
7. CUADRO DE PRESUPUESTOS PARCIALES
8. PRESUPUESTO DE EJECUCIÓN POR CONTRATA



# DOCUMENTO 1: MEMORIA

DESARROLLO DE UNA APLICACIÓN DE  
REALIDAD AUMENTADA PARA  
TELÉFONO MÓVIL PARA LA  
LOCALIZACIÓN DE LA ARTERIA FEMORAL  
MEDIANTE SECUENCIAS DE ECOGRAFÍA

AUTORA: CORAL BLANQUER PIÑA

TUTOR: DAVID MORATAL PÉREZ

COTUTOR: AGUSTÍN FERNÁNDEZ CISNAL

TITULACIÓN: GRADO EN INGENIERÍA BIOMÉDICA

CURSO: 2021/22

# ÍNDICE

1. MOTIVACIÓN, OBJETIVOS Y JUSTIFICACIÓN .....	7
2. INTRODUCCIÓN .....	8
2.1 Sistema circulatorio.....	8
2.1.1. <i>Arteria femoral</i> .....	8
2.2. Localización sitio punción arteria femoral .....	10
2.3. Ecografía.....	10
2.4. Realidad Aumentada .....	12
3. MATERIALES .....	15
3.1. Software .....	15
3.1.1. <i>Vuforia</i> .....	15
3.1.2. <i>Unity</i> .....	15
3.1.3. <i>OBS</i> .....	15
3.1.4. <i>VLC</i> .....	16
3.1.5. <i>Nginx</i> .....	16
3.1.6. <i>Ffmpeg</i> .....	16
3.1.7. <i>Android Studio</i> .....	16
3.1.8. <i>PuTTY</i> .....	17
3.2. Hardware.....	17
3.2.1. <i>Capturadora de vídeo</i> .....	17
3.2.2. <i>Cable HDMI</i> .....	18
3.2.3. <i>Ordenador portátil</i> .....	18
3.2.4. <i>Teléfono móvil</i> .....	18
3.2.5. <i>Raspberry Pi 2</i> .....	19
3.2.6. <i>Router</i> .....	19
3.2.7. <i>Ecógrafo</i> .....	20
3.2.8. <i>Sonda ultrasonidos</i> .....	20
4. MÉTODOS.....	21
4.1. Cubo AR.....	21
4.2. Un <i>ImageTarget</i> .....	27
4.3. Más de un <i>ImageTarget</i> .....	30
4.4. Reproducción del vídeo: forma 1.....	35

DESARROLLO DE UNA APLICACIÓN DE REALIDAD AUMENTADA PARA TELÉFONO MÓVIL PARA LA  
LOCALIZACIÓN DE LA ARTERIA FEMORAL MEDIANTE SECUENCIAS DE ECOGRAFÍA

4.5. Reproducción del vídeo: forma 2.....	36
5. RESULTADOS Y DISCUSIÓN.....	43
5.1. Cubo AR.....	43
5.2. Un <i>ImageTarget</i> .....	44
5.3. Más de un <i>ImageTarget</i> .....	45
5.4. Reproducción del vídeo: forma 1.....	48
5.5. Reproducción del vídeo: forma 2.....	50
6. CONCLUSIONES .....	54
7. LÍNEAS FUTURAS .....	55
8. BIBLIOGRAFÍA.....	56

## ÍNDICE DE FIGURAS

Figura 1. Anatomía de la arteria femoral [5].....	9
Figura 2. Localización de la arteria femoral [1].....	10
Figura 3. Comparación entre realidad virtual (izquierda) y realidad aumentada (derecha). El verde representa el mundo real mientras el naranja representa los objetos virtuales [13]. .....	12
Figura 4. Sistema ClarifEye de Philips [14]. .....	14
Figura 5. Sistema experimental de realidad aumentada de Philips [15]. .....	14
Figura 6. Cubo realizado en Vuforia [ <a href="https://mergeedu.com/cube">https://mergeedu.com/cube</a> ]. .....	21
Figura 7. Jerarquía del trabajo con el cubo [Realización Propia]. .....	22
Figura 8. Implementación del reproductor de Youtube [Realización Propia].....	22
Figura 9. Dimensionamiento y posicionamiento de los botones [Realización Propia]. .....	23
Figura 10. Poner la función al botón [Realización Propia]. .....	25
Figura 11. Añadir el script al proyecto de Unity [Realización Propia]. .....	25
Figura 12. Código QR realizado [ <a href="https://es.qr-code-generator.com/">https://es.qr-code-generator.com/</a> ] .....	27
Figura 13. Añadir base de datos de Vuforia a Unity. El nombre de la base de datos y el nombre de la imagen son las que he especificado en Vuforia [Realización Propia]. .....	28
Figura 14. Transformaciones del plano [Realización Propia]. .....	28
Figura 15. Escena de Unity con un código QR [Realización Propia]. .....	29
Figura 16. Estructura y previsualización del proyecto. En él se pueden ver cuatro botones: arriba a la izquierda el botón para empezar a reproducir, arriba a la derecha el botón para pausar el vídeo, abajo a la izquierda el botón para dejar de seguir al target y abajo a la derecha el botón para volver a seguir al target [Realización Propia]. .....	29
Figura 17. Colores de los botones [Realización Propia]. .....	30
Figura 18. Códigos QR utilizados en la ampliación. De izquierda a derecha: QR2, QR3 y QR4 [ <a href="https://es.qr-code-generator.com/">https://es.qr-code-generator.com/</a> ] [ <a href="https://www.the-qr-code-generator.com/es/">https://www.the-qr-code-generator.com/es/</a> ] [ <a href="https://qrcode.tec-it.com/es">https://qrcode.tec-it.com/es</a> ]. .....	30
Figura 19. Estructura del proyecto con dos targets [Realización Propia]. .....	31
Figura 20. Escena del proyecto con dos códigos QR [Realización Propia]. .....	31
Figura 21. Escena proyecto tres QR [Realización Propia]. .....	32
Figura 22. Estructura del proyecto con tres QR [Realización Propia]. .....	32
Figura 23. Escena del proyecto con cuatro QR [Realización Propia].....	33
Figura 24. Estructura del proyecto con cuatro QR [Realización Propia]. .....	33
Figura 25. Escena del proyecto con cuatro QR método 2 [Realización Propia]. .....	34
Figura 26. Estructura del proyecto con cuatro QR método 2 [Realización Propia]. .....	34

DESARROLLO DE UNA APLICACIÓN DE REALIDAD AUMENTADA PARA TELÉFONO MÓVIL PARA LA  
LOCALIZACIÓN DE LA ARTERIA FEMORAL MEDIANTE SECUENCIAS DE ECOGRAFÍA

Figura 27. Implementación del VLC en Unity [Realización Propia].....	36
Figura 28. Implementación del Plugin UMP en el proyecto [Realización Propia].....	41
Figura 29. Estructura del proyecto final [Realización Propia].....	41
Figura 30. Resultado Cubo AR visto desde el teléfono móvil [Realización Propia].....	43
Figura 31. Resultado Cubo AR visto desde fuera [Realización Propia]. ....	44
Figura 32. Resultados de un código QR. A la izquierda la sonda está completamente horizontal y a la derecha está levantada e inclinada [Realización Propia]. ....	45
Figura 33. Resultados de dos códigos QR [Realización Propia].....	45
Figura 34. Resultados del proyecto con tres QR [Realización Propia]. ....	46
Figura 35. Resultados del proyecto con cuatro QR [Realización Propia]. ....	47
Figura 36. Resultados del proyecto con cuatro QR método 2 [Realización Propia]. ....	48
Figura 37. Resultados de retransmitir ordenador-ordenador vistos desde dentro. A la izquierda tenemos el VLC y a la derecha el OBS [Realización Propia]. ....	48
Figura 38. Resultados de retransmitir ordenador-ordenador vistos desde fuera. A la izquierda el ordenador del que captamos la señal y a la derecha el ordenador que la recibe [Realización Propia]. ....	49
Figura 39. Resultados de la retransmisión ordenador-móvil [Realización Propia]. ....	49



## ÍNDICE DE TABLAS

Tabla 1. Coordenadas (X; Y; Z) del proyecto de cuatro QR método b donde IT corresponde a ImageTarget y P corresponde a Plano [Realización Propia].....	34
Tabla 2. Parámetros del ffmpeg utilizados (en negrita están los introducidos) y sus significados [Realización Propia].....	38
Tabla 3. Parámetros del ffmpeg utilizados para disminuir la latencia aún más (en negrita están los introducidos) y sus significados [Realización Propia].....	40
Tabla 4. Parámetros del ffmpeg utilizados (en negrita están los introducidos) y el tiempo de retraso en segundos [Realización Propia].....	50
Tabla 5. Parámetros del ffmpeg utilizados para disminuir aún más la latencia (en negrita están los introducidos) y el tiempo de retraso en segundos [Realización Propia].	52

## 1.MOTIVACIÓN, OBJETIVOS Y JUSTIFICACIÓN

Es esencial realizar una precisa localización de la arteria para que su punción sea correcta, puesto que entre las complicaciones del cateterismo por acceso femoral se encuentran [1,2]:

- **Hemorragia:** pérdida de sangre a través del orificio de punción y suele ser la más frecuente.
- **Hematoma:** agrupación de sangre en una zona fuera de la arteria.
- **Pseudoaneurisma:** lesión en la pared de la arteria que causa su ruptura y la acumulación de sangre en los tejidos. Si es mayor de dos centímetros, requiere de una intervención para repararlo.
- **Fístula arteriovenosa:** comunicación continua entre la arteria y la vena. Aparece al hacer múltiples intentos de punción.
- **Trombosis arterial:** producción de un trombo en la arteria.
- **Dissección arterial.**
- **Infección:** suele aparecer en caso de punciones repetidas en un mismo lugar.
- **Reacción vagal:** el dolor durante la punción puede derivar en hipotensión y bradicardia.

El uso de ultrasonidos para la localización disminuye el número de intentos, el tiempo de intervención, el riesgo de punción de la vena y la incidencia de las complicaciones [3].

Según un estudio realizado en el Hospital Universitario Nuestra Señora de la Candelaria en Santa Cruz de Tenerife donde se analizaron 917 cateterismos cardíacos a través de la femoral efectuados entre mayo de 2011 y abril de 2012, 70 pacientes sufrieron complicaciones. Las complicaciones más frecuentes fueron los hematomas y el pseudoaneurisma [4].

Por lo que la motivación para realizar este Trabajo Fin de Grado subyace en poder aportar una herramienta novedosa como lo es la realidad aumentada para poder reducir las posibles complicaciones derivadas de una incorrecta punción de la arteria femoral, además de ser una herramienta para que los residentes de cirugía vascular puedan aprender este procedimiento de forma más sencilla y rápida.

El objetivo de este Trabajo Fin de Grado es el crear una aplicación de realidad aumentada para un teléfono móvil con sistema operativo Android capaz de reconocer una imagen y proyectar un vídeo además de seguir tal imagen en el espacio real, reproducir en tiempo real la señal del ecógrafo sobre la piel del paciente y dejar de seguir la imagen cuando se apriete un botón para que así el cirujano pueda dejar la sonda y coger el catéter sin dejar de ver la imagen del ecógrafo.

## 2. INTRODUCCIÓN

### 2.1 Sistema circulatorio

El sistema circulatorio de un humano es un circuito cerrado que está formado por el corazón y los vasos sanguíneos (arterias, venas y capilares) y cuya función es transportar la sangre a los distintos órganos del cuerpo para que estos reciban los nutrientes necesarios [5].

El corazón es la bomba que aporta la presión suficiente para que la sangre llegue a los distintos órganos, las arterias se encargan de llevar la sangre desde el corazón hasta el resto del cuerpo, los capilares son donde se produce el intercambio de nutrientes y otros productos y las venas conducen la sangre de vuelta al corazón [5].

En cuanto al corazón, este está dividido en cuatro cavidades, aurícula derecha e izquierda, ventrículo derecho e izquierdo. Las primeras reciben la sangre y las segundas impulsan la sangre. La sangre llega a la aurícula derecha a través de las venas cavas, pasa al ventrículo derecho por la válvula tricúspide y de este, a través de la válvula y las arterias pulmonares, llega a los pulmones donde la sangre se oxigena. Posteriormente, llega a la aurícula izquierda por las venas pulmonares, pasa al ventrículo izquierdo a través de la válvula mitral y es distribuida al organismo mediante la válvula aórtica y la arteria aorta [5] [6].

Por tanto, las funciones del aparato circulatorio son llevar el oxígeno y los nutrientes a las distintas células, transportar hormonas que actúan como mensajeras, contribuir a la defensa del cuerpo al trasladar leucocitos (glóbulos blancos) e inmunoglobulinas (anticuerpos) y regula la temperatura corporal [5].

Se pueden distinguir tres tipos de arteria según su calibre: grandes, medianas y arteriolas. Las arterias de gran calibre son las que se encuentran más cerca del corazón y también reciben el nombre de elásticas debido a que en su composición predominan las fibras elásticas. Las de mediano calibre están más alejadas y también se denominan arterias musculares porque tienen una gran cantidad de fibras musculares lisas en sus paredes. Las arteriolas son mucho más pequeñas, su diámetro oscila las micras y la presencia de fibras musculares en sus paredes disminuye conforme se aproximan a los capilares [5].

#### 2.1.1. Arteria femoral

El riego arterial de las extremidades inferiores depende de la arteria ilíaca externa, que al pasar por el muslo recibe el nombre de arteria femoral que desciende por la parte anterior interna y al pasar por detrás de la rodilla se continúa con la arteria poplítea [5]. La arteria es el elemento más externo y situada a su lado está la vena femoral; los músculos fundamentales que la rodean son el sartorio, el aductor largo y el aductor mayor [5] (Figura 1).

DESARROLLO DE UNA APLICACIÓN DE REALIDAD AUMENTADA PARA TELÉFONO MÓVIL PARA LA  
LOCALIZACIÓN DE LA ARTERIA FEMORAL MEDIANTE SECUENCIAS DE ECOGRAFÍA

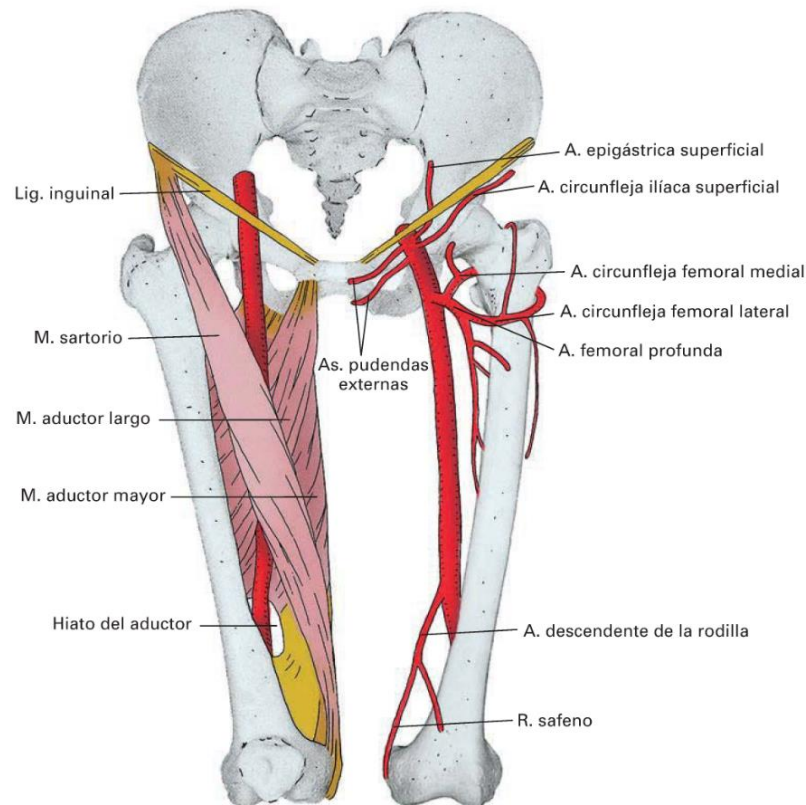


Figura 1. Anatomía de la arteria femoral [5].

La arteria femoral se divide en distintas ramas: la arteria femoral profunda que es la rama más gruesa de la femoral, la principal arteria del muslo y nace de la cara posterior de la femoral; las arterias perforantes compuestas por tres o cuatro vasos que atraviesan por delante los músculos aductores y llegan a la cara posterior. Las arterias circunflejas femoral medial y lateral son ramificaciones de la femoral profunda. Las arterias pudendas externas nacen de la cara interna de la femoral [5].

Se utiliza esta arteria como acceso arterial para técnicas endovasculares por su gran calibre, su conexión con el corazón, por verse menos afectada por la patología ateromatosa y por la facilidad de acceso. La punción se puede realizar en la bifurcación de las arterias profunda y superficial o en la arteria superficial; siendo más frecuente en esta última por la dificultad de acceso a la profunda. La arteria femoral común se puede punzar también pero no se suele realizar porque la hemostasia (proceso para detener hemorragias) es peor [7]. Para conseguir una buena hemostasia durante el procedimiento, basta con comprimir contra la cabeza del fémur [1].

DESARROLLO DE UNA APLICACIÓN DE REALIDAD AUMENTADA PARA TELÉFONO MÓVIL PARA LA  
LOCALIZACIÓN DE LA ARTERIA FEMORAL MEDIANTE SECUENCIAS DE ECOGRAFÍA

## 2.2. Localización sitio punción arteria femoral

Las referencias anatómicas para la punción de la arteria femoral que se toman, tal y como se puede visualizar en la Figura 2, son la espina iliaca, la sínfisis del pubis y el ligamento inguinal que se encuentra entre estas dos estructuras. Una vez localizadas, se colocan dos dedos (3 cm) partiendo del ligamento y así se tiene asegurada la punción de la femoral común [1].

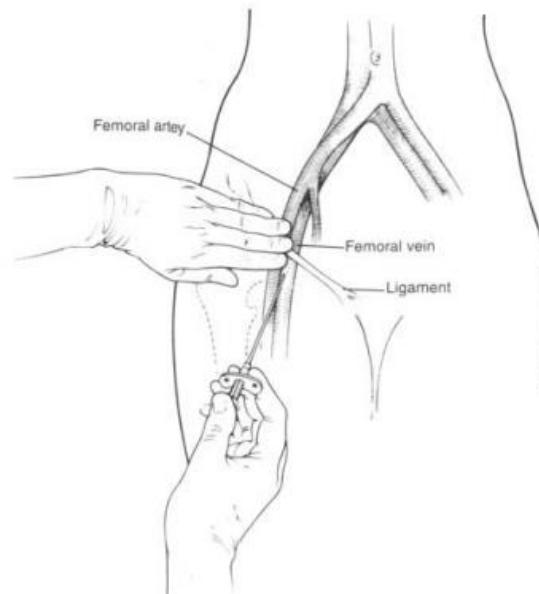


Figura 2. Localización de la arteria femoral [1].

En el caso de querer pinchar en la derecha, con la mano izquierda se palpa la arteria y con otra mano se coge la aguja y pincha entre los dos dedos. Se debe profundizar de dos a cinco centímetros correspondientes a la distancia de la piel al vaso sanguíneo [3].

## 2.3. Ecografía

Una ecografía es una prueba diagnóstica que utiliza ultrasonidos (frecuencia mayor a 20 kHz) para crear imágenes médicas. Al tratarse de ondas acústicas, no se emite radiación y pueden ser utilizados en cualquier momento y lugar para observar estructuras en tiempo real [8].

Los ecógrafos están compuestos por un transductor, una unidad de procesamiento y una pantalla. Los transductores tienen un cristal piezoeléctrico que se encarga de emitir los pulsos de haces de ultrasonidos y de recibir aquellos que son reflejados por las distintas estructuras corporales. Estos últimos son procesados por el ecógrafo y traducidos a una tonalidad de la escala de grises para formar la imagen. Cuando los ultrasonidos pasan por los distintos medios del cuerpo humano, sufren fenómenos físicos de atenuación (pierde amplitud), reflexión (se refleja según el ángulo de incidencia), refracción (cambia de dirección) y bioefectos (efectos sobre los tejidos). La cantidad de reflexión y refracción de los haces depende del tipo de tejido con el que se encuentren. Por tanto,

DESARROLLO DE UNA APLICACIÓN DE REALIDAD AUMENTADA PARA TELÉFONO MÓVIL PARA LA  
LOCALIZACIÓN DE LA ARTERIA FEMORAL MEDIANTE SECUENCIAS DE ECOGRAFÍA

cuanta más amplitud reciba el ecógrafo de vuelta, más blanco se ve, y de esta forma se pueden distinguir las estructuras. La unidad de procesamiento calcula la profundidad midiendo el tiempo que transcurre desde que se emite la señal hasta que es devuelta [9].

En la actualidad, los cirujanos están optando por procedimientos mínimamente invasivos para reducir la probabilidad de infección y reducir el tiempo de hospitalización y recuperación, por lo que, cada vez se utilizan más los ecógrafos portátiles para dirigir la colocación de los catéteres durante estos tratamientos [10].

Centrándose más en la cirugía vascular, el uso de la ecografía permite visualizar los vasos sanguíneos, detectar patologías en sus paredes y trombos en las venas. Si se utiliza el modo Doppler, se es capaz incluso, de detectar cambios en el patrón y velocidad del flujo de la sangre, siendo esto indicador de la gravedad de una enfermedad. Además, se pueden utilizar los ultrasonidos para evaluar previamente una lesión y estimar si es necesaria una cirugía o no, también para el seguimiento después de realizar una cirugía [10].

Entre otras patologías vasculares que se pueden detectar con ecografía se encuentran la estenosis arterial (estrechamiento de la arteria) que su clasificación se hace en base a la medición del diámetro o del área; el síndrome de atrapamiento poplíteo, enfermedad quística de la adventicia de la arteria poplíteo (inflamación quística de una capa de la pared arterial); aneurismas (dilatación anormal de los vasos que produce un abultamiento) e insuficiencia venosa [10].

Para localizar la arteria femoral mediante ultrasonidos, se sitúa el transductor en la zona situada entre la sínfisis pubiana y la espina iliaca, se localiza el ligamento inguinal y se desplaza transversalmente la sonda hasta dar con la femoral, a unos 2.5 cm por debajo del ligamento (misma localización que con la técnica tradicional), tras una buena ubicación de la arteria en el ecógrafo se mantiene en esa posición y se mide la profundidad en la pantalla [3] [11].

Por tanto, la ecografía durante este procedimiento permite, entre otras cosas, detectar los vasos más fácilmente, tener una mayor visualización de la aguja y su introducción en los vasos durante la punción, controlar las condiciones de la cateterización y valorar las posibles complicaciones que puedan surgir [7].

La calcificación de la pared de las arterias femorales secundaria a la enfermedad aterosclerótica es común [12]. Tanto la comprensión como la utilización de dispositivos hemostáticos tienen alta tasa de fallo y posibles complicaciones si la punción se ha realizado en una zona calcificada. La ecografía permite identificar la localización de las placas calcificadas y guiar la punción hacia una zona libre de estas placas, de forma que permitirá una mejor hemostasia posterior [12].

## 2.4. Realidad Aumentada

La realidad aumentada (AR) es una variación de la realidad virtual (VR), con la diferencia de que en la segunda la persona está inmersa en un mundo virtual y no puede interactuar con el real, mientras que, en la primera, el usuario puede ver el mundo real con objetos virtuales superpuestos a él, es decir, la AR complementa la realidad mientras que la VR la reemplaza [12]. Esta diferencia puede verse en la Figura 3.

Por lo que, con la realidad aumentada, una persona puede visualizar el mundo real y los objetos virtuales a la vez, como un todo coherente. Lo que quiere decir que, en el entorno de la AR, un objeto virtual tiene un lugar fijo en mundo real y mientras que no se modifique, permanecerá en el mismo lugar [13].

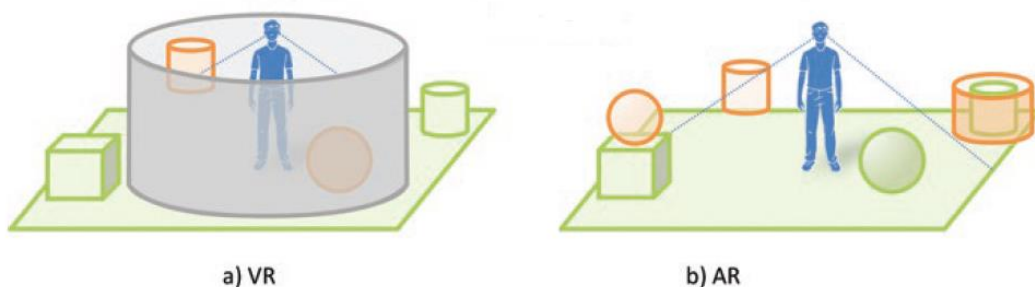


Figura 3. Comparación entre realidad virtual (izquierda) y realidad aumentada (derecha). El verde representa el mundo real mientras el naranja representa los objetos virtuales [13].

En el presente trabajo, se utiliza el seguimiento basado en cámara que consiste en utilizar los objetos que capta la cámara para determinar la orientación y la posición de los objetos virtuales. Además, se utilizan marcadores para este seguimiento, puesto que son fácilmente reconocibles por sus colores, formas, contrastes, etc. [13].

Si se comparan marcadores de color con los blancos y negros, estos últimos ofrecen ciertas ventajas, puesto que pueden ser extraídos de cualquier imagen. Pueden ser cuadrados o redondos con un margen completamente negro o blanco. Los criterios para su selección suelen ser su estabilidad, velocidad de reconocimiento y el número de marcas distinguibles [13].

Por norma general, los marcadores deben cumplir ciertas condiciones. Deben ser completamente visibles a la cámara para poder ser reconocidos, sin embargo, si se usa un patrón redundante, puede ser detectado si es parcialmente visible gracias a esa redundancia. También, si es muy grande, solo una parte será visible cuando la cámara está muy cerca de él, lo que se traduce en una imposibilidad de seguimiento; pero si es muy pequeño, se produce un reconocimiento incorrecto de patrones (si la cámara está muy lejos) porque se detectan pocos píxeles [13].

Además, el ángulo entre la cámara y el marcador y la situación de la iluminación también tienen un gran impacto en el seguimiento. Si el ángulo es demasiado plano, los valores calculados cambian mucho y si la iluminación es demasiado brillante u oscura las zonas blancas y negras del marcador

DESARROLLO DE UNA APLICACIÓN DE REALIDAD AUMENTADA PARA TELÉFONO MÓVIL PARA LA  
LOCALIZACIÓN DE LA ARTERIA FEMORAL MEDIANTE SECUENCIAS DE ECOGRAFÍA

dejan de tener el contraste suficiente como para ser distinguidas entre sí, lo que resulta en una falta de reconocimiento del marcador [13].

Por tanto, en el presente Trabajo Fin de Grado, se utiliza el seguimiento basado en marcadores, puesto que pueden crearse rápida y fácilmente imprimiéndolos y pueden aplicarse a los objetos, tal y como se comenta en los apartados de métodos, resultados y discusión [13].

A continuación, se describen algunos conceptos relacionados con la AR:

- **Tracking:** cálculo de la posición y la orientación. Por ejemplo, en las aplicaciones de uso exterior, el GPS suele proporcionar este contexto [13].
- **Registro:** anclaje correcto del contenido virtual sobre la realidad [13].
- **Visualización:** a partir de la transformación resultante del registro geométrico y de la perspectiva de la cámara, se renderiza el contenido virtual (se crea el gráfico) que se superpone a la imagen recibida por la cámara [13].
- **Salida:** las imágenes superpuestas se muestran en una pantalla que suele estar conectada a la cámara. Esta pantalla puede ser de un dispositivo móvil, una tableta o unas gafas de realidad aumentada [13].

En el ámbito médico, la realidad aumentada se usa como soporte para los procedimientos quirúrgicos, especialmente para aquellos mínimamente invasivos. De esta forma, información que normalmente está separada puede mostrarse en paralelo y en el campo de visión del cirujano [13]. A continuación, se describe un ejemplo de un sistema comercial de la casa *Philips*.

El *ClarifEye* (Figura 4) es un sistema de navegación quirúrgica de realidad aumentada para cirugía de columna desarrollado por *Philips*. Es capaz de facilitar una planificación detallada y una guía eficaz para la colocación de los tornillos de forma específica [14]. Entre sus características se encuentran:

- **Ver la anatomía:** aporta una planificación de alta precisión y la información de la navegación gracias a la tomografía computarizada (TC) de baja dosis y alta calidad que es usada. El *ClarifEye* utiliza una exploración intraoperatoria para crear un modelo 3D de la columna en tiempo real con segmentación automática [14].
- **Reducir la dosis de radiación:** al facilitarse la visualización, el procedimiento puede realizarse de forma más rápida, reduciendo así la dosis al paciente y al personal [14].
- **Planificación:** aporta una ayuda para planificar la localización de los tornillos a insertar, puesto que, al seleccionar la posible ubicación, se muestran las vistas transversales además de calcular la trayectoria [14].
- **Optimizar el flujo de trabajo:** se realiza un seguimiento automático de los marcadores cutáneos y se corrige la imagen virtual en función de sus posiciones. No requiere registro manual ni marcos de referencia invasivos [14].



DESARROLLO DE UNA APLICACIÓN DE REALIDAD AUMENTADA PARA TELÉFONO MÓVIL PARA LA LOCALIZACIÓN DE LA ARTERIA FEMORAL MEDIANTE SECUENCIAS DE ECOGRAFÍA

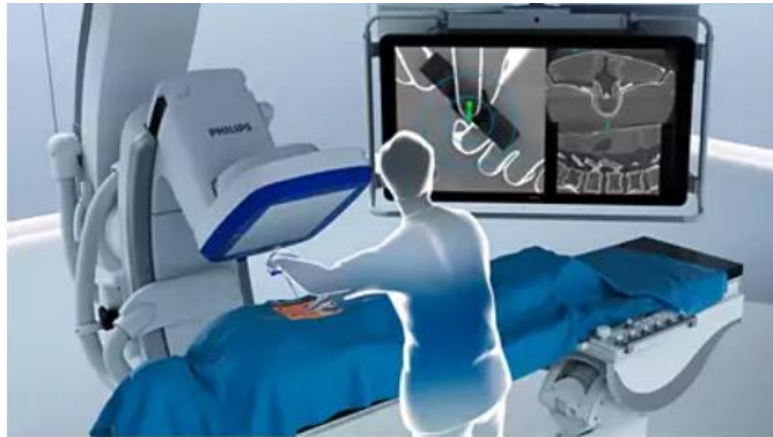


Figura 4. Sistema ClarifEye de Philips [14].

Otro sistema siendo desarrollado por *Philips* (Figura 5) se trata de una solución de realidad aumentada que permite mostrar modelos en dos y tres dimensiones de los órganos en tiempo real y manejarlos con comandos de voz o gestos. Además, permite ver otra información del paciente para saber su estado durante la intervención. Este sistema se basa en la cirugía guiada por imagen combinada con las gafas de realidad virtual/aumentada *HoloLens*. Por tanto, esta técnica mejorará y agilizará el proceso quirúrgico además de reducir los costes hospitalarios al ser las intervenciones más rápidas y eficientes [15].



Figura 5. Sistema experimental de realidad aumentada de Philips [15].

### 3. MATERIALES

#### 3.1. Software

##### 3.1.1. *Vuforia*

*Vuforia* es una plataforma de desarrollo de aplicaciones de realidad aumentada y realidad mixta que permite un seguimiento robusto y un buen rendimiento en diversos dispositivos como por ejemplo teléfonos móviles, y cascos de realidad mixta como las *HoloLens* de *Microsoft*. *Vuforia* tiene la posibilidad de integrarse en *Unity*, lo que permite crear aplicaciones para los sistemas operativos *Android* y *iOS* de forma sencilla mediante el sistema de arrastrar y soltar [16].

Un marcador, o *target*, en realidad aumentada es una imagen que es registrada por la aplicación y que desencadena información en la aplicación. Cuando la cámara del dispositivo, con la aplicación en uso, reconoce el marcador, se activa la visualización del contenido virtual sobre el *target*. Entre los distintos tipos de marcadores, se pueden encontrar, los códigos QR, los marcadores físicos reflectantes, los objetivos de imagen, las etiquetas 2D y los objetos en tres dimensiones [16].

Los objetivos de imagen (o *ImageTargets*), como se ha comentado, son un tipo de marcador y están compuestos por una imagen que se registra manualmente en la aplicación (puede ser cualquier imagen) [16].

##### 3.1.2. *Unity*

*Unity* (en el presente trabajo se ha usado la versión 2020.3.16f1) es una plataforma para crear contenidos en tres dimensiones en tiempo real. Ofrece un conjunto de soluciones de *software* para crear contenidos interactivos en dos y tres dimensiones en tiempo real para teléfonos móviles, tabletas, ordenadores y dispositivos de realidad virtual y aumentada [17].

Es importante que durante la instalación se seleccione y se descargue el *SDK* y el *NDK* de *Android* [17].

##### 3.1.3. *OBS*

*OBS* es un *software* libre y de código abierto para la grabación y retransmisión de vídeo en directo. Entre las funcionalidades de este programa informático se encuentran la captura y mezcla de señales de audio y vídeo a tiempo real en alta calidad; también cuenta con opciones de configuración potentes y con una interfaz amigable [18].

#### 3.1.4. VLC

*VLC* es un reproductor multimedia libre, de código abierto que puede ser usado en múltiples plataformas y que reproduce casi todos los tipos de archivos como *DVD* y diversos protocolos de transmisión como *rtmp* o *ump* [19].

#### 3.1.5. Nginx

*Nginx* en versión *gryphon* es un software de código abierto que ofrece un servidor web/proxy que mejora el rendimiento, la seguridad y la fiabilidad de las aplicaciones en las que es usado. Se usa esta versión porque tiene el módulo necesario para transmitir por *rtmp* [20].

#### 3.1.6. Ffmpeg

*Ffmpeg* es un conversor de vídeo y audio muy rápido y que además tiene la capacidad de grabar y capturar una fuente de audio y vídeo, decodificarla, codificarla, combinarla y transmitirla en tiempo real [21] [22].

Ofrece tres herramientas principales en forma de código fuente que son *ffmpeg* para el procesamiento, *ffprobe* para la información y *ffplay* para la reproducción [22].

#### 3.1.7. Android Studio

*Android Studio* es un entorno de desarrollo integrado que sirve para desarrollar aplicaciones que serán usadas en dispositivos con el sistema operativo *Android*. Entre sus funciones se encuentra una compilación flexible, un emulador rápido y facilidad de integración con *GitHub* y plantillas de código entre otras [23].

### 3.1.8. PuTTY

*PuTTY*, creado por Simon Tatham para *Windows*, es un *software* de código abierto que utiliza *SSH* y *telnet*. Está disponible con el código fuente y fue desarrollado y apoyado por voluntarios. *SSH* es un protocolo de comunicación de red que permite la comunicación entre dos ordenadores mediante protocolos como *http* y que se manden información entre ellos. Una característica de *SSH* es que esta comunicación está encriptada lo que lo hace adecuado para su uso en redes inseguras [24] [25].

## 3.2. Hardware

### 3.2.1. Capturadora de vídeo

Para poder capturar el vídeo del ecógrafo y poder tenerlo en el ordenador portátil se utilizó una capturadora de vídeo cuya entrada es un puerto *HDMI* y cuya salida es un puerto *USB 2.0* con las siguientes especificaciones [26]:

- **Alcance:** puede grabar en tiempo real cualquier tipo de imagen como por ejemplo las imágenes médicas. Permite una transmisión de alta velocidad sin demora y, además, puede utilizar funciones integradas del *software* para mejorar la captura.
- **Compacto y portátil:** no requiere de alimentación, es compacto y al ser de pequeño tamaño es fácil de transportar.
- **No controlador:** no requiere de un controlador para capturar el vídeo.
- **Puerto USB 2.0:** este puerto le permite transmitir a alta velocidad fácilmente para aplicaciones como *Twitch*, *Youtube*, *OBS*, *Potplayer* y *VLC*.
- **Compatibilidad:** la tarjeta de captura es compatible con programas como *VLC*, *OBS* o *Amcap* y con los sistemas operativos *Windows*, *Android* y *iOS*.
- **Tarjeta de captura:** es capaz de capturar señales de vídeo y audio *HDMI* y transmitir las a un ordenador o teléfono inteligente.
- **Marca y referencia del fabricante:** *Yummici – Mnioky*.
- **Peso:** 20 g.
- **Tasa de fotogramas (o frame rate):** 1080P HD y 60 fps. Con una resolución de entrada de 3840x2160 30 Hz y una resolución de salida de 1920x1080 30 Hz.

DESARROLLO DE UNA APLICACIÓN DE REALIDAD AUMENTADA PARA TELÉFONO MÓVIL PARA LA  
LOCALIZACIÓN DE LA ARTERIA FEMORAL MEDIANTE SECUENCIAS DE ECOGRAFÍA

### 3.2.2. Cable HDMI

Para conectar el ecógrafo con la capturadora de vídeo se ha utilizado un cable HDMI 4K 2.0 macho/macho de alta velocidad y con canal de retorno de audio con las siguientes especificaciones [27]:

- **Velocidad de transferencia de datos:** 18 GB por segundo.
- **Material:** PVC.
- **Longitud:** 1.8 m.
- **Peso:** 0.17 kg.
- Soporta hasta 2160x4096 60Hz de tasa de fotogramas.
- **Compatibilidad:** es compatible con *Mac iOS* y *Windows*.
- **Número de pieza del fabricante:** MTBHDB2018.

### 3.2.3. Ordenador portátil

Para la realización del presente Trabajo Fin de Grado se ha utilizado un ordenador portátil con las siguientes especificaciones [28]:

- **Marca:** *Lenovo Legion 5 15IMH05*.
- **Pantalla:** con un tamaño de 15.6 pulgadas, resolución de pantalla: *FullHD 1920x1080* píxeles, 120Hz.
- **Sistema operativo:** *Windows 10*.
- **Procesador:** *Intel Core i7-10750Hz*.
- **Tamaño memoria RAM:** 16 GB.
- **Tarjeta gráfica:** GTX 1650.
- **Velocidad CPU:** 5 GHz.
- **Disco duro:** 512 GB SSD.
- **Referencia del fabricante:** 82AU00MFSP.

### 3.2.4. Teléfono móvil

Para probar la aplicación se ha utilizado un teléfono móvil con las siguientes especificaciones [29]:

- **Marca y modelo:** *Xiaomi Redmi Note 10*.
- **Almacenamiento y RAM:** 4 GB + 128 GB.
- **Pantalla:** es una pantalla *DotDisplay AMOLED* de 6.43 pulgadas con una resolución de 2400x1080.
- **Procesador:** *Qualcomm Snapdragon 678*.
- **Batería y carga:** 5000 mAh y 33 W.

DESARROLLO DE UNA APLICACIÓN DE REALIDAD AUMENTADA PARA TELÉFONO MÓVIL PARA LA  
LOCALIZACIÓN DE LA ARTERIA FEMORAL MEDIANTE SECUENCIAS DE ECOGRAFÍA

- **Cámara trasera:** cuenta con una cámara cuádruple de 48 MP + 8 MP + 2 MP + 2 MP.
- **Redes compatibles:** 4G, Wi-Fi y Bluetooth 5.0.
- **Sistema operativo:** MIUI 12 basado en *Android* 11.

### 3.2.5. Raspberry Pi 2

Este dispositivo fue utilizado en las últimas etapas puesto que al ser prácticamente un ordenador puede reemplazar a uno. Sus características son [30]:

- **Procesador** Broadcom BCM2836 de 900 MHz ARM Cortex-A7 de cuatro núcleos con GPU.
- La **GPU** cuenta con tecnología Open GL ES 2.0, *hardware* acelerado OpenVG y admite imágenes de alta resolución. Tiene una capacidad de 1 Gpixel/s con filtrado DMA.
- SDRAM LPDDR2 de 1 GB.
- **Salida de vídeo** HD 1080p y salida de vídeo compuesto.
- **Salida de audio** estéreo.
- Conector hembra *Ethernet*.
- Conector hembra de *HDMI* 1.3 y 1.4.
- Conector hembra de salida de vídeo compuesto de 3.5mm.
- 4 conectores hembra *USB* 2.0.
- Conector MPI CSI-2 de 15 vías.
- Conector para tarjeta MicroSD.
- **Sistema operativo** *Linux*.
- **Fuente de alimentación** de cinco voltios a dos amperios a través de conector hembra microUSB.
- **Dimensiones:** 86x56x20 mm.

### 3.2.6. Router

Para hacer las conexiones se utilizó un *router* *Netgear* N300 con dos cables *ethernet*, las características de este primero son [31]:

- **Modelo:** JWNR2010v3.
- Sin conexión a Internet.
- **Dimensiones:** 17.3x12x3.3 cm.
- **Procesador:** Intel.
- **Conexión inalámbrica:** 802.11b.
- **Protocolo de seguridad:** WPA2-PSK.
- **Número de puertos:** 5 *Fast Ethernet* (1 WAN y 4 LAN).
- **Ancho de banda de puerto LAN:** 10/100 megabits por segundo.
- **Velocidad de transferencia de datos:** 300 Megabits por segundo.

DESARROLLO DE UNA APLICACIÓN DE REALIDAD AUMENTADA PARA TELÉFONO MÓVIL PARA LA  
LOCALIZACIÓN DE LA ARTERIA FEMORAL MEDIANTE SECUENCIAS DE ECOGRAFÍA

### 3.2.7. Ecógrafo

- **Modelo:** Ecógrafo Philips EPIQ 7 con sonda lineal vascular L12-3 [32].
- Puede adquirir imágenes *nSIGHT*, es decir, crea la imagen con la resolución óptima por debajo del nivel de píxel [32].
- Incorpora la tecnología *PureWave* como transductor, lo que aumenta su eficiencia.
- Cuenta con *iSCAN* en tiempo real, lo que se traduce en una optimización de la imagen captada de forma automática [32].
- Al tener *Xmatrix*, tiene los modos de transducción 2D, 3D/4D, *Live xPlane*, *Live MPR*, *MPR*, *Doppler*, *Doppler a color* y *CPA* [32].
- Tiene función multimodalidad *DICOM*, puede integrar distintas modalidades para su visualización y comparación [32].
- Reconstrucción de haces de precisión en tiempo real, es capaz de realizar menos operaciones de transmisión para obtener una mayor tasa de fotogramas y cuenta con un ancho de banda muy amplio y una excepcional relación señal/ruido [33].

### 3.2.8. Sonda ultrasonidos

- **Modelo:** sonda lineal vascular L12-3 [34].
- **Frecuencia de operación:** 3 a 12 MHz [34].

## 4. MÉTODOS

### 4.1. Cubo AR

El primer paso que se realizó fue descargar *Unity* y *Unity Hub*, este último es un gestor de proyectos de Unity, además, de crear una cuenta de *Vuforia* y descargarlo.

Primero, se pensó que lo mejor sería utilizar un *target* cuboideo que se situaría encima de la sonda del ecógrafo mediante un dispositivo que se diseñaría y se imprimiría posteriormente. Para hacer las pruebas, se escogió un cubo prediseñado con características similares a las de un código QR y se construyó en *Vuforia* para formalizarlo como *target* (Figura 6).



Figura 6. Cubo realizado en Vuforia [<https://mergeedu.com/cube>].

Para configurar *Unity*, se creó, nada más entrar, un proyecto en *Unity* con la plantilla para teléfono móvil y tras realizar esto, lo primero que hay que cambiar es la plataforma, hay que seleccionar *Android* antes de realizar cualquier otra operación, esto es fundamental porque si no se cambia, luego no se va a poder crear la aplicación para un teléfono móvil con este sistema operativo.

Una vez que se ha realizado el paso anterior, se incorpora *Vuforia* al proyecto y se introduce la licencia. Posteriormente, se incorpora una cámara de Realidad Aumentada (*GameObject* > *VuforiaEngine* > *AR Camera*) y también un *MultiTarget* (*GameObject* > *VuforiaEngine* > *MultiTarget*) porque no es solamente una imagen, son seis. Para poder incluir el *MultiTarget* desde *Vuforia* se descarga la base de datos que contiene el *target* deseado y se importa. El cubo se situó en las coordenadas (0; 0; 0) para que sirviera como referencia.

A continuación, para poder reproducir un vídeo, se creó un plano (*GameObject* > *3D Object* > *Plane*) y se situó debajo y dentro del *MultiTarget* previamente creado (Figura 7), de esta forma, el plano sabe que debe aparecer al ver el *target* y seguirlo. Como es una prueba, se posicionó el plano encima de la cara del cubo que se puede ver seleccionada en la Figura 6, para ello, el cubo ocupa tres puntos de coordenadas, por lo que su coordenada se situó 0,1 puntos por encima para que no coincidiera exactamente con el cubo y no se viera; después se giró el plano 180º en el eje 'Y' para que se viera desde arriba y se escaló con estas cifras: (3,4361; 1,7158; 1,7158). Se eligió este tamaño con relación a las dimensiones del cubo.



DESARROLLO DE UNA APLICACIÓN DE REALIDAD AUMENTADA PARA TELÉFONO MÓVIL PARA LA  
LOCALIZACIÓN DE LA ARTERIA FEMORAL MEDIANTE SECUENCIAS DE ECOGRAFÍA

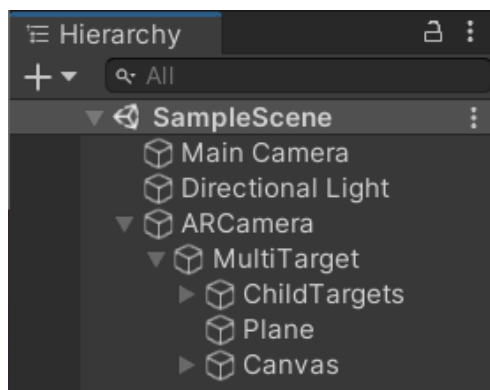


Figura 7. Jerarquía del trabajo con el cubo [Realización Propia].

Para poder reproducir un vídeo sobre el plano, se le incorporó un reproductor de vídeos de *Youtube* para ver si funcionaría reproducir vídeos que no estuvieran en la memoria del ordenador. En la Figura 8 se puede observar su implementación en *Unity*. Además, para que funcionara, hay que añadirle textura al plano, en este caso se usó un *Video Renderer*.

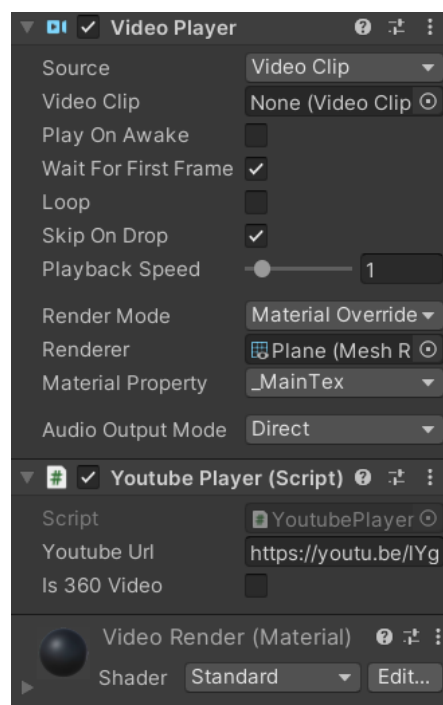


Figura 8. Implementación del reproductor de Youtube [Realización Propia].

Como se quiere que el plano deje de seguir al *target* cuando se pulse un botón, se crearon botones para ir probando cómo se realizaban y cómo se hacían funcionar. Para crearlos, es mejor hacerlo desde la ventana *Game* de *Unity*, porque si se crean en la escena, se pueden mover y no están fijos. El crearlo en esta primera ventana, lo hace más fácil porque se ponen las dimensiones deseadas y en la orientación (vertical u horizontal) que se quiere (como por ejemplo 2960x1440) y de esta forma se tiene una vista parecida a la del teléfono móvil. Todas estas configuraciones pueden ser observadas en la Figura 9.

DESARROLLO DE UNA APLICACIÓN DE REALIDAD AUMENTADA PARA TELÉFONO MÓVIL PARA LA  
LOCALIZACIÓN DE LA ARTERIA FEMORAL MEDIANTE SECUENCIAS DE ECOGRAFÍA

Al crearlos (*GameObject > UI > Button*) también se crea un *Canvas* que es donde se sitúan los botones. Para redimensionarlos y posicionarlos, simplemente hay que pulsar sobre ellos e introducir los datos (como se puede ver en la Figura 9) o se puede redimensionar manualmente con las herramientas de *Unity*. Además, se puede cambiar el texto y su tamaño al pulsar sobre la pestaña *Text*.



Figura 9. Dimensionamiento y posicionamiento de los botones [Realización Propia].

Para controlar botones se utilizó un *script*, que puede verse en el Código 1, que se les fue introducido a los botones como puede verse en la Figura 10. Se puede apreciar que hay cuatro botones y que el método para hacerlos es: primero creando dichos botones, luego haciendo que no sean interactivables nada más se inicie la aplicación sino que sean interactivables cuando el vídeo esté preparado y finalmente se crea una función para cada botón. Para poder darle funcionalidad a los botones, primeramente hay que añadir el *script* al proyecto, hay que añadirlo en el objeto que se quiere (en este caso en el *Canvas* que es donde están los botones) simplemente apretando en *Add Component* y seleccionando el *Script* deseado (Figura 11). Además, a este, hay que añadirle el plano donde está el *Youtube Player* y los botones a los sitios correspondientes como se puede observar en la Figura 11. Luego, en el botón mismo, en la sección *On Click* (Figura 10) se selecciona el sitio donde está el código (se recuerda al lector que está situado en el *Canvas*) y se añade la función que realiza el botón al ser pulsado.

DESARROLLO DE UNA APLICACIÓN DE REALIDAD AUMENTADA PARA TELÉFONO MÓVIL PARA LA  
LOCALIZACIÓN DE LA ARTERIA FEMORAL MEDIANTE SECUENCIAS DE ECOGRAFÍA

```
using UnityEngine; using UnityEngine.Video; using UnityEngine.UI;
namespace YoutubePlayer
{
    public class VideoControl : MonoBehaviour
    {
        public YoutubePlayer youtubePlayer;
        VideoPlayer videoPlayer;
        public Button BtPlay;
        public Button BtPause;
        public Button BtReset;

        private void Awake()
        {
            BtPlay.interactable = false;
            BtPause.interactable = false;
            BtReset.interactable = false;
            videoPlayer = youtubePlayer.GetComponent<VideoPlayer>();
            videoPlayer.prepareCompleted += VideoPlayerPreparedCompleted;
        }
        void VideoPlayerPreparedCompleted(VideoPlayer source)
        {
            BtPlay.interactable = source.isPrepared;
            BtPause.interactable = source.isPrepared;
            BtReset.interactable = source.isPrepared;
        }
        public async void Prepare()
        {
            print("Cargando video..");
            try
            {
                await youtubePlayer.PrepareVideoAsync();
                print("Video Cargado");
            }
            catch
            {
                print("ERROR video no cargado");
            }
        }
        public void PlayVideo()
        {
            videoPlayer.Play();
        }
        public void PauseVideo()
        {
            videoPlayer.Pause();
        }
        public void ResetVideo()
        {
            videoPlayer.Stop();
            videoPlayer.Play();
        }
        void OnDestroy()
        {
            videoPlayer.prepareCompleted -= VideoPlayerPreparedCompleted;
        }
    }
}
```

Código 1. Script para controlar el Reproductor de Youtube con los botones de Reproducir, Preparar, Resetear y Pausar  
[[https://youtu.be/Jn\\_Wxx24Yak](https://youtu.be/Jn_Wxx24Yak)].

DESARROLLO DE UNA APLICACIÓN DE REALIDAD AUMENTADA PARA TELÉFONO MÓVIL PARA LA  
LOCALIZACIÓN DE LA ARTERIA FEMORAL MEDIANTE SECUENCIAS DE ECOGRAFÍA

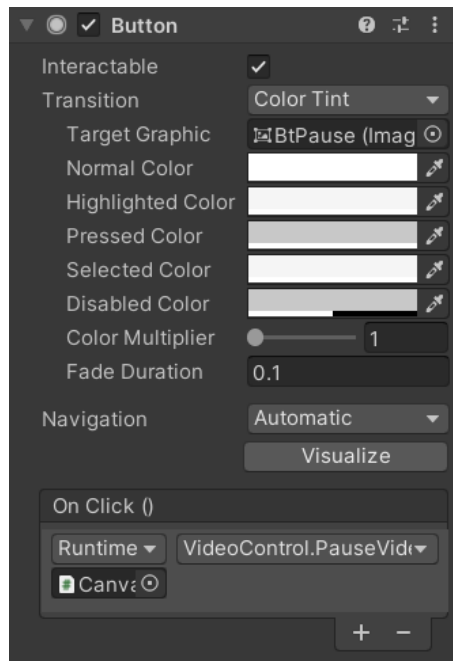


Figura 10. Poner la función al botón [Realización Propia].

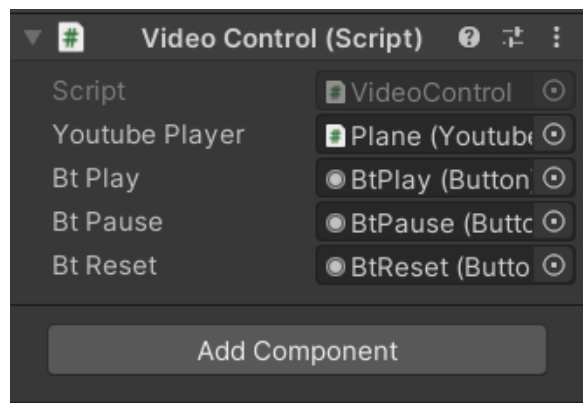


Figura 11. Añadir el script al proyecto de Unity [Realización Propia].

Como uno de los objetivos es tener botones que permitan al usuario parar de perseguir al *target* y se vuelva a seguirlo a conveniencia, dos botones fueron creados del mismo modo explicado anteriormente (se pueden observar en la Figura 9) y se les puso el Código 2:

```
using UnityEngine;
using UnityEngine.Video;
using UnityEngine.UI;
using UnityEngine.Events;
using Vuforia;
namespace YoutubePlayer
{
    public class VideoControl : MonoBehaviour
    {
        public YoutubePlayer youtubePlayer;
        VideoPlayer videoPlayer;
        public Button BtPlay;
        public Button BtPause;
```

DESARROLLO DE UNA APLICACIÓN DE REALIDAD AUMENTADA PARA TELÉFONO MÓVIL PARA LA  
LOCALIZACIÓN DE LA ARTERIA FEMORAL MEDIANTE SECUENCIAS DE ECOGRAFÍA

```
public Button BtReset;
public Button BtSTrack;
public Button BtTracking;

private void Awake()
{
    BtPlay.interactable = false;
    BtPause.interactable = false;
    BtReset.interactable = false;
    BtSTrack.interactable = false;
    BtTracking.interactable = false;
    videoPlayer = youtubePlayer.GetComponent<VideoPlayer>();
    videoPlayer.prepareCompleted += VideoPlayerPreparedCompleted;
}
void VideoPlayerPreparedCompleted(VideoPlayer source)
{
    BtPlay.interactable = source.isPrepared;
    BtPause.interactable = source.isPrepared;
    BtReset.interactable = source.isPrepared;
    BtSTrack.interactable = source.isPrepared;
    BtTracking.interactable = source.isPrepared;
}
public async void Prepare()
{
    print("Cargando video..");
    try
    {
        await youtubePlayer.PrepareVideoAsync();
        print("Video Cargado");
    }
    catch
    {
        print("ERROR video no cargado");
    }
}
public void PlayVideo()
{
    videoPlayer.Play();
}
public void PauseVideo()
{
    videoPlayer.Pause();
}
public void ResetVideo()
{
    videoPlayer.Stop();
    videoPlayer.Play();
}
public void StopTrack()
{
    TrackerManager.Instance.GetTracker<ObjectTracker>().Stop();
}
public void StartTrack()
{
    TrackerManager.Instance.GetTracker<ObjectTracker>().Start();
}
void OnDestroy()
{

```

DESARROLLO DE UNA APLICACIÓN DE REALIDAD AUMENTADA PARA TELÉFONO MÓVIL PARA LA  
LOCALIZACIÓN DE LA ARTERIA FEMORAL MEDIANTE SECUENCIAS DE ECOGRAFÍA

```
        videoPlayer.prepareCompleted -= VideoPlayerPreparedCompleted;  
    }  
}
```

Código 2. Script para controlar el Reproductor de Youtube con la adición de los botones para parar y restaurar el target  
[<https://stackoverflow.com/questions/23823580/how-do-i-stop-tracking-in-vuforia-for-unity>].

#### 4.2. Un *ImageTarget*

Como se comentará en el apartado de resultados del anterior paso, se decidió cambiar el cubo por una sola imagen, más concretamente por un código QR por su diseño suficientemente diferenciador y con características únicas. Para ello, se creó uno que puede ser observado en la Figura 12, se fotocopió con unas dimensiones 5x5 cm y se pegó en un trozo de papel doblado con dimensiones 17,5x7,5 cm que simulaba ser una sonda por el poco acceso a esta.

Las dimensiones no son al azar, de acuerdo con (*QR Code Tracking Overview - Mixed Reality | Microsoft Docs*, n.d.) el tamaño mínimo para que un código QR sea reconocido es de 5 cm y el máximo es de 10 cm [35], por lo que, por las dimensiones de la sonda, cinco centímetros es perfecto, puesto que más grande no cabría bien. Además, el dispositivo de captura debe de estar a una distancia cercana para poder detectar el código.

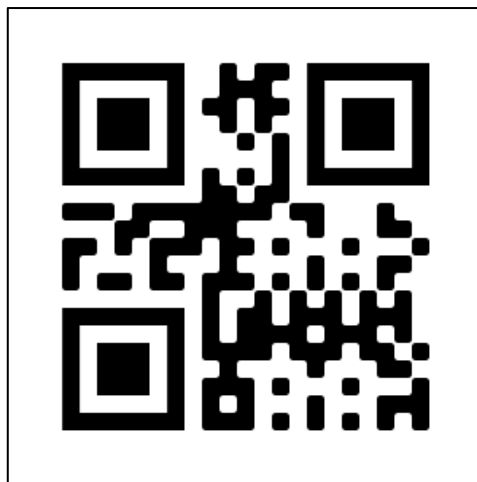


Figura 12. Código QR realizado [<https://es.qr-code-generator.com/>]

En cuanto a la estructura del proyecto de *Unity*, es básicamente la misma con algunas modificaciones que se comentan a continuación. La primera es que se pasó de un *MultiTarget* a un *ImageTarget* (*GameObject* > *VuforiaEngine* > *ImageTarget*) e igualmente hay que descargarse e importar la base de datos que contiene la imagen que se quiere de *Vuforia* y se tiene que introducir en el *ImageTarget* tal y como se puede observar en la Figura 13 (al seleccionar el *ImageTarget* aparece la opción descrita en la imagen).

DESARROLLO DE UNA APLICACIÓN DE REALIDAD AUMENTADA PARA TELÉFONO MÓVIL PARA LA LOCALIZACIÓN DE LA ARTERIA FEMORAL MEDIANTE SECUENCIAS DE ECOGRAFÍA

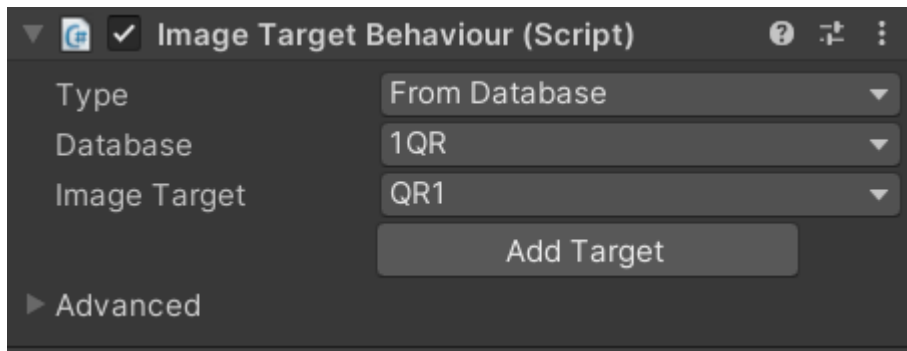


Figura 13. Añadir base de datos de Vuforia a Unity. El nombre de la base de datos y el nombre de la imagen son las que he especificado en Vuforia [Realización Propia].

El siguiente paso, fue crear el plano donde se reproducirá el vídeo (se crea de la misma forma que se ha especificado en el apartado 4.1) pero se le realizaron algunas modificaciones para adaptarlo a lo que se desea (Figura 14). Para ello, se separó el plano del código QR, para que no se reproduzca encima de este último, sino para que se reproduzca en la piel del paciente, para ello, se midió la distancia desde donde se pondría la pegatina (el centro de esta) hasta la punta de la sonda (con las dimensiones de la 'sonda' de prueba, es decir, el trozo de papel) y se convirtió esta distancia a coordenadas de *Unity* mediante sencillas reglas de tres. Además, se rotó 90° en el eje 'X' de *Unity* para que estuviera paralelo a la punta de la sonda porque en la realidad, la parte que toca al paciente hace un ángulo aproximado de 90°; también se rotó 180° en el eje 'Z' para que la imagen se viera hacia el teléfono, hacia los ojos. En cuanto a las dimensiones, se le dieron unas provisionales hasta que se tuviesen las medidas exactas del tamaño del vídeo y fueron elegidas en relación con el tamaño de la pegatina. Todas estas transformaciones se pueden ver en la Figura 14 y en la Figura 15 que muestra la escena correspondiente a todo lo que se ha nombrado.

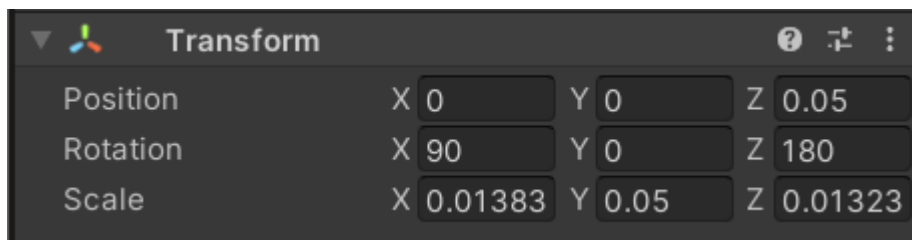


Figura 14. Transformaciones del plano [Realización Propia].

DESARROLLO DE UNA APLICACIÓN DE REALIDAD AUMENTADA PARA TELÉFONO MÓVIL PARA LA  
LOCALIZACIÓN DE LA ARTERIA FEMORAL MEDIANTE SECUENCIAS DE ECOGRAFÍA



Figura 15. Escena de Unity con un código QR [Realización Propia].

Seguidamente, se crearon los botones y se les dieron funcionalidad, este paso se realiza exactamente como se ha descrito en el apartado 4.1. Lo que cambia un poco es donde poner el *script* que controla los botones, puesto que está todo en uno mismo. Los botones para controlar el seguimiento del *target*, su *script* se puso en la cámara AR porque parecía ser lo adecuado al ser la cámara la que reconoce el *ImageTarget* y si se le dice a esta que deje de seguirlo, lo hará. Mientras que para los otros botones se deben poner en el *Canvas*. Por tanto, la estructura de esta parte quedaría como se puede ver en la Figura 16.

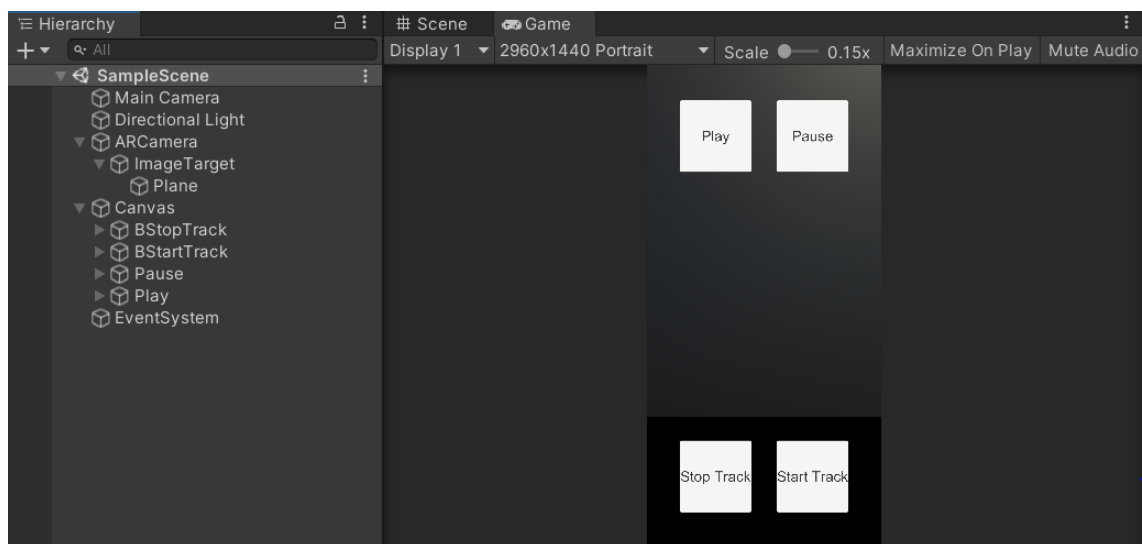


Figura 16. Estructura y previsualización del proyecto. En él se pueden ver cuatro botones: arriba a la izquierda el botón para empezar a reproducir, arriba a la derecha el botón para pausar el vídeo, abajo a la izquierda el botón para dejar de seguir al *target* y abajo a la derecha el botón para volver a seguir al *target* [Realización Propia].



DESARROLLO DE UNA APLICACIÓN DE REALIDAD AUMENTADA PARA TELÉFONO MÓVIL PARA LA  
LOCALIZACIÓN DE LA ARTERIA FEMORAL MEDIANTE SECUENCIAS DE ECOGRAFÍA

Una cosa que se implementó en esta parte que no en la anterior, fue cambiar los colores de los botones, porque se hacía un poco complicado diferenciar si estaba activo, pulsado, etc. Por ello, como se puede apreciar en la Figura 17, el color normal es blanco, el color cuando se pasa por encima de él (sin pulsar) es rojo, mientras se presiona cambia a un gris oscuro, cuando está presionado es de un color rosa y si está inhabilitado es de un gris claro.

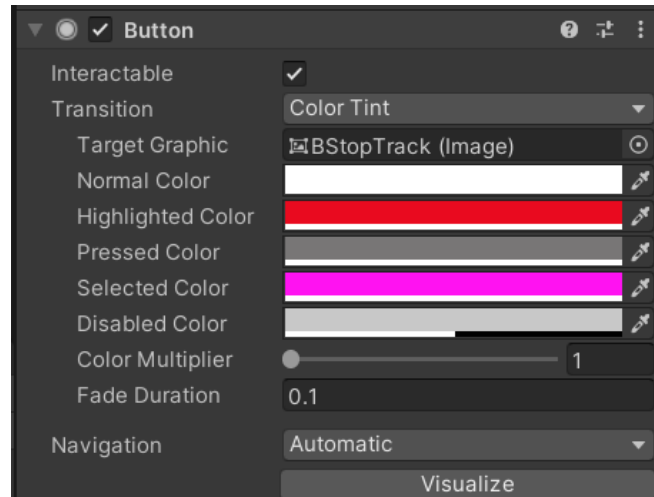


Figura 17. Colores de los botones [Realización Propia].

#### 4.3. Más de un *ImageTarget*

Se pensó en añadir más códigos QR para que en caso de que se perdiese la visión de uno, los otros suplieran a este, es decir, que el vídeo siguiera a diversos *targets*. Para ello, se crearon tres códigos QR más a parte del de la Figura 12 y se les dieron nombre (se pueden observar en la Figura 18) y se construyeron distintos proyectos en los que se variaron el número de estos. Primero se realizó con dos, luego con tres y por último con cuatro.

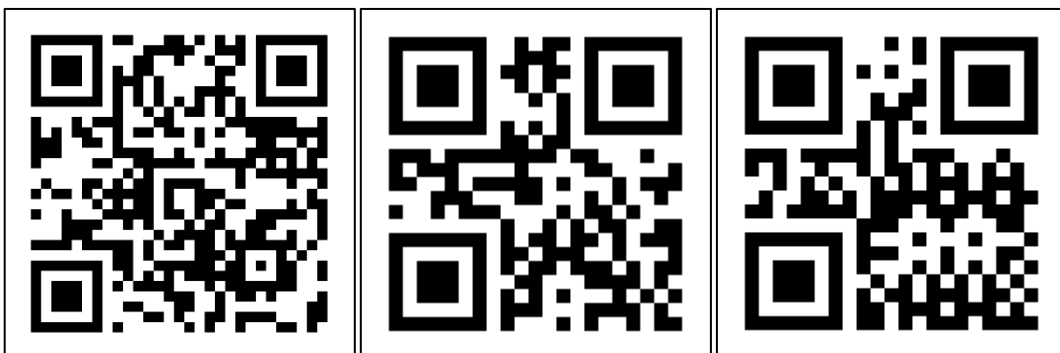


Figura 18. Códigos QR utilizados en la ampliación. De izquierda a derecha: QR2, QR3 y QR4 [<https://es.qr-code-generator.com/>] [<https://www.the-qr-code-generator.com/es/>] [<https://qrcode.tec-it.com/es/>].

DESARROLLO DE UNA APLICACIÓN DE REALIDAD AUMENTADA PARA TELÉFONO MÓVIL PARA LA  
LOCALIZACIÓN DE LA ARTERIA FEMORAL MEDIANTE SECUENCIAS DE ECOGRAFÍA

Primero se realizó con dos códigos QR, se utilizó el primero descrito en la sección 4.2 en la posición descrita y el QR2 a diez centímetros de el QR1 en horizontal, puesto que de esta forma toda la cara superior de la sonda estaría cubierta por algún *target*.

Para trasladar todo esto a *Unity*, se hace como se ha comentado previamente, pero cambiando algunas cosas. Primero, se deben crear dos *ImageTarget* y darle a cada uno la imagen que se decida. Segundo, para hacer que el plano siga a los *target*, se introdujo uno dentro del otro y el plano dentro del primero, a la misma altura que el QR2, como se puede observar en la Figura 19.

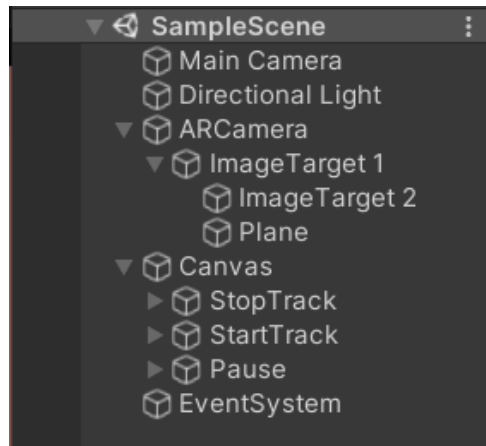


Figura 19. Estructura del proyecto con dos targets [Realización Propia].

Como el plano está situado en el mismo nivel que para un solo *target*, no hace falta cambiar nada. Finalmente, de los botones tampoco se cambió nada, puesto que este subapartado no se enfoca en la funcionalidad de los botones. Por lo que la escena queda como se puede apreciar en la Figura 20.

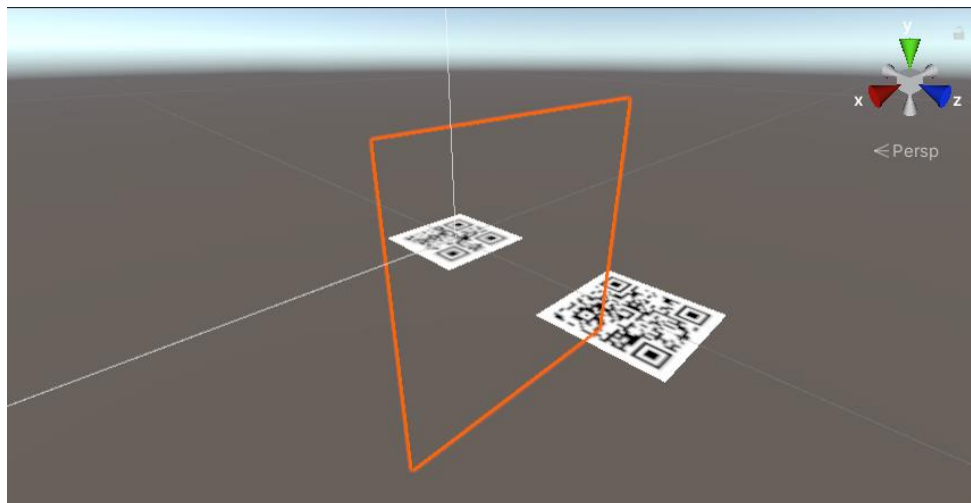


Figura 20. Escena del proyecto con dos códigos QR [Realización Propia].

DESARROLLO DE UNA APLICACIÓN DE REALIDAD AUMENTADA PARA TELÉFONO MÓVIL PARA LA  
LOCALIZACIÓN DE LA ARTERIA FEMORAL MEDIANTE SECUENCIAS DE ECOGRAFÍA

Ahora se prueba con tres, los dos primeros más el QR3, el QR1 y el QR2 se sitúan en la misma posición descrita para dos *targets* y el último se posiciona al final de la sonda, a 15 cm o 0,15 en *Unity* del origen de coordenadas (el centro del primer QR) y se gira 90º para que mire hacia el exterior (esta distribución puede ser apreciada en la Figura 21). Por lo que respecta a la estructura, es similar a la anterior solo hay que añadir el QR3 dentro del *ImageTarget 1*, tal y como está el dos (Figura 22).

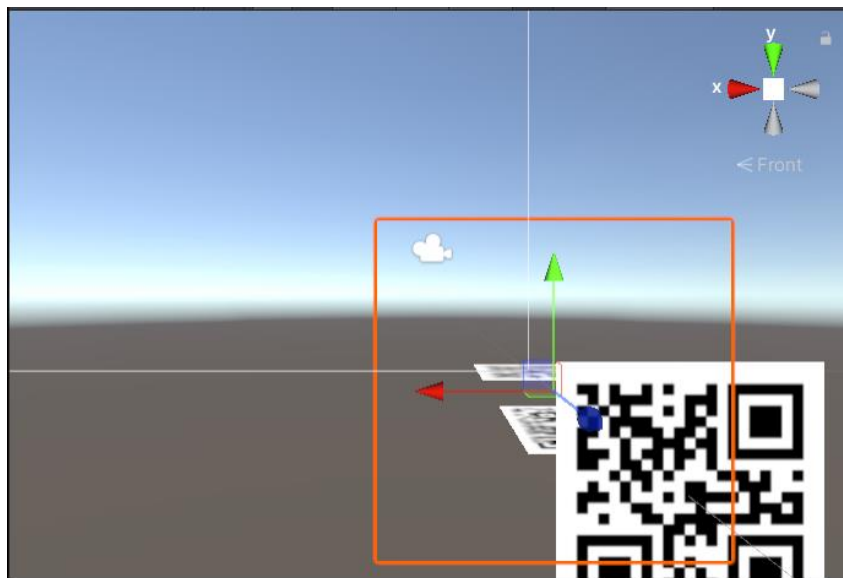


Figura 21. Escena proyecto tres QR [Realización Propia].

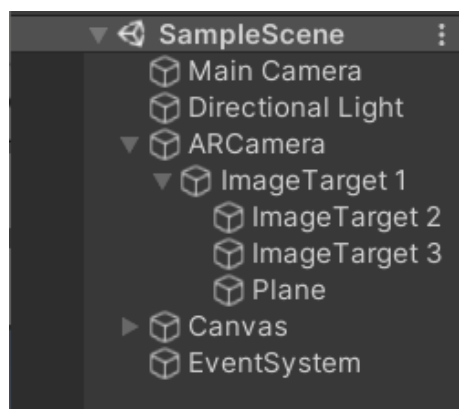


Figura 22. Estructura del proyecto con tres QR [Realización Propia].

Ahora, se probó con cuatro imágenes, para esto, se utilizó todos los códigos QR de la Figura 18. El primero y el segundo se dejaron iguales, el tercero se bajó un poco para que el inicio de la pegatina coincidiera con las otras y el cuarto se puso entre los dos primeros, más concretamente a 4 cm del origen de coordenadas (se recuerda que es el centro del *ImageTarget* número uno) y a la misma altura que el tercero (en la coordenada -0,025 del eje 'Y' en *Unity*) y movido en el eje 'X' para que esté al borde de la sonda, en su lateral izquierdo (en la coordenada 0,035 en *Unity*) (Figura 23). En cuanto a la estructura, simplemente hay que añadir el *ImageTarget 4* a la altura del dos y del tres (Figura 24).

DESARROLLO DE UNA APLICACIÓN DE REALIDAD AUMENTADA PARA TELÉFONO MÓVIL PARA LA LOCALIZACIÓN DE LA ARTERIA FEMORAL MEDIANTE SECUENCIAS DE ECOGRAFÍA

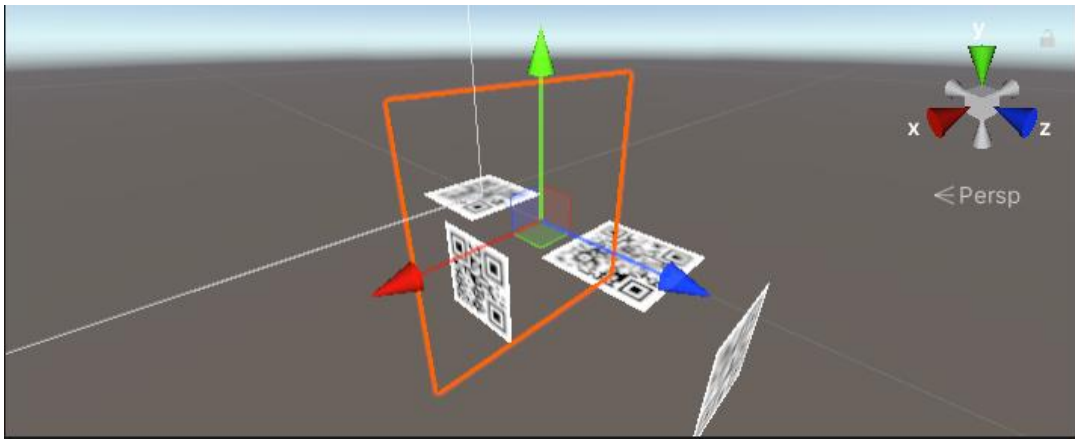


Figura 23. Escena del proyecto con cuatro QR [Realización Propia].

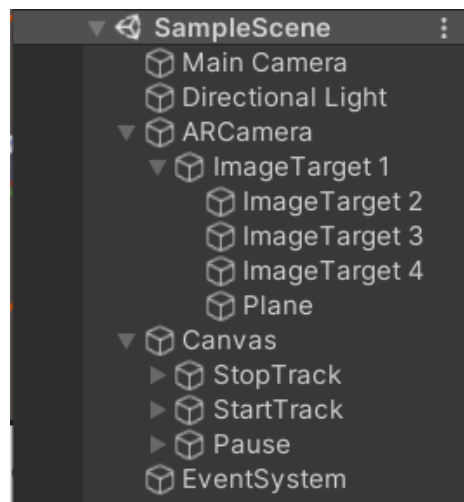


Figura 24. Estructura del proyecto con cuatro QR [Realización Propia].

A continuación, se prueba otro método para reproducir el vídeo con cuatro códigos QR, para ello se realizó lo mismo que con el otro método con el mismo número de imágenes, es decir, se cogieron los cuatro *ImageTargets* en la misma posición y se cambió un poco el proyecto. En vez de usar un solo plano para todas las imágenes, se utilizó uno para cada una y se intentó que se superpusiesen. La teoría es que solo habrá un plano que siempre se mostrará, independientemente de cuántos *targets* haya. Como hay que superponer cada plano y las coordenadas del plano son respecto a las coordenadas de su *ImageTarget* se complica un poco realizar esta tarea. Las coordenadas de cada imagen con las de sus respectivos planos se pueden observar en la Tabla 1, la escena se puede observar en la Figura 25 y la estructura en la Figura 26.

DESARROLLO DE UNA APLICACIÓN DE REALIDAD AUMENTADA PARA TELÉFONO MÓVIL PARA LA LOCALIZACIÓN DE LA ARTERIA FEMORAL MEDIANTE SECUENCIAS DE ECOGRAFÍA

Tabla 1. Coordenadas (X; Y; Z) del proyecto de cuatro QR método b donde IT corresponde a ImageTarget y P corresponde a Plano [Realización Propia].

Objeto	Posición (X; Y; Z)	Rotación (X; Y; Z)
IT 1	(0; 0; 0)	(0; 0; 0)
P 1	(0; 0; 0,5)	(90; 0; 180)
IT 2	(0; 0; 0,1)	(0; 0; 0)
P 2	(0; 0; 0,15)	(90; 0; 180)
IT 3	(0; -0,025; 0.15)	(90; 0; 0)
P 3	(0; -0,15; -0.025)	(0; 0; 180)
IT 4	(0,035; -0,025; 0,04)	(0; 0; -90)
P 4	(-0,025; -0,035; 0,01)	(90; 0; 180)

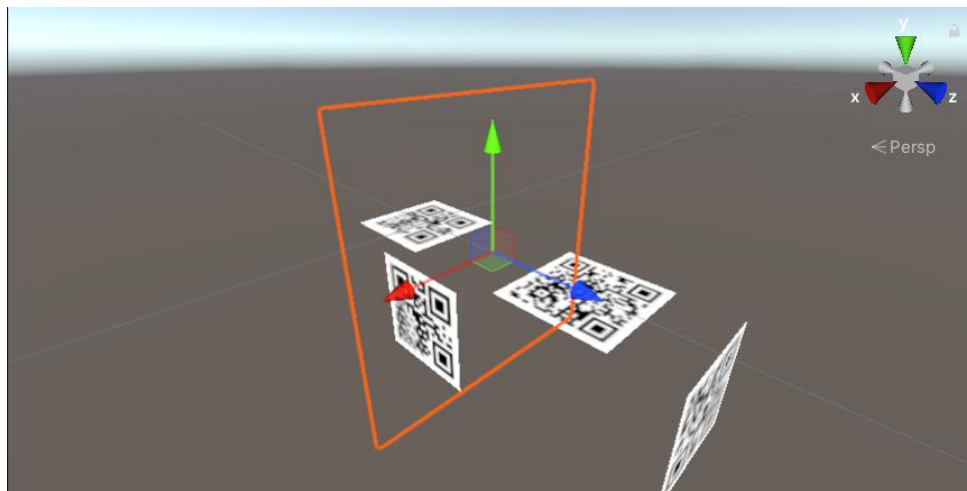


Figura 25. Escena del proyecto con cuatro QR método 2 [Realización Propia].

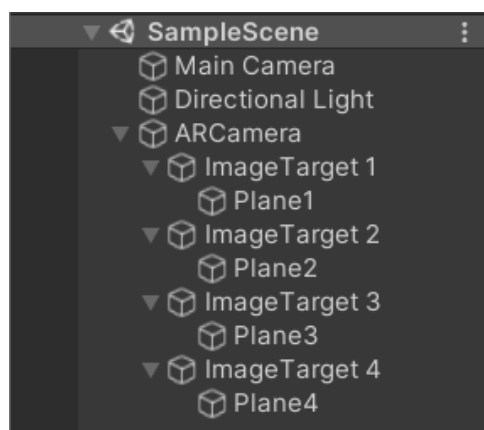


Figura 26. Estructura del proyecto con cuatro QR método 2 [Realización Propia].

#### 4.4. Reproducción del vídeo: forma 1

Para poder reproducir el vídeo del ecógrafo en el plano en tiempo real se aplicó el siguiente método. Con la capturadora de vídeo, se coge la señal del ecógrafo a través de un *HDMI* y se pasa al ordenador a través del puerto *USB*; una vez que llega se recoge por *OBS* y se retransmite. Luego, *Unity* la capta y la reproduce donde es deseada. A continuación, se describe más detalladamente paso a paso.

Primeramente, para probar que se captura bien la señal y se reproduce bien, lo primero que se hizo fue intentar captar la señal de otro ordenador portátil (fuente de señal *HDMI*) y reproducirla en el ordenador mediante *VLC* porque este es capaz de reproducir vídeo en *stream*. Los pasos a realizar son: primero configurar el *OBS* para que recoja la señal; para ello, se aprieta el signo '+' de Fuentes y se selecciona Dispositivo de Captura de Video, luego se selecciona crear uno nuevo, se le da nombre y en la posterior pestaña en dispositivo se elige *USB*. De esta forma, el dispositivo es reconocido de forma inmediata y se reproduce a pantalla completa, se deja de esta forma puesto que interesa que ocupe toda la pantalla.

Segundo, hay que configurar la salida, la retransmisión, para ello, lo que se hace es retransmitirlo por *rtmp* que es un protocolo de transmisión que nos permite hacer *stream* en una red local. Se hace de este modo puesto que no interesa que la información llegue a Internet y pueda ser accesible a otros usuarios ya que no deja de ser información médica que es confidencial. Para configurarlo, simplemente en Servicio Personalizado introducimos la siguiente línea en el servidor (no es necesaria una clave de retransmisión): `rtmp://localhost/live` y esta misma es la que hay que introducir en *VLC*. Pero antes de iniciar la transmisión, hay que iniciar *nginx* que proporciona el servidor *live*.

El siguiente paso para ver si funciona, es poder reproducir esta misma señal en el *VLC* para el teléfono móvil, simplemente hay que introducirle la misma línea (`rtmp://localhost/live`) pero cambiando *localhost* por la dirección IP del ordenador que retransmite, y estar conectado al mismo *router*.

Para poder hacerlo por *Unity*, se descargó el *plugin* de *VLC* de *Unity* desde la *asset store* y se importó. La estructura del proyecto es la misma que el de un solo QR por los resultados obtenidos de los apartados 4.2 y 4.3.

Al introducir el *VLC* a *Unity*, se añade en el plano y se introducen los objetos que se pueden ver en la Figura 27. Donde la pantalla es el plano, con esto se indica donde se va a reproducir el vídeo; luego, la textura es el *VideoRender* que, como se ha comentado antes, permite al plano poder reproducir vídeo y finalmente el camino es la misma dirección que ha sido introducida anteriormente en el teléfono móvil.

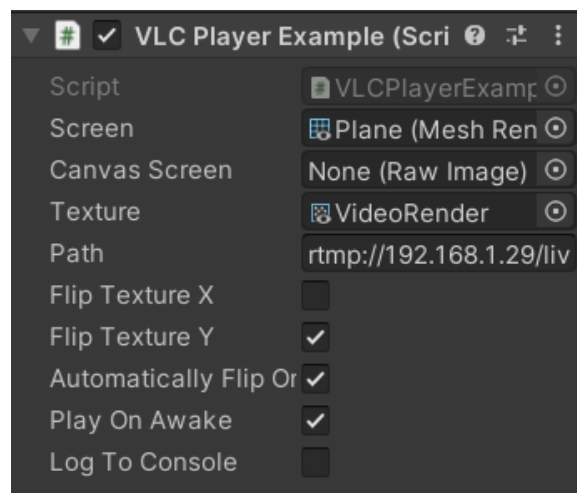
DESARROLLO DE UNA APLICACIÓN DE REALIDAD AUMENTADA PARA TELÉFONO MÓVIL PARA LA  
LOCALIZACIÓN DE LA ARTERIA FEMORAL MEDIANTE SECUENCIAS DE ECOGRAFÍA

Figura 27. Implementación del VLC en Unity [Realización Propia].

Como no funcionó, se instaló otro *plugin* para enviar vídeo para probar si de otra forma funciona, se trata de *UMP* para *iOS* y *Android* que es un reproductor universal de media codificado en *C* y basado en *VLC* y *FFmpeg*, además puede reproducir vídeo en el editor de *Unity* y tiene una fácil, rápida y flexible implementación. También puede reproducir vídeos de *Youtube*, la mayoría de los formatos de vídeo y vídeos en directo [36]. Para probar que era efectivo, se creó un proyecto nuevo (para que otros *plugins* no interfieran) y se puso una escena de ejemplo.

#### 4.5. Reproducción del vídeo: forma 2

Para solucionar el problema del retraso, se creó una red mediante un *router* y una *raspberrypi*, dado que, de esta forma, no se tendría una codificación mediante *software*, sino que se tendría una mediante *hardware* que es más rápida. Lo primero para hacerlo, es conectar el *router* mediante *ethernet* al ordenador y a la *raspberrypi*, luego se une el ordenador a la *raspberrypi* mediante el *HDMI* y la capturadora para poder coger la señal (no es lo ideal, pero se hace de esta forma porque no hay más medios) y se enchufan tanto el *router* como la *raspberrypi* a la fuente de alimentación.

Segundo, se quiere que la *raspberrypi* tenga una dirección IP fija para que no se tenga que cambiar la dirección de la retransmisión en vivo (se recalca al lector que ahora el que se encarga de retransmitir en una red local la señal de vídeo es este dispositivo) para ello se entra en la configuración del *router* y su IP es fijada a la que este le da originalmente: 192.168.1.2.

Tercero, se accede a la *raspberrypi* a través de *PuTTY* poniendo la dirección IP y dejando el puerto 22. Una vez dentro, después de haber puesto el usuario y la contraseña, se usa *ffmpeg* para coger el vídeo y retransmitirlo. Para comenzar, se usa la línea de código que puede verse en Código 3.

DESARROLLO DE UNA APLICACIÓN DE REALIDAD AUMENTADA PARA TELÉFONO MÓVIL PARA LA  
LOCALIZACIÓN DE LA ARTERIA FEMORAL MEDIANTE SECUENCIAS DE ECOGRAFÍA

```
ffmpeg -f v4l2 -video_size 640x480 -i /dev/video0 -codec:v h264_omx -b:v 512k -f matroska  
- | cvlc --sout '#http{mux=ffmpeg{mux=flv},dst=:8000/}'
```

Código 3. Línea de código de ffmpeg usada [<https://forums.raspberrypi.com/viewtopic.php?t=256572>]

Donde la barra vertical '|' separa la entrada de la salida. El resto de los parámetros significan:

- **-f:** para forzar el formato del archivo de entrada o de salida. Normalmente es detectado automáticamente para los de entrada y para los de salida se adivina a partir de su extensión, por lo que esta opción no es necesaria en la mayoría de los casos, pero se ha puesto para asegurarnos [22].
- **V4l2:** es una API de captura de vídeo para Linux.
- **-video\_size:** para indicar las dimensiones del vídeo de entrada/salida, en este caso se ha establecido en 640x480 porque parecía una adecuada dimensión que no consume tiempo [22].
- **-i:** para indicar que a continuación se introduce la URL correspondiente al vídeo de entrada [22].
- **Dev:** dispositivo de entrada. **Video0:** nombre de la capturadora de vídeo (el actual dispositivo de entrada).
- **-codec:** muestra todos los códecs conocidos por 'liavcodec', se utiliza para indicar el flujo de bits de los medios de comunicación [22].
- **-codec:v:** de esta forma se configura el códec de vídeo, se le dice de que la entrada es de este tipo [22].
- **H264\_omx:** codificación en *hardware* de la *raspberrypi* (no se puede cambiar porque viene por defecto y si se hace no funciona) [22].
- **-b:v 512k:** para ajustar la tasa de bits de vídeo del archivo de salida, en este caso es 512 kbits/s y se utiliza esta porque es la más pequeña posible para que no sucedan cosas extrañas [22].
- **Matroska:** es un formato contenedor abierto, un archivo informático que puede contener una cantidad muy grande de vídeo dentro de un solo archivo (al igual que pasa con h264\_omx, este parámetro no se puede cambiar) [37].
- **Cvlc:** se utiliza para hacer vídeo LAN, para hacer *stream* en red local. Es el *VLC* y debe ser este porque *Unity* lo utiliza para reproducir vídeos.
- **Sout:** para indicar la ruta de salida de la señal [22].
- **http:** protocolo de transferencia utilizado.
- **Mux:** demultiplexación. **Flv:** tipo de archivo de salida (vídeo *flash*) [22].
- **Dst =:8000:** puerto de salida.

Para comprobar que funcionaba, se introdujo la dirección en *VLC* (<http://192.168.1.2:8000>).



DESARROLLO DE UNA APLICACIÓN DE REALIDAD AUMENTADA PARA TELÉFONO MÓVIL PARA LA  
LOCALIZACIÓN DE LA ARTERIA FEMORAL MEDIANTE SECUENCIAS DE ECOGRAFÍA

A continuación, para poder disminuir latencia, se quitó el audio mediante **-an** porque no es necesario tener audio en la aplicación, también se bajó el caché de VLC (cuanto más caché, más memoria y más retraso). Para poder bajar aún más el retraso, se hicieron distintas pruebas con distintos parámetros y se midió el retraso. Se probaron primeramente uno a uno y estos pueden ser observados Tabla 2.

Tabla 2. Parámetros del ffmpeg utilizados (en negrita están los introducidos) y sus significados [Realización Propia].

Parámetro	Significado
ffmpeg <b>-fflags nobuffer</b> -f v4l2 -video_size 640x480 -an -i /dev/video0 -codec:v h264_omx -b:v 512k -f matroska -   cvlc - --sout '#http{mux=ffmpeg{mux=flv},dst=:8000/}'	Reduce la latencia introducida por el almacenamiento en búfer durante el análisis inicial de los flujos de entrada [38].
ffmpeg <b>-flags low_delay</b> -f v4l2 -video_size 640x480 -an -i /dev/video0 -codec:v h264_omx -b:v 512k -f matroska -   cvlc - --sout '#http{mux=ffmpeg{mux=flv},dst=:8000/}'	Fuerza un retraso bajo [38].
ffmpeg <b>-fflags nobuffer -flags low_delay</b> -f v4l2 -video_size 640x480 -an -i /dev/video0 -codec:v h264_omx -b:v 512k -f matroska -   cvlc - --sout '#http{mux=ffmpeg{mux=flv},dst=:8000/}'	
ffmpeg <b>-itsoffset offset</b> -f v4l2 -video_size 640x480 -an -i /dev/video0 -codec:v h264_omx -b:v 512k -f matroska -   cvlc - --sout '#http{mux=ffmpeg{mux=flv},dst=:8000/}'	Ajusta el desplazamiento del tiempo de entrada [22]. Si el valor es negativo, adelanta los flujos de entrada esa cantidad de segundos.
ffmpeg <b>-max_delay delay</b> -f v4l2 -video_size 640x480 -an -i /dev/video0 -codec:v h264_omx -b:v 512k -f matroska -   cvlc - --sout '#http{mux=ffmpeg{mux=flv},dst=:8000/}'	Impone el retraso máximo de reordenación [39].
ffmpeg <b>-thread_queue_size size</b> -f v4l2 -video_size 640x480 -an -i /dev/video0 -codec:v h264_omx -b:v 512k -f matroska -   cvlc - --sout '#http{mux=ffmpeg{mux=flv},dst=:8000/}'	Fuerza a que algunos fotogramas no sean leídos y se aplica a la entrada posterior [40].
ffmpeg -f v4l2 -video_size 640x480 -an - <b>thread_queue_size size</b> -i /dev/video0 -codec:v h264_omx -b:v 512k -f matroska -   cvlc - --sout '#http{mux=ffmpeg{mux=flv},dst=:8000/}'	

DESARROLLO DE UNA APLICACIÓN DE REALIDAD AUMENTADA PARA TELÉFONO MÓVIL PARA LA  
LOCALIZACIÓN DE LA ARTERIA FEMORAL MEDIANTE SECUENCIAS DE ECOGRAFÍA

ffmpeg -thread_queue_size size -f v4l2 -video_size 640x480 -an -thread_queue_size size -i /dev/video0 -codec:v h264_omx -b:v 512k -f matroska -   cvlc - --sout '#http{mux=ffmpeg{mux=flv},dst=:8000/}'	
ffmpeg -f v4l2 -video_size 640x480 -an -i /dev/video0 -codec:v h264_omx -tune zerolatency -b:v 512k -tune zerolatency -f matroska -   cvlc - --sout '#http{mux=ffmpeg{mux=flv},dst=:8000/}'	Fuerza que la latencia sea cero.
ffmpeg -f v4l2 -video_size 640x480 -an -i /dev/video0 -codec:v h264_omx -tune zerolatency -b:v 512k -f matroska -   cvlc - --sout '#http{mux=ffmpeg{mux=flv},dst=:8000/}'	
ffmpeg -f v4l2 -video_size 640x480 -an -i /dev/video0 -codec:v h264_omx -preset velocity -b:v 512k -f matroska -   cvlc - --sout '#http{mux=ffmpeg{mux=flv},dst=:8000/}'	Proporciona la relación entre la velocidad de compresión y la de codificación [41].

Posteriormente, se unieron aquellos parámetros que más disminuían la latencia para ver si reducía aún más y, por tanto, el código que quedaría sería el que puede leerse en Código 4.

```
ffmpeg -flags nobuffer -itsoffset -0.0001 -thread_queue_size 320 -f v4l2 -video_size 640x480 -an -thread_queue_size 320 -i /dev/video0 -codec:v h264_omx -tune zerolatency -preset superfast -b:v 512k -tune zerolatency -f matroska - | cvlc - --sout '#http{mux=ffmpeg{mux=flv},dst=:8000/}'
```

Código 4. Línea de código de ffmpeg con los parámetros unidos [Realización Propia].

Cabe destacar que se ha intentado reducir el *framerate* para que así la latencia fuera más baja, para poder hacer esto, la única forma que parecía funcionar era usar **-r** en la parte de entrada, después de especificar el tamaño del vídeo y **fps=fps=número** en la parte de salida después de cvlc. En ambos comandos hay que poner la misma cantidad, por defecto es 30 fps. Se cambia a 12 y a 25.

Además, se prueban los parámetros de la Tabla 3 para intentar reducir aún más la latencia, se prueban solos con el código de partida y con el Código 4.

DESARROLLO DE UNA APLICACIÓN DE REALIDAD AUMENTADA PARA TELÉFONO MÓVIL PARA LA LOCALIZACIÓN DE LA ARTERIA FEMORAL MEDIANTE SECUENCIAS DE ECOGRAFÍA

Tabla 3. Parámetros del ffmpeg utilizados para disminuir la latencia aún más (en negrita están los introducidos) y sus significados [Realización Propia].

Parámetro	Significado
<code>Ffmpeg -fflags nobuffer -flags low_delay \ <b>-strict experimental</b> -f v4l2 -video_size 640x480 -an -i /dev/video0 -codec:v h264_omx -b:v 512k -f matroska -   cvlc - --sout '#http{mux=ffmpeg{mux=flv},dst=:8000/}'</code>	<i>-strict</i> especifica como seguir estrictamente los estándares y con <i>experimental</i> permite realizar cosas no estandarizadas y utilizar decodificadores y codificadores experimentales [38].
<code>ffmpeg -flags nobuffer <b>-flags low_delay \ -strict experimental</b> -itsoffset -0.0001 -thread_queue_size 320 -f v4l2 -video_size 640x480 -an -thread_queue_size 320 -i /dev/video0 -codec:v h264_omx -tune zerolatency -preset superfast -b:v 512k -tune zerolatency -f matroska -   cvlc - --sout '#http{mux=ffmpeg{mux=flv},dst=:8000/}'</code>	
<code>Ffmpeg <b>-fflags nobuffer -fflags discardcorrupt -flags low_delay \ -avioflags direct</b> -f v4l2 -video_size 640x480 -an -i /dev/video0 -codec:v h264_omx -b:v 512k -f matroska -   cvlc - --sout '#http{mux=ffmpeg{mux=flv},dst=:8000/}'</code>	Reduce el <i>buffering</i> de forma agresiva [38].
<code>ffmpeg -flags nobuffer <b>-fflags discardcorrupt -flags low_delay \ -avioflags direct</b> -itsoffset -0.0001 -thread_queue_size 320 -f v4l2 -video_size 640x480 -an -thread_queue_size 320 -i /dev/video0 -codec:v h264_omx -tune zerolatency -preset superfast -b:v 512k -tune zerolatency -f matroska -   cvlc - --sout '#http{mux=ffmpeg{mux=flv},dst=:8000/}'</code>	

También, cabe destacar que se ha intentado cambiar el protocolo de comunicación a *rtmp* o *rtsp* para ver si así la latencia se reduce.

Una vez implementado todo esto, se debe incorporar a *Unity*. Para ello, se crea un nuevo proyecto y se le dota de lo básico descrito en los apartados previos (una cámara AR, un *ImageTarget* correspondiente al QR1, un plano y el *Canvas* con los botones para parar de seguir al *target* y para reanudar su seguimiento). Además, se le incorpora el *plugin UMP* usado en la sección 4.4 y se le indica donde debe reproducir el contenido (el plano) y el *link* de donde obtenerlo (<http://192.168.1.2:8000>) tal y como se puede observar en la Figura 28. Por tanto, la estructura final del proyecto comentado se puede observar más claramente en la Figura 29.

DESARROLLO DE UNA APLICACIÓN DE REALIDAD AUMENTADA PARA TELÉFONO MÓVIL PARA LA LOCALIZACIÓN DE LA ARTERIA FEMORAL MEDIANTE SECUENCIAS DE ECOGRAFÍA

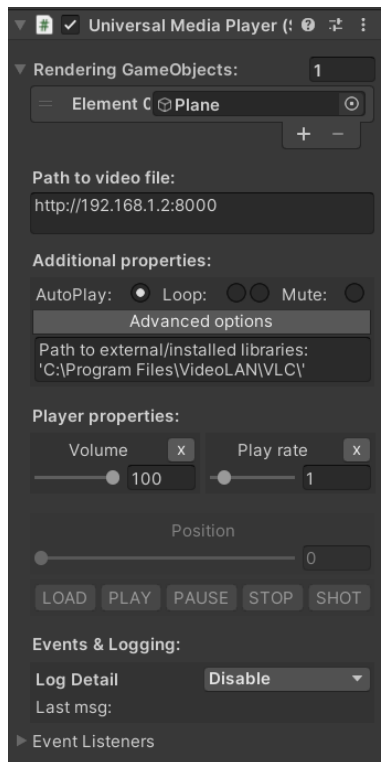


Figura 28. Implementación del Plugin UMP en el proyecto [Realización Propia].

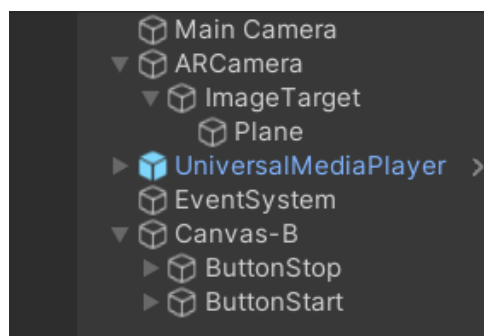


Figura 29. Estructura del proyecto final [Realización Propia].

Finalmente, para la funcionalización de los botones, se creó un nuevo código que puede ser observado en el Código 5. Básicamente es una reducción de los códigos anteriores puesto que ahora se tienen menos botones y no es necesario un reproductor de *Youtube*. Además, el componente con el *script* se añade a la cámara y se procede como en el apartado 4.2.

DESARROLLO DE UNA APLICACIÓN DE REALIDAD AUMENTADA PARA TELÉFONO MÓVIL PARA LA  
LOCALIZACIÓN DE LA ARTERIA FEMORAL MEDIANTE SECUENCIAS DE ECOGRAFÍA

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.Video;
using UnityEngine.UI;
using UnityEngine.Events;
using Vuforia;

public class ButtonControls : MonoBehaviour
{
    public Button BtStopTracker;
    public Button BtStartTracker;

    // Start is called before the first frame update
    void Start()
    {
        BtStartTracker.interactable = false;
        BtStopTracker.interactable = false;
    }
    // Update is called once per frame
    void Update()
    {
        BtStartTracker.interactable = true;
        BtStopTracker.interactable = true;
    }

    public void StopTrack()
    {
        TrackerManager.Instance.GetTracker<ObjectTracker>().Stop();
    }

    public void StartTrack()
    {
        TrackerManager.Instance.GetTracker<ObjectTracker>().Start();
    }
}
```

*Código 5. Script para el control de los botones [Realización Propia]*

Brevemente se quiere mencionar que, para los botones, se intentó poner otras líneas de código para parar el seguimiento y que se quedase en el sitio. Estas líneas son las que se pueden observar en el Código 6, con el que se intenta obtener la posición del plano cuando se aprieta el botón y asignarle esta como permanente.

```
// Coger la posición actual: pos = (assign) new position
    Vector3 currentposition = transform.position;
    transform.position = currentposition; //asignamos una nueva posición
al plano
    // transform.position = new Vector3();// asignamos una nueva posición
al plano
```

*Código 6. Líneas de código para dejar de seguir al target calculando su posición [Realización Propia].*

## 5. RESULTADOS Y DISCUSIÓN

### 5.1. Cubo AR

En las Figuras 30 y 31 se pueden contemplar los resultados obtenidos de la aplicación de este procedimiento.



Figura 30. Resultado Cubo AR visto desde el teléfono móvil [Realización Propia].

DESARROLLO DE UNA APLICACIÓN DE REALIDAD AUMENTADA PARA TELÉFONO MÓVIL PARA LA  
LOCALIZACIÓN DE LA ARTERIA FEMORAL MEDIANTE SECUENCIAS DE ECOGRAFÍA

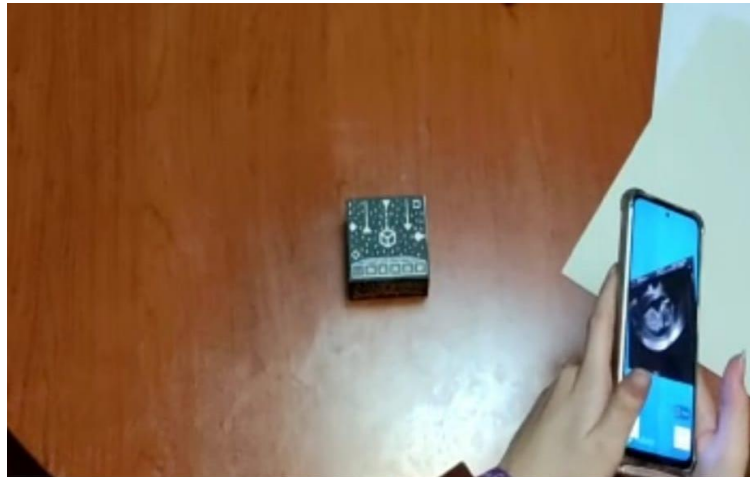


Figura 31. Resultado Cubo AR visto desde fuera [Realización Propia].

Como se ha podido observar, sí se reproduce el vídeo al ser el cubo localizado y al mover el teléfono móvil. Cuando se pulsa el botón la imagen desaparece, por lo que no es lo esperado. A pesar de ello, se creyó que el cubo era demasiado aparatoso y que llegaría a molestar, además de que el sistema que se estaba pensando para sujetarlo solo podía usarse en un tipo de sonda específica puesto que se diseñaría a medida. Por esto, se decidió cambiar el cubo (*MultiTarget*) por una sola imagen (*ImageTarget*) que se podría fotocopiar en hoja de pegatinas y se podría pegar en cualquier sonda a una distancia exacta y calculada desde el inicio (por donde se transmiten los ultrasonidos). Esta opción abarata también los costes puesto que se pasa de un dispositivo impreso en 3D a una imagen que se puede imprimir en cualquier impresora.

## 5.2. Un *ImageTarget*

Los resultados obtenidos pueden ser apreciados en la Figura 32 que, aunque no son perfectos, no están mal, puesto que el vídeo es visible y sigue al *target*, los botones se ven exactamente como se habían diseñado y los nuevos funcionan según lo previsto. El problema es la distancia, el lugar donde se sitúa el plano, no es el adecuado puesto que cuando la sonda está en horizontal, este sí que se ve a la distancia que toca, pero en cuanto se inclina la sonda el plano aparece por encima del código QR, aunque puede ser por perspectiva y se debería comprobar en la realidad con el cirujano para ver la suya. Además, cuando se pulsa el botón de parar el seguimiento, el plano desaparece, lo que no es lo que pretende conseguir (este último resultado es esperable puesto que no se ha cambiado el *script*).

DESARROLLO DE UNA APLICACIÓN DE REALIDAD AUMENTADA PARA TELÉFONO MÓVIL PARA LA  
LOCALIZACIÓN DE LA ARTERIA FEMORAL MEDIANTE SECUENCIAS DE ECOGRAFÍA

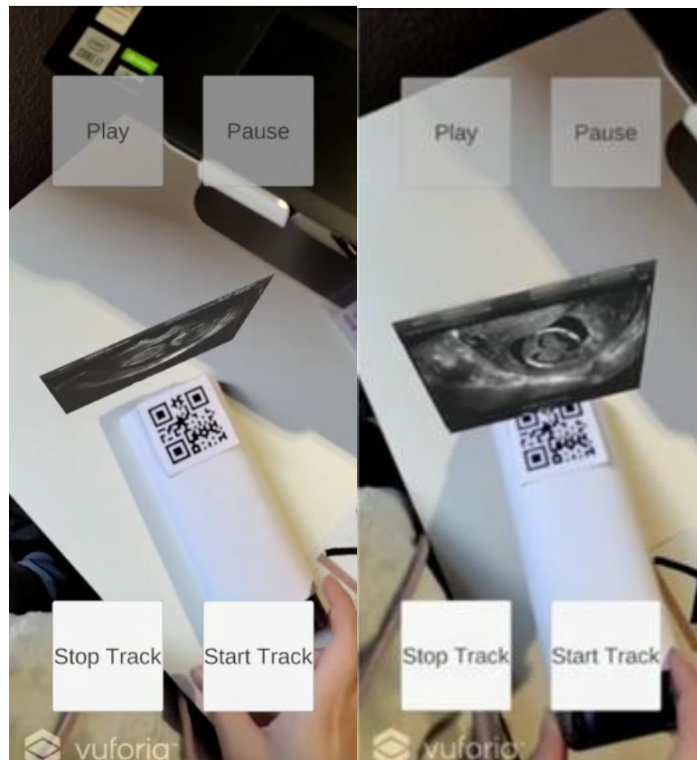


Figura 32. Resultados de un código QR. A la izquierda la sonda está completamente horizontal y a la derecha está levantada e inclinada [Realización Propia].

### 5.3. Más de un *ImageTarget*

Los resultados con dos *ImageTargets* se observan en la Figura 33.



Figura 33. Resultados de dos códigos QR [Realización Propia].



DESARROLLO DE UNA APLICACIÓN DE REALIDAD AUMENTADA PARA TELÉFONO MÓVIL PARA LA  
LOCALIZACIÓN DE LA ARTERIA FEMORAL MEDIANTE SECUENCIAS DE ECOGRAFÍA

Se obtienen los mismos resultados que para el apartado de un solo *target* y este método no parece funcionar puesto que cuando se tapa el QR1, la imagen desaparece cuando debería quedar y seguir al QR2, por lo que se puede decir que no funciona y que otro método sería más efectivo.

En cuanto a los resultados con tres códigos QR, que pueden ser observados en la Figura 34, se puede decir que pasa lo mismo descrito con dos *ImageTarget*, no se realiza exactamente su función y el plano se superpone a los *targets* si la sonda está muy inclinada, aunque de nuevo, esto puede ser debido a la perspectiva.



Figura 34. Resultados del proyecto con tres QR [Realización Propia].

Por lo que respecta a los resultados de usar cuatro *targets*, estos pueden ser observados en la Figura 35 y pasa lo mismo que con los dos anteriores proyectos, no parece que siga a todos los objetivos y también el plano se sobrepone un poco. Por lo que se puede decir que este método que se ha usado en estos últimos tres desarrollos no funciona y debe buscarse otro. Además, el tamaño del QR3 y del QR4 es pequeño para lo necesario, puesto que como se ha comentado en la sección de métodos, deben tener un tamaño en específico y, la iluminación afecta en gran cantidad porque en las Figuras 34 y 35, se pueden observar sombras del propio cuerpo o del dispositivo que hace que no sea reconocible el *ImageTarget*.

DESARROLLO DE UNA APLICACIÓN DE REALIDAD AUMENTADA PARA TELÉFONO MÓVIL PARA LA  
LOCALIZACIÓN DE LA ARTERIA FEMORAL MEDIANTE SECUENCIAS DE ECOGRAFÍA



Figura 35. Resultados del proyecto con cuatro QR [Realización Propia].

En lo que concierne a los resultados del segundo método con cuatro códigos QR, que pueden ser observados en la Figura 36, no son nada parecidos a los esperados porque salen dos planos cuando debería salir uno solo. Además, uno de ellos se retrasa y va a trompicones, se gira, hace cosas extrañas, etc. Por lo que no es una solución nada óptima y no se va a utilizar. Por tanto, esta idea se descarta y para avanzar a la siguiente sección (apartados 4.4 y 4.5), como no es un objetivo el tener distintos *targets*, lo restante del presente trabajo se centró en conseguir reproducir la imagen del ecógrafo en tiempo real porque es una parte esencial.

Como se ha visto, con un solo código QR, la aplicación funciona bastante bien y por la posición del teléfono móvil, siempre está visible; también es la solución más sencilla y, por eso, se utiliza este para intentar conseguir el objetivo específico principal.

DESARROLLO DE UNA APLICACIÓN DE REALIDAD AUMENTADA PARA TELÉFONO MÓVIL PARA LA LOCALIZACIÓN DE LA ARTERIA FEMORAL MEDIANTE SECUENCIAS DE ECOGRAFÍA



Figura 36. Resultados del proyecto con cuatro QR método 2 [Realización Propia].

#### 5.4. Reproducción del vídeo: forma 1

En las Figuras 37 y 38 se puede observar el resultado de los pasos descrito en el apartado de métodos correspondiente y se puede apreciar que efectivamente funciona y se puede retransmitir y captar *rtmp*.

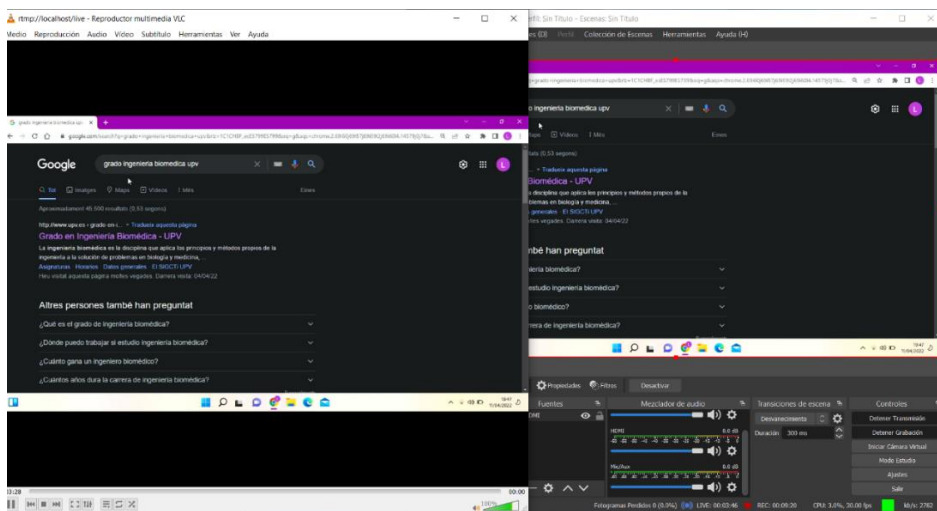


Figura 37. Resultados de retransmitir ordenador-ordenador vistos desde dentro. A la izquierda tenemos el VLC y a la derecha el OBS [Realización Propia].

DESARROLLO DE UNA APLICACIÓN DE REALIDAD AUMENTADA PARA TELÉFONO MÓVIL PARA LA LOCALIZACIÓN DE LA ARTERIA FEMORAL MEDIANTE SECUENCIAS DE ECOGRAFÍA

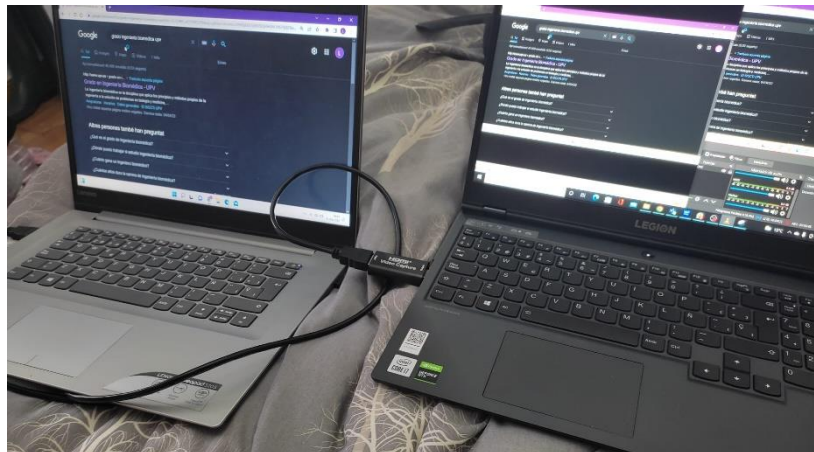


Figura 38. Resultados de retransmitir ordenador-ordenador vistos desde fuera. A la izquierda el ordenador del que captamos la señal y a la derecha el ordenador que la recibe [Realización Propia].

Como se puede ver en la Figura 39, la señal se puede reproducir en el teléfono móvil, por lo que el protocolo de transmisión escogido es adecuado para la retransmisión.

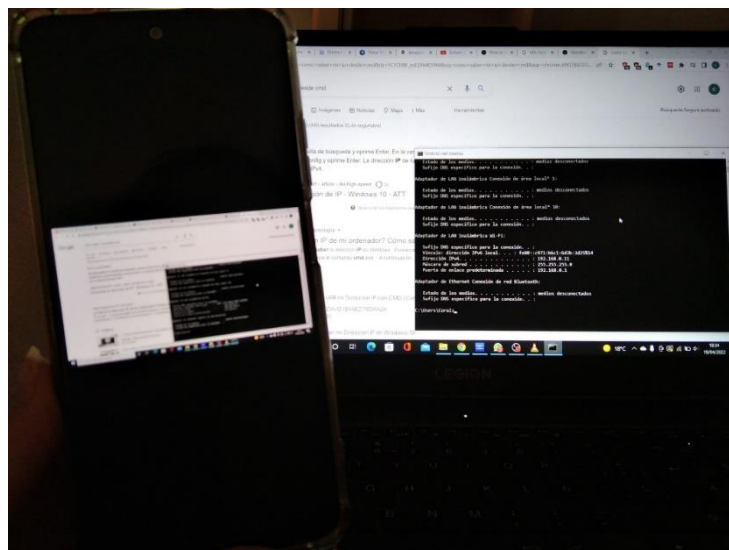


Figura 39. Resultados de la retransmisión ordenador-móvil [Realización Propia].

A la hora de probar la aplicación, esta si funcionaba en el ordenador, la imagen se veía donde se supone que debe, pero una vez se pasó al teléfono móvil, no se mostraba el vídeo, sino que en su lugar se veía el *VideoRenderer*.

La escena de prueba con *UMP* se reproduce tanto en el ordenador como en la aplicación *VLC* del teléfono, sin embargo, al construir la aplicación, la imagen no se reproduce, pero si se escucha el sonido, por lo que se puede decir que sí que funciona pero que la conexión wifi no es lo suficientemente fuerte como para proveer la imagen (puesto que el vídeo de prueba proviene de Internet).

DESARROLLO DE UNA APLICACIÓN DE REALIDAD AUMENTADA PARA TELÉFONO MÓVIL PARA LA  
LOCALIZACIÓN DE LA ARTERIA FEMORAL MEDIANTE SECUENCIAS DE ECOGRAFÍA

El problema de todo este conjunto (Capturadora-*OBS-Unity*) es que el retraso es considerable, es de aproximadamente treinta segundos, lo que es una cantidad que no permitiría al cirujano un control en tiempo real del sistema.

## 5.5. Reproducción del vídeo: forma 2

Como resultado del utilizar el Código 3, la imagen se podía ver perfectamente, pero parecía tener un retraso que se calculó y era de 15 segundos, menos que con la solución mediante codificación *software*, pero todavía bastante elevado. Con la misma línea, pero quitando la señal de audio del vídeo entrante, se obtuvo un retraso de 9 s.

Los resultados de las pruebas con distintos parámetros pueden ser observados en la Tabla 4.

Tabla 4. Parámetros del *ffmpeg* utilizados (en negrita están los introducidos) y el tiempo de retraso en segundos [Realización Propia].

Parámetro	Tiempo de retraso (s)	
<i>ffmpeg -fflags nobuffer</i> -f v4l2 -video_size 640x480 -an -i /dev/video0 -codec:v h264_omx -b:v 512k -f matroska -   cvlc --sout '#http{mux=ffmpeg{mux=flv},dst=:8000/}'	6,77	
<i>ffmpeg -flags low_delay</i> -f v4l2 -video_size 640x480 -an -i /dev/video0 -codec:v h264_omx -b:v 512k -f matroska -   cvlc --sout '#http{mux=ffmpeg{mux=flv},dst=:8000/}'	7	
<i>ffmpeg -fflags nobuffer -flags low_delay</i> -f v4l2 -video_size 640x480 -an -i /dev/video0 -codec:v h264_omx -b:v 512k -f matroska -   cvlc --sout '#http{mux=ffmpeg{mux=flv},dst=:8000/}'	7,14	
<i>ffmpeg -itsoffset offset</i> -f v4l2 -video_size 640x480 -an -i /dev/video0 -codec:v h264_omx -b:v 512k -f matroska -   cvlc --sout '#http{mux=ffmpeg{mux=flv},dst=:8000/}'	<b>offset</b>	Tiempo(s)
	-0.001	6,99
	-0.01	7
	-0.0001	6,4
	-0.00001	7
<i>ffmpeg -max_delay delay</i> -f v4l2 -video_size 640x480 -an -i /dev/video0 -codec:v h264_omx -b:v 512k -f matroska -   cvlc --sout '#http{mux=ffmpeg{mux=flv},dst=:8000/}'	<b>delay</b>	Tiempo(s)
	500 000 (0,5 s)	7,35
	50 000	6,9
	5000	7,62
	5 000 000	7,06
<i>ffmpeg -thread_queue_size size</i> -f v4l2 -video_size 640x480 -an -i /dev/video0 -codec:v h264_omx -b:v 512k -f matroska -   cvlc --sout '#http{mux=ffmpeg{mux=flv},dst=:8000/}'	<b>size</b>	Tiempo(s)
	320	7,12

DESARROLLO DE UNA APLICACIÓN DE REALIDAD AUMENTADA PARA TELÉFONO MÓVIL PARA LA LOCALIZACIÓN DE LA ARTERIA FEMORAL MEDIANTE SECUENCIAS DE ECOGRAFÍA

	512	7,27
ffmpeg -f v4l2 -video_size 640x480 -an -thread_queue_size size -i /dev/video0 -codec:v h264_omx -b:v 512k -f matroska -   cvlc - --sout '#http{mux=ffmpeg{mux=flv},dst=:8000/}'	size	Tiempo(s)
	320	7,2
	512	7,3
ffmpeg -thread_queue_size size -f v4l2 -video_size 640x480 -an -thread_queue_size size -i /dev/video0 - codec:v h264_omx -b:v 512k -f matroska -   cvlc - --sout '#http{mux=ffmpeg{mux=flv},dst=:8000/}'	size	Tiempo(s)
	320	6,9
	512	7,12
ffmpeg -f v4l2 -video_size 640x480 -an -i /dev/video0 - codec:v h264_omx -tune zerolatency -b:v 512k -tune zerolatency -f matroska -   cvlc - --sout '#http{mux=ffmpeg{mux=flv},dst=:8000/}'	6,6	
ffmpeg -f v4l2 -video_size 640x480 -an -i /dev/video0 - codec:v h264_omx -tune zerolatency -b:v 512k -f matroska -   cvlc - --sout '#http{mux=ffmpeg{mux=flv},dst=:8000/}'	7	
ffmpeg -f v4l2 -video_size 640x480 -an -i /dev/video0 - codec:v h264_omx -preset velocity -b:v 512k -f matroska -   cvlc - --sout '#http{mux=ffmpeg{mux=flv},dst=:8000/}'	velocity	Tiempo(s)
	ultrafast	7
	superfast	6,91
	veryfast	6,91

Con la configuración del Código 4, se obtiene una latencia de 6,6 segundos que es menor que el original, lo que es una mejora, pero lo ideal sería conseguir un retraso sobre los milisegundos puesto que, ya que no se puede tener latencia cero, lo más cercano a esta situación sería tener la mínima posible que haga que el retardo no sea perceptible al ojo humano.

Los resultados del cambio de *framerate* son para 12 fps nueve segundos de latencia y para 25 se obtuvo once segundos lo que quiere decir que, al reducir la tasa de fotogramas, se está aumentando la latencia, que es lo contrario a lo que se quiere y por eso, no se introduce este comando en el código final.

La Tabla 5 refleja los resultados correspondientes a la Tabla 3 del apartado 4.5.

## DESARROLLO DE UNA APLICACIÓN DE REALIDAD AUMENTADA PARA TELÉFONO MÓVIL PARA LA LOCALIZACIÓN DE LA ARTERIA FEMORAL MEDIANTE SECUENCIAS DE ECOGRAFÍA

Tabla 5. Parámetros del *ffmpeg* utilizados para disminuir aún más la latencia (en negrita están los introducidos) y el tiempo de retraso en segundos [Realización Propia].

Parámetro	Tiempo de latencia (s)
<code>ffmpeg -fflags nobuffer -flags low_delay \ -strict experimental -f v4l2 -video_size 640x480 -an -i /dev/video0 -codec:v h264_omx -b:v 512k -f matroska -   cvlc - --sout '#http{mux=ffmpeg{mux=flv},dst=:8000/}'</code>	6,95
<code>ffmpeg -fflags nobuffer -flags low_delay \ -strict experimental -itsoffset -0.0001 -thread_queue_size 320 -f v4l2 -video_size 640x480 -an -thread_queue_size 320 -i /dev/video0 -codec:v h264_omx -tune zerolatency -preset superfast -b:v 512k -tune zerolatency -f matroska -   cvlc - --sout '#http{mux=ffmpeg{mux=flv},dst=:8000/}'</code>	6,97
<code>ffmpeg -fflags nobuffer -fflags discardcorrupt -flags low_delay \ -avioflags direct -f v4l2 -video_size 640x480 -an -i /dev/video0 -codec:v h264_omx -b:v 512k -f matroska -   cvlc - --sout '#http{mux=ffmpeg{mux=flv},dst=:8000/}'</code>	7,16
<code>ffmpeg -fflags nobuffer -fflags discardcorrupt -flags low_delay \ -avioflags direct -itsoffset -0.0001 -thread_queue_size 320 -f v4l2 -video_size 640x480 -an -thread_queue_size 320 -i /dev/video0 -codec:v h264_omx -tune zerolatency -preset superfast -b:v 512k -tune zerolatency -f matroska -   cvlc - --sout '#http{mux=ffmpeg{mux=flv},dst=:8000/}'</code>	7,93

Al intentar reducir la latencia con *rtsp* o *rtmp* ocurrieron errores que hicieron que no fuera posible implementarlo puesto que el componente de *VLC* no lo reconocía. Por eso, como protocolo de transporte se usa *http*.

Como se puede comprobar en la Tabla 5, la latencia no es menor que los 6,6 segundos que se consiguen con el Código 4, por lo que se descarta cualquier modificación a este. De modo que, por la falta de tiempo y recursos, el presente trabajo se queda con este retraso, puesto que no se es capaz de reducirlo con los recursos disponibles.

Los resultados correspondientes a la implementación en *Unity* mostraron que el vídeo es visible perfectamente cuando la aplicación es probada en el ordenador (con el consiguiente retraso que se ha mencionado). Sin embargo, la aplicación no funciona del todo en el teléfono móvil, lo que puede deberse a un problema de conectividad wifi (se vuelve a mencionar que la *raspberry pi* y el móvil deben estar conectados al *router*, el primero por ethernet y el segundo por wifi; también que no importa la IP del segundo).

En cuanto a los botones, la primera opción descrita lo único que conseguía era que desapareciera el plano al apretar el botón *Stop* y que reapareciera al pulsar el *Start*, lo que no es lo que quiere hacer y es porque de esta forma se está deshabilitando el *target*, lo que lo hace desaparecer, tal y como sucedía con los apartados 5.1 y 5.2. En el apartado de líneas futuras, se comenta una alternativa para que funcione.

DESARROLLO DE UNA APLICACIÓN DE REALIDAD AUMENTADA PARA TELÉFONO MÓVIL PARA LA  
LOCALIZACIÓN DE LA ARTERIA FEMORAL MEDIANTE SECUENCIAS DE ECOGRAFÍA

La segunda opción, no resulta efectiva, puesto que no se puede fijar el plano a unas coordenadas del mundo real cuando se quita el *target*. Puesto que si al plano, a lo que no es real, le quitas su referencia con el mundo real (el *target*), el primero pierde la orientación, no sabe dónde está, por lo que, si se mueve el teléfono móvil, el vídeo se va a desplazar con él. Cabe destacar que fijar un objeto al mundo real sin una referencia no es realidad aumentada. Igualmente, en el apartado de líneas futuras, se da una posible solución a este problema.



## 6. CONCLUSIONES

En el presente Trabajo Fin de Grado se ha desarrollado una aplicación de realidad aumentada para el teléfono móvil que permite observar la imagen del ecógrafo sobre la piel del paciente y seguir a la sonda en su recorrido. Además, con una serie de botones se puede llegar a dejar de seguir al *target* cuando el usuario así lo desee.

En vista a los resultados obtenidos se puede concluir que ha sido posible alcanzar el objetivo principal dado que la aplicación ha podido ser creada, compilada y ejecutada en un teléfono móvil con sistema operativo tipo *Android*. En cuanto a los objetivos específicos:

- El código QR utilizado como *target* es reconocido y al realizarse esta acción se proyecta el plano en el lugar elegido. Aunque este patrón de blanco y negro se ve muy afectado por la iluminación ambiental por lo que necesita luz lo suficiente buena como para que el contraste sea suficiente.
- No se ha logrado reproducir en tiempo real la señal proporcionada por el ecógrafo, puesto que hay una latencia muy superior a la máxima permitida para aplicaciones médicas (menor de 100 ms). Este retraso se debe a todos los equipos, *hardware* y *software* utilizados, puesto que cada uno añade su latencia.
- Tampoco se ha conseguido, por muy poco, que el plano donde se reproduce el vídeo deje de seguir al *target*. Actualmente al pulsar el botón el plano desaparece puesto que se está deshabilitando el código QR. Por el contrario, los botones son operativos y están integrados en una interfaz de usuario sencilla y amigable a todos los usuarios.

En resumen, el presente Trabajo Fin de Grado es una buena primera aproximación para alcanzar la solución que se quiere obtener, con realidad aumentada, al problema planteado, dado los recursos disponibles y el tiempo limitado.

## 7. LÍNEAS FUTURAS

Como se ha comentado en el apartado de conclusiones, el presente trabajo es una primera aproximación por lo que deja una ventana abierta de posibilidades a futuro, entre las cuales se encuentran:

- Hacer que el plano donde la ecografía es reproducida se adapte a la forma de la pierna del paciente, esto sería de gran ayuda puesto que se ajusta mejor a la anatomía y es más realista por tener mayor grado de solape.
- Como ya se ha comentado en diversas ocasiones, una vez que se deja de seguir al *target*, el teléfono móvil no se puede mover, puesto que, de hacerlo, el plano se movería con él. Por tanto, para solucionarlo, se podría poner otro *target* en la pierna del paciente y así el vídeo podría pasar de seguir a uno a seguir el otro, pudiendo de esta forma, moverse el usuario con libertad ya que la referencia con la realidad no se perdería.
- Para dejar de seguir al objetivo, en vez de inhabilitarlo, encontrar otra solución que funcionase, tal vez con la propiedad *transform*.
- Para no necesitar de un ordenador para lanzar el comando que hace que la *raspberrypi* empiece a retransmitir lo que le llega por la capturadora, escribir en un *script* estas líneas y utilizar el comando *screen* que permite ejecutar un comando en segundo plano.
- Ahora, la distancia del paciente a la imagen está predefinida y es fija, pero una posible mejora sería que se pusiera automáticamente o que el cirujano lo pueda ajustar a su conveniencia mediante, por ejemplo, un *scrollbar*.
- Se ha comentado de utilizar varios *ImageTargets*, lo cual sería una buena idea por explotar, se podría probar con distintas combinaciones para ver si funcionaría mejor que con uno solo. Además, se podría probar a poner uno a la izquierda, cosa que en el presente escrito no se ha realizado, para las personas zurdas.
- Ya que la aplicación es para un sistema operativo en concreto, no estaría mal adaptarla para *iOS* puesto que también existe bastante población que cuenta con esta tecnología.

## 8. BIBLIOGRAFÍA

- [1] Oyarbide, F. A., Vicente, M. J. M., García, M. G., & Zarrabeitia, I. L. Capítulo III Vías de acceso vascular percutáneo Tema 10 Vía femoral
- [2] *Seudoaneurisma: ¿qué lo causa?* - Mayo Clinic. (n.d.). Fecha último acceso: 11 de junio de 2022, de <https://www.mayoclinic.org/es-es/tests-procedures/cardiac-catheterization/expert-answers/pseudoaneurysm/faq-20058420>
- [3] Varon, S. S. L. B., Artigues, S. A. P., Garcia, S. C. N., Eiriz, S. M. R., Britel, R. B., & Mollo, M. G. T. (2021). Punción arterial y venosa ¿es útil utilizar ultrasonido?. *Seram*, 1(1).
- [4] Álvarez-Acosta, L., Rodríguez-Esteban, M., Fariña Ruíz, A. P., Pérez-Pérez, H., Grillo-Pérez, J. J., & Hernández-Afonso, J. (2014). Anatomía angiográfica femoral y complicaciones derivadas del cateterismo cardiaco. *Angiología*, 66(1), 4–10. <https://doi.org/10.1016/J.ANGIO.2013.10.008>
- [5] García-Porrero, J. A., Hurlé, J. M., & Benítez Padilla, G. (2013). *Anatomía humana*. 1010.
- [6] Luis, J., & López, R. (n.d.). *Válvulas cardíacas: funcionamiento y enfermedades*.
- [7] Puerta, C. V. (2006). *Procedimientos endovasculares*. C. Vaquero.
- [8] *Ecografía: Prueba de laboratorio de MedlinePlus*. (n.d.). Fecha último acceso: 9 de junio de 2022, de <https://medlineplus.gov/spanish/pruebas-de-laboratorio/ecografia/>
- [9] Borrego, R. (n.d.). *Fundamentos básicos de ecografía*.
- [10] Thrush, A., Hartshorne, T., Deane, C. R., & Goss, D. (2011). *Ecografía vascular: cómo, por qué y cuándo*. Elsevier.
- [11] *Cómo hacer una canulación de la arteria femoral, bajo guía ecográfica - Cuidados críticos - Manual MSD versión para profesionales*. (n.d.). Fecha último acceso: 17 de junio de 2022, de <https://www.msdmanuals.com/es-es/professional/cuidados-criticos/cómo-hacer-procedimientos-vasculares-centrales/cómo-hacer-una-canulación-de-la-arteria-femoral,-bajo-guía-ecográfica>
- [12] Rocha-Singh, K. J., Zeller, T., & Jaff, M. R. (2014). Peripheral arterial calcification: prevalence, mechanism, detection, and clinical implications. *Catheterization and Cardiovascular Interventions : Official Journal of the Society for Cardiac Angiography & Interventions*, 83(6). <https://doi.org/10.1002/CCD.25387>
- [13] Doerner Wolfgang Broll Paul Grimm Bernhard Jung Eds, R. (n.d.). *Virtual and Augmented Reality (VR/AR)*.
- [14] *ClarifEye | Navegación quirúrgica de RA | Philips*. (n.d.). Fecha último acceso: 10 de junio de 2022, de <https://www.philips.es/healthcare/product/NCVD246/clarifeye-augmented-reality-surgical-navigation>
- [15] *Philips presenta su innovadora tecnología de realidad aumentada aplicada a la cirugía mínimamente invasiva*. (n.d.). Fecha último acceso: 10 de junio de 2022, de <https://www.philips.es/a-w/about/news/archive/standard/news/press/2020/20201402-philips-presents-its-innovative-augmented-reality-technology-applied-to-minimally-invasive-surgery.html>

## DESARROLLO DE UNA APLICACIÓN DE REALIDAD AUMENTADA PARA TELÉFONO MÓVIL PARA LA LOCALIZACIÓN DE LA ARTERIA FEMORAL MEDIANTE SECUENCIAS DE ECOGRAFÍA

- [16] *Vuforia - Unity Manual*. (n.d.). Fecha último acceso: 16 de mayo de 2022, de <https://docs.unity3d.com/es/2018.4/Manual/vuforia-sdk-overview.html>
- [17] *Unity Real-Time Development Platform | 3D, 2D VR & AR Engine*. (n.d.). Fecha último acceso: 17 de mayo de 2022, de <https://unity.com/>
- [18] *Open Broadcaster Software | OBS*. (n.d.). Fecha último acceso: 17 de mayo de 2022, de <https://obsproject.com/es>
- [19] *Descarga oficial del Reproductor multimedia VLC, el mejor reproductor de código abierto - VideoLAN*. (n.d.). Fecha último acceso: 17 de mayo de 2022, de <https://www.videolan.org/vlc/index.es.html>
- [20] *Advanced Load Balancer, Web Server, & Reverse Proxy - NGINX*. (n.d.). Fecha último acceso: 17 de junio de 2022, de <https://www.nginx.com/#>
- [21] *Builds - CODEX FFmpeg @ gyan.dev*. (n.d.). Fecha último acceso: 13 de mayo de 2022, de <https://www.gyan.dev/ffmpeg/builds/>
- [22] *ffmpeg Documentation*. (n.d.). Fecha último acceso: 2 de junio de 2022, de <https://ffmpeg.org/ffmpeg.html>
- [23] *Introducción a Android Studio | Desarrolladores de Android | Android Developers*. (n.d.). Fecha último acceso: 19 de junio de 2022, de <https://developer.android.com/studio/intro?hl=es-419>
- [24] *Download PuTTY - a free SSH and telnet client for Windows*. (n.d.). Fecha último acceso: 1 de junio de 2022, de <https://www.putty.org/>
- [25] *What is SSH and how do I use it? | Information Services Division - UCL – University College London*. (n.d.). Fecha último acceso: 1 de junio de 2022, de <https://www.ucl.ac.uk/isd/what-ssh-and-how-do-i-use-it>
- [26] *Capturadora De Video,HDMI a USB 2.0 Vídeo Game Capture 1080P 60FPS, Yummici Transmisión en Vivo de Transmisión de Vídeo para Juegos, Transmisión, Enseñanza, Videoconferencia : Amazon.es: Informática*. (n.d.). Fecha último acceso: 10 de mayo de 2022, de [https://www.amazon.es/gp/product/B08K43M3ZB/ref=ppx\\_yo\\_dt\\_b\\_asin\\_title\\_o02\\_s00?ie=UTF8&psc=1](https://www.amazon.es/gp/product/B08K43M3ZB/ref=ppx_yo_dt_b_asin_title_o02_s00?ie=UTF8&psc=1)
- [27] *Cable Maillon HDMI 4K 2.0 Gold Connector- High Speed | BC Black 3 m*. (n.d.). Fecha último acceso: 10 de mayo de 2022, de [https://www.mediamarkt.es/es/product/\\_cable-maillon-hdmi-4k-2-0-gold-connector-high-speed-bc-black-3-m-1441385.html](https://www.mediamarkt.es/es/product/_cable-maillon-hdmi-4k-2-0-gold-connector-high-speed-bc-black-3-m-1441385.html)
- [28] *Lenovo Legion 5 - Ordenador Portátil Gaming 15.6" FullHD 120Hz (Intel Core i7-10750H, 16GB RAM, 512GB SSD, Nvidia GTX1650 4GB, FreeDOS), Negro - Teclado QWERTY Español : Amazon.es: Informática*. (n.d.). Fecha último acceso: 10 de mayo de 2022, de [https://www.amazon.es/gp/product/B093WZ9BNG/ref=ppx\\_yo\\_dt\\_b\\_asin\\_title\\_o04\\_s00?ie=UTF8&th=1](https://www.amazon.es/gp/product/B093WZ9BNG/ref=ppx_yo_dt_b_asin_title_o04_s00?ie=UTF8&th=1)
- [29] *Redmi Note 10 | Xiaomi España / Mi.com*. (n.d.). Fecha último acceso: 10 de mayo de 2022, de <https://www.mi.com/es/product/redmi-note-10/specs>
- [30] *Raspberry Pi 2 Model B+ (1GB) Adafruit 2461030 | BricoGeek.com*. (n.d.). Fecha último acceso: 1 de junio de 2022, de <https://tienda.bricogeek.com/descatalogado/718-raspberry-pi-2-model-b-1gb.html>

DESARROLLO DE UNA APLICACIÓN DE REALIDAD AUMENTADA PARA TELÉFONO MÓVIL PARA LA  
LOCALIZACIÓN DE LA ARTERIA FEMORAL MEDIANTE SECUENCIAS DE ECOGRAFÍA

- [31] *Netgear JWNR2010-100PES - Router con tecnología WiFi N300 (300 Mbit/s, 5 Puertos Ethernet y 2 Antenas externas de 5dBi) : Netgear: Amazon.es: Informática.* (n.d.). Fecha último acceso: 1 de junio de 2022, de <https://www.amazon.es/Netgear-JWNR2010-100PES-tecnología-Ethernet-externas/dp/B00I4UPYHC>
- [32] *Philips - EPIQ 7 Sistema de ultrasonido premium.* (n.d.). Fecha último acceso: 7 de junio de 2022, de <https://www.philips.es/healthcare/product/HC795200/epiq-7-ultrasound-system>
- [33] Karl. (n.d.). *Exploring nSIGHT Imaging-a totally new architecture for premium ultrasound Philips EPIQ ultrasound system.*
- [34] *Sonda de ultrasonido Philips L12-3 - Venta y Reparación - Ecosondas.* (n.d.). Fecha último acceso: 7 de junio de 2022, de <https://wsa-ecosondas.es/philips-l12-3-compact/>
- [35] *QR code tracking overview - Mixed Reality | Microsoft Docs.* (n.d.). Fecha último acceso: 19 de junio de 2022, de <https://docs.microsoft.com/en-us/windows/mixed-reality/develop/advanced-concepts/qr-code-tracking-overview>
- [36] *UMP Pro (Android, iOS) | Video | Unity Asset Store.* (n.d.). Fecha último acceso: 19 de junio de 2022, de <https://assetstore.unity.com/packages/tools/video/ump-pro-android-ios-83283>
- [37] *Matroska Element Specification.* (n.d.). Fecha último acceso: 19 de junio de 2022, de <https://www.matroska.org/technical/elements.html#track>
- [38] *How to minimize the delay in a live streaming with ffmpeg - Stack Overflow.* (n.d.). Fecha último acceso: 2 de junio de 2022, de <https://stackoverflow.com/questions/16658873/how-to-minimize-the-delay-in-a-live-streaming-with-ffmpeg>
- [39] *FFmpeg Protocols Documentation.* (n.d.). Fecha último acceso: 19 de junio de 2022, de <https://ffmpeg.org/ffmpeg-protocols.html>
- [40] *alsa - Correct usage of thread\_queue\_size in ffmpeg - Stack Overflow.* (n.d.). Fecha último acceso: 19 de junio de 2022, de <https://stackoverflow.com/questions/61723571/correct-usage-of-thread-queue-size-in-ffmpeg>
- [41] *EncodingForStreamingSites – FFmpeg.* (n.d.). Fecha último acceso: 2 de junio de 2022, de <https://trac.ffmpeg.org/wiki/EncodingForStreamingSites>



# DOCUMENTO 2: PRESUPUESTO

DESARROLLO DE UNA APLICACIÓN DE  
REALIDAD AUMENTADA PARA  
TELÉFONO MÓVIL PARA LA  
LOCALIZACIÓN DE LA ARTERIA FEMORAL  
MEDIANTE SECUENCIAS DE ECOGRAFÍA

AUTORA: CORAL BLANQUER PIÑA

TUTOR: DAVID MORATAL PÉREZ

COTUTOR: AGUSTÍN FERNÁNDEZ CISNAL

TITULACIÓN: GRADO EN INGENIERÍA BIOMÉDICA

CURSO: 2021/22

## ÍNDICE DEL PRESUPUESTO

1.INTRODUCCIÓN.....	3
2. CUADRO DE MANO DE OBRA.....	3
3. CUADRO DE MATERIALES.....	4
4. CUADRO DE MAQUINARIA.....	4
5. CUADRO DE PRECIOS UNITARIOS.....	5
6. CUADRO DE PRECIOS DESCOMPUESTOS .....	6
7. CUADRO DE PRESUPUESTOS PARCIALES.....	9
8. PRESUPUESTO DE EJECUCIÓN POR CONTRATA .....	9



DESARROLLO DE UNA APLICACIÓN DE REALIDAD AUMENTADA PARA TELÉFONO MÓVIL PARA LA  
LOCALIZACIÓN DE LA ARTERIA FEMORAL MEDIANTE SECUENCIAS DE ECOGRAFÍA

## 1. INTRODUCCIÓN

En el presente documento de presupuestos se calculan los costes de realizar el Trabajo Fin de Grado para comprobar su viabilidad económica a la hora de implementarlo en la vida real. A continuación, se van a mostrar el cuadro de mano de obra, el de materiales, el de maquinaria, los descompuestos, los unitarios, los parciales y el presupuesto de ejecución por contrata. Para ello, se han clasificado los recursos utilizados en mano de obra, materiales y maquinas con sus correspondientes códigos que pueden verse en la Tabla 6.

Tabla 6. Códigos de la clasificación de recursos [Realización Propia]

Código	Significado
<b>MO</b>	Mano de obra
<b>MAT</b>	Materiales
<b>MAQ</b>	Maquinaria
<b>%AUX</b>	Otros costes

## 2. CUADRO DE MANO DE OBRA

En esta sección, se tiene en cuenta el coste de las personas implicadas en el proyecto. Estas son, un profesor universitario como tutor, un cirujano vascular como cotutor, un ingeniero sénior como apoyo y una ingeniera biomédica junior como autora.

Para el cálculo de sus salarios se tuvo en cuenta que en un año hay 104 días de fin de semana, 22 de vacaciones y 12 festivos, lo que deja un total de 227 días laborables con una jornada laboral de 8 horas. En cuanto al sueldo base, se estimaron, en base a la media, respectivamente: 45.000 €, 60.000 €, 38.000 € y 25.000 €.

Tabla 7. Cuadro mano de obra [Realización Propia].

Núm.	Código	Denominación de la mano de obra	Precio	Horas	Total (€)
<b>1</b>	MO.Cirujano	Cirujano vascular	33,00	19,500 h	643,50
<b>2</b>	MO.Tutor	Tutor TFG	25,00	16,500 h	412,50
<b>3</b>	MO.IngS	Ingeniero Sénior	21,00	18,000 h	378,00
<b>4</b>	MO.IngJ	Ingeniero Biomédico Júnior	13,77	300,000 h	4.131,00
<b>Total mano de obra:</b>					<b>5.565,00</b>

DESARROLLO DE UNA APLICACIÓN DE REALIDAD AUMENTADA PARA TELÉFONO MÓVIL PARA LA  
LOCALIZACIÓN DE LA ARTERIA FEMORAL MEDIANTE SECUENCIAS DE ECOGRAFÍA

## 3. CUADRO DE MATERIALES

Para el cálculo del presupuesto solo se tuvo en cuenta los materiales de pago, es decir, aquellos softwares de acceso libre como *Vuforia*, *OBS*, *VLC*, *Nginx*, *Ffmpeg*, *Android Studio* y *PuTTY* no se incluyeron.

Tabla 8. Cuadro de materiales [Realización Propia].

Núm.	Código	Denominación del material	Precio	Cantidad	Total (€)
1	MAT.TM	Teléfono móvil	9,26	27,000 h	250,02
2	MAT.UVLC	Plugin VLC para Unity	4,71	19,000 h	89,49
3	MAT.Ord	Ordenador portátil	3,53	283,000 h	998,99
4	MAT.Unity	Licencia Unity	3,14	122,000 h	383,08
5	MAT.UUMP	Plugin UMP para Unity	1,52	32,000 h	48,64
6	MAT.Rou	Router	2,35	51,000 h	119,85
7	MAT.RP	Raspberry Pi	1,96	51,000 h	99,86
8	MAT.Office	Licencia Office 365	0,73	95,000 h	69,35
9	MAT.CV	Capturadora de vídeo	0,11	71,000 h	7,81
10	MAT.HDMI	Cable HDMI	0,07	71,000 h	4,97
11	MAT.Impr	Impresión	0,05	5,000 u	0,10
<b>Total materiales:</b>					<b>2.072,26</b>

## 4. CUADRO DE MAQUINARIA

Para el cálculo de este cuadro, se tuvo en cuenta que se trata de maquinaria médica por lo que está en funcionamiento todos los días del año y las veinticuatro horas del día. También que su vida útil es de 10 años.

Tabla 9. Cuadro maquinaria [Realización Propia].

Núm.	Código	Denominación de la maquinaria	Precio	Cantidad	Total (€)
1	MAQ.Eco	Ecógrafo	25,00	3,000 h	75,00
2	MAQ.Sonda	Sonda ecógrafo	25,00	3,000 h	75,00
<b>Total maquinaria:</b>					<b>150,00</b>

DESARROLLO DE UNA APLICACIÓN DE REALIDAD AUMENTADA PARA TELÉFONO MÓVIL PARA LA  
LOCALIZACIÓN DE LA ARTERIA FEMORAL MEDIANTE SECUENCIAS DE ECOGRAFÍA

## 5. CUADRO DE PRECIOS UNITARIOS

Tabla 10. Cuadro precios unitarios [Realización Propia].

Nº	Designación	Importe	
		En cifra (Euros)	En letra (Euros)
<b>1 CAPÍTULO 1. Definición del proyecto</b>			
1.1	h Reunión inicial con el director del TFG	<b>78,37 €</b>	SETENTA Y OCHO EUROS CON TREINTA Y SIETE CÉNTIMOS
1.2	h Reuniones de seguimiento	<b>783,73 €</b>	SETECIENTOS OCHENTA Y TRES EUROS CON SETENTA Y TRES CÉNTIMOS
1.3	h Reunión con cirujano para concretar detalles	<b>317,02 €</b>	TRESCIENTOS DIECISIETE EUROS CON DOS CÉNTIMOS
1.4	h Reuniones de planificación detallada	<b>113,91 €</b>	CIENTO TRECE EUROS CON NOVENTA Y UN CÉNTIMOS
<b>2 CAPÍTULO 2. Desarrollo de la herramienta</b>			
2.1	u Cubo AR	<b>655,40 €</b>	SEISCIENTOS CINCUENTA Y CINCO EUROS CON CUARENTA CÉNTIMOS
2.2	u Un <i>ImageTarget</i>	<b>354,90 €</b>	TRESCIENTOS CINCUENTA Y CUATRO EUROS CON NOVENTA CÉNTIMOS
2.3	u Más de un <i>ImageTarget</i>	<b>913,31 €</b>	NOVECIENTOS TRECE EUROS CON TREINTA Y UN CÉNTIMOS
2.4	u Reproducción del vídeo: forma 1	<b>1.439,29 €</b>	MIL CUATROCIENTOS TREINTA Y NUEVE EUROS CON VEINTINUEVE CÉNTIMOS
2.5	u Reproducción de vídeo: forma 2	<b>1.628,99 €</b>	MIL SEISCIENTOS VEINTIOCHO EUROS CON NOVENTA Y NUEVE CÉNTIMOS
<b>3 CAPÍTULO 3. Redacción y defensa del TFG</b>			
3.1	h Redacción	<b>1.308,00 €</b>	MIL TRESCIENTOS OCHO EUROS
3.2	h Correcciones	<b>256,79 €</b>	DOSCIENTOS CINCUENTA Y SEIS EUROS CON SETENTA Y NUEVE CÉNTIMOS
3.3	h Preparación de la defensa	<b>654,00 €</b>	SEISCIENTOS CINCUENTA Y CUATRO EUROS

DESARROLLO DE UNA APLICACIÓN DE REALIDAD AUMENTADA PARA TELÉFONO MÓVIL PARA LA  
LOCALIZACIÓN DE LA ARTERIA FEMORAL MEDIANTE SECUENCIAS DE ECOGRAFÍA

## 6. CUADRO DE PRECIOS DESCOMPUESTOS

Tabla 11. Cuadro de precios descompuestos

N.º	Código	Ud.	Descripción	Total
<b>CAPÍTULO 1. Definición del proyecto</b>				
1.1	01.01	h	Reunión inicial con el director del TFG	
	MO.Tutor		1,000 h Tutor TFG	25,00 €
	MO.Cirujano		1,000 h Cirujano vascular	33,00 €
	MO.IngJ		1,000 h Ingeniero Biomédico Júnior	13,77 €
	%AUX01		5,000 % Recursos Auxiliares	71,77 €
			4,000 % Costes indirectos	75,36 €
<b>Precio total por hora</b>				<b>78,37 €</b>
1.2	01.02	h	Reuniones de seguimiento	
	MO. Tutor		10,000 h Tutor TFG	25,00 €
	MO.Cirujano		10,000 h Cirujano vascular	33,00 €
	MO.IngJ		10,000 h Ingeniero Biomédico Júnior	13,77 €
	%AUX01		5,000 % Recursos Auxiliares	717,70 €
			4,000 % Costes indirectos	753,59 €
<b>Precio total por hora</b>				<b>783,73 €</b>
1.3	01.03	h	Reunión con cirujano para concretar detalles	
	MO.Cirujano		3,000 h Cirujano vascular	33,00 €
	MO.IngJ		3,000 h Ingeniero Biomédico Júnior	13,77 €
	MAQ.Eco		3,000 h Ecógrafo	25,00 €
	MAQ.Sonda		3,000 h Sonda ecógrafo	25,00 €
	%AUX01		5,000 h Recursos Auxiliares	290,31 €
			4,000 % Costes indirectos	304,83 €
<b>Precio total por h</b>				<b>317,02€</b>
1.4	01.04	h	Reuniones de planificación detallada	
	MO.IngJ		3,000 h Ingeniero Biomédico Júnior	13,77 €
	MO.IngS		3,000 h Ingeniero Sénior	21,00 €
	%AUX01		5,000 % Recursos Auxiliares	104,31 €
			4,000 % Costes indirectos	304,83 €
<b>Precio total por h</b>				<b>113,91 €</b>
<b>CAPÍTULO 2. Desarrollo de la herramienta</b>				
2.1	02.01	u	Cubo AR	
	MO.IngJ		30,000 h Ingeniero Biomédico Júnior	13,77 €
	MAT.Unity		17,000 h Licencia Unity	3,14 €
	MAT.Ord		30,000 h Ordenador portátil	3,53 €
	MAT.TM		3,000 h Teléfono móvil	9,26 €
	MAT.Impr		1,000 u Impresión	0,02 €
	%AUX01		5,000 % Recursos Auxiliares	600,18 €
			4,000% Costes indirectos	630,19 €
<b>Precio total por unidad</b>				<b>655,40 €</b>

DESARROLLO DE UNA APLICACIÓN DE REALIDAD AUMENTADA PARA TELÉFONO MÓVIL PARA LA  
LOCALIZACIÓN DE LA ARTERIA FEMORAL MEDIANTE SECUENCIAS DE ECOGRAFÍA

2.2	02.02	u	Un <i>ImageTarget</i>	
	MAT.Unity		12,000 h Licencia Unity	3,14 € 37,68 €
	MO.IngJ		15,000 h Ingeniero Biomédico Júnior	13,77 € 206,55 €
	MAT.Ord		15,000 h Ordenador portátil	3,53 € 52,95 €
	MAT.TM		3,000 h Teléfono móvil	9,26 € 27,78 €
	MAT.Impr		2,000 u Impresión	0,02 € 0,04 €
	%AUX01		5,000 % Recursos Auxiliares	325,00 € 16,25 €
			4,000% Costes indirectos	341,25 € 13,65 €
			<b>Precio total por unidad</b>	<b>354,90 €</b>
2.3	02.03	u	Más de un <i>ImageTarget</i>	
	MAT.Unity		33,000 h Licencia Unity	3,14 € 103,62 €
	MO.IngJ		37,000 h Ingeniero Biomédico Júnior	13,77 € 509,49 €
	MAT.Ord		37,000 h Ordenador portátil	3,53 € 130,61 €
	MAT.TM		10,000 h Teléfono móvil	9,26 € 92,60 €
	MAT.Impr		2,000 u Impresión	0,02 € 0,04 €
	%AUX01		5,000 % Recursos Auxiliares	836,36 € 41,82 €
			4,000% Costes indirectos	878,18 € 35,13 €
			<b>Precio total por unidad</b>	<b>913,31 €</b>
2.4	02.04	u	Reproducción del vídeo: forma 1	
	MO.IngJ		45,000 h Ingeniero Biomédico Júnior	13,77 € 619,65 €
	MO.IngS		10,000 h Ingeniero Sénior	21,00 € 210,00 €
	MAT.Ord		45,000 h Ordenador portátil	3,53 € 158,85 €
	MAT.CV		20,000 h Capturadora de vídeo	0,11 € 2,20 €
	MAT.HDMI		20,000 h Cable HDMI	0,07 € 1,40 €
	MAT.Unity		40,000 h Licencia Unity	3,14 € 125,60 €
	MAT.UVLC		19,000 h Plugin VLC para Unity	4,71 € 89,49 €
	MAT.UUMP		12,000 h Plugin UMP para Unity	1,52 € 18,24 €
	MAT.TM		10,000 h Teléfono móvil	9,26 € 92,60 €
	%AUX01		5,000 % Recursos Auxiliares	1.318,03 € 65,90 €
			4,000% Costes indirectos	1.383,93 € 55,36 €
			<b>Precio total por unidad</b>	<b>1.439,29 €</b>
2.5	02.05	u	Reproducción de vídeo: forma 2	
	MAT.Ord		61,000 h Ordenador portátil	3,53 € 215,33 €
	MAT.TM		1,000 h Teléfono móvil	9,26 € 9,26 €
	MAT.CV		51,000 h Capturadora de vídeo	0,11 € 5,61 €
	MAT.HDMI		51,000 h Cable HDMI	0,07 € 3,57 €
	MAT.RP		51,000 h Raspberry Pi	1,96 € 99,96 €
	MAT.Rou		51,000 h Router	2,35 € 119,85 €
	MAT.Unity		20,000 h Licencia Unity	3,14 € 62,80 €
	MAT.UUMP		20,000 h Plugin UMP para Unity	1,52 € 30,40 €
	MO.IngJ		51,000 h Ingeniero Biomédico Júnior	13,77 € 839,97 €
	MO.IngS		5,000 h Ingeniero Sénior	21,00 € 105,00 €
	%AUX01		5,000 % Recursos Auxiliares	1.491,75 € 74,59 €
			4,000 % Costes indirectos	1.566,34 € 62,65 €
			<b>Precio total por unidad</b>	<b>1.628,99 €</b>

DESARROLLO DE UNA APLICACIÓN DE REALIDAD AUMENTADA PARA TELÉFONO MÓVIL PARA LA  
LOCALIZACIÓN DE LA ARTERIA FEMORAL MEDIANTE SECUENCIAS DE ECOGRAFÍA

CAPÍTULO 3. Redacción y defensa del TFG				
3.1	03.01	h	Redacción	
	MO.Tutor		2,000 h Tutor TFG	25,00 € 50,00 €
	MO.Cirujano		2,000 h Cirujano vascular	33,00 € 66,00 €
	MO.IngJ		60,000 h Ingeniero Biomédico Júnior	13,77 € 826,20 €
	MAT.Ord		60,000 h Ordenador portátil	3,53 € 211,80 €
	MAT.Office		60,000 h Licencia Office 365	0,73 € 43,80 €
	%AUX01		5,000 % Recursos Auxiliares	1.197,80 € 59,89 €
			4,000 % Costes indirectos	1.257,69 € 50,31 €
<b>Precio total por hora</b>				<b>1.308,00 €</b>
3.2	03.02	h	Correcciones	
	MO.Tutor		2,500 h Tutor TFG	25,00 € 62,50 €
	MO.Cirujano		2,500 h Cirujano vascular	33,00 € 82,50 €
	MO.IngJ		5,000 h Ingeniero Biomédico Júnior	13,77 € 68,85 €
	MAT.Ord		5,000 h Ordenador portátil	3,53 € 17,65 €
	MAT.Office		5,000 h Licencia Office 365	0,73 € 3,65 €
	%AUX01		5,000 % Recursos Auxiliares	235,15 € 11,76 €
			4,000 % Costes indirectos	246,91 € 9,88 €
<b>Precio total por hora</b>				<b>256,79 €</b>
3.3	03.03	h	Preparación de la defensa	
	MO.Tutor		1,000 h Tutor TFG	25,00 € 25,00 €
	MO.Cirujano		1,000 h Cirujano vascular	33,00 € 33,00 €
	MO.IngJ		30,000 h Ingeniero Biomédico Júnior	13,77 € 413,10 €
	MAT.Ord		30,000 h Ordenador portátil	3,53 € 105,90 €
	MAT.Office		30,000 h Licencia Office 365	0,73 € 21,90 €
	%AUX01		5,000 % Recursos Auxiliares	598,90 € 29,95 €
			4,000 % Costes indirectos	628,85 € 25,15 €
<b>Precio total por h</b>				<b>654,00 €</b>

DESARROLLO DE UNA APLICACIÓN DE REALIDAD AUMENTADA PARA TELÉFONO MÓVIL PARA LA  
LOCALIZACIÓN DE LA ARTERIA FEMORAL MEDIANTE SECUENCIAS DE ECOGRAFÍA

## 7. CUADRO DE PRESUPUESTOS PARCIALES

Tabla 12. Cuadro de presupuestos parciales

N.º	Ud.	Descripción	Cantidad	Precio (€)	Importe (€)
<b>Presupuesto parcial nº01: CAPÍTULO 1. Definición del proyecto</b>					
01.01	H	Reunión inicial con el director del TFG	1,00	78,37	78,37
01.02	H	Reuniones de seguimiento	1,00	783,73	783,73
01.03	H	Reunión con cirujano para concretar detalles	1,00	317,02	317,02
01.04	H	Reuniones de planificación detallada	1,00	113,91	113,91
<b>Total Presupuesto parcial nº01</b>					<b>1.293,03</b>
<b>Presupuesto parcial nº02: CAPÍTULO 2. Desarrollo de la herramienta</b>					
02.01	U	Cubo AR	1,00	655,40	655,40
02.02	U	Un <i>ImageTarget</i>	1,00	354,90	354,90
02.03	U	Más de un <i>ImageTarget</i>	1,00	913,31	913,31
02.04	U	Reproducción del vídeo: forma 1	1,00	1.439,29	1.439,29
02.05	U	Reproducción del vídeo: forma 2	1,00	1.628,99	1.628,99
<b>Total Presupuesto parcial nº02</b>					<b>4.991,89</b>
<b>Presupuesto parcial nº03: CAPÍTULO 3. Redacción y defensa del TFG</b>					
03.01	H	Redacción	1,00	1.308,00	1.308,00
03.02	H	Correcciones	1,00	256,79	256,79
03.03	H	Preparación de la defensa	1,00	654,00	654,00
<b>Total Presupuesto parcial nº03</b>					<b>2.218,79</b>

## 8. PRESUPUESTO DE EJECUCIÓN POR CONTRATA

Tabla 13. Presupuesto de ejecución por contrata

Capítulo	Importe (€)
1.CAPÍTULO 1. Definición del proyecto	1.293,03
2.CAPÍTULO 2. Desarrollo de la herramienta	4.991,89
3.CAPÍTULO 3. Redacción y defensa del TFG	2.218,79
<b>Presupuesto de ejecución material</b>	<b>8.503,71</b>
13 % de gastos generales	1.105,48
6% de beneficio industrial	510,22
<b>Suma</b>	<b>10.119,41</b>
21 % IVA	2.125,08
<b>Presupuesto de ejecución por contrata</b>	<b>12.244,49</b>

Asciende el presupuesto de ejecución por contrata a la expresada cantidad de DOCE MIL DOSCIENTOS CUARENTA Y CUATRO EUROS CON CUARENTA Y NUEVE CÉNTIMOS.