



UNIVERSIDAD DE BURGOS  
ESCUELA POLITÉCNICA SUPERIOR  
Grado en Ingeniería Informática



**TFG del Grado en Ingeniería  
Informática**

**título del TFG  
Documentación Técnica**



Presentado por Saúl Martín Ibáñez  
en Universidad de Burgos — 21 de marzo  
de 2020

Tutor: Pedro Luis Sánchez Ortega  
Álvar Arnaiz González



---

# Índice general

---

<b>Índice general</b>	<b>I</b>
<b>Índice de figuras</b>	<b>III</b>
<b>Índice de tablas</b>	<b>IV</b>
<b>Apéndice A Plan de Proyecto Software</b>	<b>1</b>
A.1. Introducción . . . . .	1
A.2. Planificación temporal . . . . .	1
A.3. Estudio de viabilidad . . . . .	4
<b>Apéndice B Especificación de Requisitos</b>	<b>7</b>
B.1. Introducción . . . . .	7
B.2. Objetivos generales . . . . .	7
B.3. Catalogo de requisitos . . . . .	8
B.4. Especificación de requisitos . . . . .	8
<b>Apéndice C Especificación de diseño</b>	<b>9</b>
C.1. Introducción . . . . .	9
C.2. Diseño de datos . . . . .	9
C.3. Diseño procedimental . . . . .	9
C.4. Diseño arquitectónico . . . . .	9
<b>Apéndice D Documentación técnica de programación</b>	<b>11</b>
D.1. Introducción . . . . .	11
D.2. Estructura de directorios . . . . .	11
D.3. Manual del programador . . . . .	11

D.4. Compilación, instalación y ejecución del proyecto . . . . .	16
D.5. Pruebas del sistema . . . . .	16
<b>Apéndice E Documentación de usuario</b>	<b>17</b>
E.1. Introducción . . . . .	17
E.2. Requisitos de usuarios . . . . .	17
E.3. Instalación . . . . .	17
E.4. Manual del usuario . . . . .	17
<b>Bibliografía</b>	<b>19</b>

---

## Índice de figuras

---

D.1. Paso 1: Seleccionar la versión de Unity . . . . .	12
D.2. Paso 2: Seleccionar . . . . .	13
D.3. Instalación Vuforia en Unity paso 1. . . . .	14
D.4. Instalación Vuforia en Unity paso 2. . . . .	15

---

## Índice de tablas

---

## Apéndice A

---

# Plan de Proyecto Software

---

### A.1. Introducción

Para la organización del trabajo se ha utilizado GitHub<sup>1</sup> con la extensión de ZenHub<sup>2</sup> para facilitar el seguimiento de las issues gracias a las opciones que ofrece la extensión.

En el siguiente enlace se encuentra el repositorio del trabajo: [https://github.com/smi0010/TFG\\_Herramientas\\_Realidad\\_Aumentada](https://github.com/smi0010/TFG_Herramientas_Realidad_Aumentada).

### A.2. Planificación temporal

En un principio, se ha llevado a cabo una reunión a la semana para cada sprint, en la que se exponían los avances realizados durante el sprint de la semana y la planificación del siguiente.

A continuación se explicarán un resumen de los sprints que se han ido desarrollando durante el proyecto.

#### Sprint 0 (15/01/2020-22/01/2020)

Durante la primera reunión, estuvimos hablando sobre el objetivo del proyecto, de que visión teníamos sobre él, y de hacia donde le podríamos dirigir. Las tareas que se hicieron fueron sobre la creación del repositorio y la instalación de L<sup>A</sup>T<sub>E</sub>X para la documentación. También comencé a investigar

---

<sup>1</sup><https://github.com>

<sup>2</sup><https://www.zenhub.com/>

sobre diferentes herramientas de realidad aumentada de forma general, para poder compararlas y ver las ventajas y desventajas que tienen.

### **Sprint 1 (22/01/2020-29/01/2020)**

Durante este sprint, hemos acordado ir mejorando y completando la investigación respecto las herramientas de realidad aumentada que tenía seleccionadas. Para este sprint decidí centrarme en las herramientas Vuforia y Mergecube. También como uno de los objetivos del TFG es trabajar con Unity para la creación de ejemplos/proyectos, he incluido una pequeña introducción sobre Unity y los pasos necesarios para su instalación.

### **Sprint 2 (29/01/2020 - 5/02/2020)**

Durante el segundo sprint me centré en documentar sobre las diferentes técnicas de realidad aumentada que se utilizan por la mayoría de herramientas. También en documentar las herramientas de ArCore y ArKit.

Otro de los puntos que tratamos, fue sobre posibles ideas para la elaboración de un ejemplo en el que poner a prueba las herramientas AR. Surgió la idea de realizar una especie de juego o aplicación «educativa» que pueda servir como inicio a la programación. Aunque esta idea nos dimos cuenta que depende de cómo la enfoquemos puede ser demasiado compleja y amplia como para hacer un ejemplo sencillo.

### **Sprint 3 (5/02/2020 - 12/02/2020)**

Me centré en la documentación de las herramientas de Kudan y 8thwall. Debido que estuve enfermo gran parte de la semana no pude avanzar más.

### **Sprint 4 (12/02/2020 - 26/02/2020)**

Durante este sprint decidí centrarme en las herramientas de Wikitude, OpenCV, ZapWorks, y en realizar también correcciones y mejoras de la documentación que el tutor había señalado. También estuve pensando en posibles ideas para realizar un ejemplo aplicando las herramientas de realidad aumentada.

### **Sprint 5 (26/02/2020 - 04/03/2020)**

Se estuvo hablando respecto al ejemplo, para empezar ya con él. Sobre la idea de hacer un pequeño juego en el que hay un «laberinto» y que el



usuario tuviera que resolver el camino moviéndose por comandos de una forma similar a una programación por blocking pero simplificado, todo esto aplicado con las herramientas de realidad aumentada.

Dado que aún la idea parecía algo incompleta y complicada de ajustar a las herramientas de realidad aumentada, vamos a intentar primero un ejemplo mas sencillo, este sería uno en la que poder ver las fases de crecimiento en una «planta». Durante el sprint probare a poder usar la cámara de forma que sea como una regadera, así cuando este encima del Mercube donde estaría la planta virtual, se simule en realidad aumentada, que cae agua desde la cámara del teléfono hacia la planta. Otra forma pensada mas simplificada, sería simplemente añadir en pantalla un botón, que al presionarle se simule que cae agua sobre la planta sin tener que mover el teléfono.

### **Sprint 6 (04/03/2020 - 11/03/2020)**

Durante la reunión de este sprint mostré los primeros pasos que había realizado del ejemplo a desarrollar en Unity para aplicar herramientas AR. También comenté mis ideas sobre como continuar y los puntos en que me quería centrar durante este sprint.

Centrarme en poder detectar las partículas que simulan ser agua, cuando estas colisionan con un objeto determinado. Y que contando el número de contactos poder determinar la progresión para poder pasar a otra etapa o nivel. También hacer uso del sensor giroscopio del móvil, para determinar con la inclinación del dispositivo la cantidad de partículas que caen.

Como la idea del ejemplo sería seguir el crecimiento de una planta en realidad aumentada desde un punto educativo, pensamos en que aparte de poder echarle agua, también se podría cambiar entre diferentes opciones, como abonos, sulfatos, etc. También se podrían considerar establecer normas del tipo, escoger la temporada en la que se planta, si necesita que le quiten malas hiervas, etc. Por lo que habría que incluir una forma de cambiar entre esas diferentes opciones.

Otro de los aspectos que he estado mirando, es mejorar el posicionamiento virtual del objeto 3D en el marcador. Ya que con la primera configuración que he probado, cuando se mueve la cámara (el móvil) el objeto se mueve de su posición demasiado y no queda estable.

### **Sprint 7 (11/03/2020 - 18/03/2020)**

Durante la reunión de este sprint , mostré los avances que había realizado sobre el ejemplo de la aplicación. También les explique que estuve realizando varias pruebas añadiendo las funcionalidades del SDK de MergeCube en la aplicación. Consiguiendo mejorar los resultados con el uso conjunto de Vuforia + Mergecube, que anteriormente usando unicamente Vuforia.

Otro de los aspectos tratados fue sobre la herramienta de CoSpace, la necesidad de explicar las ventajas de su uso, para explicar su viabilidad económica.

Durante la semana realice algunos avances en la documentación, y estuve solucionando un bug en el uso del giroscopio. El bug en cuestión se traba de que el eje de gravedad de las partículas, no coincidía con el eje de gravedad normal. El problema resulto ser que, el parámetro «Simulation Space» debía de estar con la opción Word. Otro problema de este bug, era que el eje Z, no tenía la interpretación deseada para nuestro ejemplo. Para esto en el script del giroscopio, se modifiko para invertir el eje Z.

### **Sprint 8 (18/03/2020 - 25/03/2020)**

## **A.3. Estudio de viabilidad**

### **Viabilidad económica**

En este apartado se evaluara la viabilidad económica de las herramientas que se han investigado.

CoSpaces, ofrece varias tarifas aparte de la básica que es gratuita, la más barata de 69,99€ anuales y la mas cara es de 1349,99€ anuales.

La tarifa básica, ofrece la creación de una clase con capacidad para 30 personas (alumnos y profesores). En esta clase virtual, el profesor puede crear unicamente una tarea y asignarse al alumnado de forma individual o por grupos. El número máximo de profesores por clase en la tarifa básica es de uno.

La tarifa básica, permite la creación de un máximo de dos espacios, es decir un máximo de dos proyectos. En la creación de ejemplos, estos se encuentran limitados a unos pocos modelos 3D de CoSpace, y a un limitado número de opciones de programación por CoBlocks. En la tarifa básica no es posible realizar proyectos para el MergeCube.

Con la tarifa Pro de 69,99€, el número de personas máximas para una clase, aumenta en 5, permitiendo también que haya mas de un profesor en la clase. Permite crear mas de una clase. En el apartado de espacios, la tarifa Pro, permite crear mas de dos espacios, y en estos espacios da acceso a todos los modelos 3D de CoSpace, y permite utilizar en la programación todos los bloques de CoBlocks y scripts. Si se desea utilizar el Meregecube, la licencia asciende hasta los 79.99€.

El resto de tarifas que posee CoSpaces, su principal característica es el número de personas simultaneas para una clase que puede tener, subiendo en la ultima tarifa hasta 400 personas adicionales.

La tarifa que sería recomendable se trata de la primera ofertada en el plan Pro, ya que aunque se vaya a utilizar con un número de alumnos superior al ofertado, sería posible organizarse por turnos.

**Costes Personal****Costes Material**

Mergecube...

**Costes Totales****Viabilidad legal**



## *Apéndice B*

---

# Especificación de Requisitos

---

### B.1. Introducción

En el siguiente apartado se especificarán los requisitos y objetivos del ejemplo que se ha desarrollado.

### B.2. Objetivos generales

- El objetivo principal es realizar una aplicación móvil de realidad aumentada, desarrollándola con algunas de las herramientas de realidad aumenta vistas.
- Realizar la aplicación, teniendo como principal marcador el MergeCube.
- Poder ser una aplicación de ámbito educativo.
- Se propone simular el crecimiento de una planta, pudiendo ver las diferentes etapas del desarrollo de la planta.
- Que pueda usarse en el mayor número de dispositivos compatible posibles.

## B.3. Catalogo de requisitos

### Requisitos Funcionales

- **RF 1:** Menú en el que configurar diferentes modos.
  - **RF 1.1:** Seleccionar entre diferentes modos de juego.
  - **RF 1.2:** Ajustar tiempos de cada fase, .
  - **RF 1.3:** Seleccionar posibles tipos de plantas.
- **RF 2:** Botones para seleccionar las diferentes herramientas(regar, abonar, etc).
- **RF 3:** Botón para volver al menú de selección.
- **RF 4:** Una interfaz que indique el progreso de cada fase del juego.
- **RF 5:** Uso del giroscopio del dispositivo, para simular tener gravedad correctamente orientada.
- **RF 6:** Posibilidad de moverse alrededor del objeto representado por realidad aumentada.
- **RF 7:** Ofrecer información educativa implementada dentro del juego.
- **RF 8:** Ofrecer una Ayuda de usuario o tutorial.
- **RF 9:** Poder cambiar de idioma.

## B.4. Especificación de requisitos

## *Apéndice C*

---

# **Especificación de diseño**

---

- C.1. Introducción
- C.2. Diseño de datos
- C.3. Diseño procedimental
- C.4. Diseño arquitectónico





## *Apéndice D*

---

# Documentación técnica de programación

---

## D.1. Introducción

En este apartado se explicaran los pasos para la instalación y ejecución de las herramientas con las que se deberá trabajar.

## D.2. Estructura de directorios

## D.3. Manual del programador

### Unity

Instalación.

Para Instalar el motor Unity es necesario descargar Unity Hub: <https://store.unity.com/es/download-nuo>

Unity Hub se trata de un launcher desde el que se pueden descargar varias versiones de Unity simultáneamente, pudiendo escoger la que mejor se acomode a las necesidades del usuario. También ofrece una serie de tutoriales para iniciarse en el desarrollo de Unity. Por último ofrece un listado de los proyectos del usuario, así como a que versión pertenecen, pudiendo escoger con que versión de Unity desean ejecutarlos.

En la pestaña de installs, pulsando el botón «Add», saldrá una ventana donde poder escoger la versión deseada, al pasar el siguiente paso deberemos

## APÉNDICE D. DOCUMENTACIÓN TÉCNICA DE PROGRAMACIÓN

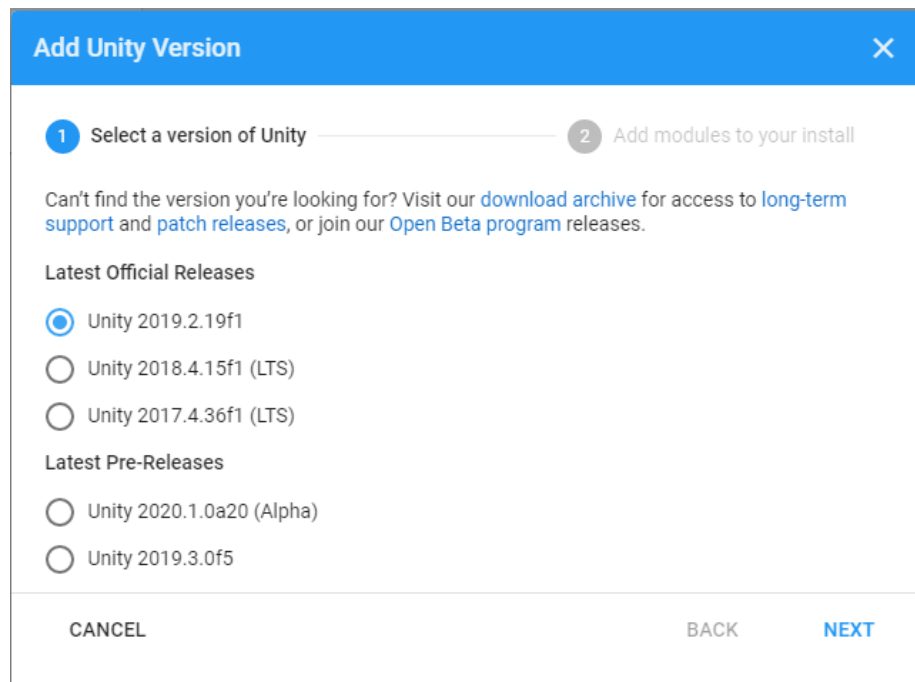


Figura D.1: Paso 1: Seleccionar la versión de Unity

escoger los módulos complementarios para Unity, por el momento sería necesario el de Android, para poder pasar nuestros proyectos a una aplicación Android, importante desplegar las opciones del módulo y seleccionar ambas, pues para la compilación de una aplicación es estrictamente necesario el APK de Android.

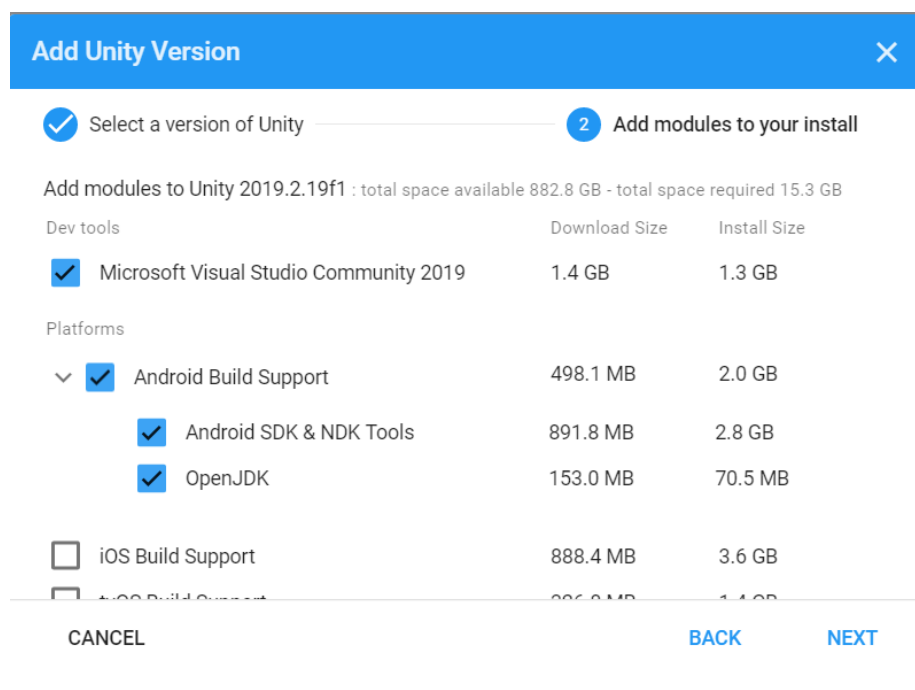


Figura D.2: Paso 2: Seleccionar

Una vez terminada la instalación ya es posible comenzar a crear proyectos. Para poder descargar y usar Assets desde la tienda de Unity es necesario tener una cuenta de usuario.

Dado que en el ejemplo realizado en Unity se utiliza Vuforia como herramienta de realidad aumentada, es necesario añadir dicha herramienta al proyecto de Unity en el se trabajará. Para esto hay varios métodos posibles:

- Desde la pestaña de Edit-Project Settings, en el apartado Player-XR Settings, nos encontraremos con varias herramientas que se han instalado por defecto con Unity, marcamos el check de Vuforia y se incluirá automáticamente en nuestro proyecto.

## APÉNDICE D. DOCUMENTACIÓN TÉCNICA DE PROGRAMACIÓN

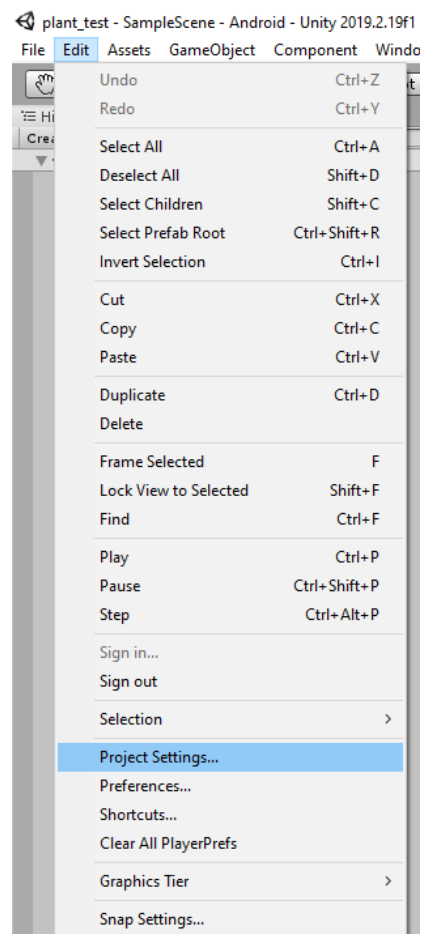


Figura D.3: Instalación Vuforia en Unity paso 1.

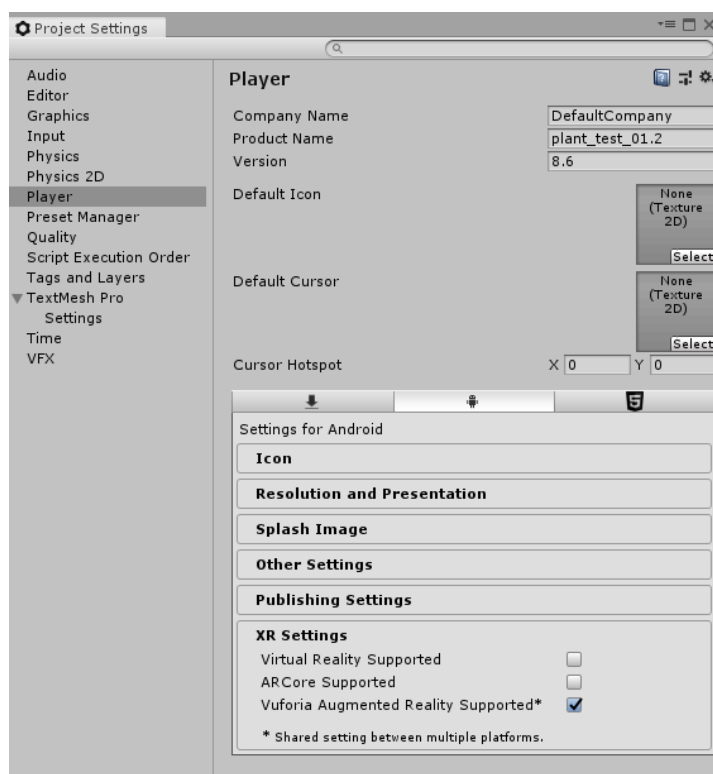


Figura D.4: Instalación Vuforia en Unity paso 2.

## *APÉNDICE D. DOCUMENTACIÓN TÉCNICA DE PROGRAMACIÓN*

- Otra opción posible es, desde la pestaña de la tienda de Assests de Unity, podemos buscar Vuforia Core Samples. Aquí la podremos encontrar de forma gratuita para descargar, y el único proceso que tendremos que seguir es presionar el botón de descargar y seguidamente el botón de importar, de esta manera Vuforia quedará importado en nuestro proyecto. Si seguimos estos pasos también se habrán incorporado al proyecto algunos ejemplos ya contruidos de las distintas funcionalidades de la herramienta.
- Por ultimo, con el SDK de Vuforia para Unity, el cual le podemos descargar desde **developer Vuforia** si tenemos una cuenta de usuario (Una cuenta gratuita es suficiente). Una vez que le hemos descargado, podemos importarlo en el proyecto ejecutándolo mientras tenemos el proyecto Unity abierto, o simplemente arrastrando el ejecutable dentro del proyecto abierto.

### **D.4. Compilación, instalación y ejecución del proyecto**

### **D.5. Pruebas del sistema**

## *Apéndice E*

---

# Documentación de usuario

---

### E.1. Introducción

### E.2. Requisitos de usuarios

El ejemplo desarrollado se trata de una aplicación Android, por lo que el primer requisito necesario es disponer de un Dispositivo Android, con una versión Android 6.0 (Marshmallow) o superior.

Ademas deberá cumplir los siguientes requisitos<sup>[1]</sup>:

- CPU Quad Core.
- 1-2GB de memoria RAM.
- Sensor de Giróscopio.
- Se recomienda una cámara que permita grabar mínimo en 720p.

También es necesario disponer de un MergeCube, es valido tanto en papel como el original.

### E.3. Instalación

### E.4. Manual del usuario





---

## Bibliografía

---

- [1] Vuforia supported devices.