

Memoria del proyecto

3DView: Visualizar con Realidad Aumentada un modelo 3D a partir de un código QR.

Trabajo de Fin de Grado
INGENIERÍA INFORMÁTICA



**VNiVERSIDAD
D SALAMANCA**

Julio de 2021

Autor:

Alfredo Fernández Sánchez

Tutores:

Jesús Lucas Natal (HP)

Sara Rodríguez González

Fernando de la Prieta Pintado

CERTIFICADO DE LOS TUTORES

Dña. Sara Rodríguez González, profesora del departamento de Informática y Automática, D. Fernando De la Prieta Pintado, profesor del departamento de Informática y Automática, y D. Jesús Lucas Natal, empleado de HP SCDS.

Hacen constar:

Que el trabajo titulado “3DView: Visualizar con Realidad Aumentada un modelo 3D a partir de un código QR”, que se presenta, ha sido realizado por Alfredo Fernández Sánchez para la superación de la asignatura Proyecto de Fin de Grado de la Titulación Ingeniería Informática de la Universidad de Salamanca.

En Salamanca, a 6 de septiembre de 2021

RESUMEN

En la actualidad es cada vez más común el uso de la Realidad Aumentada por parte de empresas dedicadas a muy diversos campos, como pueden ser la enseñanza, la medicina, o los deportes. También el uso de códigos QR (del inglés Quick Response code) ha aumentado a lo largo de los años, bien para publicidad o bien para aportar cualquier tipo de información adicional.

El proyecto ha sido realizado como Trabajo de Fin de Grado de la Universidad de Salamanca y forma parte del Observatorio Tecnológico de HP. (SCDS, 2021). Se ha desarrollado con el motor de videojuegos Unity y el lenguaje de programación orientado a objetos C#. El proyecto consiste en una aplicación móvil, la cual ofrece la posibilidad de escanear códigos QR (específicos de la app) y mediante un rastreador de imágenes colocar con Realidad Aumentada un modelo 3D encima del código.

Además de la construcción de la aplicación, se ha construido un servidor. Se trata de un servidor apache, implementado en un Debian 10 Buster. De este modo la aplicación no tiene un gran tamaño, ya que los modelos están alojados en el servidor. Los únicos modelos que tendrá serán los descargados.

El sistema ofrece múltiples posibilidades. Los modelos que se muestran mediante Realidad Aumentada pueden ser usados para enseñanza, información, etc.

Por último, se han puesto en práctica todos los conocimientos adquiridos en el grado de Ingeniería informática, con especial énfasis en las asignaturas de Ingeniería del Software.

Todas las tareas llevadas a cabo están recogidas en este documento y en sus cinco anexos.

Palabras clave: Realidad Aumentada, Unity, Código QR, Modelo 3D

SUMMARY

In this day and age execution of augmented reality by a big variety of companies has become usual. Education, medicine or sports are among those fields that take advantage of this technology. In addition to this, the usage of Quick Response (QR) codes has proliferated in the advertising environment.

This research work was conducted as part of a Final Degree Project, and it is part of the HP Technological Observatory. The project has been developed using the video game engine called "Unity" and the programming language aimed to objects C#. It is a smartphone app, that thanks to augmented reality and an image search, allows users to scan QR codes to create rotation movement and different scales.

In addition to this, a server has been created. It uses an Apache server, implemented in a Debian 10 Buster. The app will not be heavy since all the available models are stored in an external server. It will only store those models that had been downloaded.

The system offers a wide variety of possibilities and the models created through augmented reality can be used for teaching, to share information...

In short, all the knowledge acquired through the Informatics Engineering Degree has been directed and used for this project, paying special attention to the modules and subjects like Software Engineering.

All the tasks and work are described and attached to this file and to the five appendixes.

Key words: Augmented reality, Unity, QR Code, 3D Model

Contenido

1 INTRODUCCIÓN.....	2
1.1 ANTECEDENTES	3
1.1.1 REALIDAD AUMENTADA	3
1.1.2 CÓDIGO QR.....	4
1.1.3 GOOGLE ARTICLE	4
1.2 ACTUALIDAD	4
1.2.1 VISUARTECH	4
1.2.2 SKETCHFAB	5
1.2.3 QUIVERVISION	6
2 OBJETIVOS	7
2.1 OBJETIVOS DEL SISTEMA	7
2.2 OBJETIVOS PERSONALES.....	8
3 CONCEPTOS TEÓRICOS.....	9
3.1 REALIDAD AUMENTADA	9
3.2 CÓDIGO QR	9
3.3 UNITY 3D	10
3.4 MODELO 3D	11
4 TÉCNICAS Y HERRAMIENTAS	12
4.1 TÉCNICAS Y HERRAMIENTAS USADAS EN EL SERVIDOR.....	12
4.1.1 APACHE	12
4.2 TÉCNICAS Y HERRAMIENTAS USADAS EN LA APLICACIÓN	13
4.2.1 C#.....	13
4.2.2 UNITY 3D	13
4.2.3 VISUAL STUDIO CODE	14
4.2.4 ZXING	14
4.2.5 AR FOUNDATION	14
4.2.6 ASSETBUNDLES.....	15
4.3 HERRAMIENTAS CASE.....	15
4.3.1 REM.....	15
4.3.2 MICROSOFT PROJECT.....	15
4.3.3 VISUAL PARADIGM	15

4.3.4 EZESTIMATE.....	16
4.4 HERRAMIENTAS DE CONTROL DE VERSIONES.....	16
4.4.1 GITLAB.....	16
5 ASPECTOS RELEVANTES DEL DESARROLLO	18
5.1 MARCO DE TRABAJO	18
5.2 PLANIFICACIÓN TEMPORAL.....	19
5.3 ESPECIFICACIÓN DE REQUISITOS	20
5.3.1 PARTICIPANTES.....	20
5.3.2 OBJETIVOS DEL SISTEMA.....	21
5.3.3 REQUISITOS DE INFORMACIÓN.....	22
5.3.4 REQUISITOS FUNCIONALES.....	23
5.3.5 REQUISITOS NO FUNCIONALES	25
5.4 ANÁLISIS DE REQUISITOS.....	26
5.4.1 MODELO DE DOMINIO.....	26
5.4.2 REALIZACIÓN DE CASOS DE USO	29
5.4.3 CLASES DE ANÁLISIS.....	29
5.4.5 VISTA ARQUITECTÓNICA DEL MODELO DE ANÁLISIS.....	30
5.5 DISEÑO DEL SISTEMA	30
5.5.1 PATRONES ARQUITECTÓNICOS.....	31
5.5.2 SUBSISTEMAS DE DISEÑO	32
5.5.3 VISTA ARQUITECTÓNICA DEL MODELO DE DISEÑO	33
5.5.4 REALIZACION DE CASOS DE USO DISEÑO	34
5.5.5 MODELO DE DESPLIEGUE.....	34
5.6 IMPLEMENTACIÓN	35
5.7 PRUEBAS	35
5.8 FUNCIONALIDAD DEL SISTEMA.....	37
6 LIMITACIONES DE LA APLICACIÓN	40
7 CONCLUSIONES Y LÍNEAS DE TRABAJO FUTURAS.....	41
7.1 CONCLUSIONES	41
7.2 LÍNEAS DE TRABAJO FUTURAS.....	42
8 REFERENCIAS	43

INDICE DE TABLAS

Tabla 1. Participante	21
Tabla 2. Objetivo del sistema	22
Tabla 3. Requisito de información	23
Tabla 4. Actor administrador	24
Tabla 5. Modificar AssetBundle	25
Tabla 6. Concurrencia	25

ÍNDICE DE FIGURAS

Figura 1. Google Article.....	4
Figura 2. Visuartech	5
Figura 3. Sketchfab.....	5
Figura 4. Quivervision	6
Figura 5. Realidad Aumentada	9
Figura 6. Código QR.....	9
Figura 7. Logo Unity 3D	10
Figura 8. Modelo 3D.....	11
Figura 9. /var/www/html/Assets.....	12
Figura 10. Bitvise ftp	12
Figura 11. Bitvise root	13
Figura 12. Proyecto Unity 3D.....	13
Figura 13. Zxing	14
Figura 14. AR Foundation	15
Figura 15. Proceso Unificado	18
Figura 25. PU Transición. Iteración 10	20
Figura 26. Diagrama de paquetes	23
Figura 27. Modelo de dominio	26
Figura 28. Gesticular Rotación.....	29
Figura 29. Gestión de modelos	29
Figura 30. Arquitectura del modelo de análisis.....	30
Figura 31. MVC.....	31
Figura 32. Subsistema de diseño	32
Figura 33. Escanear Código QR.....	34
Figura 34. Modelo de despliegue	34
Figura 35. Unity API Level.....	35
Figura 36. API Compatibility	36
Figura 37. Unity Build Settings.....	36

1 INTRODUCCIÓN

En la actualidad vivimos rodeados de tecnología, tanto es así que hasta podemos llegar a “aumentarla”. Es decir, usar la realidad en la que vivimos para insertar modelos gráficos. Esto se consigue mediante Realidad Aumentada, sus usos son muy amplios, desde publicidad a enseñanza, medicina o videojuegos. Cada día está más presente en nuestras vidas, tanto es así que en cualquier informativo podemos ver su uso.

Este proyecto va a consistir en el desarrollo de una aplicación que permita al usuario escanear un código QR y mediante Realidad Aumentada muestre un modelo 3D. También debe permitir al usuario una vez descargado el modelo, interactuar con él haciendo de movimientos de rotación y escala.

A lo largo de este documento se irán exponiendo las técnicas y habilidades usadas para la elaboración del sistema final. La estructura es la siguiente:

- **Antecedentes:** se hará una pequeña introducción de su historia y su uso de dos de los términos más importantes del proyecto, como son la Realidad Aumentada y los códigos QR. También se expondrá una de las aplicaciones relacionadas con el proyecto. Esta hace uso también de la Realidad Aumentada y modelos 3D.
- **Actualidad:** se mostrarán algunas aplicaciones actuales relacionadas con este sistema. Estas aplicaciones que se exponen hacen uso de Realidad Aumentada, modelos 3D, escaneo de imágenes o marcadores en la realidad.
- **Objetivos:** en este apartado se explicarán los objetivos de la aplicación, estos deberán ser estudiados en un principio y al final del proyecto tendrán que cuadrar con los objetivos iniciales. También se expondrán objetivos personales del porqué de la elección del proyecto.
- **Conceptos teóricos:** se explicarán todos los conceptos, de forma teórica, de todos los que estén relacionados con algún subsistema de la aplicación. En este caso se explicarán los más generales como pueden ser los códigos QR o de el desarrollo de la aplicación como bibliotecas usadas, por ejemplo, XZing.
- **Técnicas y herramientas:** en este apartado se detallan los sistemas, aplicaciones, herramientas, plataformas y demás técnicas que se han usado en el desarrollo del proyecto. Se van a dividir en dos bloques, las técnicas usadas en el servidor y las técnicas usadas en la aplicación.
- **Aspectos relevantes del desarrollo:** aspectos relevantes que se han llevado a cabo durante el desarrollo. Estos pueden ser los procesos que se han seguido, en este caso el proceso unificado, por lo que se van detallando cada apartado de este proceso. También como se ha llevado el desarrollo del sistema, en este caso por casos de uso, así que se van exponiendo todos los detalles de este tipo de metodología. Por otro lado, también están los patrones de diseño que se han usado, como MVC u Observer. Por último, el estado actual del sistema.

- **Limitaciones de la aplicación:** se exponen las limitaciones que tiene la aplicación en el estado actual. También se comentan posibles soluciones para esas limitaciones.
- **Conclusiones y líneas de trabajo futuras:** se explican las conclusiones sacadas del desarrollo del proyecto. En la versión que esta la aplicación hay varias líneas futuras que se exponen también en este apartado.
- **Referencias:** en este apartado hay un listado con todas las referencias que se van haciendo a lo largo del informe, así como la información de la bibliografía o sitio web al que se refieren.

Para completar aún más este documento se han realizado los siguientes anexos:

- **Anexo I – Planificación temporal:** estudio y estimación del tiempo que se llevara a cabo para la realización del proyecto, Así como la estimación del esfuerzo estimado.
- **Anexo II – Especificación y análisis de requisitos:** recoge la especificación de los requisitos necesarios del sistema, así como su análisis. Contiene apartados de requisitos de información, funcionales y no funcionales, todos ellos detallados. También tiene un modelo de dominio para poder hacerse una idea de la magnitud del problema.
- **Anexo III – Diseño del sistema software:** recoge la fase de diseño de un sistema software. Tiene explicados los patrones arquitectónicos y de diseño que se han usado en la aplicación, así como una vista más detallada de todos los casos de uso recogidos en el anexo II.
- **Anexo IV – Documentación técnica:** documento que recogerá la información para una mejor comprensión del código realizado. Se explicará todas las clases y ficheros que tiene el sistema.
- **Anexo V – Manual de usuario:** manual para el usuario final para que pueda ver las funcionalidades del sistema, así experimentara una mejor experiencia. Contiene toda la funcionalidad del sistema, así como sus pasos para poder hacer uso de esta.

1.1 ANTECEDENTES

1.1.1 REALIDAD AUMENTADA

La realidad aumentada (RA) es el término que se usa para describir al conjunto de tecnologías que permiten que un usuario visualice parte del mundo real a través de un dispositivo tecnológico con información gráfica añadida por este. El dispositivo, o conjunto de dispositivos, añaden información virtual a la información física ya existente, es decir, una parte virtual aparece en la realidad. De esta manera los elementos físicos

tangibles se combinan con elementos virtuales, creando así una realidad aumentada en tiempo real.

1.1.2 CÓDIGO QR

Un código es la evolución del código de barras. Es un módulo para almacenar información en una matriz de puntos o en un código de barras bidimensional. La matriz se lee en el dispositivo móvil por un lector específico (lector de QR) y de forma inmediata nos lleva a una aplicación en internet y puede ser un mapa de localización, un correo electrónico, una página web o un perfil en una red social. Fue creado en 1994 por la compañía japonesa Denso Wave. Presenta tres cuadrados en las esquinas que permiten detectar la posición del código al lector.

1.1.3 GOOGLE ARTICLE

En 2018 Google crea Google Article. Es un visor de modelos 3D mediante realidad aumentada. Está disponible en PC para cualquier navegador, y para en el teléfono móvil, aparece un botón AR en la esquina inferior derecha y al pulsarlo se activa la cámara del teléfono. El usuario puede entonces colocar el objeto en el suelo para tener una idea de lo grande que es algo en el mundo real.



Figura 1. Google Article

1.2 ACTUALIDAD

En la actualidad existen numerosas aplicaciones que hacen uso de la Realidad Aumentada. En este punto se exponen varias aplicaciones relacionadas con el proyecto, ya que hacen uso de marcadores y/o interacción con modelos 3D. A continuación, se van a mostrar algunas de ellas.

1.2.1 VISUARTECH

Visuarteched Augmented & Virtual reality es una empresa especializada en desarrollo de soluciones y aplicaciones de realidad aumentada y realidad virtual a medida para la arquitectura, ingeniería, diseño de producto y fabricación.

Visuarteched App ha sido desarrollada para la arquitectura, ingeniería, diseño, constructores, promotores, decoradores, diseño de interior, PLV, etc. Pone al alcance un inmenso y atractivo potencial hasta ahora nunca visto. (Visuarteched, s.f.)

Usa marcadores de la vida real para colocar los modelos 3D.

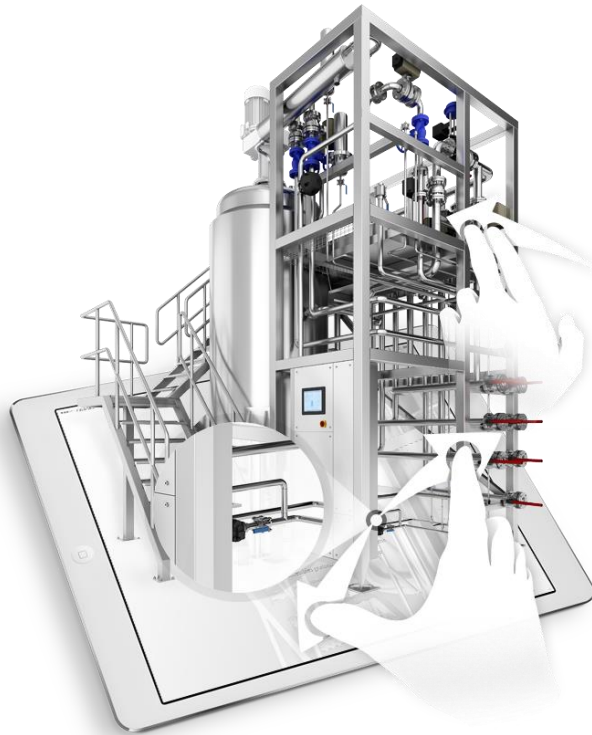


Figura 2. Visuartech

1.2.2 SKETCHFAB

Sketchfab es una aplicación que permite al usuario a partir de una biblioteca de modelos 3D, presentarlos con Realidad Aumentada en un plano. El plano sobre el que se va a presentar el modelo le selecciona el usuario.

Esta app está disponible para cualquier plataforma, cualquier navegador y sus modelos permiten cualquier tipo de formato. (Sketchfab, s.f.)



Figura 3. Sketchfab

1.2.3 QUIVERVISION

Quivervision es una aplicación que usa la realidad aumentada para presentar dibujos de la vida real como modelos 3D. Estos dibujos están disponibles en su plataforma, una vez descargados se deben pintar, después se escanean y por último se presentan con Realidad Aumentada para la interacción con dichos modelos. (Quivervision, 2017)



Figura 4. Quivervision

2 OBJETIVOS

En este apartado se exponen los objetivos que tiene que cumplir el sistema para la correcta realización del Trabajo de Fin de Grado. Se van a detallar los objetivos del sistema, los objetivos técnicos y los objetivos personales que deberán cumplirse a lo largo de este proceso.

2.1 OBJETIVOS DEL SISTEMA

Los objetivos del sistema son las especificaciones que deberá cumplir este, para poder satisfacer los requisitos del usuario:

- **Detección del entorno:** el sistema deberá poder acceder a la cámara del usuario para poder obtener información del entorno. La información del entorno que se obtiene es una textura por cada fotograma, esta textura será analizada posteriormente.
- **Decodificación códigos QR:** el sistema deberá analizar cada textura que venga de la cámara. Esta búsqueda se hará hasta que se encuentre algún código QR en el entorno. Para poder decodificar un código QR se hace uso de la biblioteca XZing y la interfaz IBarcodeReader.
- **Búsqueda de marcadores:** el sistema tendrá integrado una biblioteca de marcadores. Estos marcadores son códigos QR recortados con los puntos en común que tienen estos, hay tres de diferentes tamaños. Mediante el módulo de seguimiento de imágenes de ARTrackedImageManager, el sistema estará en constante búsqueda de estos marcadores.
- **Gestión de marcadores:** el sistema permitirá al administrador de la aplicación poder gestionar la biblioteca de marcadores. Esta gestión tiene como objetivos poder añadir, quitar o modificar marcadores.
- **Gestión de modelos:** el sistema deberá permitir al administrador del proyecto gestionar los modelos del servidor. Esta gestión deberá tener una alta, baja o modificación de algún modelo, con su respectivo AssetBundle que será el que se aloje en el servidor.
- **Gestión de visualización:** el sistema gestionará la visualización de los modelos 3D con Realidad Aumentada. Esta visualización se consigue gracias al módulo ARTrackedImageManager, que gestiona la aparición, movimiento o eliminación de los modelos 3D.
- **Gestión de interacción:** el sistema deberá permitir al usuario movimientos de rotación y escala en el modelo. Toda esta gestión se hace gracias a dos scripts específicos para ello, que han sido configurados con las herramientas de modelaje de Unity.
- **Gestión de herramientas:** el sistema pondrá a disposición del usuario un menú con varias opciones, una de información otra de ayuda y una última de configuración. Esta es la otra funcionalidad de la aplicación.

2.2 OBJETIVOS PERSONALES

Como en todo proyecto existen unas motivaciones personales para la realización de tal sistema, de esta forma se podrá llevar a cabo hasta ver cumplidos los objetivos anteriores. Estos motivos personales son:

- Formar parte del Observatorio tecnológico de HP, trabajando así con profesionales del sector informático y poder crecer personalmente.
- Poner en práctica todos los conocimientos obtenidos en el grado en Ingeniería informática durante los cursos de este.
- Trabajar con la Realidad Aumentada ya que es una tecnología que cada año que pasa va aumentando sus posibilidades de uso.
- Estudiar y analizar cómo funcionan los códigos QR y así poder trabajar con ellos en este proyecto y en los que vendrán.
- Trabajar con herramientas de control de versiones y con sus diferentes opciones de planificación de trabajo.
- Obtener conocimientos técnicos necesarios para el desarrollo del proyecto, en concreto:
 - Conocimiento del entorno Unity. Este objetivo es esencial ya que todo el proyecto se basa en el motor de videojuegos Unity. Se deberá tener un manejo de su entorno principal, así como de sus diferentes configuraciones.
 - Conocimiento del lenguaje de programación C#. Este objetivo es importante ya que todos los scripts que se han realizado están programados en C#. De esta forma se puede optimizar de forma correcta los diferentes módulos de la aplicación.
 - Conocimiento de entornos Linux. En este proyecto se ha usado Debian 10 pero valdría cualquiera de los entornos disponibles, ya que lo que se requiere es un manejo general para poder crear y optimizar un servidor apache.

3 CONCEPTOS TEÓRICOS

Ahora se van a exponer todos los conceptos teóricos que tienen relación con el proyecto para así facilitar el entorno que rodea este sistema. También se expondrán algunos problemas que se han encontrado en el proceso, así como su solución.

3.1 REALIDAD AUMENTADA

Consiste en insertar figuras gráficas en el mundo real. Puede usarse para mostrar objetos para informar, enseñar o publicidad.



Figura 5. Realidad Aumentada

3.2 CÓDIGO QR

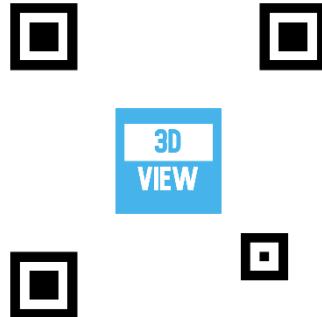
El código QR es la evolución del código de barras que todos conocemos. Es una matriz de puntos la cual al leerla nos muestra un texto, una url, un correo electrónico...

Consta de 3 cuadrados en sus esquinas para la localización de este, así nos permite una alta velocidad de procesamiento.



Figura 6. Código QR

Al tener estas partes en común, se decidió hacer marcadores comunes para no tener que almacenar todos los códigos QR de todos los modelos en la biblioteca. Así se soluciona un problema de almacenamiento. Estos marcadores tienen el siguiente aspecto:



También para una mayor velocidad de procesamiento del rastreador de imágenes, se añadió un logo en el centro del código, ya que este tipo de prácticas los hacen actualmente muchas empresas.

3.3 UNITY 3D

Unity es un motor de videojuegos multiplataforma. Tiene varias opciones de compilación como aplicación de escritorio, Android, Mac Os... (Unity, s.f.)



Figura 7. Logo Unity 3D

3.4 MODELO 3D

Un modelo 3D presenta un objeto tridimensional usando una colección de puntos en el espacio.



Figura 8. Modelo 3D

Estos modelos 3D eran un problema a la hora de subirse o descargarse del servidor, debido a su extensión. Para solucionar esto, se usó el tipo de objeto creado por parte de Unity, que son los AssetBundles. Este concepto se explicará en los siguientes apartados.

4 TÉCNICAS Y HERRAMIENTAS

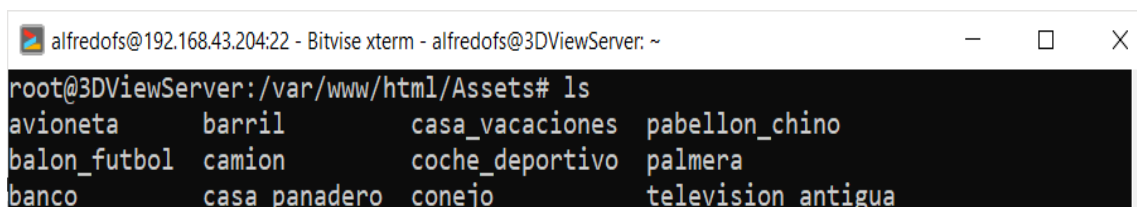
A continuación, se van a presentar las técnicas usadas para el desarrollo del sistema, tanto en el servidor como en la aplicación.

4.1 TÉCNICAS Y HERRAMIENTAS USADAS EN EL SERVIDOR

4.1.1 APACHE

Es un software de servidor web multiplataforma de código abierto y gratuito. Es desarrollado y mantenido por una comunidad abierta de desarrolladores.

Una vez instalado, mediante la herramienta gratuita Bitwise ssh client, se han copiado los AssetBundles en el directorio /var/www/html.



```
alfredofs@192.168.43.204:22 - Bitwise xterm - alfredofs@3DViewServer: ~
root@3DViewServer:/var/www/html/Assets# ls
avioneta      barril        casa_vacaciones  pabellon_chino
balon_futbol  camion       coche_deportivo  palmera
banco         casa_panadero conejo           television_antigua
```

Figura 9. /var/www/html/Assets

Esta copia se realiza mediante el protocolo FTP (File Transfer Protocol). Además, tiene la opción de abrir una consola y así poder administrar el servidor mediante superusuario.

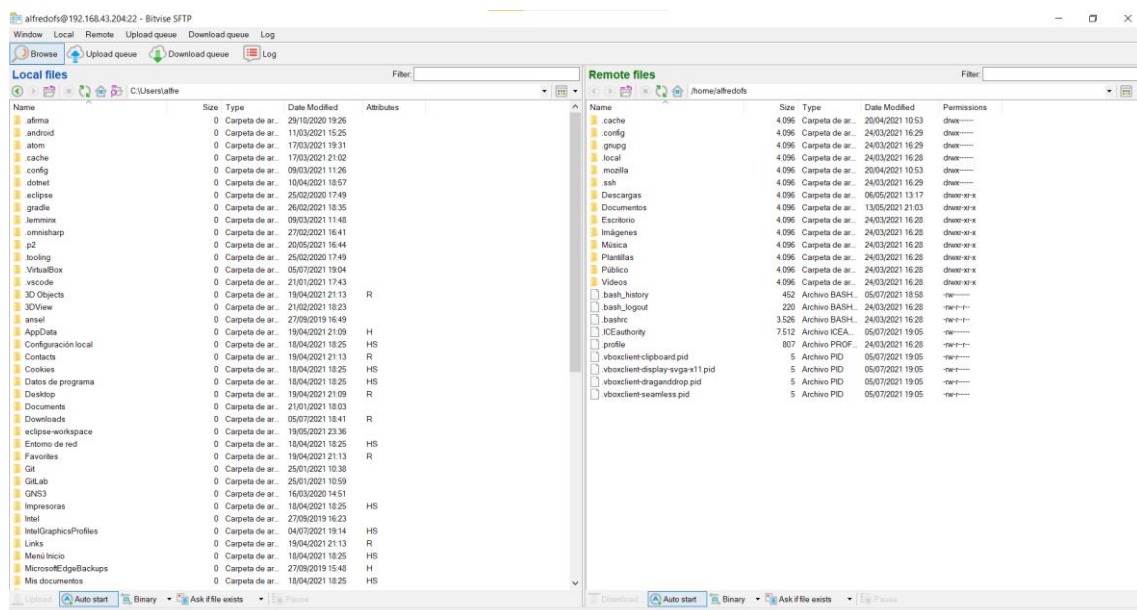


Figura 10. Bitwise ftp



```
alfredofs@192.168.43.204:22 - Bitvise xterm - alfredofs@3DViewServer: ~
alfredofs@3DViewServer:~$ su
Contraseña:
root@3DViewServer:/home/alfredofs#
```

Figura 11. Bitvise root

4.2 TÉCNICAS Y HERRAMIENTAS USADAS EN LA APLICACIÓN

4.2.1 C#

Lenguaje de programación multiparadigma desarrollado por Microsoft como parte de su plataforma .NET. Su sintaxis deriva de C/C++. Utiliza el modelo de objetos muy similar a Java. (Microsoft, C# Documentation, 2021)

C# admite los conceptos de encapsulación, herencia y polimorfismo.

4.2.2 UNITY 3D

Como ya se ha explicado, Unity es un motor de videojuegos. Se ha utilizado para crear una aplicación Android, y mediante sus múltiples opciones se ha creado el proyecto. El resultado final de este es el siguiente:

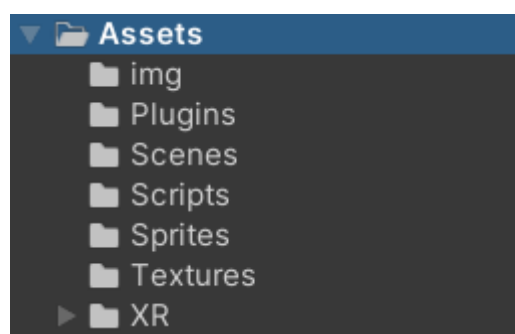


Figura 12. Proyecto Unity 3D

Donde en cada carpeta hay lo siguiente:

- img: dentro se encuentran los marcadores de la aplicación.
- Plugins: esta la biblioteca usada para el rastreador de imágenes y la librería ZXing, para decodificar códigos QR.
- Scenes: hay una escena, la del proyecto principal.
- Scripts: dentro están los cinco scripts creados:

- ControladorMenu: para manejar las opciones de menú.
- EscalarModelo: para la escalación del modelo 3D.
- EscanearQR: controla la cámara principal del dispositivo para el escaneo de imágenes.
- ImageManager: controla el modelo 3D para su visualización e interacción.
- RotarModelo: para la rotación del modelo 3D.
- Sprites: dentro están las diferentes imágenes para la interfaz de la aplicación, ya sean los botones o el logo de esta.
- Textures: dentro se encuentran materiales.

4.2.3 VISUAL STUDIO CODE

Visual studio code es un editor de código fuente desarrollado por Microsoft. Se ha elegido este entorno de desarrollo debido a que tiene soporte para depuración y control integrado de Git.

4.2.4 ZXING

Zxing es una librería de código abierto multiformato implementada en Java pero válida para otros lenguajes. Mediante esta librería se ha podido escanear códigos QR.

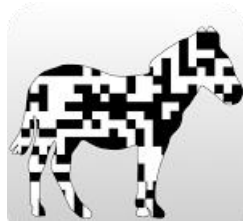


Figura 13. Zxing

4.2.5 AR FOUNDATION

AR Foundation es un marco de trabajo creado especialmente para el desarrollo de realidad aumentada que te permite generar experiencias enriquecidas y luego implementarlas en diferentes dispositivos móviles.

Para usar AR Foundation también necesita instalar paquetes en el dispositivo de destino, en este caso ha sido ARCore XR para Android. (Unity, s.f.)

Con ARFoundation podemos tener:

- Seguimiento de imágenes, en este caso marcadores
- Detección de planos.
- Anclaje
- Seguimiento de imágenes 2D, y 3D

Se ha usado esta tecnología por delante de otras como Vuforia debido a que ya está implementado dentro de Unity y no es necesario bases de datos externas para el seguimiento de imágenes, sino que creamos estas bases dentro de la aplicación.

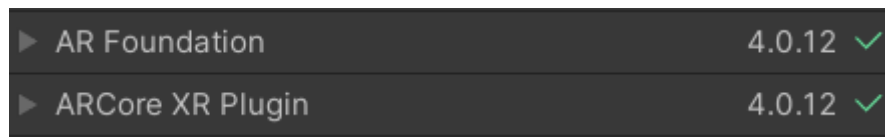


Figura 14. AR Foundation

4.2.6 ASSETBUNDLES

Un AssetBundle es un archivo de almacenamiento que contiene activos sin código específicos, como modelos. La gran ventaja es que se pueden cargar en tiempo de ejecución. Se han utilizado debido a que son muy útiles para contenido descargable.

Siguiendo la guía de Unity3D sobre AssetBundles se han implementado. (Unity D. , 2021)

El único cambio más relevante es el cambio de plataforma en el BuildTarget, donde en este caso se ha tenido que poner Android.

4.3 HERRAMIENTAS CASE

A continuación, se muestra las herramientas CASE (Computer Aided Software Engineering) utilizadas. Estas son destinadas a aumentar la productividad en el ciclo de desarrollo de software. En este caso para cálculo de costos, planificación y diseño software.

4.3.1 REM

REquirements Management es una herramienta de gestión de requisitos diseñada para soportar la fase de ingeniería de requisitos. Ha sido construido siguiendo la metodología de Duran y Bernardez. (Sevilla, s.f.)

4.3.2 MICROSOFT PROJECT

Es un software de administración de proyectos diseñado por Microsoft. Permite el desarrollo de planes, asignación de recursos a tareas, dar seguimiento al progreso, administrar presupuesto, etc. (Microsoft, Microsoft 365, s.f.)

4.3.3 VISUAL PARADIGM

Herramienta que permite crear diagramas del modelado UML, tales como diagramas de clase, diagramas de comunicación, diagramas de casos de uso, etc. (Paradigm, s.f.)

Para su uso se ha escogido la licencia disponible de la USAL.

4.3.4 EZESTIMATE

Es una herramienta que permite realizar la estimación de esfuerzo de un proyecto.

4.4 HERRAMIENTAS DE CONTROL DE VERSIONES

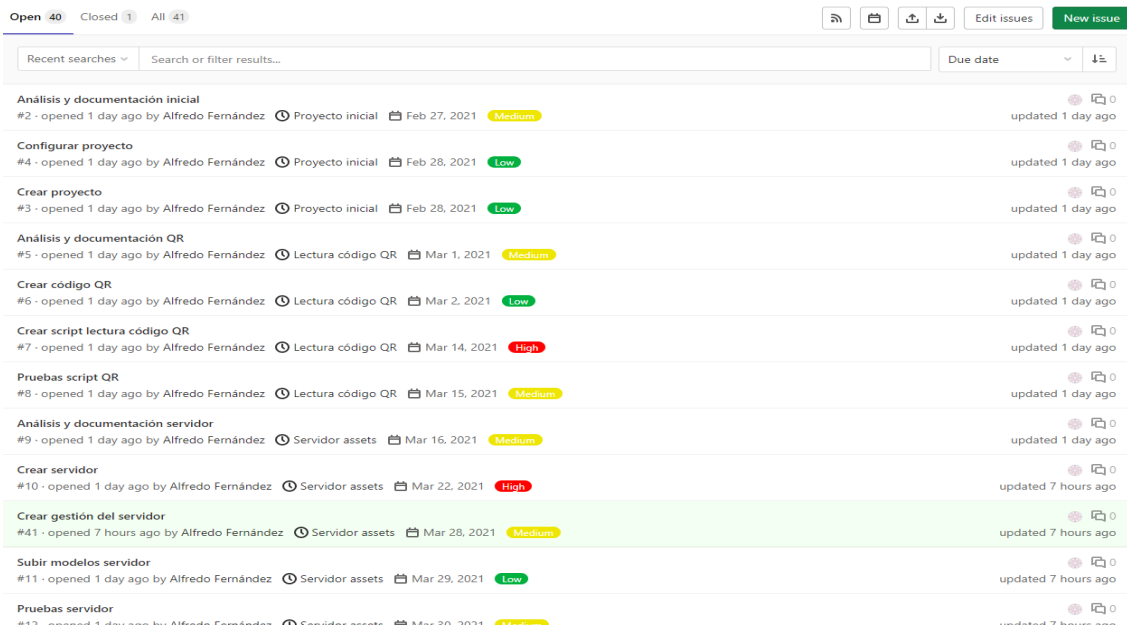
En este apartado se describe de la herramienta utilizada para control de versiones. Estas herramientas son de las más importantes en el desarrollo del software, ya que nos permite tener una copia en la nube de la versión que deseemos y poder recuperar versiones anteriores.

4.4.1 GITLAB

Es una plataforma DevOps completa, entregada como una sola aplicación. Gitlab permite tener:

- Una fuente de verdad: gestionar proyectos, no herramientas.
- Todo continuo: permite trabajar de forma continua de manera eficiente.
- Seguridad en tiempo real: brinda una seguridad automatizada.

Además, en Gitlab se puede tener una planificación y control en todo momento de las tareas y los hitos de la aplicación. Se han creado varias tareas e hitos para llevar a cabo un control del desarrollo software. Estos han sido:



Issue Title	ID	Status	Priority	Due Date	Updated
Análisis y documentación inicial	#2	Open	Medium	Feb 27, 2021	updated 1 day ago
Configurar proyecto	#4	Open	Low	Feb 28, 2021	updated 1 day ago
Crear proyecto	#3	Open	Low	Feb 28, 2021	updated 1 day ago
Análisis y documentación QR	#5	Open	Medium	Mar 1, 2021	updated 1 day ago
Crear código QR	#6	Open	Low	Mar 2, 2021	updated 1 day ago
Crear script lectura código QR	#7	Open	High	Mar 14, 2021	updated 1 day ago
Pruebas script QR	#8	Open	Medium	Mar 15, 2021	updated 1 day ago
Análisis y documentación servidor	#9	Open	Medium	Mar 16, 2021	updated 1 day ago
Crear servidor	#10	Open	High	Mar 22, 2021	updated 7 hours ago
Crear gestión del servidor	#41	Open	Medium	Mar 28, 2021	updated 7 hours ago
Subir modelos servidor	#11	Open	Low	Mar 29, 2021	updated 7 hours ago
Pruebas servidor	#12	Open	Medium	Mar 30, 2021	updated 7 hours ago

Figura 13. GitLab tareas

Open 10	Closed 0	All 10	Filter by milestone name	Due soon	New milestone
Proyecto inicial					
Feb 26, 2021–Feb 28, 2021		3 Issues · 0 Merge Requests		0% complete	Close Milestone
Upcoming HP-SCDS / Observatorio / 2020-2021 / USAL-3DView					
Lectura código QR					
Mar 1, 2021–Mar 15, 2021		4 Issues · 0 Merge Requests		0% complete	Close Milestone
Upcoming HP-SCDS / Observatorio / 2020-2021 / USAL-3DView					
Servidor assets					
Mar 16, 2021–Mar 23, 2021		5 Issues · 0 Merge Requests		0% complete	Close Milestone
Upcoming HP-SCDS / Observatorio / 2020-2021 / USAL-3DView					
Visualización modelo 3D					
Mar 24, 2021–Apr 7, 2021		4 Issues · 0 Merge Requests		0% complete	Close Milestone
Upcoming HP-SCDS / Observatorio / 2020-2021 / USAL-3DView					
Interacción modelo 3D					
Apr 8, 2021–Apr 15, 2021		5 Issues · 0 Merge Requests		0% complete	Close Milestone
Upcoming HP-SCDS / Observatorio / 2020-2021 / USAL-3DView					
Interfaz app					
Apr 16, 2021–Apr 30, 2021		6 Issues · 0 Merge Requests		0% complete	Close Milestone
Upcoming HP-SCDS / Observatorio / 2020-2021 / USAL-3DView					
3DView beta					
May 1, 2021–May 5, 2021		2 Issues · 0 Merge Requests		0% complete	Close Milestone
Upcoming HP-SCDS / Observatorio / 2020-2021 / USAL-3DView					

Figura 14. GitLab hitos

5 ASPECTOS RELEVANTES DEL DESARROLLO

5.1 MARCO DE TRABAJO

Para la realización del proyecto se han seguido las directrices del Proceso Unificado. Es un marco de trabajo genérico para diferentes tipos de organizaciones, diferentes tamaños de proyecto, etc.

Este tipo de proceso tiene las siguientes características:

- Conducido por casos de uso
- Basado en componentes
- Centrado en la arquitectura
- Iterativo e incremental

El Proceso Unificado se repite a lo largo de una serie de ciclos. Cada ciclo consta de cuatro fases, estas finalizan con un hito:

- **Inicio:** se define el alcance del proyecto
- **Elaboración:** se planifica el proyecto, se elaboran la mayoría de los casos de uso y la arquitectura del sistema.
- **Construcción:** se empieza a construir el sistema.
- **Transición:** se consigue una versión entregable del producto. A este se le corrigen problemas y se hacen mejoras.

El Proceso Unificado tiene diferentes disciplinas, estas son:

- Modelado de negocio
- Requisitos
- Análisis
- Diseño
- Implementación
- Pruebas

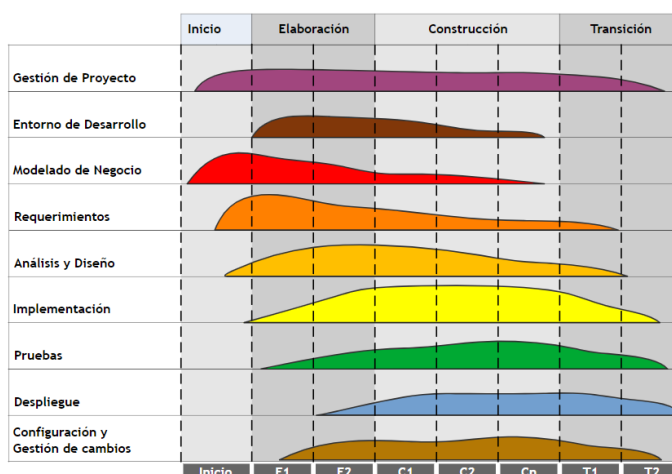


Figura 15. Proceso Unificado

En el caso de nuestro proyecto se ha usado el proceso unificado de la siguiente manera:

- En cuanto a las características del PU:
 - Se ha dividido la funcionalidad del sistema en casos de uso, estos son en total 24 casos. Están divididos en paquetes según el tipo de característica que se gestiona.
 - Se basa en componentes dependiendo de la funcionalidad, ya sea el componente que se encarga de escanear el código QR, el que gestiona el modelo dependiendo del marcador...
 - Se centra en la arquitectura ya que desde una vista arquitectónica general se puede ver cómo hay dos sistemas, la aplicación y el servidor, y toda la arquitectura que compone a estos.
 - Iterativo ya que cada fase se ha realizado varias veces, en total hay 10 iteraciones, e incremental ya que el volumen de trabajo y del sistema ha ido creciendo a lo largo de las iteraciones.
- Las cuatro fases en las que se divide un ciclo del PU se han usado la siguiente manera:
 - Inicio: esta fase está dividida en dos iteraciones, en ambas se puede ver como trata más sobre gestión, modelado y requerimientos del proyecto. (Figura 16). De esta fase cabe destacar la reunión con los tutores para una explicación de cómo llevar a cabo el proyecto y la elaboración de los objetivos, donde se expone que se va a conseguir con este sistema.
 - Elaboración: esta fase consta de dos iteraciones. Las disciplinas que se empiezan a usar son las de análisis, diseño e implementación, sin olvidar modelado de negocio. De aquí es destacable la elaboración de los requisitos del sistema, así como los casos de uso y las diferentes vistas del modelo.
 - Construcción: esta fase tiene 4 iteraciones. En estas ya se ira implementando el sistema, así como realizando pruebas con esas implementaciones. Cabe destacar de esta fase la implementación de los diferentes módulos y las diferentes pruebas que se han ido realizando.
 - Transición: en esta parte quedaría refinar la aplicación, así como el despliegue de esta en un entorno real para su verificación. De esta parte hay que destacar las diferentes pruebas que se han ido realizando tanto de la app como del servidor, así como la elaboración de los diferentes anexos y esta memoria.

5.2 PLANIFICACIÓN TEMPORAL

Para la planificación temporal se ha usado la herramienta Microsoft Project. Mediante esta planificación podemos concretar que tareas tenemos que llevar a cabo y cuáles serán la duración de estas. Así podemos hacernos una idea de cuanto nos va a llevar el desarrollo del proyecto.

En la siguiente figura se detalla la iteración que más horas ha durado. En esta iteración se realizan los anexos y la memoria final.

▼ Transición	22 días	4/5/2021	156 horas	
> Iteración 9	11 días	4/5/2021	107 horas	
Hito Iteración 9	0 días	18/5/2021		
▼ Iteración 10	11 días	19/5/2021	49 horas	
▼ Requisitos	3 días	19/5/2021	24 horas	
Elaboración memoria	1 día	19/5/2021	8 horas	
Elaboración Anexo V	1 día	20/5/2021	8 horas	AS Alfredo Fernández
Elaboración Anexo VI	1 día	21/5/2021	8 horas	AS Alfredo Fernández
▼ Análisis	2 días	24/5/2021	2 horas	
Estudio Anexos	1 día	24/5/2021	1 hora	AS Alfredo Fernández
Estudio memoria	1 día	25/5/2021	1 hora	AS Alfredo Fernández
▼ Diseño	2 días	26/5/2021	2 horas	
Planificación memoria	1 día	26/5/2021	1 hora	
Planificación Anexos	1 día	27/5/2021	1 hora	AS Alfredo Fernández
▼ Implementación	3 días	28/5/2021	20 horas	
Elaboración memoria	1 día	28/5/2021	8 horas	AS Alfredo Fernández
Elaboración Anexos V y VI	2 días	31/5/2021	12 horas	AS Alfredo Fernández
▼ Pruebas	1 día	2/6/2021	1 hora	
Refinamiento de servidor	1 día	2/6/2021	1 hora	AS Alfredo Fernández
Hito Iteración 10	0 días	2/6/2021		
Hito fin transición	0 días	2/6/2021		

Figura 16.PU Transición. Iteración 10

Para más información acerca de la planificación temporal se puede ver el **Anexo I Plan de proyecto**.

5.3 ESPECIFICACIÓN DE REQUISITOS

Para la especificación de requisitos se ha utilizado la metodología de Duran y Bernárdez.

Se ha realizado el catálogo con los requisitos de información, funcionales, no funcionales.

A continuación, se va a exponer un ejemplo de cada una de las partes. Para más información se puede mirar el **Anexo II Especificación y análisis de requisitos**

5.3.1 PARTICIPANTES

Los participantes son:

- Alfredo Fernández Sánchez: alumno y desarrollador del proyecto.

- Jesús Lucas Natal: tutor del proyecto.
- Sara Rodríguez González: tutora académica del proyecto.
- Fernando de la Prieta Pintado: tutor académico del proyecto.

Participante	Alfredo Fernández Sánchez
Organización	Universidad de Salamanca
Rol	Programador
Es desarrollador	Sí
Es cliente	No
Es usuario	No
Comentarios	Ninguno

Tabla 1. Participante

Para cada uno de los participantes del proyecto se ha creado una tabla con los diferentes datos que hay en ella. En este caso está la tabla del alumno.

5.3.2 OBJETIVOS DEL SISTEMA

Los objetivos del sistema son:

- **Detección del entorno:** el sistema deberá acceder a la cámara del dispositivo móvil y así poder obtener información del entorno.
- **Decodificación códigos QR:** el sistema deberá mediante la utilización de la biblioteca XZing, decodificar los códigos QR que haya en el entorno.
- **Búsqueda de marcadores:** el sistema deberá buscar mediante el módulo de seguimiento de imágenes los marcadores que tiene integrados.
- **Gestión de marcadores:** el sistema deberá permitir al administrador de la aplicación gestionar la librería de marcadores para poder añadir o quitar marcadores.
- **Gestión de modelos:** el sistema deberá permitir al administrador de la aplicación gestionar las altas, bajas y modificaciones de los modelos del servidor.
- **Gestión de visualización:** el sistema deberá gestionar la visualización del modelo 3D mediante Realidad Aumentada.
- **Gestión de interacción:** el sistema deberá gestionar la interacción del usuario con el modelo, permitiendo movimientos de rotación y escala.
- **Gestión de herramientas:** el sistema deberá permitir al usuario varias herramientas para informarse o configurar aspectos de esta.

OBJ-0007	Gestión de interacción
Versión	1.0
Autores	• Alfredo Fernández Sánchez

Fuentes	<ul style="list-style-type: none"> • Fernando de la Prieta Pintado • Jesús Lucas Natal • Sara Rodríguez González
Descripción	El sistema deberá <i>gestionar la interacción del usuario con el modelo, permitiendo movimientos de rotación y escala</i>
Subobjetivos	Ninguno
Importancia	importante
Urgencia	inmediatamente
Estado	validado
Estabilidad	alta
Comentarios	Ninguno

Tabla 2. Objetivo del sistema

En esta tabla se detalla la gestión de la interacción. Esta gestión es muy importante ya que trata todo lo relacionado con la rotación y escala del modelo. Sin esta gestión no habría interacción del usuario, y es uno de los requisitos principales del proyecto.

5.3.3 REQUISITOS DE INFORMACIÓN

Es el tipo de requisito que el sistema debe almacenar. Estos son:

- **Información código QR:** el sistema deberá almacenar la información correspondiente al código QR actual. En concreto el texto que lleva asociado.
- **Información sobre posición del marcador:** el sistema deberá ir guardando la posición actual del marcador en cada frame. En concreto las posiciones X, Y, Z del espacio.
- **Información sobre modelos:** el sistema deberá guardar los correspondientes AssetBundles donde se encuentran los modelos. En concreto el nombre del modelo.
- **Información sobre modelos descargados:** el sistema deberá guardar información sobre los modelos descargados. En concreto el nombre y el modelo.
- **Información sobre gestos de usuario:** el sistema deberá poder administrar los gestos del usuario para poder aplicar estos cambios al modelo. En concreto el número de dedos que hay sobre la pantalla y la posición de estos.

IRQ-0002	Información sobre posición del marcador
Versión	1.0
Autores	<ul style="list-style-type: none"> • Alfredo Fernández Sánchez
Fuentes	<ul style="list-style-type: none"> • Fernando de la Prieta Pintado • Jesús Lucas Natal • Sara Rodríguez González

Dependencias	Ninguno	
Descripción	El sistema deberá almacenar la información correspondiente a <i>El sistema deberá ir guardando la posición actual del marcador en cada frame</i> . En concreto:	
Datos específicos	<ul style="list-style-type: none"> • Posición X • Posición Y • Posición Z 	
Tiempo de vida	Medio	Máximo
Ocurrencias simultáneas	Medio	Máximo
Importancia	importante	
Urgencia	inmediatamente	
Estado	validado	
Estabilidad	alta	
Comentarios	Ninguno	

Tabla 3. Requisito de información

Este ejemplo de requisito de información trata sobre la información de la posición del modelo. Es una de las informaciones más importantes ya que sin ella, no se puede colocar el modelo 3D mediante Realidad Aumentada. Guardando bien esta información se puede colocar el modelo y moverse a la posición que se mueva el marcador.

5.3.4 REQUISITOS FUNCIONALES

Estos requisitos tratan sobre cómo se tiene que comportar el sistema en una situación determinada. Para tener una visión inicial del sistema se ha realizado un diagrama de paquetes:

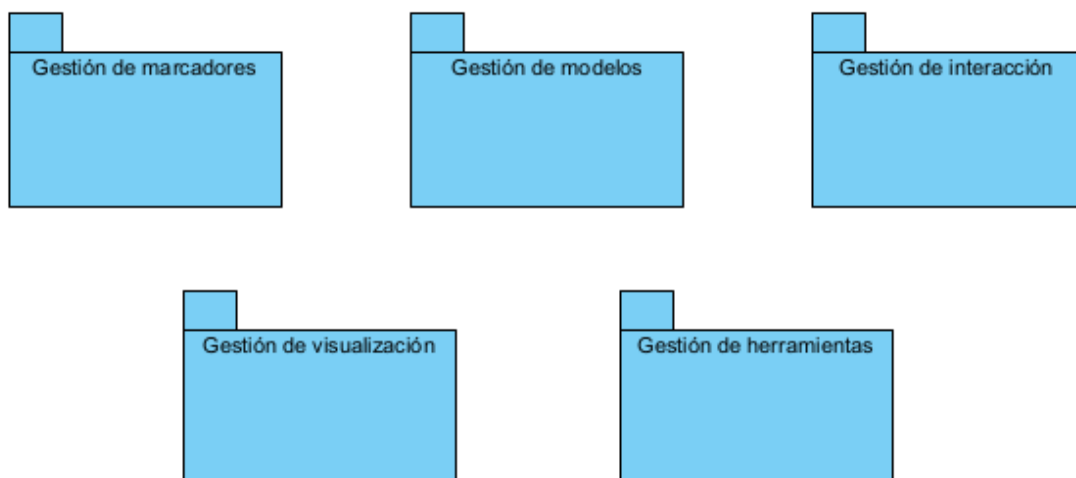


Figura 17. Diagrama de paquetes

Se puede ver en la figura como hay 5 paquetes para las diferentes gestiones, cada uno de ellos gestiona un subsistema diferente. A lo largo de este apartado se irán poniendo ejemplos de cada paquete.

Dentro de los Requisitos funcionales se encuentran también la definición de los diferentes actores que hay en el sistema. A continuación, podemos ver un actor del sistema:

ACT-0001	Administrador
Versión	1.0
Autores	<ul style="list-style-type: none"> • Alfredo Fernández Sánchez
Fuentes	<ul style="list-style-type: none"> • Fernando de la Prieta Pintado • Jesús Lucas Natal • Sara Rodríguez González
Descripción	Este actor representa <i>al administrador de la aplicación, el cual se encargará de esta y del servidor de activos</i>
Comentarios	Ninguno

Tabla 4. Actor administrador

Este actor será el encargado de gestionar cosas como los marcadores, modelos...

Después de determinar los actores del sistema, se han detallado los casos de uso. Estos vienen con los pasos que ha de seguir el actor y el sistema, por ejemplo:

UC-0012	Modificar AssetBundle	
Versión	1.0	
Autores	<ul style="list-style-type: none"> • Alfredo Fernández Sánchez 	
Fuentes	<ul style="list-style-type: none"> • Fernando de la Prieta Pintado • Jesús Lucas Natal • Sara Rodríguez González 	
Dependencias	<ul style="list-style-type: none"> • [OBJ-0005] Gestión de modelos 	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando <i>el administrador modificará un AssetBundle</i>	
Precondición		
Secuencia normal	Paso	Acción
	1	El actor Administrador (ACT-0001) <i>modificara un modelo</i>
	2	El sistema <i>ejecutará de nuevo el script de los AssetBundle</i>
	3	El sistema <i>modificara el AssetBundle</i>

	4	El actor Administrador (ACT-0001) <i>subirá de nuevo el AssetBundle al servidor</i>
	5	El sistema <i>actualizara el AssetBundle en el servidor</i>
Postcondición		
Excepciones	Paso	Acción
	4	Si <i>error de conexión</i> , el sistema <i>rechazara la conexión</i> , a continuación, este caso de uso
Rendimiento	Paso	Tiempo máximo
	-	-
Frecuencia esperada		
Importancia	importante	
Urgencia	inmediatamente	
Estado	validado	
Estabilidad	alta	
Comentarios	Ninguno	

Tabla 5. Modificar AssetBundle

En este caso de uso se gestiona uno de los usos más importantes, que es gestionar los modelos del servidor en caso de que den error o simplemente por modificar algún aspecto de este. En cada caso de uso viene detallado cada uno de los pasos a seguir.

5.3.5 REQUISITOS NO FUNCIONALES

Los requisitos no funcionales sirven imponer al sistema a desarrollar restricciones relacionadas con aspectos principalmente de calidad. Por ejemplo:

NFR-0002	Concurrencia
Versión	1.0
Autores	<ul style="list-style-type: none"> • Alfredo Fernández Sánchez
Fuentes	<ul style="list-style-type: none"> • Fernando de la Prieta Pintado • Jesús Lucas Natal • Sara Rodríguez González
Dependencias	Ninguno
Descripción	El sistema deberá <i>funcionar de manera concurrente, aunque haya varios usuarios accediendo al mismo tiempo al servidor</i>
Importancia	importante
Urgencia	inmediatamente
Estado	validado
Estabilidad	alta
Comentarios	Ninguno

Tabla 6. Concurrencia

En este caso se trata la concurrencia para que, si dos usuarios acceden a la vez al servidor a la descarga de algún modelo, este pueda satisfacer la demanda simultáneamente sin errores.

Hay otros requisitos no funcionales como acceso a la cámara que se le hay que facilitar desde el propio dispositivo móvil.

5.4 ANÁLISIS DE REQUISITOS

En el análisis de requisitos se ha hecho una investigación exhaustiva de los casos de uso determinados. Para ello se ha llevado a cabo un modelo de dominio, realización de casos de uso, clases de análisis y una vista de la arquitectura del modelo de análisis.

5.4.1 MODELO DE DOMINIO

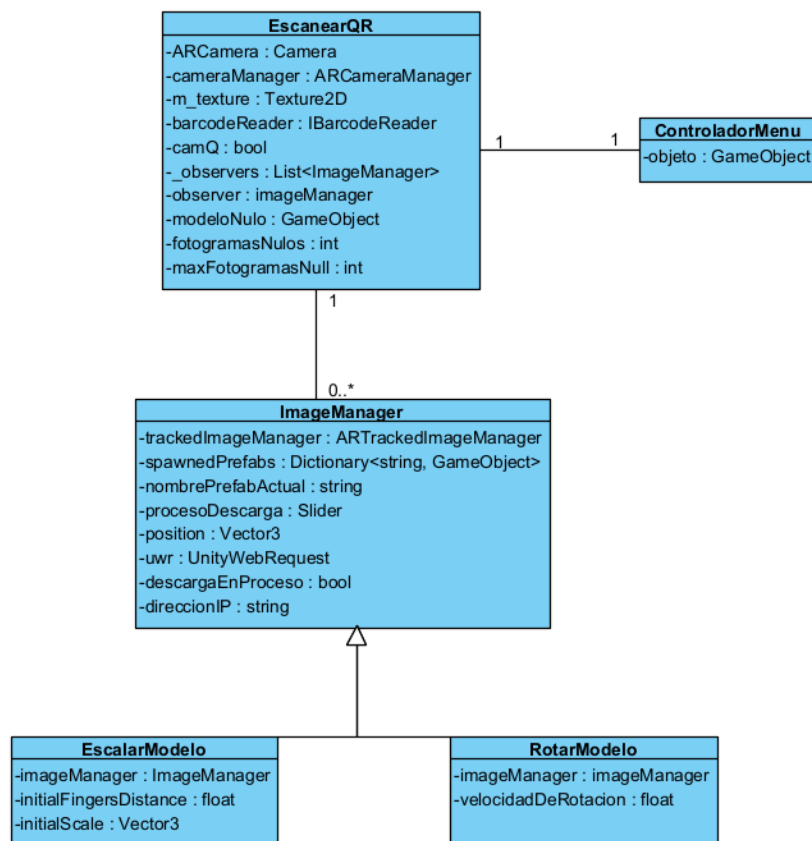


Figura 18. Modelo de dominio

En el diagrama del modelo de dominio podemos diferenciar entre las siguientes clases:

- EscanearQR: clase que accede a la cámara del dispositivo y se encarga de analizar las imágenes obtenidas. Para ello tiene los siguientes atributos:
 - ARCamera: objeto que almacena la cámara del dispositivo.
 - cameraManager: instancia que obtiene la cámara del dispositivo.

- m_texture: textura que almacena un fotograma de la cámara.
- barcodeReader: instancia de la clase IBarcodeReader encargada de analizar la textura obtenida.
- camQ: booleano encargado de gestionar si se ha podido obtener la textura de la cámara o si por el contrario no ha sido posible
- _observers: lista creada para poder diseñar el patrón Observer. Esta lista contiene ImageMangers los cuales se irán añadiendo para así poder notificar si se ha encontrado algún QR con el modelo adecuado.
- observer: objeto creado para poder pasar mediante el panel de Unity, la instancia del ImageManager del proyecto.
- modeloNulo: GameObject creado como nulo, el cual se pasará si se han obtenido al máximo de fotogramas nulos indicados. De esta manera hacemos desaparecer el modelo 3D cuando el código QR haya desaparecido.
- fotogramasNulos: contador que se va incrementando cada vez que un fotograma o bien no tiene un código QR o bien el código QR no ha podido decodificarse.
- maxFotogramasNulos: número máximo de fotogramas nulos que puede haber antes de que se le pase al ImageManager un modelo nulo.
- ControladorMenu: clase creada para poder manejar el comportamiento de los botones de la interfaz gráfica.
- ImageManager: clase creada para manejar el tratamiento de los modelos, así como su descarga del servidor.
 - trackedImageManager: instancia de la clase ARTrackedImageManager que maneja todo lo relacionado con la realidad aumentada. Este objeto maneja el control de los marcadores, así como su búsqueda. De los métodos que nos ofrece podemos obtener varias cosas funcionales, como la posición del marcador o su desaparición.
 - spawnedPrefabs: diccionario creado para poder almacenar los modelos que se han descargado desde el servidor. Se ira guardando como clave el nombre del modelo y como valor el gameobject del modelo en sí.
 - nombrePrefabActual: string que controla que el nombre del último modelo pasado coincida con el modelo que se está manejando en ese momento. Con este nombre se puede o bien guardarlo como clave en el diccionario o bien comprobar que, si ya está guardado, obtener el modelo o la función más importante que es descargar el modelo gracias a este string.
 - procesoDescarga: slider creada para poder presentar en pantalla el proceso que lleva de descarga. Este se ira llenando, dependiendo del proceso de descarga que haya en ese momento.

- position: vector tridimensional creado para almacenar la posición del marcador en el espacio. Así una vez se descargue el modelo o se obtenga del diccionario, podrá colocarse en la posición en la cual está el código QR.
- uwr: instancia de la clase UnityWebRequest necesaria para descargar el AssetBundle del servidor. También con este objeto se va obteniendo el proceso de descarga para ir completando la slider.
- descargaEnProceso: booleano que controla que, si hay una descarga en proceso,
- direccionIP: string creado para almacenar la dirección IP que meta el usuario. Esta se obtiene del objeto que se ha creado en el panel de opciones de Unity.
- EscalarModelo:
 - ImageManager: objeto que obtiene la instancia creada de la clase ImageManager. De esta forma se obtiene el modelo actual que hay en pantalla.
 - initialFingersDistance: float que almacena la distancia inicial entre dos puntos de la pantalla donde ha pulsado el usuario. De esta forma se obtiene la posición inicial y despegar los dedos de la pantalla se obtiene el tanto por ciento de lo que hay que escalar.
 - initialScale: Vector que almacena la escala actual que hay en el sistema.
- RotarModelo:
 - ImageManager: objeto que obtiene la instancia creada de la clase ImageManager. De esta forma se obtiene el modelo actual que hay en pantalla.
 - velocidadDeRotacion: float que almacena la velocidad que se quiere aplicar a la rotación.

5.4.2 REALIZACIÓN DE CASOS DE USO

Se han creado los diagramas de secuencia necesarios para describir los pasos a seguir cada actor en cada caso de uso. A continuación, se expone un ejemplo de diagrama de secuencia:

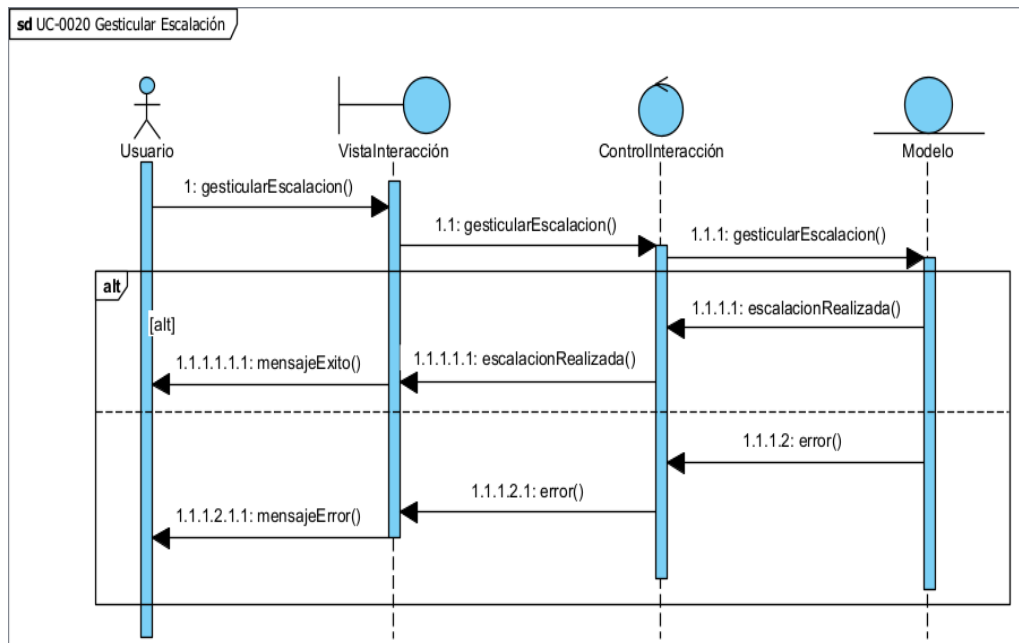


Figura 19. Gesticular Rotación

En este diagrama de secuencia se puede ver los pasos que debe realizar el usuario con el sistema para poder realizar una escalación. Hay un actor que es el usuario, una entidad de vista para la interacción, una entidad de control para controlar que se está realizando de forma correcta, y una entidad de modelo, que se refiere al modelo 3D que se quiere escalar.

5.4.3 CLASES DE ANÁLISIS

En las clases de análisis se han realizado diagramas de comunicación que relacionan las clases que se han expuesto en el apartado anterior:

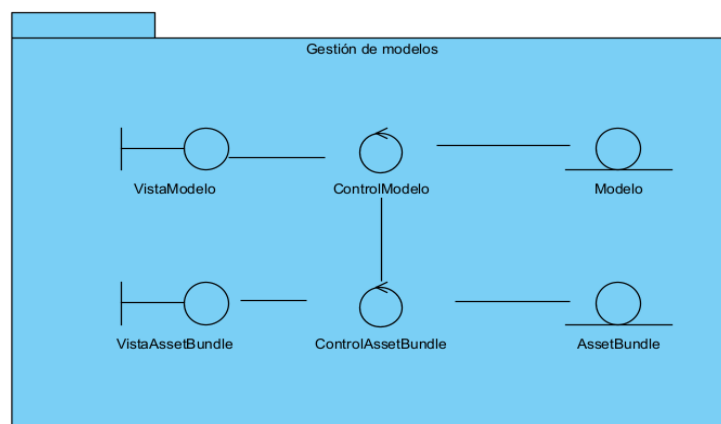


Figura 20. Gestión de modelos

En este diagrama de secuencia se puede ver la relación que existe entre los AssetBundles y los modelos. Cada modelo o AssetBundle tiene su propia vista y entidad. Estos se comunican entre sus entidades de control para poder extraer de un AssetBundle el modelo o partiendo de un modelo crear un AssetBundle.

5.4.5 VISTA ARQUITECTÓNICA DEL MODELO DE ANÁLISIS

En este apartado se muestra de forma general todas las clases de análisis que se han creado:

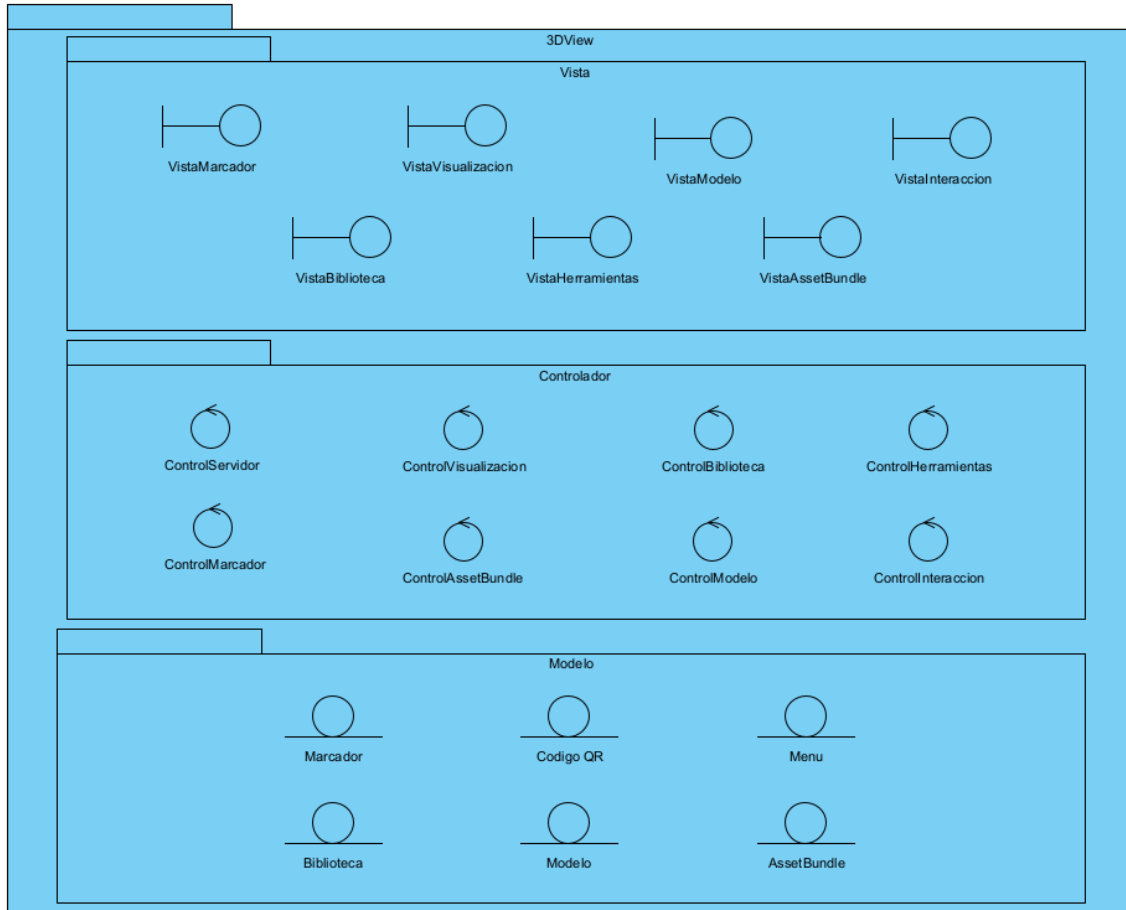


Figura 21. Arquitectura del modelo de análisis

Con esta vista arquitectónica del modelo de análisis se puede ver todos los objetos que hay de cada entidad. De esta forma se puede hacer una vista general de todos los participantes que hay en el sistema.

5.5 DISEÑO DEL SISTEMA

Durante el diseño del sistema se consigue una aproximación muy cercana a la aplicación final. Este va a tener los nombres de las clases, métodos, ... y todo lo que se necesite para la construcción del sistema.

A continuación, se va a poner un ejemplo de cada apartado, para más información se puede consultar el **Anexo III Diseño del sistema software**.

5.5.1 PATRONES ARQUITECTÓNICOS

En este apartado se hace hincapié en los patrones arquitectónicos y de diseño que se han utilizado para el diseño del sistema software.

El patrón utilizado para la arquitectura ha sido el Modelo Vista Controlador (MVC), es un patrón de arquitectura software que separa los componentes software de la aplicación en tres grupos:

- **Modelo:** contiene los datos del sistema. Es independiente de la interacción con el usuario. Almacena datos y así satisface las peticiones del controlador.
- **Vista:** representa los datos que ve el usuario final. A cada vista se asocia con un modelo, pero pueden existir varias vistas para el mismo modelo.
- **Controlador:** se encarga de manejar las instrucciones que hace el usuario y así dar funcionalidad al sistema.

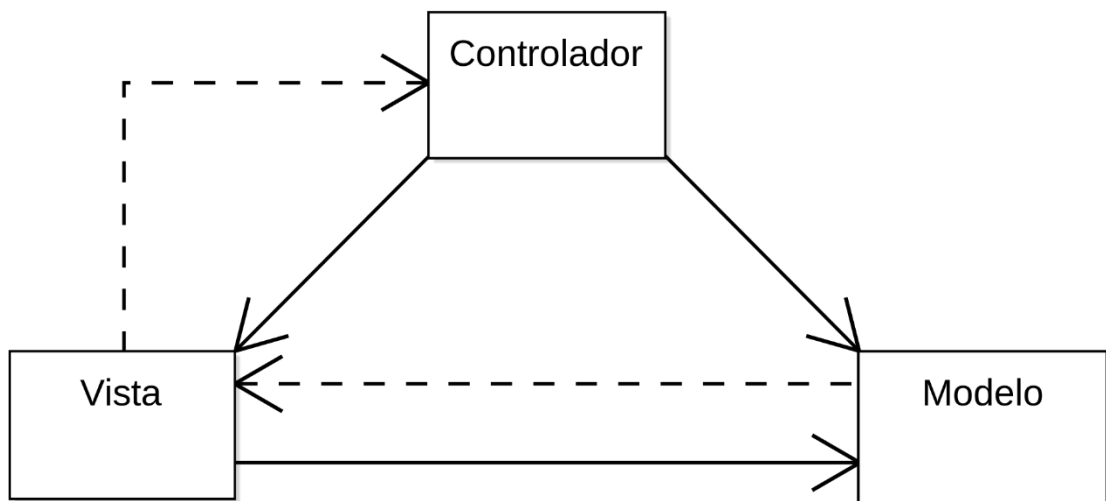


Figura 22. MVC

En este proyecto este patrón ha servido para diferenciar los siguientes subsistemas:

- **Modelo:** se ha usado para almacenar en el servidor los diferentes AssetBundles para su posterior descarga. En el proyecto Unity sirven para guardar los datos relativos a un modelo de una escena.
- **Vista:** nos permite mostrar los datos al usuario, es decir, el modelo 3D mediante Realidad Aumentada. De esta forma permitimos la interacción con el modelo.
- **Controlador:** encargado de controlar tres partes diferentes del sistema. Por una parte, tenemos el controlador del servidor, encargado de la conexión HTTP con la aplicación móvil, así como la seguridad del servidor. Por otro

lado, está el controlador de Unity que nos sirve para poder manejar la escena final. Por último, está el controlador de la app que nos permite manejar la interacción con el modelo, así como controlar el flujo de trabajo.

5.5.2 SUBSISTEMAS DE DISEÑO

Para la realización de subsistemas de diseño se ha dividido la aplicación en sistemas más pequeños a desarrollar. Se compone principalmente de dos paquetes:

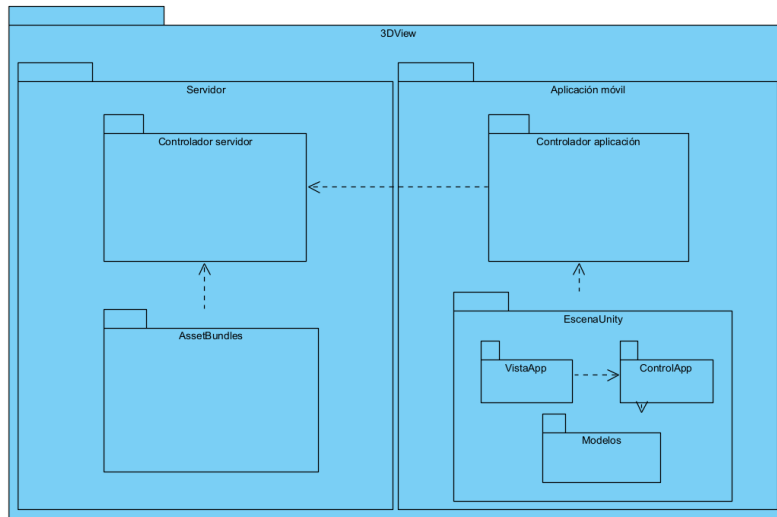


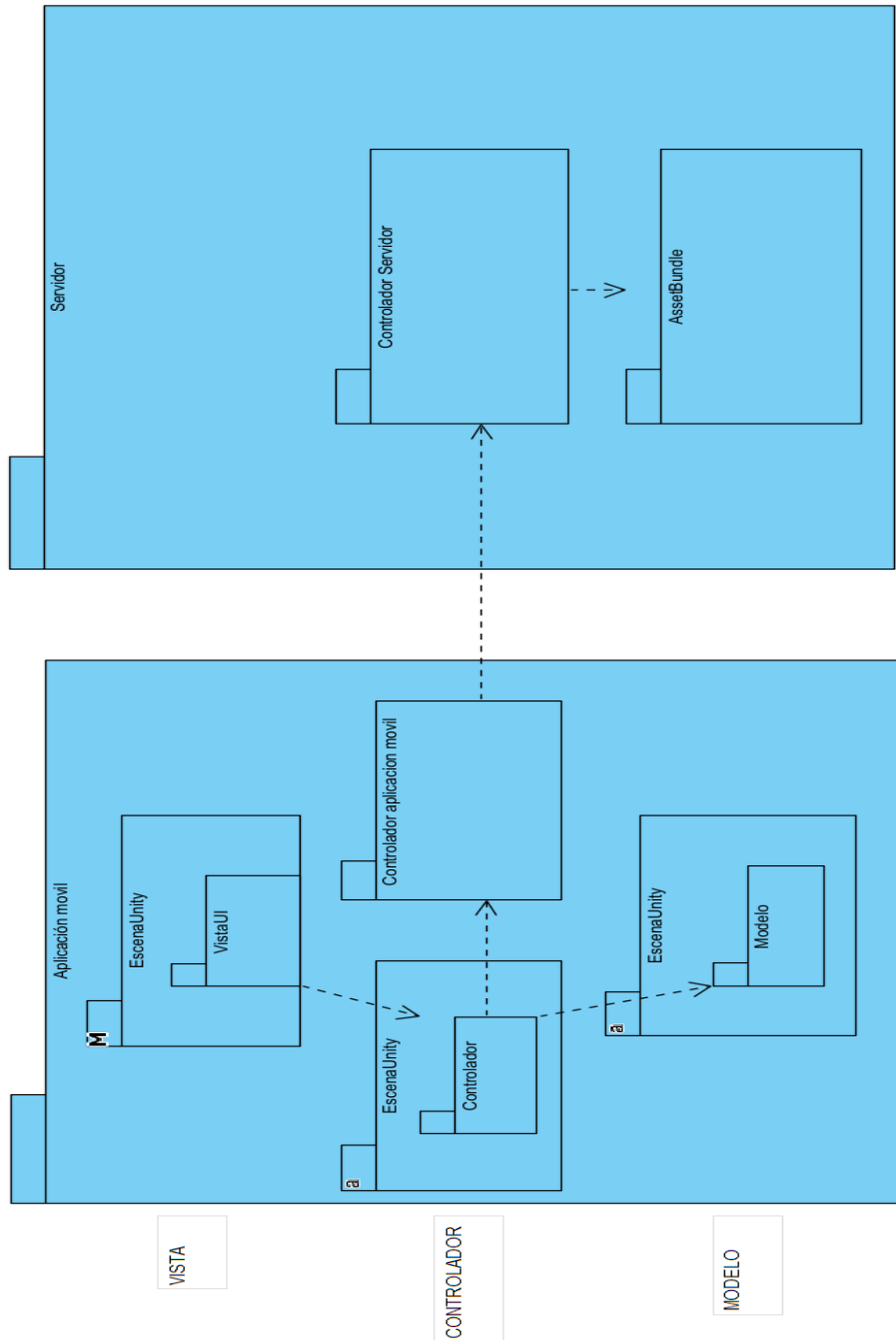
Figura 23. Subsistema de diseño

Estos subsistemas son:

- **Servidor:** dentro del servidor están los Assetbundles, creados y gestionados para poder ser subidos al servidor. Todo esto controlado por otro subsistema como es el Controlador.
- **Aplicación móvil:** que depende principalmente del servidor para poder obtener modelos. También está la escena de Unity en la cual se gestiona e inicializa todo el sistema.

5.5.3 VISTA ARQUITECTÓNICA DEL MODELO DE DISEÑO

Este apartado trata sobre el MVC. Se hace un esquema para ver como quedaría el subsistema:



Se puede ver cómo están colocados todos los subsistemas del patrón MVC explicados en el apartado anterior.

5.5.4 REALIZACION DE CASOS DE USO DISEÑO

En este apartado se han realizado los diagramas de secuencia con los métodos y clases del sistema final:

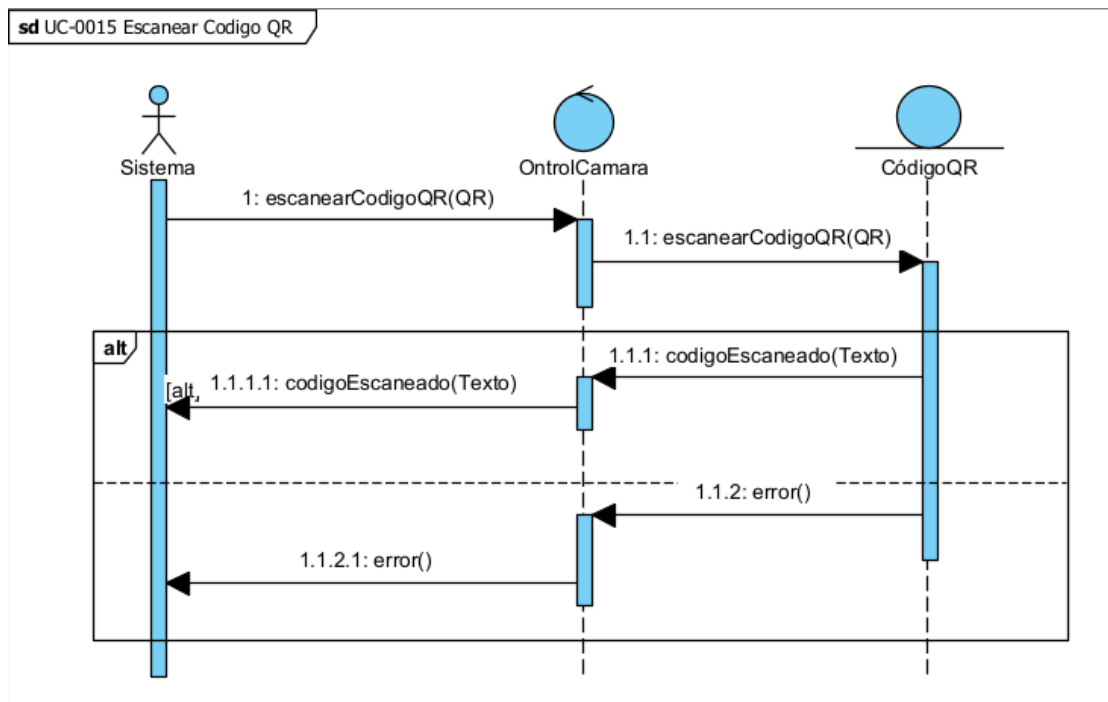


Figura 24. Escanear Código QR

En estos diagramas de secuencia se han creado ya con los métodos que han sido implementados y con sus respectivos parámetros.

5.5.5 MODELO DE DESPLIEGUE

Con el modelo de despliegue vemos el resultado de los nodos desarrollados y el modelo de comunicación entre los mismos:

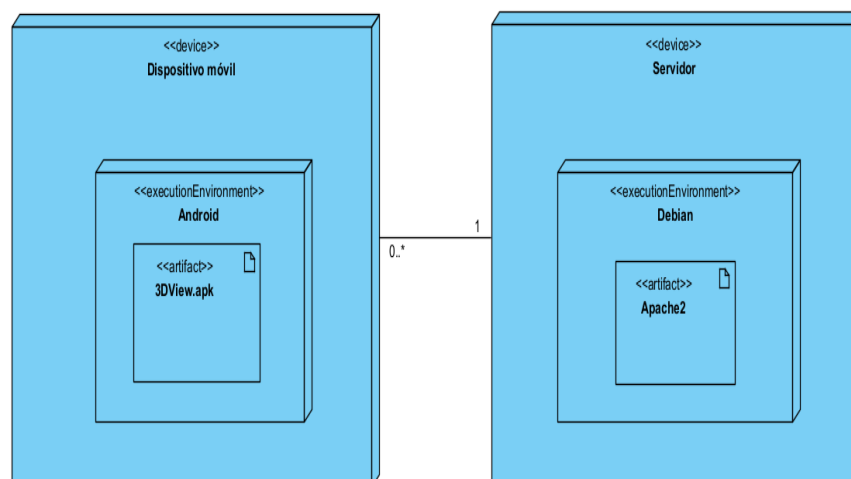


Figura 25. Modelo de despliegue

En el modelo se pueden diferenciar dos dispositivos:

- Dispositivo móvil: el cual hace referencia al smartphone del usuario en cuestión. Será el encargado de instalar la aplicación en él. Para ello se necesita un entorno de ejecución, en este caso será Android, ya que así se estableció en la fase inicial de requisitos. Así podrá instalar 3DView.apk.
- Servidor: encargado de almacenar los modelos. Deberá tener un protocolo HTTP para su conexión por internet. El entorno de ejecución elegido ha sido Debian 10 Buster ya que es flexible y seguro. Con todo esto ya se puede instalar un servidor gratuito APACHE2.

5.6 IMPLEMENTACIÓN

Durante la implementación de la aplicación se ha llevado a cabo la codificación del sistema. Podemos dividir la tarea en:

- App: se ha ido codificando la aplicación en Visual Studio, apoyándose en la herramienta Unity para ir viendo el resultado final.
- Servidor: se ha ido ajustando el servidor final para no tener problemas al descargar el código.

Para una visión más detallada de todas las clases y demás métodos que se han realizado se puede ver el **Anexo IV Documentación técnica**.

5.7 PRUEBAS

Para ir comprobando las distintas funcionalidades del sistema se han ido realizando pruebas aisladas. Estas pruebas han sido de dos tipos:

- O bien en el entorno Unity para comprobar una parte de un subsistema.
- O bien exportando el proyecto al móvil y su posterior prueba en vivo.

Estas han sido muy importantes para poder seguir con el desarrollo de la app.

Como ejemplo de prueba se mostrará como se ha ejecutado la aplicación en un dispositivo móvil, paso a paso:

1. **Configurar Unity:** para que se pueda exportar a Android hay que configurar el entorno Unity. Esta configuración es la siguiente:
 - a. Project Settings->Player Settings->Player->Identification. De esta forma seleccionamos que se soporte para una versión 7.0 o superior.

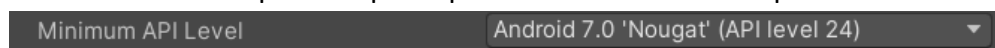


Figura 26. Unity API Level

- b. Project Settings->Player Settings->Player->Configuration. Para seleccionar .NET igual o superior a 4.0
2. **Exportar a Android:** se necesita exportar el proyecto a un tipo de fichero reconocido por Android. Para ello en la pestaña File->Build Settings:



Figura 27. API Compatibility

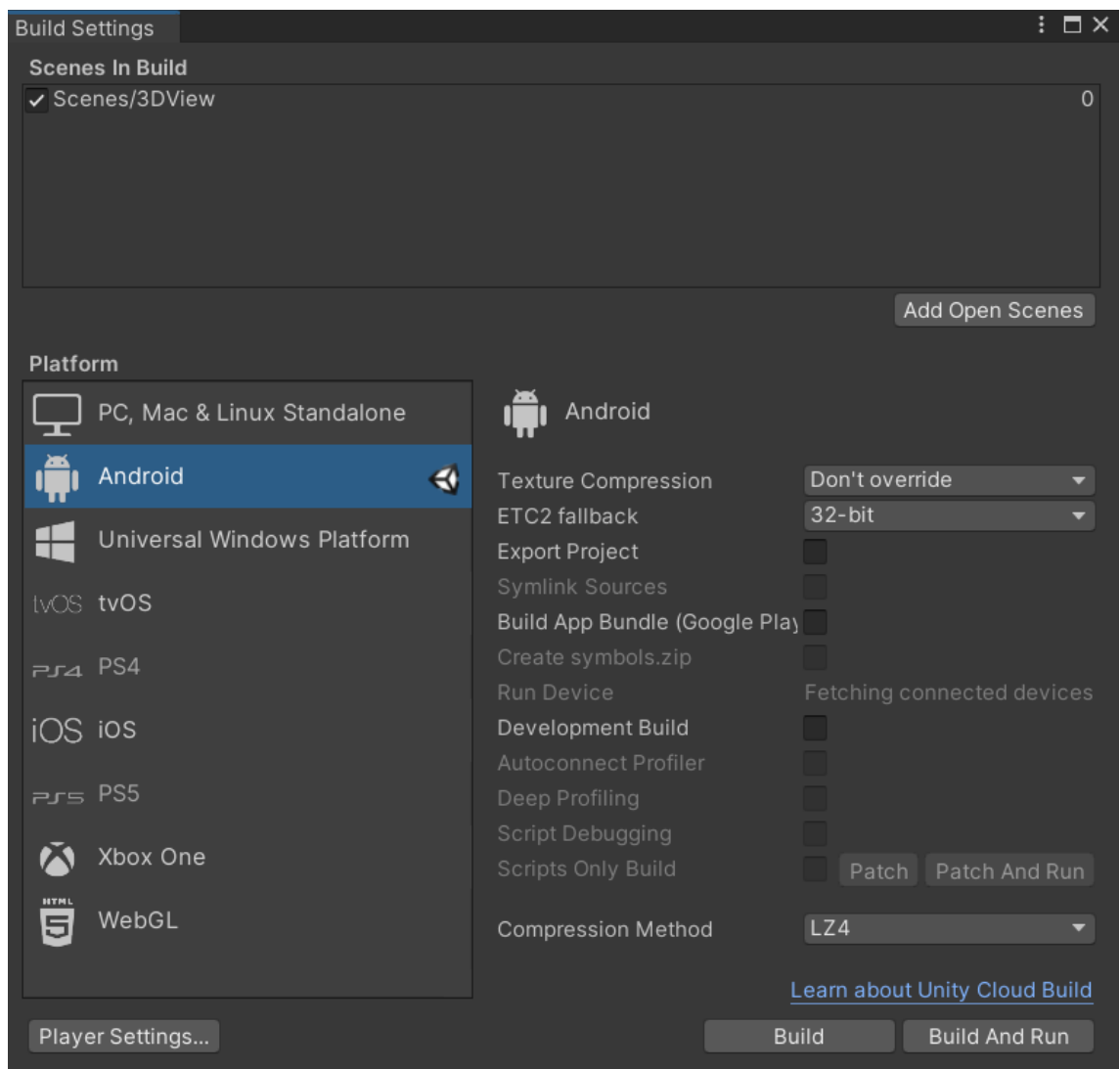


Figura 28. Unity Build Settings

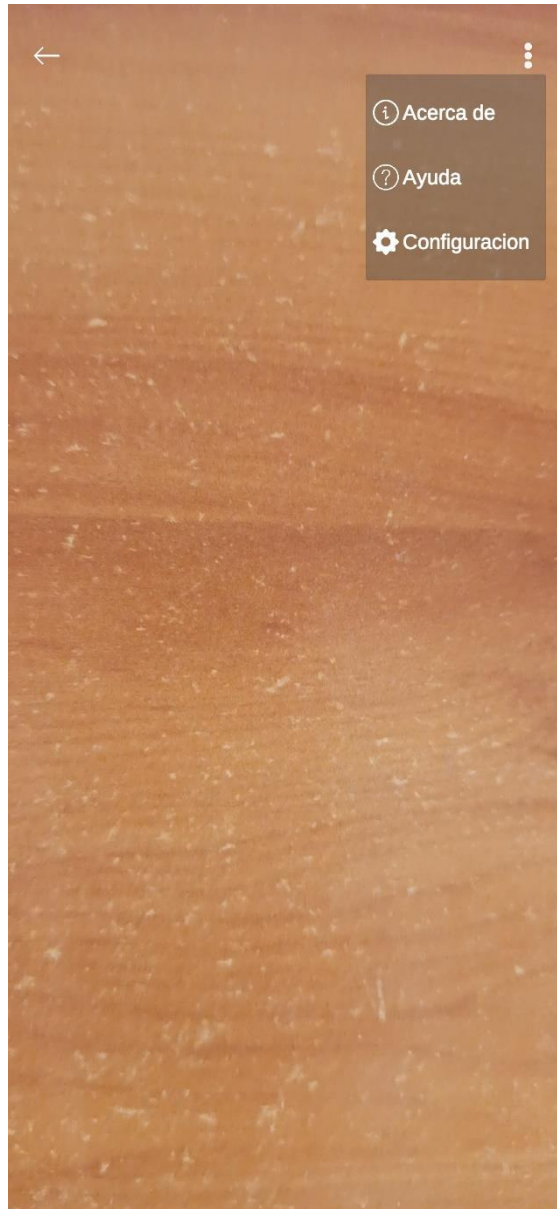
De esta forma se selecciona la opción Android.

3. **Instalar apk:** el fichero creado con extensión .apk se transfiere al dispositivo móvil en el que se va a probar y se procede a su instalación.
4. **Probar aplicación:** se tiene que dar acceso a la cámara por parte de la aplicación. A continuación, ya se puede probar la funcionalidad.

5.8 FUNCIONALIDAD DEL SISTEMA

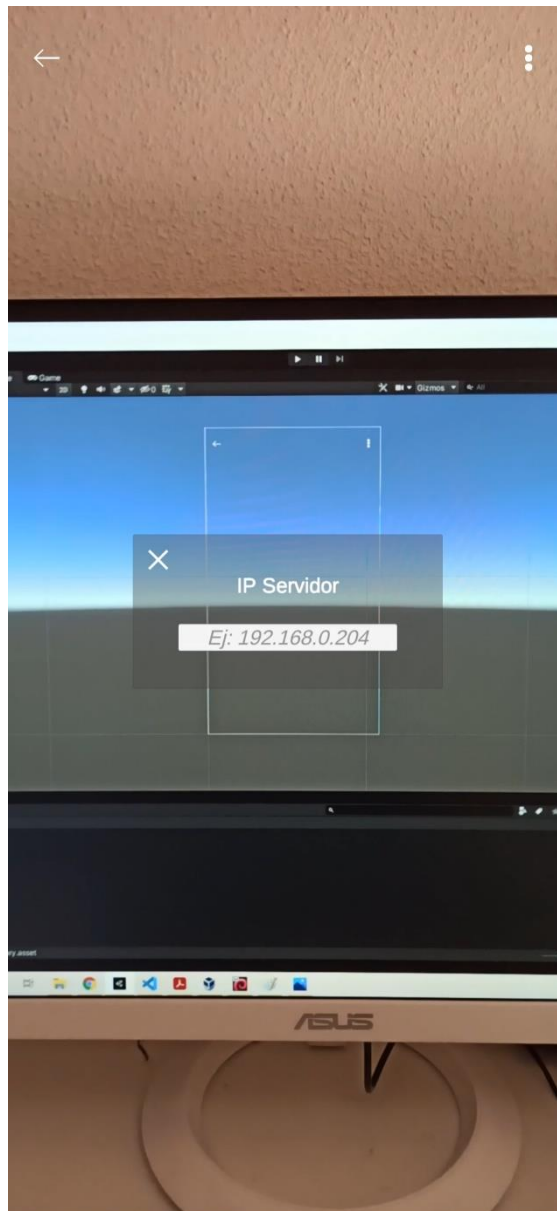
Toda la funcionalidad del sistema viene detallada en el **Anexo V Manual de usuario**. A continuación, se muestran los apartados más relevantes de la funcionalidad final del sistema.

Al iniciar la aplicación podemos ver la pantalla principal, esta tiene la siguiente apariencia:

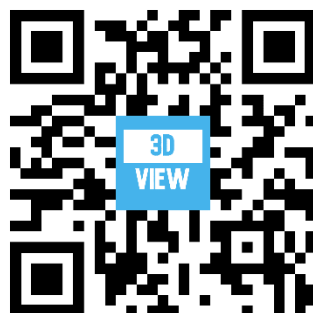


Se puede ver un botón para salir de la aplicación y uno para desplegar un menú. La opción más importante es la de configuración en la cual se escribe la IP del servidor donde están alojados los modelos.

La pantalla de configuración es la siguiente:



Después de escribir la dirección IP del servidor, ya se puede usar la aplicación, para ello habrá que encontrar un código QR específico de la aplicación. Estos códigos tienen el siguiente aspecto:



Al colocar la cámara del móvil sobre el QR, se analizará el contenido de este y se usará como ancla del modelo 3D en el mundo real. Una vez que se descargue el modelo:



Ahora ya se podrán realizar movimientos de rotación o escala con el modelo 3D. También es posible analizar otro código QR y así se descargará otro modelo.

6 LIMITACIONES DE LA APLICACIÓN

Como todo prototipo, este también tiene sus limitaciones. Se van a explicar a continuación cuales son las más significativas:

- La aplicación debe usarse de forma local. El servidor apache no tiene un dominio público propio, por lo que se debe usar de forma local si queremos que funcione correctamente. Por eso se hace uso de una opción de configuración para poder introducir la IP del servidor que se le haya dado a este de forma dinámica.
- No se hace uso de una memoria cache. Los modelos descargados se irán almacenando, pero al cerrar la aplicación perdemos la información de estos. Si se implementase una memoria cache se podrían tener los N últimos más usados disponibles para un uso inmediato.
- Velocidad de descarga. En este caso se depende de la red en la que nos encontremos ya que cada modelo tiene un tamaño, así dependemos directamente de la velocidad de la red.

7 CONCLUSIONES Y LÍNEAS DE TRABAJO FUTURAS

7.1 CONCLUSIONES

Una vez finalizado el proyecto se puede ver como se han cumplido los objetivos de la aplicación, principalmente:

- **Detección del entorno:** este objetivo se ha cumplido gracias a la creación de un objeto ARCamera y mediante el método TryAcquireLastestCpulmage se intenta obtener la última imagen de la cámara del dispositivo.
- **Decodificación códigos QR:** este objetivo como ya se ha dicho se cumple gracias a la librería XZing y a la clase BarCodeReader. La imagen que se ha obtenido de la cámara se convierte a una textura en 2D y después se procesa.
- **Búsqueda de marcadores:** la búsqueda de marcadores se realiza suscribiéndose a los eventos del rastreador.
- **Gestión de marcadores:** este objetivo se puede realizar en el entorno grafico de Unity.
- **Gestión de modelos:** este objetivo se ha obtenido siguiendo la siguiente guía de Unity (Unity D. , 2021)
- **Gestión de visualización:** para la visualización se ha vuelto a hacer uso de la clase ARTrackedImageManager. Al cambiar el marcador, se actualiza el modelo.
- **Gestión de interacción:** este objetivo se cumple gracias a los scripts EscalarModelo y RotarModelo. Se obtiene la instancia del modelo que está en pantalla y se hace la transformación.
- **Gestión de herramientas:** esta gestión se realiza en el editor de Unity con la creación de un canvas y los diferentes objetos que se insertan en él. Toda esta información ha sido consultada en el libro de Marc Lidon. (Mañas, 2018)

Por último, se van a mostrar los objetivos personales que se han cumplido:

- Formar parte del Observatorio tecnológico de HP, tanto es así que se ha trabajado junto a Jesús Lucas Natal, Software Specialist de HP SCDS.
- Poner en práctica todos los conocimientos obtenidos en el grado en Ingeniería informática, de esta forma se ha podido diseñar y construir una aplicación móvil.
- Trabajar con la Realidad Aumentada, creando un proyecto que hace uso de esta tecnología.
- Estudiar y analizar cómo funcionan los códigos QR, de esta forma se ha podido crear y escanear, códigos QR personales.
- Trabajar con herramientas de control de versiones, teniendo así el proyecto alojado en GitLab.
- Obtener conocimientos técnicos necesarios para el desarrollo del proyecto, en concreto:

- Conocimiento del entorno Unity. Manejando todas las opciones del creador y del sistema de exportación de proyectos.
- Conocimiento del lenguaje de programación C#. Aumentando los conocimientos que previamente se tenían.
- Conocimiento de entornos Linux. Haciendo uso práctico de los conocimientos que se tenían de este tipo de SSOO.

7.2 LÍNEAS DE TRABAJO FUTURAS

Las líneas de trabajo futuras vienen ligadas a las limitaciones de la aplicación. Estas pueden ser:

- Obtención de un dominio para hacer público el servidor, así sería posible descargar los modelos del servidor desde cualquier parte del mundo. De esta forma se ahorra el paso al usuario de tener que configurar la dirección IP, bastaría con encontrar un código QR.
- Creación de una memoria cache para un mejor rendimiento de la aplicación. De esta manera aumentaría la velocidad ya que habría N modelos cacheados y no aumentaría el tamaño del diccionario de la app.
- Modelos 3D animados, para una mayor interacción del usuario con la aplicación. Así se tendrían varios modos de uso de la app.

8 REFERENCIAS

- Mañas, M. L. (2018). Unity 3D. En M. Lidon. Marcombo.
- Microsoft. (2021). *C# Documentation*. Obtenido de <https://docs.microsoft.com/en-us/dotnet/csharp/>
- Microsoft. *Microsoft 365*. Obtenido de <https://www.microsoft.com/es-es/microsoft-365/project/project-management-software>
- Paradigm, V. *Visual Paradigm*. Obtenido de <https://www.visual-paradigm.com/>
- Quivervision. (2017). Obtenido de <https://quivervision.com/>
- SCDS, H. (2021). *Observatorio HP. 3DView*. Obtenido de <https://hpscads.com/observatorio-hp/#3DView>
- Sevilla, U. d. *Herramienta REM*. Obtenido de http://www.lsi.us.es/descargas/descarga_programas.php?id=3
- Sketchfab. Obtenido de <https://sketchfab.com/>
- Unity. Obtenido de <https://unity.com/es>
- Unity. *ARFoundation*. Obtenido de <https://unity.com/es/unity/features/arfoundation>
- Unity, D. (30 de Agosto de 2021). *Unity Manual*. Obtenido de <https://docs.unity3d.com/Manual/AssetBundles-Workflow.html>
- Visuartech. Obtenido de <http://www.visuartech.com/>