



Trabajo Fin de Grado

Exploración de la Realidad Aumentada en la cocina mediante una experiencia para la preparación de recetas.

Exploration of Augmented Reality in the kitchen through an experience for recipe preparation

Autora

Pilar Fierro Casanova

Directores

Ana Serrano Pacheu

Diego Gutiérrez Pérez

Grado en Ingeniería Informática.

ESCUELA DE INGENIERÍA Y ARQUITECTURA
Febrero de 2025

AGRADECIMIENTOS

En primer lugar, quiero expresar mi más profundo agradecimiento a mis padres, Juan y Rosa. Papá, gracias por despertar en mí la pasión por la ingeniería, por creer siempre en mis sueños y por apoyarme en todas las decisiones que he tomado a lo largo de estos años, incluso en las más alocadas. Mamá, gracias por tu amor, tu paciencia infinita y por estar siempre a mi lado, especialmente en las innumerables horas de espera en médicos y logopedas, siempre mostrándome el valor de la perseverancia.

A mi abuelo, Domingo, cuyo recuerdo me acompaña siempre. Gracias por enseñarme el verdadero significado de amor incondicional y por confiar en que siempre lograría todo lo que me propusiera. Aunque no hayas podido ver este momento, sé que estarías orgulloso de mí.

Quiero agradecer profundamente al Dr. Héctor Vallés Varela y al Dr. Juan Royo López, cuya dedicación y compromiso han cambiado mi vida de manera trascendental. No solo me han dado la oportunidad de escuchar, sino también de vivir plenamente. Asimismo, quiero reconocer al Dr. Graeme Clark, cuyo revolucionario invento, el implante coclear, ha abierto un mundo de sonidos y posibilidades para mí. Este trabajo no es solo un logro académico, sino también un testimonio de su vocación y compromiso, demostrando cómo su esfuerzo transforma vidas.

A mi primera profesora, Noelia, y a todas las logopedas, que me han acompañado con paciencia, apoyo y dedicación a lo largo de este camino lleno de aprendizajes y desafíos. Cada sesión, cada palabra y cada avance fueron fundamentales para que hoy pueda estar aquí.

Finalmente, quisiera agradecer a todas aquellas personas que han confiado en mí, creído en mi capacidad, me han apoyado incondicionalmente y han estado a mi lado, como mis amigos. Este logro no es solo mío, sino también de todos vosotros.

Como dijo Mario Benedetti: “La perfección es una pulida colección de errores”. Gracias a cada uno de ustedes, mis errores se han convertido en aprendizajes y mis aprendizajes en logros.

Este trabajo ha sido parcialmente financiado por el Programa de Becas y Ayudas del Instituto de Investigación en Ingeniería de Aragón (I3A) y por la Cátedra URBA SER e INETUM

Exploración de la Realidad Aumentada en la cocina mediante una experiencia para la preparación de recetas.

RESUMEN

A través del desarrollo de una experiencia para dispositivos Android, este Trabajo Fin de Grado explora el potencial de la Realidad Aumentada (Augmented Reality, AR) en el ámbito culinario. Facilita la preparación de recetas al fusionar el entorno digital con el espacio físico, ofreciendo un seguimiento visual y guiado para eliminar la necesidad de recurrir a la consulta de recetarios tradicionales.

En su fase inicial, se ha diseñado un prototipo del sistema que permite a los usuarios seleccionar recetas mientras visualizan el plato final en formato 3D. A través de una interfaz intuitiva, los usuarios pueden interactuar con ingredientes virtuales, eliminarlos o seleccionarlos según las preferencias personales. Todo ello mientras recibe indicaciones claras y un temporizador que le guía en cada paso del proceso culinario.

Durante el desarrollo se han encontrado diferentes desafíos, como por ejemplo, mejorar la precisión en la detección de superficies en entornos como la cocina, que suelen ser irregulares o estar parcialmente obstruidos. Para superarlos, se ha optado por un sistema basado en marcadores visuales, mejorando la estabilidad y flexibilidad en la colocación de elementos virtuales.

Para desarrollar esta experiencia, se han analizado y seleccionado las herramientas específicas de Realidad Aumentada. El entorno de desarrollo empleado fue Unity, y las pruebas se llevaron a cabo en dispositivos Android, concretamente en la tablet Xiaomi Pad 6, ya que no se disponía de gafas de AR.

El resultado final es una experiencia que moderniza la preparación culinaria mediante el uso de la AR. En el futuro, esta solución podría integrarse en dispositivos más avanzados como gafas de AR. Por ello, la interacción del usuario en la experiencia se ha diseñado para ser lo más similar posible a la que tendría con unas gafas de AR, demostrando así como la tecnología puede transformar tareas cotidianas y abrir nuevas oportunidades en diversas áreas.

Exploration of Augmented Reality in the kitchen through an experience for recipe preparation

SUMMARY

Through the development of an experience for Android devices, this Final Degree Project explores the potential of Augmented Reality (AR) in the culinary field. It simplifies the preparation of recipes by combining the digital and physical environments, offering visual and guided assistance to eliminate the need to rely on traditional recipe books.

In its initial phase, a prototype of the system was designed, allowing users to select recipes while viewing the final dish in 3D format. Through an intuitive interface, users can interact with virtual ingredients, removing or selecting them based on personal preferences. All of this is complemented by clear instructions and a timer to guide users through each step of the cooking process.

During development, several challenges were encountered, such as improving the accuracy of surface detection in environments like kitchens, which are often uneven or partially obstructed. To address these issues, a marker-based visual system was implemented, improving the stability and flexibility of placing virtual elements.

To create this experience, specific AR tools were analyzed and selected. The development environment used was Unity, and testing was conducted on Android devices, specifically the Xiaomi Pad 6 tablet, as AR glasses were not available.

The final result is an experience that modernizes the cooking process through the use of AR. In the future, this solution could be integrated into more advanced devices such as AR glasses. For this reason, the user interaction in the experience was designed to be as similar as possible to what it would be like with AR glasses, showing how technology can transform everyday tasks and create new opportunities in various fields.

Índice

1. Introducción	7
1.1. Motivación	7
1.2. Objetivos	8
1.3. Metodología y enfoque del trabajo	9
2. Realidad Aumentada en la actualidad	11
2.1. Aplicaciones de la Realidad Aumentada en diferentes sectores	11
2.2. AR en la cocina	12
2.2.1. Kittch	12
2.2.2. Vodafone-Giga AR	13
3. Diseño y desarrollo	15
3.1. Herramientas Empleadas	15
3.2. Unity en este TFG	16
3.2.1. Editor de Unity	16
3.2.2. GameObjects y Componentes	17
3.2.3. Sistema de <i>Scripts</i>	18
3.2.4. Plantilla AR Mobile	18
3.3. Configuración inicial del proyecto	19
3.3.1. Detección de planos	20
3.4. Diseño y desarrollo del proyecto	21
3.4.1. Detección del marcador	22
3.4.2. Eliminar ingredientes mediante Raycasting	24
3.4.3. Temporizador e información de los pasos de la receta	25
3.4.4. Resultado final: una experiencia personalizada e intuitiva	26
4. Conclusiones y trabajo futuro	27
4.1. Conclusión personal	27
4.2. Futuras líneas del trabajo y mejoras	27

5. Bibliografía	29
Lista de Figuras	33
Anexos	36
A. Aplicaciones de la Realidad Aumentada	39
A.1. Ámbito Sanitario - AccuVein	39
A.2. Sector comercial - IKEA Place	40
A.3. Apple - App Medidas	41
A.4. Sector Educativo - Chromville	42
B. Experiencia de AR con Kittch	43
C. Experiencia de AR con Vodafone-Giga AR	49
D. Plantilla AR Mobile de Unity	53
E. AR Foundation	57
F. Scripts	59
F.1. <i>ARPlaneManager.cs</i>	59
F.2. <i>ARTrackedImageManager.cs</i>	62
F.3. <i>Timer.cs</i>	65
G. Interfaz de la prueba de concepto	67

Capítulo 1

Introducción

1.1. Motivación

En la actualidad, la Realidad Aumentada (AR) se encuentra en constante evolución, ofreciendo experiencias inmersivas donde el usuario se adentra en un universo digital que se fusiona con el mundo real que le rodea. Desde mi experiencia personal y profesional, considero que esta tecnología tiene el potencial de transformar nuestras vidas en actividades cotidianas tan simples como cocinar. Facilitando la preparación de recetas, eliminando la necesidad de consultar recetarios tradicionales y permitiendo al usuario ser guiado paso a paso en su entorno.

Este Trabajo Fin de Grado (TFG) se ha realizado en colaboración con el grupo de investigación *Graphics and Imaging Lab* [1] de la *Universidad de Zaragoza*, dentro de la línea de colaboración que tienen con la empresa *BSH* [2]. Donde se ha diseñado y desarrollado una experiencia de Realidad Aumentada en el ámbito culinario, enriqueciendo la preparación de recetas culinarias mediante guías interactivas con información visual y avisos en tiempo real.

Durante mis prácticas en la empresa *IMASCONO* [3], tuve la oportunidad de conocer proyectos innovadores, como la experiencia de Realidad Virtual desarrollada en colaboración con el chef *Ferran Adrià*, *Telefónica* y *Disney*. En ese proyecto, se creó un recetario para niños basado en las historias de *Disney*, *Pixar*, *Star Wars* y *Marvel*, en una aplicación educativa y divertida para unir a las familias. Inspirada por estas experiencias y motivada por mi interés en la Realidad Aumentada, así como mi deseo de hacer el proceso de cocinar recetas más atractivo, decidimos crear una experiencia en Realidad Aumentada.

El desarrollo de la experiencia se ha enfocado para dispositivos Android. En primer lugar diseñamos un prototipo inicial que permite al usuario seleccionar recetas visualizando el plato final en 3D. A través de una interfaz intuitiva, es posible interactuar con ingredientes virtuales, personalizar la receta y además se guía paso a paso junto con un temporizador integrado.

Para ello, hemos empleado herramientas como Unity y técnicas basadas en un sistema de marcadores visuales, superando los desafíos asociados a superficies como la encimera, la cual puede ser irregular o estar obstruida con objetos.

Como resultado, se obtiene una experiencia que moderniza la cocina mediante el uso de la Realidad Aumentada, haciendo el proceso más interactivo y accesible. Este proyecto también abre nuevas vías de investigación a futuro, así como oportunidades para la integración de novedades en la experiencia.

1.2. Objetivos

Para conseguir el objetivo principal de este TFG: explorar el potencial de la Realidad Aumentada como herramienta para transformar la experiencia convencional de cocinar, en colaboración con el grupo de investigación de *Graphics and Imaging Lab [1]* de la *Universidad de Zaragoza*, hemos desarrollado una prueba de concepto diseñada para ayudar al usuario mediante este tipo de tecnologías. Para ello se han marcado los siguientes objetivos a lograr:

- Familiarizarse con el entorno de desarrollo Unity y las herramientas o dispositivos relacionados con AR.
- Explorar las funcionalidades básicas de las plantillas de AR en Unity.
- Diseñar y desarrollar una experiencia interactiva de AR aplicada a recetas de cocina.
- Crear una prueba de concepto que facilite la comprensión y ejecución de recetas, haciendo el proceso lo más accesible para el mayor tipo de usuarios posibles.
- Fomentar el uso de nuevas tecnologías y habilidades digitales, impulsando nuevas tendencias en la industria alimentaria y culinaria.

Como resultado final se busca conseguir esa experiencia intuitiva lo más accesible posible y de forma didáctica.

1.3. Metodología y enfoque del trabajo

Una vez definidos los objetivos, el desarrollo de este TFG se ha organizado en las siguientes etapas para garantizar un enfoque estructurado y eficiente:

1. Investigación preliminar de las herramienta y tecnologías actuales, tanto consolidadas como emergentes relacionadas con la Realidad Aumentada.
2. Familiarización con Unity y con la plantilla que este proporciona para Realidad Aumentada, además de explorar sus posibilidades de implementación.
3. Diseño y prototipado básico, creando una estructura inicial para la prueba de concepto.
4. Implementación y desarrollo del prototipo para dispositivos Android. Realizando los ajustes necesarios según las limitaciones encontradas.
5. Elaboración de esta memoria, en la que se recoge el trabajo realizado para conseguir la prueba de concepto.

A continuación, se presenta un diagrama de *Gantt* que detalla las horas dedicadas durante los meses en los que se ha realizado este TFG ([Figura 1.1](#)).

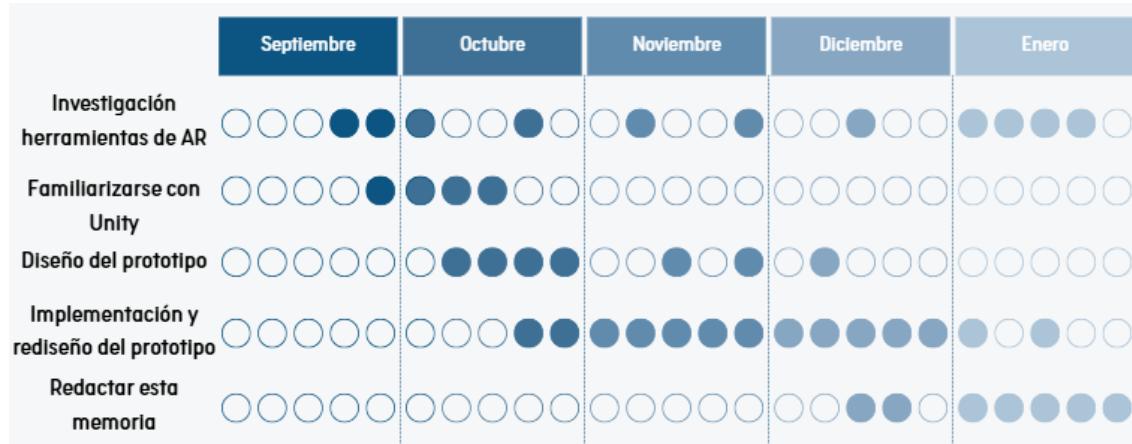


Figura 1.1: Diagrama de *Gantt* con las tareas realizadas a lo largo de los meses.

Como se aprecia en el diagrama, las fases de investigación y diseño se han desarrollado de forma continuada a lo largo del proyecto. Dado que el campo de la Realidad aumentada se encuentra en constante evolución, fue necesario realizar una investigación continua para identificar herramientas emergentes y nuevas soluciones. Por su parte, el diseño del prototipo se ha realizado en múltiples ocasiones para adaptar la experiencia a las limitaciones encontradas durante la etapa de desarrollo e implementación. Se han dedicado un total de 379 horas, de las cuales 289 corresponden a la investigación de herramientas, diseño, familiarización con Unity y la implementación.

Las 90 horas restantes corresponden a la elaboración de esta memoria, investigación de algunos aspectos abordados en ella y sus correcciones.

Capítulo 2

Realidad Aumentada en la actualidad

Esta tecnología se está posicionando como una herramienta revolucionaria, como una tendencia a futuro, encontrándose en diferentes campos o sectores. Su aplicación aporta valor en la relación entre personas y objetos, y mostrando a las empresas nuevas formas de extender la experiencia del cliente más allá del producto o servicio.

En este capítulo se describen los usos en los que podemos encontrar la Realidad Aumentada hoy en día en diferentes sectores. Además, se describen un par de aplicaciones de esta tecnología en el ámbito culinario, en el contexto de recetas de cocina.

2.1. Aplicaciones de la Realidad Aumentada en diferentes sectores

En el ámbito sanitario, [AccuVein](#) ha desarrollado un dispositivo que permite visualizar las venas de los pacientes que no se aprecian a simple vista, mejorando así la precisión de los procedimientos médicos. En el sector comercial, IKEA cuenta con una aplicación móvil llamada [IKEA Place](#), que permite a los clientes visualizar los productos del catálogo en sus propios espacios reales. Apple, por su parte, ofrece la aplicación [Medidas](#), que permite calcular las dimensiones de objetos reales a través de la Realidad Aumentada. En la misma línea, esta herramienta se está mostrando como tendencia en el sector educativo. Ejemplo de ello es [Chromville](#), una aplicación educativa donde plantillas de dibujos para colorear cobran vida en 3D al ser escaneados.

2.2. AR en la cocina

Dentro del mundo culinario, se cuenta con dos ejemplos destacados, como son Kittch y Vodafone-Giga AR.

2.2.1. Kittch

Kittch es una plataforma de streaming en directo que permite a chefs y usuarios generar ingresos mediante clases de cocina o la venta de productos. De la mano del Chef Ming Tsai y la multinacional Qualcomm, esta plataforma incorpora una experiencia de cocina inmersiva en AR.

La experiencia permite a los usuarios la posibilidad de visualizar, mediante unas gafas de AR, un vídeo donde el chef guía paso a paso en la preparación de un plato. Además, se puede vincular a la plataforma de Walmart, permitiendo añadir de forma rápida y sencilla los posibles ingredientes que hacen falta comprar.

No obstante, la experiencia cuenta con ciertas limitaciones técnicas. Las gafas de AR deben estar conectadas permanentemente mediante cable a un dispositivo móvil, en el que debe estar instalada la aplicación de Kittch. Además, parte de los gestos e interacción del usuario, como realizar algunos gestos, deben hacerse desde el teléfono y no a través de las gafas.

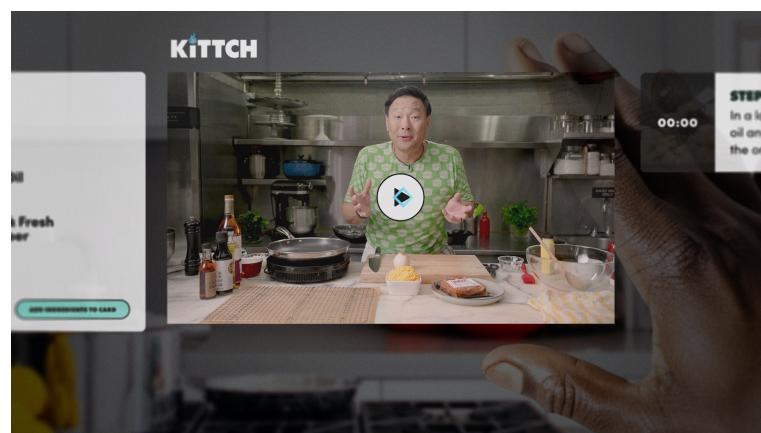


Figura 2.1: Ejemplo de experiencia AR con Kittch

En el [Anexo B](#) se pueden consultar más imágenes de la experiencia.

2.2.2. Vodafone-Giga AR

Vodafone, en colaboración con Nreal, Virtual Voyages y Optiva Media, ofrece una experiencia de AR en la cocina similar a la de Kittch, pero con la incorporación virtual del Chef Steffen Henssler, que guía al usuario apareciendo en su campo de visión.

Al igual que en el caso de Kittch, las gafas de AR necesitan estar conectadas a un dispositivo móvil mediante cable para desplegar la experiencia guiada. Sin embargo, la principal diferencia radica en el método de interacción. En este caso los usuarios no interactúan con las manos, sino que emplean la mirada para apuntar hacia botones o áreas específicas de la interfaz. Previo al inicio de la experiencia, el dispositivo móvil al que están conectadas las gafas, actúa como mando.



Figura 2.2: Ejemplo de experiencia AR con Vodafone-Giga AR

En el [Anexo C](#) se pueden consultar más imágenes de la experiencia.

Capítulo 3

Diseño y desarrollo

Este capítulo detalla el trabajo completo para alcanzar el resultado final de este TFG: el diseño y la implementación de una experiencia de Realidad Aumentada orientada a la preparación de recetas culinarias. Fusionando el entorno digital con el espacio físico del usuario, permitiéndole seguir los pasos de las recetas sin necesitar consultar los recetarios tradicionales.

Durante el trabajo realizado, dividido en varias fases, nos familiarizamos con las herramientas que hemos empleado, acompañado de una posterior investigación de las posibilidades y limitaciones que disponen dichas herramientas para dar vida a la experiencia interactiva. Esto nos ha guiado a la hora de tomar algunas decisiones de diseño, buscando soluciones alterativas para superar los desafíos encontrados.

3.1. Herramientas Empleadas

En el desarrollo de este TFG, se han empleado el siguiente conjunto de herramientas, que han facilitado su implementación:

- Unity: motor principal para desarrollar la aplicación.
- Visual Studio 2022: empleado para la escritura y depuración de *scripts*. Su integración con Unity facilita el desarrollo del código y la detección de errores.
- GitHub: ha servido como repositorio para guardar las distintas versiones de los *scripts*, pudiendo mantener un control eficaz.
- Recursos online: se han consultado tutoriales en plataformas como YouTube y la documentación oficial de Unity para resolver dudas, optimizando así el desarrollo.
- Dispositivo Android: una tablet Xiaomi Pad 6, con una versión de Android 14. Empleada para realizar las pruebas de la experiencia en un entorno real, asegurando la compatibilidad y el rendimiento.

3.2. Unity en este TFG

Unity es un motor de desarrollo de videojuegos y aplicaciones interactivas que se emplea para dar vida a proyectos en 2D, 3D, Realidad Virtual y Realidad Aumentada. A continuación, se detallan los aspectos más relevantes de su uso en este proyecto.

3.2.1. Editor de Unity

Dentro del editor de Unity se encuentran varios paneles, que facilitan la organización y manipulación del proyecto. Entre ellos destacan:

- **Project:** contiene todos los recursos (*assets*) que forman el proyecto, como texturas, materiales, *scripts* o escenas.
- **Hierarchy:** muestra la jerarquía de objetos (*GameObjects*) que componen la escena actual.
- **Scene:** representa el entorno 3D donde se colocan y editan los objetos. Permitiendo la vista previa del escenario del juego.
- **Inspector:** Muestra las propiedades del *GameObject* o *asset*, permitiendo editarlas.

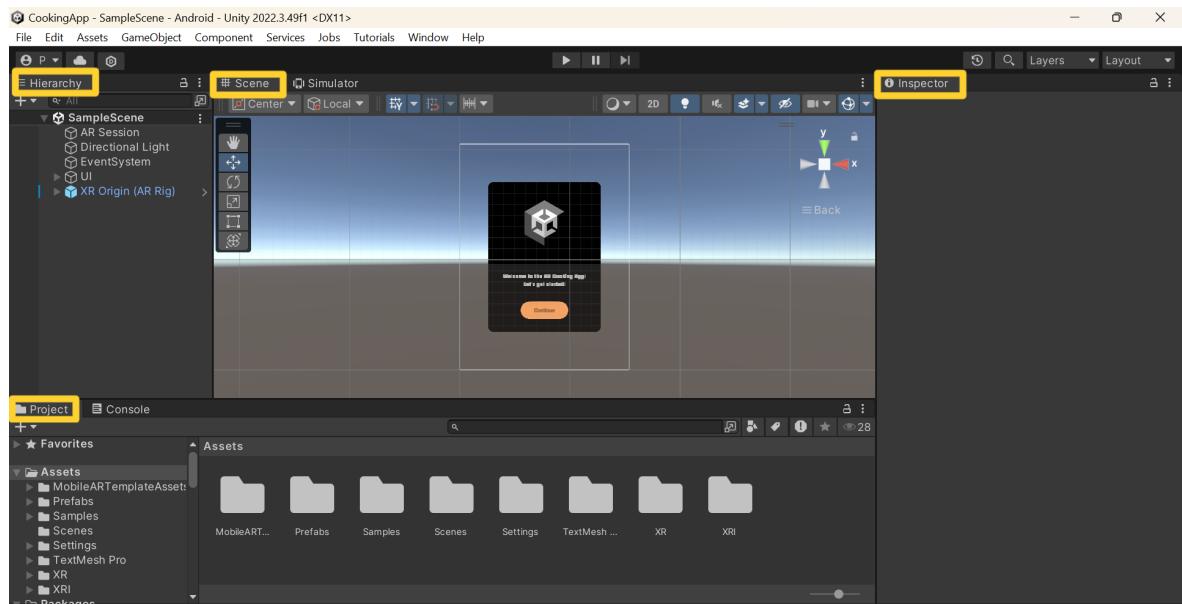


Figura 3.1: Vista del editor con los paneles **Project**, **Hierarchy**, **Scene** y **Inspector** resaltados en los recuadros amarillos.

3.2.2. GameObjects y Componentes

Todos los elementos presentes en la escena, como cubos, cámaras o personajes, son *GameObjects*. Por otro lado, un componente (*Component*) es todo aquel elemento que añade funcionalidades a los *GameObjects*, como colisiones o interactividad, y se editan desde el panel *Inspector*.

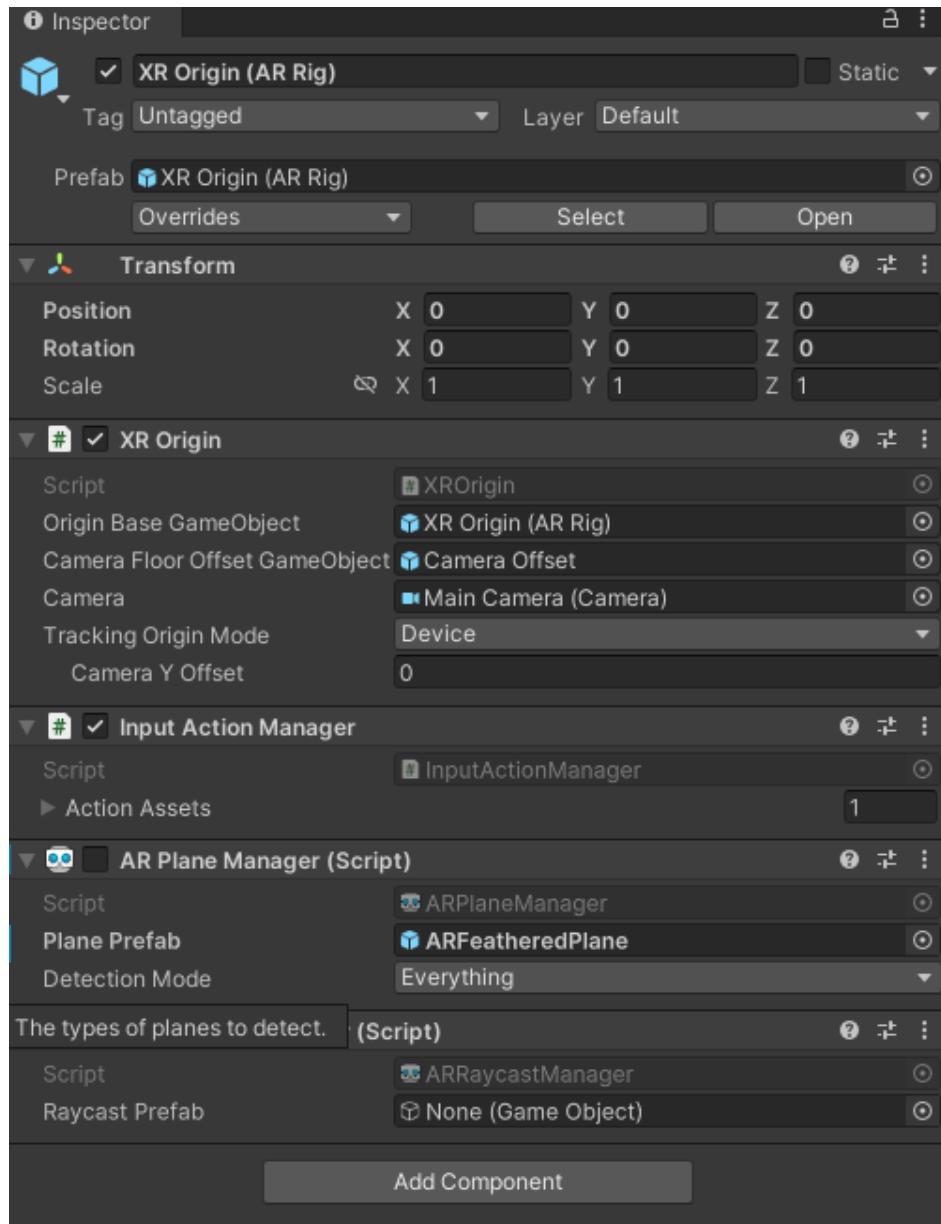


Figura 3.2: Panel *Inspector* del *GameObject* *XR Origin (AR Rig)*

3.2.3. Sistema de *Scripts*

Unity emplea **C#** como lenguaje de programación para definir el comportamiento de los objetos (*GameObjects*). Los *scripts* se adjuntan como componentes a los *GameObjects*, permitiendo definir las interacciones, movimientos o lógica del juego.

Flujo de Ejecución de los *Scripts*

Unity sigue un ciclo de vida predeterminado para ejecutar métodos dentro de un script, algunos de los más comunes son:

- **Start()**: se ejecuta cuando el objeto que tiene el script es activado.
- **Update()**: se ejecuta en cada frame del juego y se emplea para la lógica real, como la detección de entradas.

3.2.4. Plantilla AR Mobile

Al crear un proyecto nuevo en Unity de AR, se despliega una plantilla cuyos *GameObjects* y *Componentes* ya están configurados. La experiencia consiste en escanear una superficie plana con el dispositivo Android, en nuestro caso, la tablet Xiaomi Pad 6. Una vez es detectada la superficie, el usuario puede:

- Colocar objetos geométricos
- Modificar sus tamaños (agrandándolos o reduciéndolos).
- Cambiar su ubicación arrastrándolos con el dedo a la posición deseada.

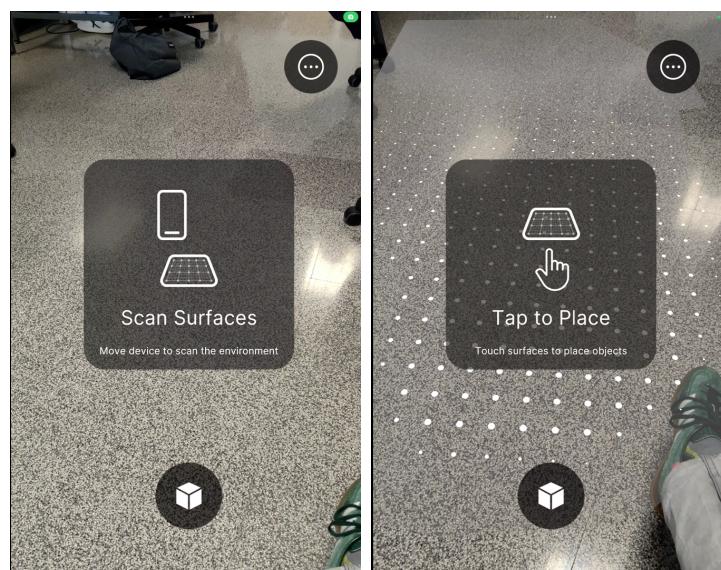


Figura 3.3: Proceso de escaneo de superficies (izquierda) y detección de superficies (derecha)

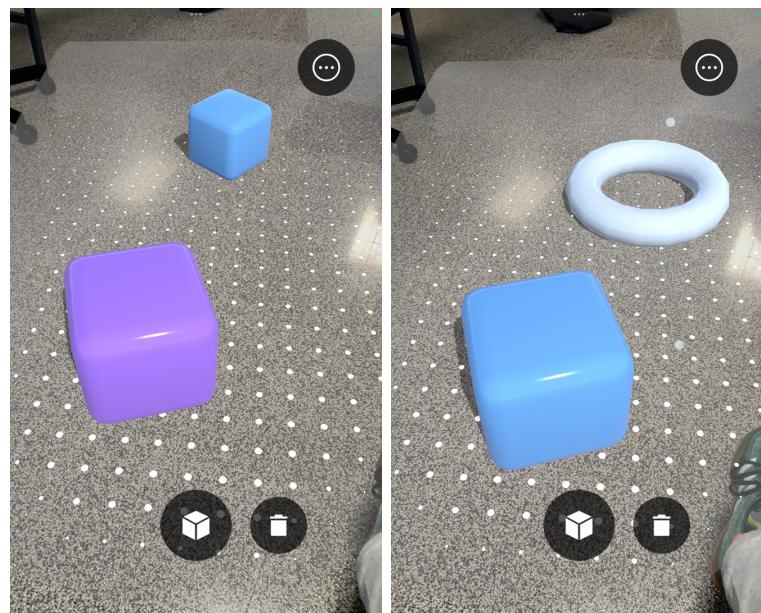


Figura 3.4: Objetos colocados sobre la superficie

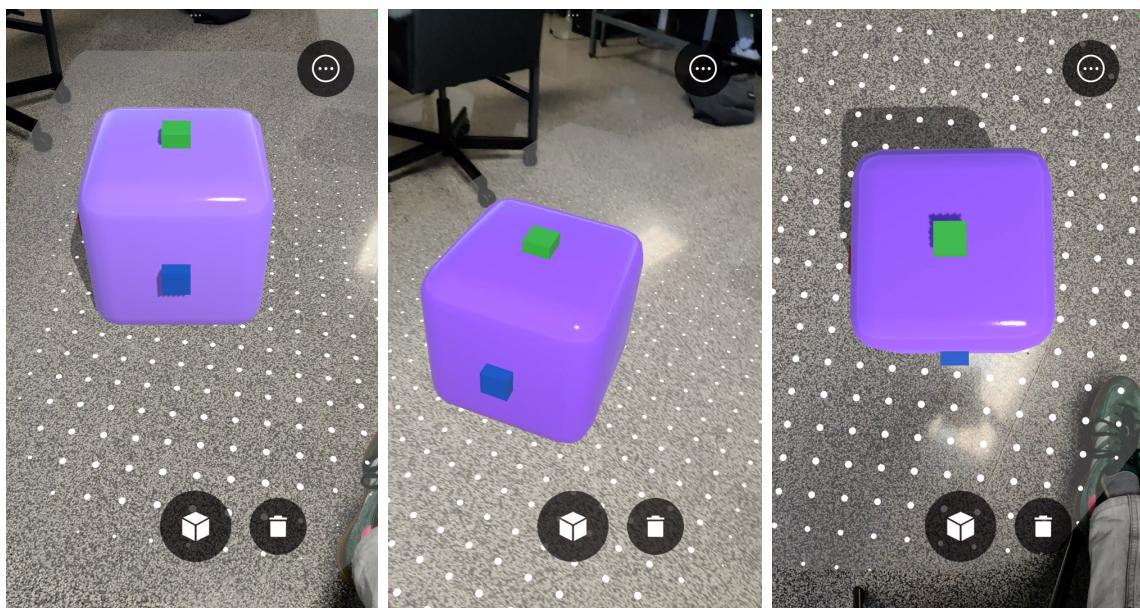


Figura 3.5: Perspectivas del cubo colocado en el plano

Se pueden consultar más imágenes de la interfaz de la plantilla en el [Anexo D](#).

3.3. Configuración inicial del proyecto

Partiendo de dicha plantilla, el primer paso fue explorar el funcionamiento de sus componentes y evaluar posibles limitaciones. Tras realizar varias pruebas, decidimos conservar únicamente la detección de planos, descartando el resto de funcionalidades no relevantes para la experiencia deseada.

Posteriormente, planteamos un boceto inicial que sirvió como referencia para definir la experiencia desde la perspectiva del usuario.



Figura 3.6: Primer boceto de la experiencia de AR desde la perspectiva del usuario

Se quiere que el usuario pueda escanear la encimera de la cocina y los fogones. La interfaz le permite escoger qué receta desea cocinar, mostrando el plato final en 3D sobre la encimera, sustituyendo las fotos tradicionales de los recetarios.

Una vez seleccionada la receta deseada, aparecen los ingredientes como elementos 3D sobre la superficie. El usuario puede seleccionar los que desea mantener o eliminar, haciendo que desaparezcan de la encimera.

A continuación, durante el proceso de cocina, se muestra un temporizador con el tiempo restante de cada paso, acompañado de instrucciones visuales como: mezclar ingredientes, encender o apagar el fuego, etc.

3.3.1. Detección de planos

El proceso de detección de planos proporcionado por [AR Foundation](#) permite identificar superficies planas en el entorno físico, como mesas, suelos o paredes. Este sistema se gestiona a través del *script ArPlaneManager*, que se asigna como componente al *GameObject: XR Origin*. Esta funcionalidad permite especificar qué tipo de superficie se desea detectar, como planos horizontales, verticales o ambos, configurándolo mediante el campo *Detection Mode*.

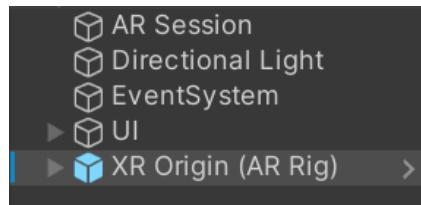


Figura 3.7: *GameObject XR Origin* en la jerarquía de la escena

Por ejemplo, en la configuración predeterminada del campo *Detection Mode* de la Figura 3.8, la opción *Everything* permite detectar tanto superficies horizontales como verticales.

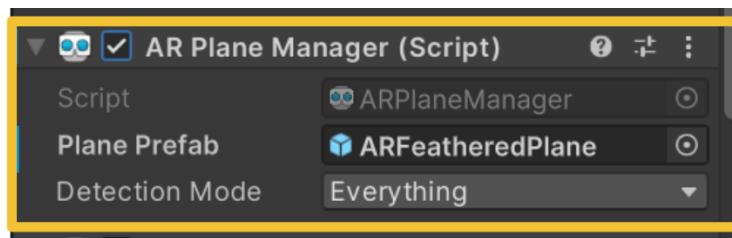


Figura 3.8: Componente *Ar Plane Manager (Script)* en el panel *Inspector*

El *script* permite asociar un *prefab* personalizado a cada plano detectado para representarlo en el espacio y poder visualizarlo (Listing F.1 en el Anexo F). A través de la propiedad *requestedDetectionMode* (Listing F.2 en el Anexo F), se configura el modo de detección de planos (vertical, horizontal o ambos) y el modo actual se puede consultar mediante la propiedad *currentDetectionMode* (Listing F.3 en el Anexo F).

También proporciona un evento *planesChanged* para reaccionar ante cambios en los planos detectados, incluye tres listas: planos añadidos, actualizados y eliminados. Esto nos permite gestionar dinámicamente los objetos detectados (Listing F.4 en el Anexo F).

3.4. Diseño y desarrollo del proyecto

Aunque el sistema es robusto, las limitaciones de espacios reducidos, como la encimera de la cocina, movimientos rápidos del dispositivo móvil o cambios bruscos en la iluminación, hacen que haya dificultades en la detección de planos y el resultado pueda ser poco fiable.

Ello nos impulsó a buscar una solución alternativa, en lugar de depender de superficies amplias y despejadas, realizar un seguimiento basado en imágenes, empleando dicha imagen como marcador donde se anclarán los objetos virtuales. De manera que al mover o rotar el marcador, el objeto virtual se sincroniza con esos movimientos. Mejorando la flexibilidad y precisión en escenarios dinámicos como una cocina.

Ahora, imaginemos el siguiente escenario: el usuario abre la aplicación y enfoca hacia el marcador. Sobre éste mismo, aparece el plato final en 3D y una vez es seleccionada la receta, se muestran los ingredientes en 3D, permitiendo elegir cuáles se desean conservar o descartar. Para eliminarlos, basta con arrastrarlos hacia un ícono de papelera. Con los ingredientes ya seleccionados, se comienza a cocinar, mostrando un temporizador con el tiempo restante para cada paso.

3.4.1. Detección del marcador

Para implementar la detección de imágenes 2D del mundo real para poder superponer contenido virtual sobre ellas, lo gestionamos a través del componente *AR Tracked Image Manager*, asignado al *GameObject*: *AR Session*. Este *GameObject* se encarga de controlar el ciclo de vida de la experiencia de la experiencia de AR, es decir, que gestiona las funcionalidades y la sincronización entre el mundo real y el virtual.

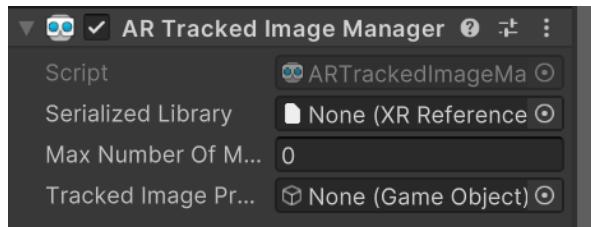


Figura 3.9: *AR Tracked Image Manager* en el panel *Inspector*

Configuración inicial de la biblioteca de imágenes

Para que el sistema pueda reconocer imágenes 2D específicas del mundo real, hay que realizar una configuración inicial mediante la creación de una *Reference Image Library* o biblioteca de imágenes de referencia que el sistema debe rastrear. Una vez creada la biblioteca de imágenes configurada, especificamos el número máximo de imágenes a rastrear y colocamos el objeto virtual que queremos colocar sobre la imagen del mundo real.

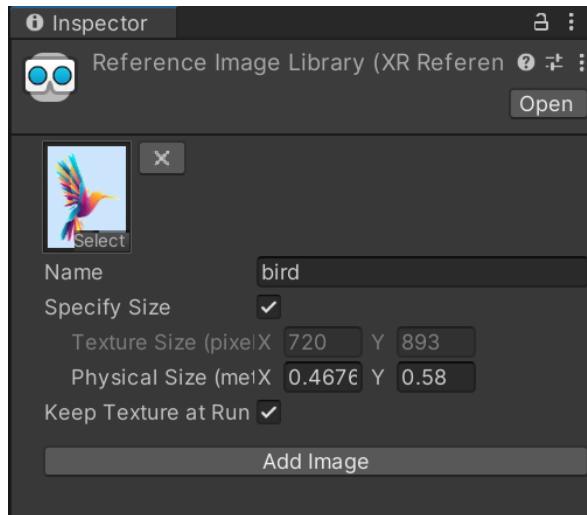


Figura 3.10: Biblioteca de imágenes empleada en el proyecto

Modificaciones para gestionar múltiples objetos por imágenes

Como se puede apreciar, presenta una limitación significativa: solo permite asociar un único objeto 3D a una imagen específica de la biblioteca. Esto implica, que para cada objeto virtual que se desee incluir en la experiencia, es necesario utilizar un marcador diferente. Esto reduce la flexibilidad del sistema, por ello se plantea un nuevo objetivo: emplear una única imagen para distintos objetos virtuales.

El componente *ARTrackedImageManager* gestiona las imágenes rastreadas. Originalmente se define un único *prefab* ([Listing F.5](#) en el [Anexo F](#)) que se instancia por cada imagen rastreada, que ha sido reemplazado por una lista de *prefabs m_TrackedImagePrefabs*, ([Listing F.6](#) en el [Anexo F](#)). De esta manera se permite instanciar múltiples objetos por cada imagen rastreada.

Originalmente se invocaba un evento *trackedImagesChanged* ([Listing F.7](#) en el [Anexo F](#)) para manejar imágenes añadidas, actualizadas o eliminadas en cada *frame*. A esto se ha agregado la lógica de instanciar todos los *prefabs* configurados en la lista de objetos virtuales (*m_TrackedImagePrefabs*), al detectar nuevas imágenes. Esto asegura que cada imagen rastreada pueda tener múltiples representaciones visuales asociadas ([Listing F.8](#) en el [Anexo F](#)).

De esta manera el componente *AR Tracked Image Manager* quedaría de la siguiente manera en el panel *Inspector*, donde se puede apreciar que se pueden asociar tantos objetos virtuales o *prefabs* a la imagen definida en la librería de referencia.

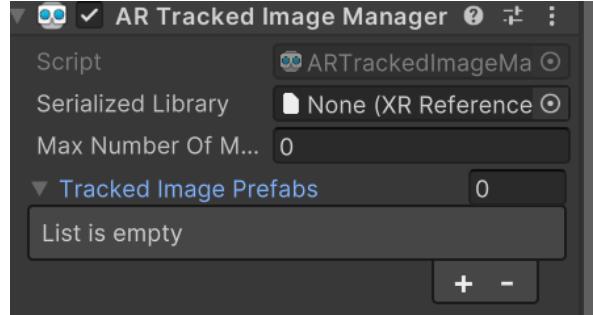


Figura 3.11: *AR Tracked Image Manager* modificado en el panel *Inspector*

En lugar de desactivar el objeto original al rastrear una imagen, según la configuración inicial, navegamos por la jerarquía de los objetos virtuales en la lista *Tracked Image Prefabs*. Activamos o desactivamos los necesarios, asegurando una mayor flexibilidad y control ([Listing F.9](#) en el [Anexo F](#)).

3.4.2. Eliminar ingredientes mediante Raycasting

En las recetas se hace una distinción entre los ingredientes obligatorios y los opcionales, por ejemplo, para los *pancakes* los ingredientes obligatorios podrían ser la harina y huevos; y los opcionales, el chocolate, sirope o fresas. De esta manera el usuario puede decidir si desea eliminar o mantener ingredientesopcionales, manteniendo la base de la receta intacta.

Los ingredientes se muestran como objetos 3D y para eliminarlos, el usuario debe mantener pulsado aquel que deseé eliminar y arrastrarlo hasta el icono de la papelera. De manera que una vez que son eliminados, dicho ingrediente desaparece de la pantalla del usuario. La técnica que permite poder hacer click en ellos, se llama *Raycasting* y consiste en lanzar un rayo desde la cámara hacia un punto en el espacio. Si este rayo choca con un objeto, se puede acceder a las componentes y propiedades, comprobando si se trata del objeto deseado.

Para ello se necesita un *Collider* que define el volumen del objeto para detectar las colisiones y un *script* adjunto al objeto que gestione la lógica de la interacción.



Figura 3.12: Componente Collider de un objeto 3D

Con este planteamiento, al eliminar el ingrediente, el objeto 3D desaparecía. De esta manera si el usuario no podía volver a introducir el ingrediente en la receta. Lo que nos llevó a re-diseñar la opción de arrastrar el ingrediente y sustituirlo por un botón de eliminar o volver a incluir el ingrediente en la receta.

3.4.3. Temporizador e información de los pasos de la receta

Una vez el usuario ya ha seleccionado la receta y los ingredientes, se muestra un temporizador en la parte superior con el tiempo estimado para completar el paso actual y se reinicia para cada paso. Se disponen de unos botones que el usuario puede realizar las siguientes acciones:

1. Pausar el cronómetro: Si necesita interrumpir el temporizador del paso actual.
2. Reanudar el cronómetro: Para continuar desde donde lo dejó.
3. Avanzar al siguiente paso: Si completa un paso antes del tiempo estimado.
4. Otro botón de pasar al siguiente paso: en caso de que el temporizador/cronómetro finalice antes de que lo haga el usuario, se quedará pausado hasta que el usuario le dé a otro botón de nuevo paso.

En la implementación de estos botones se emplean *listeners*, que permiten vincular directamente cada evento (pausar, reanudar el contador o pasar al siguiente paso) en el método `Start()` en lugar de realizar las verificaciones de variables booleanas en el `Update()`, ([Listing F.10](#) en el [Anexo F](#)).

Al emplear estos *listeners* en lugar de verificaciones de variables booleanas en el `Update()`, permite activar dichos *listeners* únicamente cuando ocurre el evento asociado. Al no realizar las comprobaciones en el `Update()`, se evitan innecesariamente que se realicen cientos de veces por segundo, ya que dicha función se ejecuta en cada *frame* del juego. Por otro lado, permite una estructura limpia y organizada, así como también, reducir errores que dependan de variables booleanas. De manera que si se desean añadir nuevos botones o funcionalidades, solo se requiere añadir un nuevo `onClick.AddListener()` en `Start()`, sin necesidad de modificar el método `Update()`, evitando sobrecarga de lógica.

3.4.4. Resultado final: una experiencia personalizada e intuitiva

Como resultado final se obtiene la experiencia de AR donde el usuario debe seleccionar la receta visualizando el plato finalizado. Posteriormente, selecciona los ingredientes que desea mantener y aquellos que desea eliminar. En el [Anexo G](#) se incluyen más imágenes de la experiencia que hemos diseñado. En la siguiente carpeta de Google Drive ([enlace](#)) se incluyen algunos vídeos de la interfaz de la experiencia diseñada en funcionamiento.

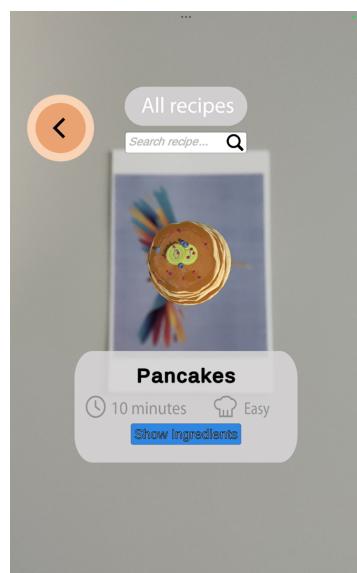


Figura 3.13: Interfaz donde se visualiza la receta

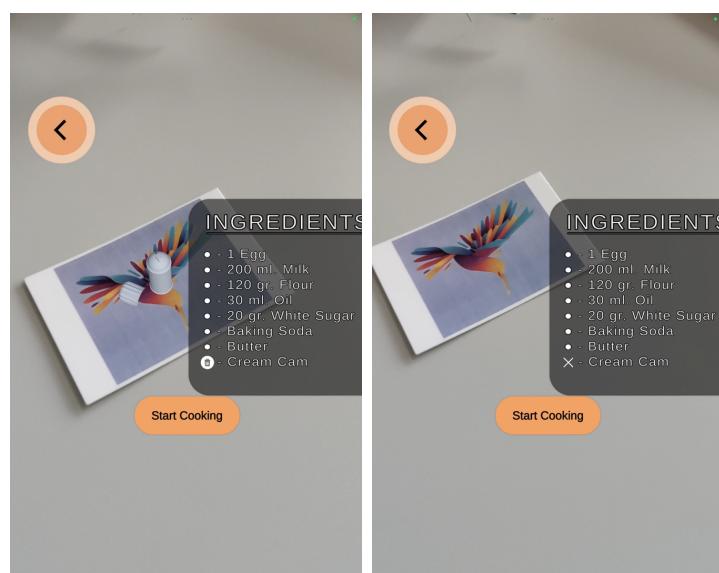


Figura 3.14: Interfaz donde aparece el ingrediente opcional (izquierdo) y una vez es eliminado de la receta (derecha)

Capítulo 4

Conclusiones y trabajo futuro

4.1. Conclusión personal

A lo largo de este TFG, he descubierto una nueva perspectiva del mundo culinario, sector que nunca antes había captado mi interés. Gracias a la oportunidad de poder haber trabajado con la Realidad Aumentada, he logrado desarrollar una apreciación por el arte de cocinar. Así como también, descubrir nuevas herramientas y aplicaciones, que van a ser fundamentales en mi futuro profesional, lo que me motiva a continuar explorando en este fascinante campo.

A nivel persona, este proyecto ha sido un proceso enriquecedor que me ha permitido rodearme de compañeros excepcionales y encontrarme un ambiente de trabajo muy positivo, una realidad que tristemente, no suele ocurrir y me siento muy afortunada de poder haberla vivido.

4.2. Futuras líneas del trabajo y mejoras

Los resultados obtenidos abren un abanico de posibilidades para futuras investigaciones en la aplicación de la Realidad Aumentada tanto en el ámbito culinario como en cualquier otro sector. A continuación se describen algunas de esas líneas de trabajo para futuras mejoras y expansiones de esta tecnología.

En primer lugar, expandir la máquina de estados del temporizador. Actualmente el sistema cuenta con una máquina de estados básica implementada con los estados *Idle*, *Running*, *StemComplete* y *RecipeComplete*, pero solo se utilizan los estados de pausa y reanudación, como si fuera una variable booleana. Se trataría de incluir *StemComplete* y *RecipeComplete* en el comportamiento actual para proporcionar una mayor flexibilidad y personalización del flujo de trabajo.

Otra línea de trabajo sería ajustar la cantidad de los ingredientes en función del número de comensales. Esto podría lograrse mediante el uso de archivos JSON, donde se almacenan los datos de los ingredientes de la receta junto con sus cantidades. De esta manera, el usuario selecciona el número de comensales y la cantidad de cada ingrediente se ajusta automáticamente mediante cálculos matemáticos al leer la información de este archivo JSON.

Asimismo, se propone la proyección de medidas en Realidad Aumentada para optimizar el uso de recipientes medidores. Mediante el uso de un recipiente o vaso concreto, proyectar una línea que indique el nivel de exacto al que debe llenarlo, mejorando la precisión en las mediciones.

Esta experiencia podría ser más interactiva mediante la integración de un asistente virtual con Inteligencia Artificial, al cual los usuarios puedan realizar consultas en tiempo real. Además, se podría incluir un sistema de lista de la compra inteligente, que permita visualizar dicha lista mientras visualiza la receta, los ingredientes o se encuentre cocinando; para poder añadir productos de manera casi automática.

Se puede ampliar sus usos a aspectos como la prevención del desperdicio de alimentos, esto podría lograrse si se pudieran reconocer los ingredientes al enfocar a la despensa o la nevera y la aplicación sugiera recetas personalizadas según los alimentos disponibles, incluso priorizando aquellos con fechas de vencimiento próximas. Esta funcionalidad combinada con las listas de compra, podrían llegar a mejorar la planificación y promueven un consumo responsable.

Por último, a largo plazo, se quiere que dispositivos que actualmente se encuentran en desarrollo, como las gafas Orion de Meta, permitan una interacción del usuario con este entorno digital fusionado con el mundo real de una forma mucho más cómoda. Sustituyendo la necesidad de dispositivos móviles y abriendo nuevas posibilidades.



Figura 4.1: Gafas Orion de Meta

Capítulo 5

Bibliografía

- [1] Graphics and Imaging Lab. Graphics and imaging lab. <https://graphics.unizar.es/>, 2025. Accedido el 21 de enero de 2025.
- [2] BSH. Bsh. <https://www.bsh-group.com/es/>, 2025. Accedido el 21 de enero de 2025.
- [3] IMASCONO. Imascono. <https://imascono.com/>, 2025. Accedido el 21 de enero de 2025.
- [4] Realidad Aumentada. Xataca. <https://www.xataka.com/basics/diferencias-entre-realidad-aumentada-realidad-virtual-y-realidad-mixta>, 2025. Accedido el 23 de enero de 2025.
- [5] Realidad Aumentada. Universitat jaume i. <https://www.youtube.com/watch?v=jOCWX9fCc0g>, 2025. Accedido el 23 de enero de 2025.
- [6] AccuVein. Accuvein. <https://www.accuvein.com/>, 2025. Accedido el 23 de enero de 2025.
- [7] AccuVein. Accuvein av500 one sheet. <https://www.accuvein.com/wp-content/uploads/2022/09/AccuVein-AV500-One-Sheet.pdf>, 2022. Accedido el 23 de enero de 2025.
- [8] IKEA Place. Ikea place. <https://www.ikea.com/es/es/customer-service/mobile-apps/>, 2025. Accedido el 23 de enero de 2025.
- [9] XatakaAndroid. Ikea place. <https://www.xatakandroid.com/aplicaciones-android/ikea-place-su-aplicacion-de-realidad-aumentada-para-decorar>, 2025. Accedido el 23 de enero de 2025.
- [10] Apple App Medidas. Apple app medidas. <https://apps.apple.com/es/app/medidas/id1383426740>, 2025. Accedido el 23 de enero de 2025.

- [11] Soporte Técnico de Apple. Utilizar la app medidas en el iphone, ipad o ipod touch. <https://support.apple.com/es-es/102468>, 2025. Accedido el 23 de enero de 2025.
- [12] Página web oficial de Chromville. Chromville. <https://chromville.com/>, 2025. Accedido el 23 de enero de 2025.
- [13] The New York Times. Kittch is a new culinary livestreaming platform. <https://www.nytimes.com/2022/03/14/dining/kittch-livestreaming.html>, 2025. Accedido el 23 de enero de 2025.
- [14] The Spoon Michael Wolf. Kittch partners with qualcomm for augmented reality-powered cooking. <https://thespoon.tech/kittch-partners-with-qualcomm-for-augmented-reality-powered-cooking/>, 2025. Accedido el 23 de enero de 2025.
- [15] Kittch. Kittch ar. <https://kittch.com/kittch/streams/kittch-ar>, 2025. Accedido el 23 de enero de 2025.
- [16] Qualcomm. qualcomm. <https://www.qualcomm.com/>, 2025. Accedido el 23 de enero de 2025.
- [17] Scarlett Evans IOT WORLD DAY. Qualcomm, kittch release ar glasses for cooking. <https://www.iotworldtoday.com/metaverse/qualcomm-kittch-release-ar-glasses-for-cooking#close-modal>, 2025. Accedido el 23 de enero de 2025.
- [18] Xataka José García. Nreal light. <https://www.xataka.com/realidad-virtual-aumentada/nreal-light-asi-funcionan-nuevas-gafas-realidad-aumentada-vodafone>, 2025. Accedido el 23 de enero de 2025.
- [19] Vodafone. Bringing augmented reality to the kitchen. <https://www.vodafone.com/news/technology/bringing-virtual-reality-kitchen>, 2025. Accedido el 23 de enero de 2025.
- [20] Vodafone (Deutschland). Giga ar app: A futuristic cooking experience. https://www.youtube.com/watch?v=u3tH3s2_mxg, 2025. Accedido el 23 de enero de 2025.

- [21] Vodafone (Deutschland). Giga ar: Virtual cooking with steffen henssler. https://www.youtube.com/watch?v=K_p6_XSnNnE, 2025. Accedido el 23 de enero de 2025.
- [22] Vodafone (Deutschland). The future of food — kitchen stories x vodafone. <https://www.youtube.com/watch?v=9C8N53L-g0U>, 2025. Accedido el 23 de enero de 2025.
- [23] Vodafone (Deutschland). Giga ar: Experience the future today with augmented reality. <https://www.youtube.com/watch?v=Wtisr0NTp1U>, 2025. Accedido el 23 de enero de 2025.
- [24] ITCO. Impacto de la realidad aumentada en la vida cotidiana. <https://itcoint.com/impacto-de-la-realidad-aumentada-en-la-vida-cotidiana/>, 2025. Accedido el 23 de enero de 2025.
- [25] Ray-Ban. Ray-ban meta smart glasses. <https://www.ray-ban.com/spain/ray-ban-meta-smart-glasses>, 2025. Accedido el 23 de enero de 2025.
- [26] Ray-Ban Meta. Smart glasses for living all in. <https://www.ray-ban.com/usa/discover-ray-ban-meta-smart-glasses/clp>, 2025. Accedido el 23 de enero de 2025.
- [27] Meta. Introducing orion, our first true augmented reality glasses. <https://about.fb.com/news/2024/09/introducing-orion-our-first-true-augmented-reality-glasses/>, 2025. Accedido el 23 de enero de 2025.
- [28] Meta. The future of wearable. <https://about.meta.com/realitylabs/orion/>, 2025. Accedido el 23 de enero de 2025.
- [29] Wikipedia. Unity (motor de videojuego). [https://es.wikipedia.org/wiki/Unity_\(motor_de_videojuego\)](https://es.wikipedia.org/wiki/Unity_(motor_de_videojuego)), 2025. Accedido el 23 de enero de 2025.
- [30] Unity. Unity. <https://unity.com/es>, 2025. Accedido el 23 de enero de 2025.
- [31] Estudio Formación. Unity. <https://www.estudioformacion.com/que-es-unity/>, 2025. Accedido el 21 de enero de 2025.
- [32] UnityDocumentation. Unity user manual. <https://docs.unity3d.com/Manual/index.html>, 2025. Accedido el 23 de enero de 2025.

- [33] MasterMoviles. Desarrollo de videojuegos con unity 5. <https://mastermoviles.gitbook.io/desarrollo-de-videojuegos-con-unity-5>, 2025. Accedido el 23 de enero de 2025.
- [34] AR Foundation. Documentación arcore. <https://developers.google.com/ar/develop/unity-arf/getting-started-ar-foundation?hl=es-419>, 2025. Accedido el 17 de enero de 2025.
- [35] AR Foundation. Unity manual. <https://docs.unity3d.com/Packages/com.unity.xr.arfoundation@6.0/manual/index.html>, 2025. Accedido el 17 de enero de 2025.
- [36] ChatGPT. Chatgpt. <https://chatgpt.com/>, 2025. Accedido el 23 de enero de 2025.

Lista de Figuras

1.1.	Diagrama de <i>Gantt</i> con las tareas realizadas a lo largo de los meses.	9
2.1.	Ejemplo de experiencia AR con Kittch	12
2.2.	Ejemplo de experiencia AR con Vodafone-Giga AR	13
3.1.	Vista del editor con los paneles Project , Hierarchy , Scene y Inspector resaltados en los recuadros amarillos.	16
3.2.	Panel <i>Inspector</i> del <i>GameObject XR Origin (AR Rig)</i>	17
3.3.	Proceso de escaneo de superficies (izquierda) y detección de superficies (derecha)	18
3.4.	Objetos colocados sobre la superficie	19
3.5.	Perspectivas del cubo colocado en el plano	19
3.6.	Primer boceto de la experiencia de AR desde la perspectiva del usuario	20
3.7.	<i>GameObject XR Origin</i> en la jerarquía de la escena	21
3.8.	Componente <i>Ar Plane Manager (Script)</i> en el panel <i>Inspector</i>	21
3.9.	<i>AR Tracked Image Manager</i> en el panel <i>Inspector</i>	22
3.10.	Biblioteca de imágenes empleada en el proyecto	23
3.11.	<i>AR Tracked Image Manager</i> modificado en el panel <i>Inspector</i>	24
3.12.	Componente <i>Collider</i> de un objeto 3D	25
3.13.	Interfaz donde se visualiza la receta	26
3.14.	Interfaz donde aparece el ingrediente opcional (izquierda) y una vez es eliminado de la receta (derecha)	26
4.1.	Gafas Orion de Meta	28
A.1.	Dispositivo de AccuVein en funcionamiento	39
A.2.	App IKEA Place en funcionamiento	40
A.3.	Logo de IKEA Place con un mueble colocado en el espacio del usuario .	40
A.4.	App Medidas del iPhone calculando las dimensiones de una maceta, un marco y la estatura de una persona	41
A.5.	Experiencia de Chromville: planta coloreada en 3D	42

A.6. Robot coloreado con Chromville en 3D	42
B.1. Dispositivo móvil con la aplicación conectado por cable a las gafas de AR	43
B.2. Usuario colocándose las gafas de AR	44
B.3. Vídeo con la receta que se puede agrandar y re-colocar en el espacio con gestos en el panel que se proporciona en la aplicación a través del dispositivo móvil	44
B.4. Control del vídeo (<i>pause</i> y <i>play</i>) a través de gestos con las manos	45
B.5. Ingredientes de la receta con la opción de añadirlos al carrito de compra de <i>Walmart</i>	45
B.6. Descripción escrita y el tiempo del paso actual	46
B.7. El usuario puede configurar un temporizador a través de la aplicación del dispositivo móvil	46
B.8. Temporizador configurado por el usuario	47
B.9. Kittch (plataforma), Spandragon spaces (desarrolladores) y TriggerXR (plataforma de XR)	47
C.1. Usuaria con las gafas en la cocina	49
C.2. Elección de la receta mediante la vista	50
C.3. Cantidad de ingredientes de la receta según el número de comensales .	50
C.4. Paso actual con la información necesaria	51
D.1. Pantalla inicial de la plantilla	53
D.2. Pantalla donde debe escanear la superficie	54
D.3. Pantalla donde debe escanear la superficie	54
D.4. Pantalla donde debe escoger el objeto a colocar	55
D.5. Objetos (dos cubos) colocados en la superficie	55
D.6. Objetos (cubo y aro) colocados en la superficie	56
D.7. Perspectivas del cubo colocado en el plano	56
G.1. Pantalla inicial de la app y principal (con el botón para mostrar las recetas)	67
G.2. Aviso de escanear tarjeta (izquierda) y plato de la receta sobre la tarjeta escaneada (derecha)	68
G.3. Ingrediente opcional mostrado (izquierda) y eliminado (derecha)	68
G.4. Temporizador con los botones <i>pause</i> , <i>play</i> y <i>nextStep</i>	69
G.5. Temporizador con la tarjeta donde se describe el paso actual	69

Lista de Fragmentos de Código

F.1.	<i>GameObject</i> m_PlanePrefab	59
F.2.	<i>requestedDetectionMode</i> - Plane Detection Mode	60
F.3.	<i>currentDetectionMode</i> - Plane Detection Mode	60
F.4.	Plane Detection Mode	61
F.5.	<i>m_TrackedImagePrefab</i> - AR Tracked Image Manager	62
F.6.	<i>m_TrackedImagePrefab</i> modificado - AR Tracked Image Manager	62
F.7.	<i>trackedImagesChanged</i> - AR Tracked Image Manager	63
F.8.	<i>trackedImagesChanged</i> modificado - AR Tracked Image Manager	64
F.9.	Gestión objetos virtuales en AR Tracked Image Manager	64
F.10.	Configuración de los botones dentro de Start()	65

Anexos

Anexos A

Aplicaciones de la Realidad Aumentada

A.1. Ámbito Sanitario - AccuVein

El 90 % de los pacientes se someten a terapia intravenosa en algún momento en sus estancias en el hospital. Por ello, AccuVein ha desarrollado una solución simple que optimiza la evaluación de venas, reducir complicaciones intravenosas y abordar las prioridades clínicas y operativas.

Este dispositivo emplea tecnología de infrarrojo cercano que se emite hacia la piel, penetrando en los tejidos superficiales del cuerpo. La hemoglobina en la sangre absorbe la luz infrarroja más que en los tejidos circundantes, creando un contraste natural entre las venas y los tejidos que las rodean.

Al apuntar el dispositivo a una distancia específica, el personal médico puede visualizar fácilmente las venas, tomar decisiones más precisas y mejorar la colocación de agujas y catéteres. Esto optimiza la atención al paciente y reduce errores.



Figura A.1: Dispositivo de AccuVein en funcionamiento

A.2. Sector comercial - IKEA Place

La aplicación móvil de IKEA place permite a los clientes colocar productos del catálogo en 3D a escala en sus propios espacios, mostrando con precisión el tamaño que ocuparían. Además, incluye un buscador visual que identifica muebles de la firma sueca vistos en otros lugares, proporcionando la información del producto al usuario.

Esta app emplea ARCore, analiza el entorno y calcula las dimensiones del espacio, mostrando los productos con la mayor precisión posible. Los usuarios pueden rotar y mover los muebles dentro de la app para probar las distintas ubicaciones, explorando detalles desde diferentes ángulos al desplazarse con su dispositivo móvil.

Este concepto mejora significativamente la experiencia de compra al eliminar la necesidad de emplear planos o mediciones manuales. Así, facilita las decisiones de compra y reduce errores en las dimensiones de los muebles adquiridos.



Figura A.2: App IKEA Place en funcionamiento



Figura A.3: Logo de IKEA Place con un mueble colocado en el espacio del usuario

A.3. Apple - App Medidas

La aplicación, desarrollada por Apple, permite calcular automáticamente las dimensiones de objetos reales, elementos rectangulares o la estatura de una persona, simplemente enfocando con el teléfono al objeto deseado. Gracias a su precisión, esta app es ideal para tareas rápidas de medición.



Figura A.4: App Medidas del iPhone calculando las dimensiones de una maceta, un marco y la estatura de una persona

A.4. Sector Educativo - Chromville

Chromville es una aplicación que fusiona la tecnología con el arte, fomentando el desarrollo de la creatividad de los niños y su motivación por el aprendizaje. Los usuarios colorean unas láminas y, posteriormente, al escanearlas con la cámara del dispositivo móvil, los objetos cobran vida en 3D. Esto permite convertir el aprendizaje en actividades dinámicas y tecnológicas.



Figura A.5: Experiencia de Chromville: planta coloreada en 3D



Figura A.6: Robot coloreado con Chromville en 3D

Anexos B

Experiencia de AR con Kittch

En este anexo se recogen las imágenes para poder entender cómo funciona la experiencia de AR que proporciona Kittch.

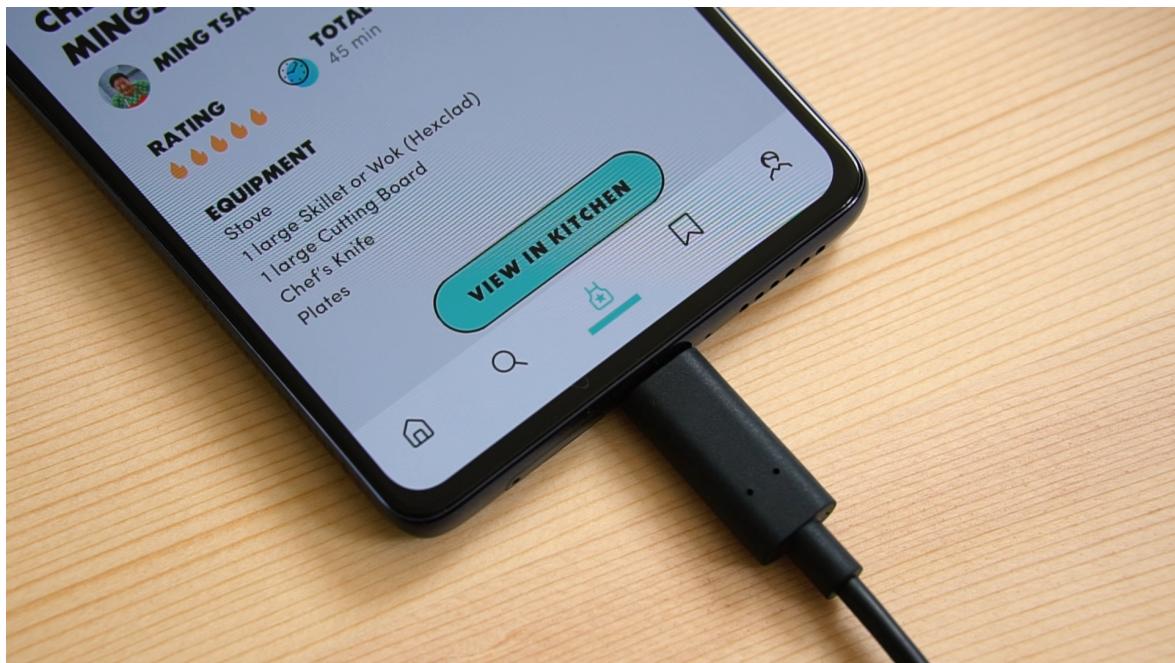


Figura B.1: Dispositivo móvil con la aplicación conectado por cable a las gafas de AR



Figura B.2: Usuario colocándose las gafas de AR

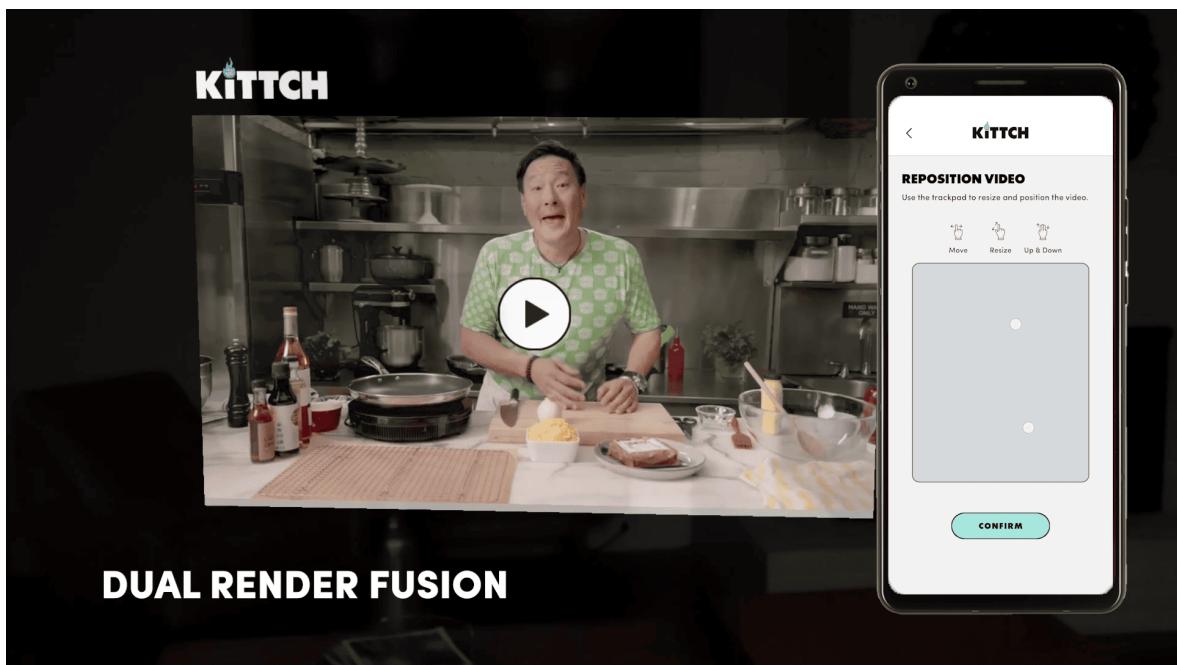


Figura B.3: Vídeo con la receta que se puede agrandar y re-colocar en el espacio con gestos en el panel que se proporciona en la aplicación a través del dispositivo móvil



Figura B.4: Control del vídeo (*pause* y *play*) a través de gestos con las manos

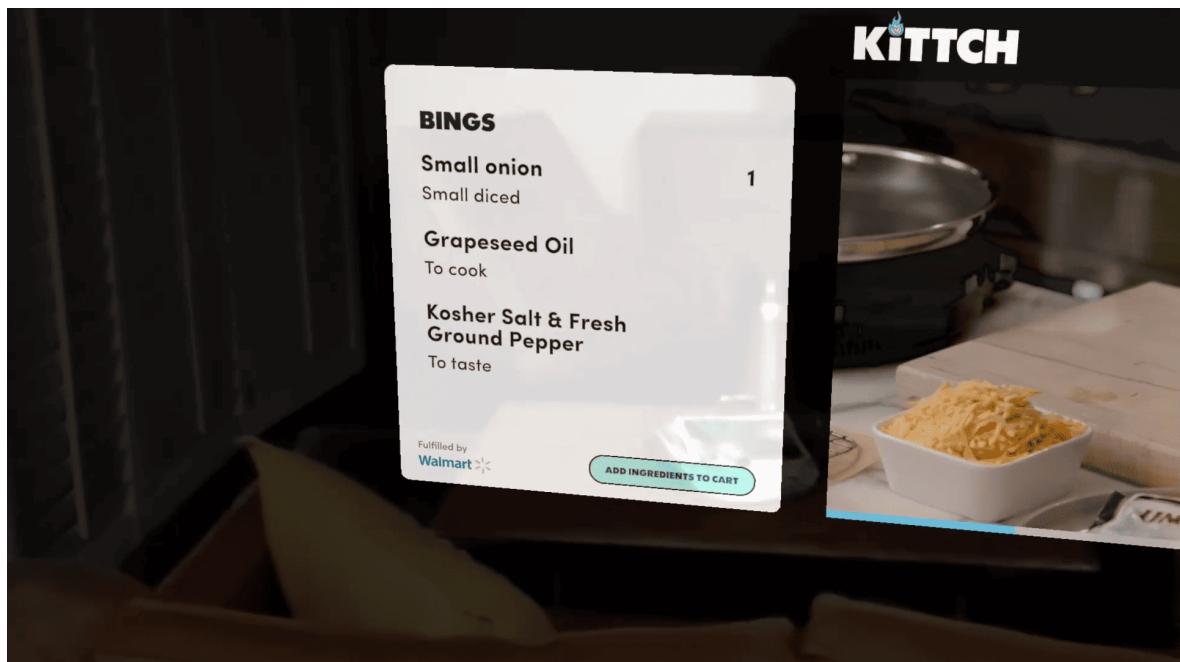


Figura B.5: Ingredientes de la receta con la opción de añadirlos al carrito de compra de *Walmart*

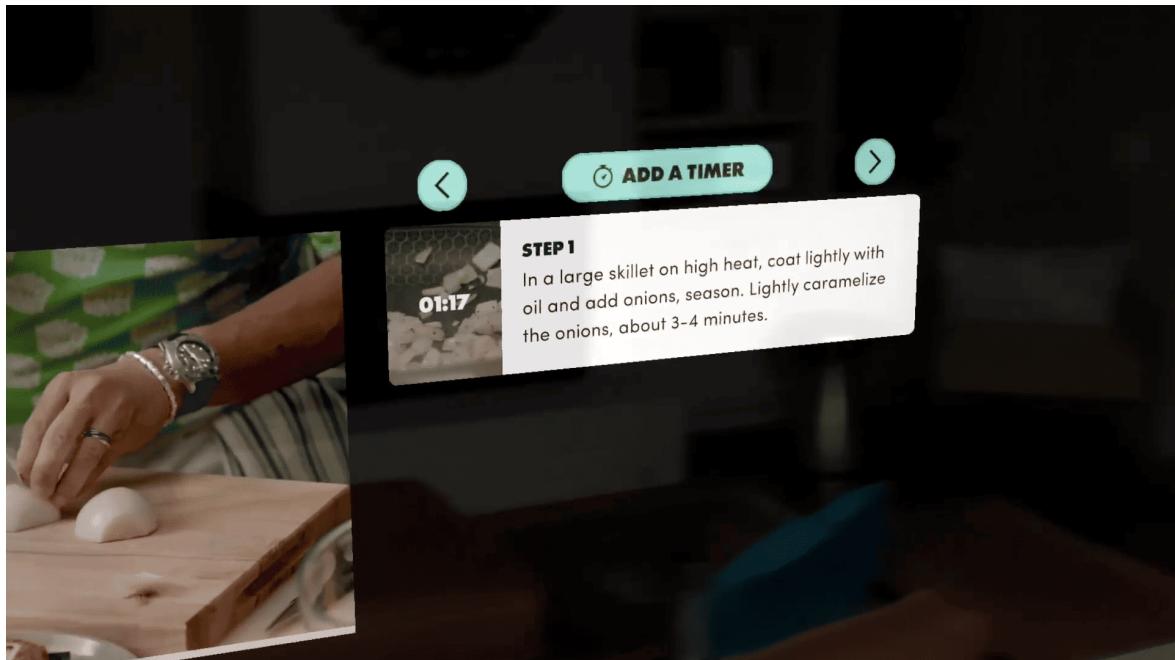


Figura B.6: Descripción escrita y el tiempo del paso actual

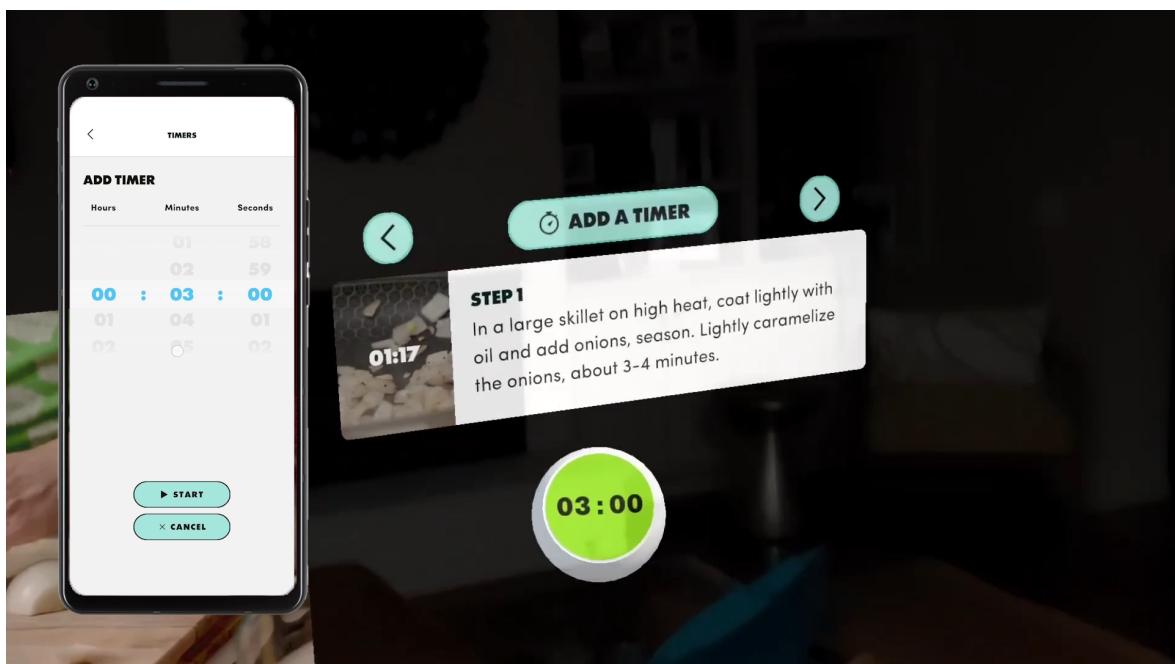
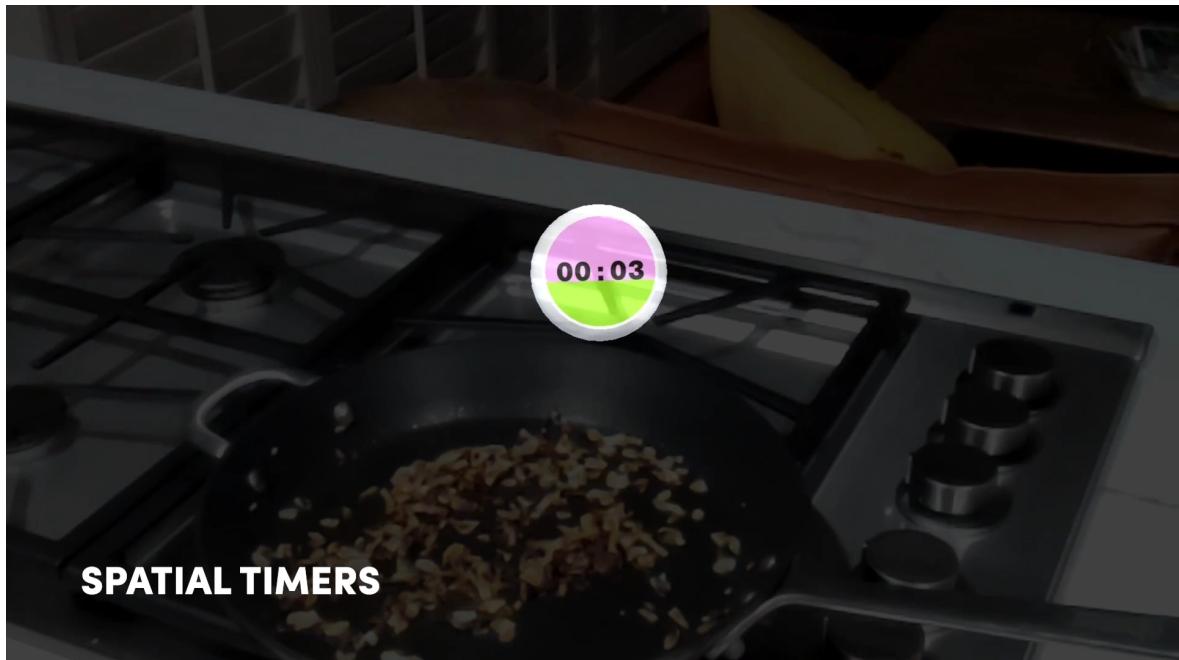


Figura B.7: El usuario puede configurar un temporizador a través de la aplicación del dispositivo móvil



SPATIAL TIMERS

Figura B.8: Temporizador configurado por el usuario

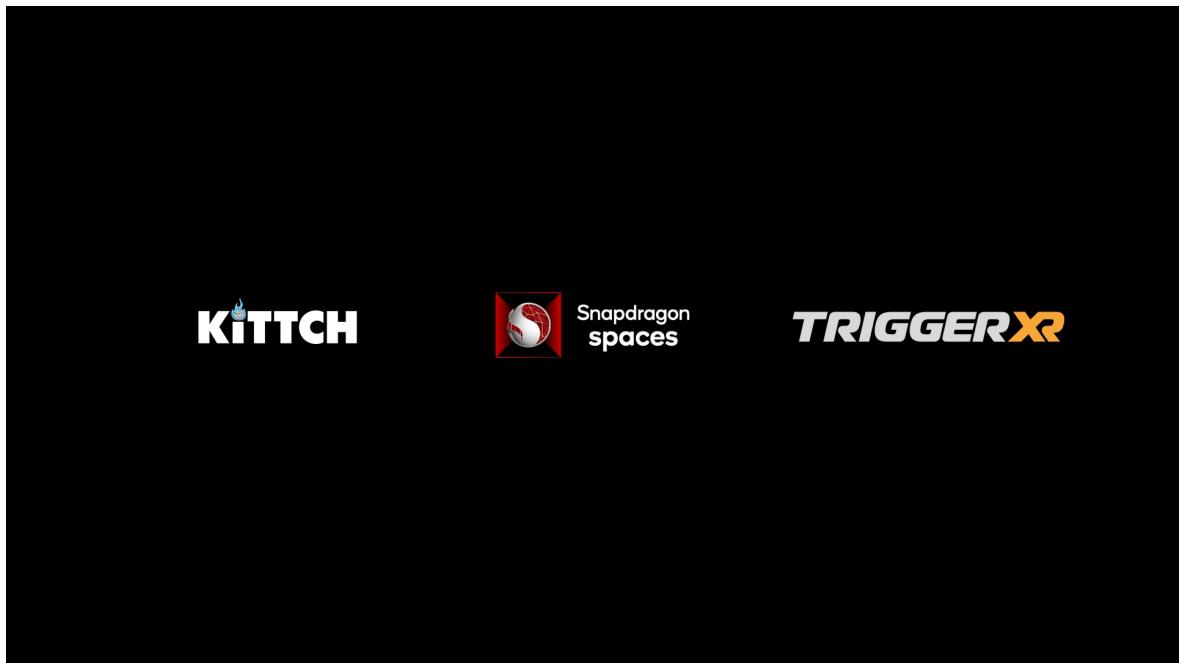


Figura B.9: Kittch (plataforma), Spandragon spaces (desarrolladores) y TriggerXR (plataforma de XR)

Anexos C

Experiencia de AR con Vodafone-Giga AR

En este anexo se recogen las imágenes para poder entender cómo funciona la experiencia de AR que proporciona Vodafone-Giga AR.



Figura C.1: Usuaria con las gafas en la cocina

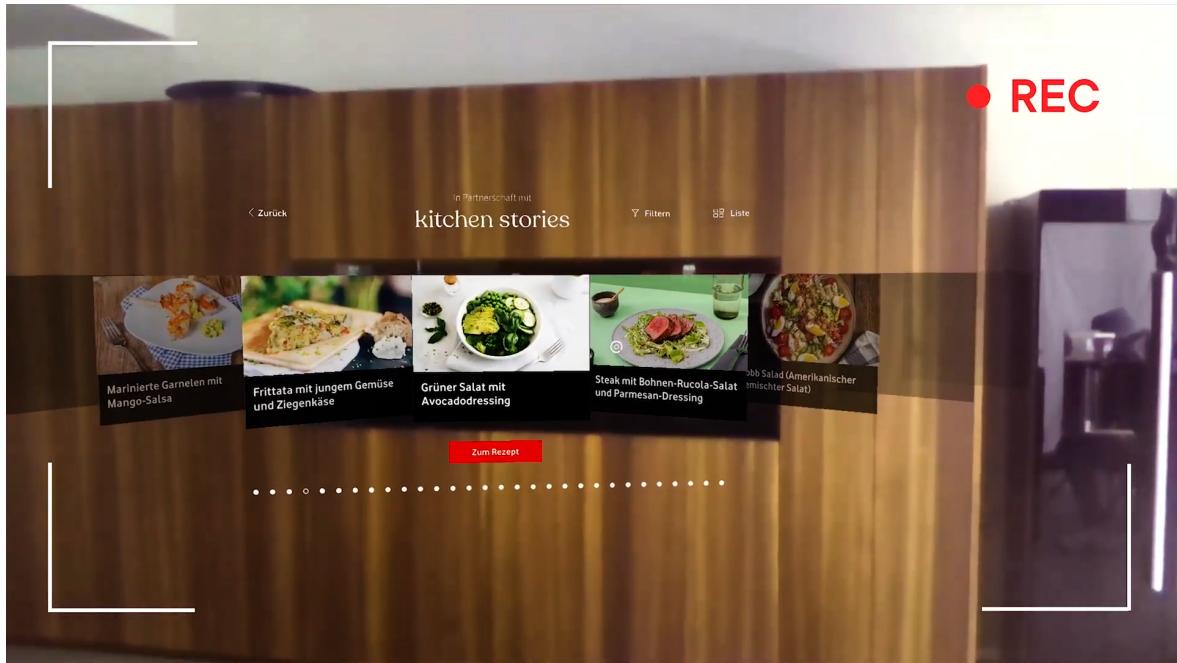


Figura C.2: Elección de la receta mediante la vista

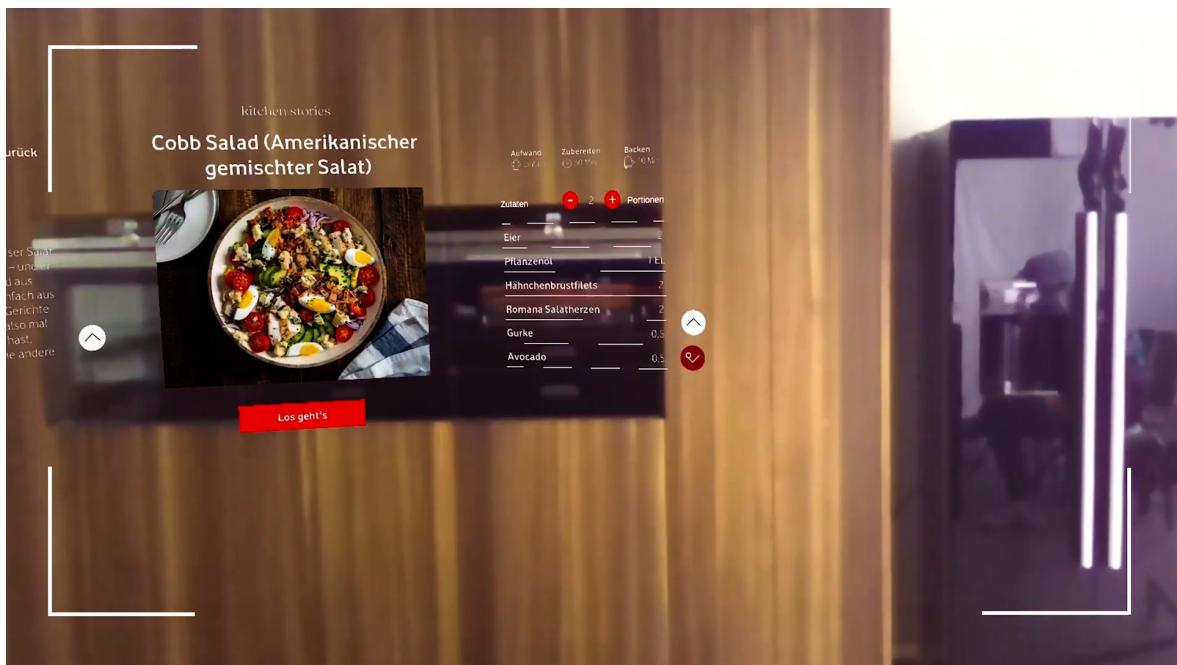


Figura C.3: Cantidad de ingredientes de la receta según el número de comensales

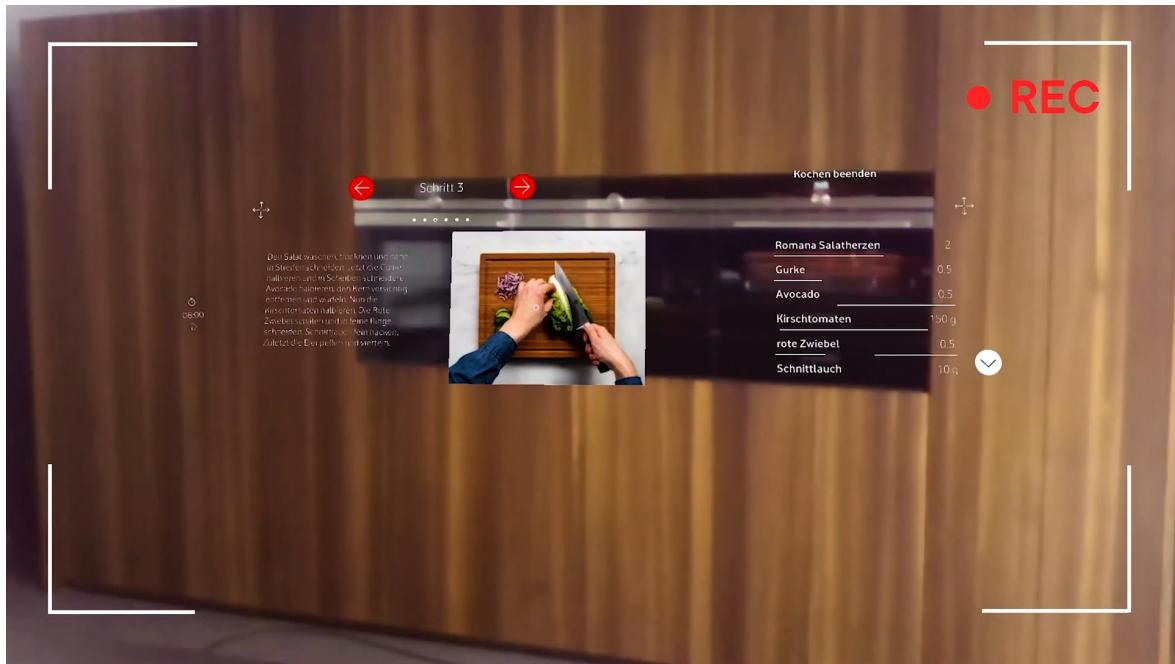


Figura C.4: Paso actual con la información necesaria

Anexos D

Plantilla AR Mobile de Unity

A continuación se muestran las imágenes donde se puede apreciar la interacción del usuario con la plantilla AR Mobile que dispone Unity. En primer lugar, una pantalla inicial que se muestra al iniciar la aplicación y donde el usuario comienza su experiencia.

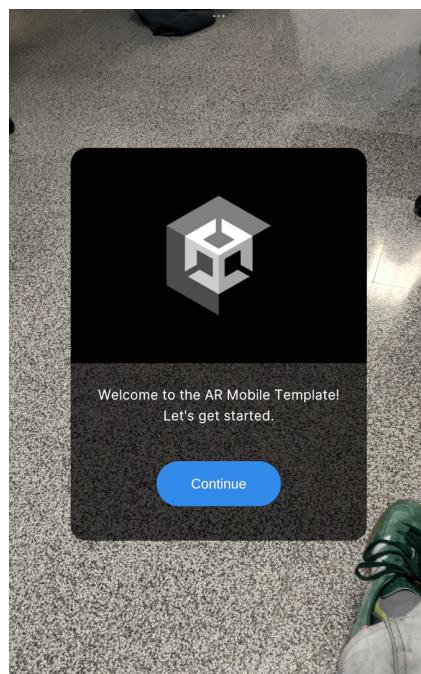


Figura D.1: Pantalla inicial de la plantilla

Posteriormente se muestra que se debe escanear la superficie. Para ello, debe mover el móvil alrededor del entorno.

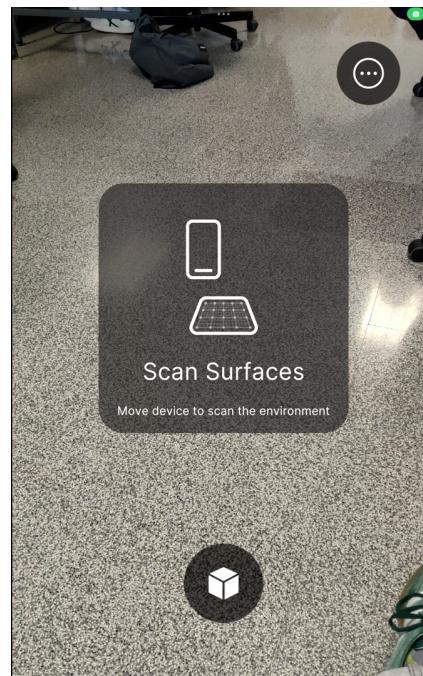


Figura D.2: Pantalla donde debe escanear la superficie

Una vez es detectada, se muestra sobre ella el plano digital donde detecta la superficie.

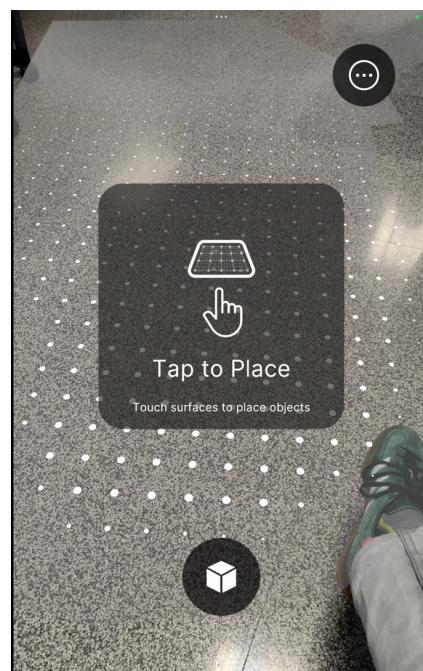


Figura D.3: Pantalla donde debe escanear la superficie

En ese plano digital puede colocarse cualquiera de los que aparecen en el menú de objetos, debe seleccionar el objeto que desea.

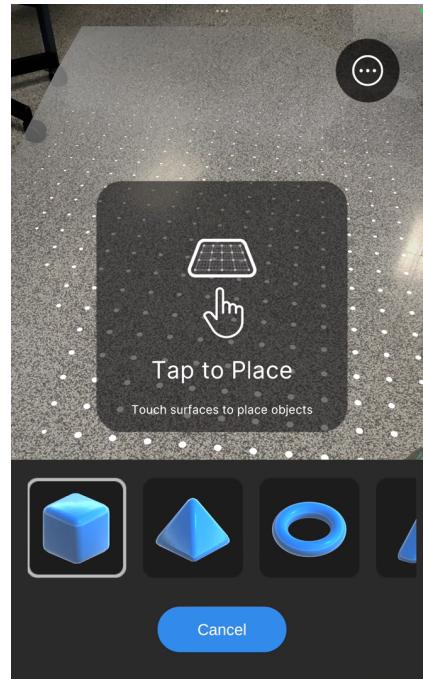


Figura D.4: Pantalla donde debe escoger el objeto a colocar



Figura D.5: Objetos (dos cubos) colocados en la superficie



Figura D.6: Objetos (cubo y aro) colocados en la superficie

A continuación se muestran varias imágenes donde se puede apreciar, que si el usuario ha colocado un objeto y se mueve, el objeto mantiene la posición en el plano como si estuviera en el mundo real.

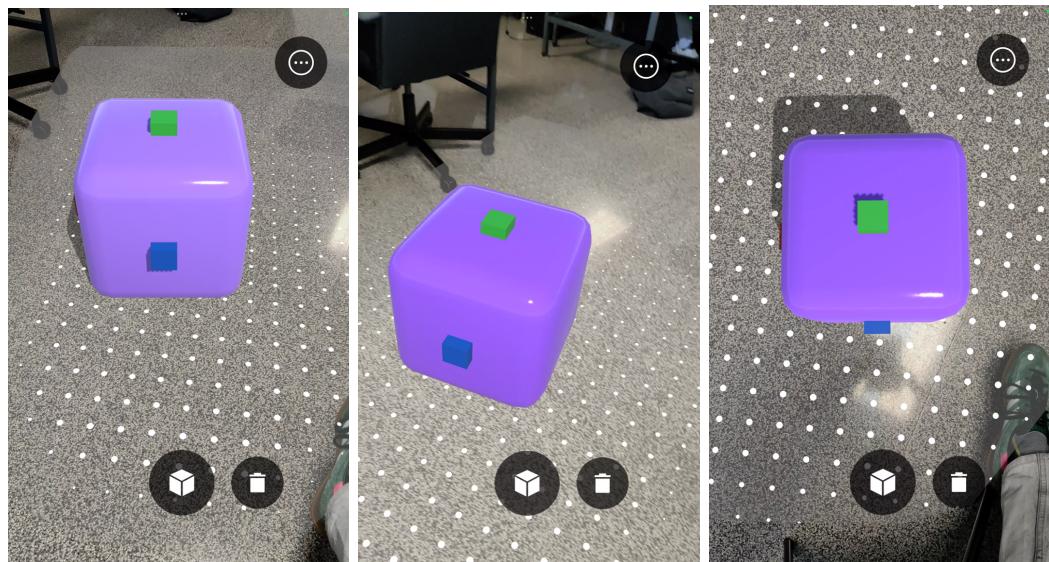


Figura D.7: Perspectivas del cubo colocado en el plano

Anexos E

AR Foundation

AR Foundation de Unity es un framework multiplataforma que te permite escribir experiencias de Realidad Aumentada una vez y luego compilarlas para dispositivos iOS o Android sin realizar cambios adicionales. El framework está disponible a través del paquete AR Foundation de Unity. [34]

En un proyecto de AR Foundation, eliges qué funciones de RA quieras habilitar agregando los componentes de administrador correspondientes a tu escena. Cuando creas y ejecutas tu aplicación en un dispositivo de AR, AR Foundation habilita estas funciones mediante el SDK de AR nativo de la plataforma, de modo que puedes crear una sola vez e implementar en las principales plataformas de AR del mundo. [35]

Anexos F

Scripts

En este anexo se encuentran los diferentes fragmentos de código a los que se han ido haciendo referencia a lo largo de este documento.

F.1. *ARPlaneManager.cs*

Listing F.1: *GameObject* m_PlanePrefab

```
1 [SerializeField]
2 [Tooltip(
3     "If not null, this prefab is instantiated for each " +
4     "detected plane." +
5     "If the prefab does not contain an AR Plane " +
6     "component , ARPlaneManager will add one.")]
7 GameObject m_PlanePrefab;
8
9 /// <summary>
10 /// Getter or setter for the Plane Prefab.
11 /// </summary>
12 public GameObject planePrefab
13 {
14     get => m_PlanePrefab;
15     set => m_PlanePrefab = value;
16 }
```

Listing F.2: *requestedDetectionMode* - Plane Detection Mode

```
1 [SerializeField, PlaneDetectionModeMask]
2 [Tooltip("The types of planes to detect.")]
3 PlaneDetectionMode m_DetectionMode = (PlaneDetectionMode)(-1);
4
5 /// <summary>
6 /// Get or set the requested plane detection mode.
7 /// </summary>
8 public PlaneDetectionMode requestedDetectionMode
9 {
10     get => subsystem?.requestedPlaneDetectionMode ??
11                     m_DetectionMode;
12     set
13     {
14         m_DetectionMode = value;
15         if (enabled && subsystem != null)
16         {
17             subsystem.requestedPlaneDetectionMode = value;
18         }
19     }
20 }
```

Listing F.3: *currentDetectionMode* - Plane Detection Mode

```
1 /// <summary>
2 /// Get the current plane detection mode in use by the
3 /// subsystem.
4 /// </summary>
5 public PlaneDetectionMode currentDetectionMode =>
6     subsystem?.currentPlaneDetectionMode ?? PlaneDetectionMode.None;
```

Listing F.4: Plane Detection Mode

```
1  /// <summary>
2  /// Invoked when planes have changed (been added, updated, or
3  /// removed).
4  /// </summary>
5  public event Action<ARPlanesChangedEventArgs> planesChanged;
6  /// <summary>
7  /// Invoked when the base class detects trackable changes.
8  /// </summary>
9  /// <param name="added">
10 ///     The list of added <see cref="ARPlane"/>s.</param>
11 /// <param name="updated">
12 ///     The list of updated <see cref="ARPlane"/>s.</param>
13 /// <param name="removed">
14 ///     The list of removed <see cref="ARPlane"/>s.</param>
15 protected override void OnTrackablesChanged(
16     List<ARPlane> added,
17     List<ARPlane> updated,
18     List<ARPlane> removed)
19 {
20     if (planesChanged != null)
21     {
22         using (new ScopedProfiler("OnPlanesChanged"))
23             planesChanged(
24                 new ARPlanesChangedEventArgs(
25                     added,
26                     updated,
27                     removed));
28     }
29 }
```

F.2. *ARTrackedImageManager.cs*

Listing F.5: *m_TrackedImagePrefab* - AR Tracked Image Manager

```
1 [SerializeField]
2 [Tooltip("If not null, instantiates this prefab for each"
3 + "detected image.")]
4 GameObject m_TrackedImagePrefab;
5
6 /// <summary>
7 /// If not null, instantiates this Prefab for each
8 /// detected image.
9 /// </summary>
10 /// <remarks>
11 /// The purpose of this property is to extend the functionality
12 /// of <see cref="ARTrackedImage"/>s. It is not the recommended
13 /// way to instantiate content associated with an
14 /// <see cref="ARTrackedImage"/>.
15 /// </remarks>
16 public GameObject trackedImagePrefab
17 {
18     get => m_TrackedImagePrefab;
19     set => m_TrackedImagePrefab = value;
20 }
21
22 /// <summary>
23 /// Get the Prefab that will be instantiated for
24 /// each <see cref="ARTrackedImage"/>.
25 /// </summary>
26 /// <returns>The Prefab that will be instantiated for each
27 /// <see cref="ARTrackedImage"/>. </returns>
28 protected override GameObject GetPrefab() =>
29     m_TrackedImagePrefab;
```

Listing F.6: *m_TrackedImagePrefab* modificado - AR Tracked Image Manager

```
1 [SerializeField]
2 [Tooltip("Prefabs para los objetos que seran instanciados por"
3 + "cada imagen rastreada.")]
4 List<GameObject> m_TrackedImagePrefabs = new List<GameObject>();
5
6 public List<GameObject> trackedImagePrefabs
7 {
8     get => m_TrackedImagePrefabs;
9     set => m_TrackedImagePrefabs = value;
10 }
```

Listing F.7: *trackedImagesChanged* - AR Tracked Image Manager

```
1  /// <summary>
2  /// Invoked once per frame with information about the
3  /// <see cref="ARTrackedImage"/>s that have changed
4  /// (that is, been added, updated, or removed). This happens
5  /// just before <see cref="ARTrackedImage"/>s are destroyed,
6  /// so you can set <c>ARTrackedImage.destroyOnRemoval</c> to
7  /// <c>false</c> from this event to suppress this behavior.
8  /// </summary>
9  public event Action<ARTrackedImagesChangedEventArgs>
10     trackedImagesChanged;
11  /// <summary>
12  /// Invokes the <see cref="trackedImagesChanged"/> event.
13  /// </summary>
14  /// <param name="added">A list of images added this
15  ///     frame.</param>
16  /// <param name="updated">A list of images updated
17  ///     this frame.</param>
18  /// <param name="removed">A list of images removed
19  ///     this frame.</param>
20 protected override void OnTrackablesChanged(
21     List<ARTrackedImage> added,
22     List<ARTrackedImage> updated,
23     List<ARTrackedImage> removed)
24 {
25     if (trackedImagesChanged != null)
26     {
27         using (new ScopedProfiler("OnTrackedImagesChanged"))
28             trackedImagesChanged(
29                 new ARTrackedImagesChangedEventArgs(
30                     added,
31                     updated,
32                     removed));
33     }
34 }
```

Listing F.8: *trackedImagesChanged* modificado - AR Tracked Image Manager

```
1 public event Action<ARTrackedImagesChangedEventArgs>
2     trackedImagesChanged;
3
4 protected override void OnTrackablesChanged(
5     List<ARTrackedImage> added,
6     List<ARTrackedImage> updated,
7     List<ARTrackedImage> removed)
8 {
9     if (trackedImagesChanged != null)
10    {
11        using (new ScopedProfiler("OnTrackedImagesChanged"))
12            trackedImagesChanged(
13                new ARTrackedImagesChangedEventArgs(
14                    added,
15                    updated,
16                    removed));
17    }
18
19    foreach (var trackedImage in added)
20    {
21        foreach (var prefab in m_TrackedImagePrefabs)
22        {
23            if (prefab != null)
24            {
25                var instantiatedObject = Instantiate(prefab,
26                                            trackedImage.transform);
27                instantiatedObject.transform.localPosition =
28                                Vector3.zero;
29                instantiatedObject.transform.localRotation =
30                                Quaternion.identity;
31            }
32        }
33    }
34 }
```

Listing F.9: Gestión objetos virtuales en AR Tracked Image Manager

```
1 GameObject trackables = GameObject.Find("Trackables");
2 trackables.transform.GetChild(0).GetChild(0).gameObject
3     .SetActive(aRScriptsControl.showPancake); // Pancakes
```

F.3. *Timer.cs*

Listing F.10: Configuración de los botones dentro de Start()

```
1 pauseButton.onClick.AddListener(pauseButtonPress);  
2 playButton.onClick.AddListener(playButtonPress);  
3 nextStepButton.onClick.AddListener(nextStepButtonPress);
```


Anexos G

Interfaz de la prueba de concepto

A continuación, se presentan las diferentes pantallas de la interfaz desarrollada. Y en el siguiente enlace a una carpeta de Google Drive se pueden consultar varios vídeos que muestran su funcionamiento.

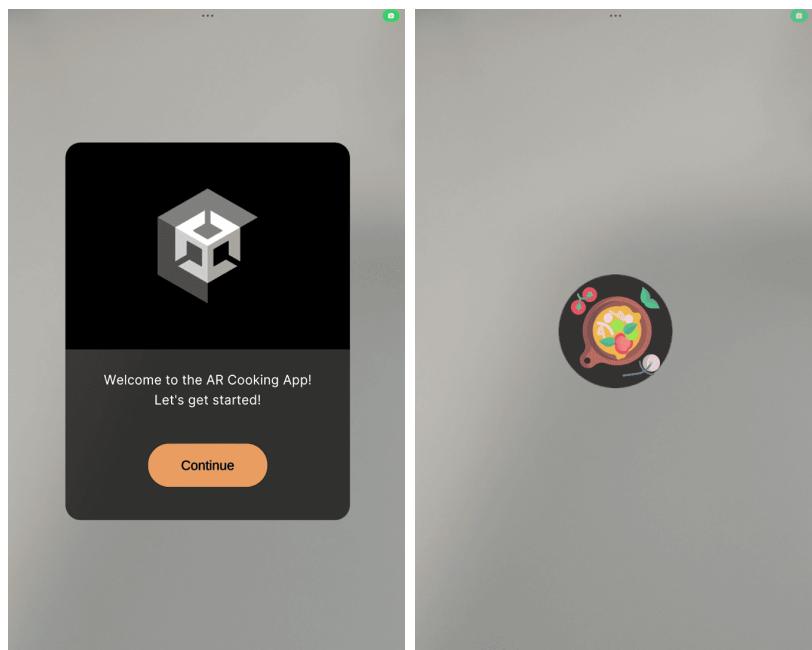


Figura G.1: Pantalla inicial de la app y principal (con el botón para mostrar las recetas)

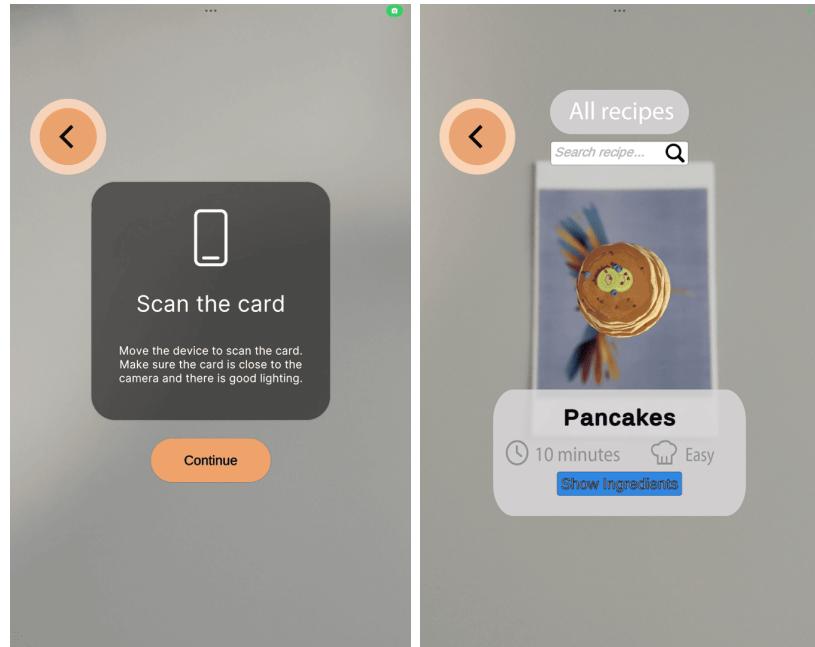


Figura G.2: Aviso de escanear tarjeta (izquierda) y plato de la receta sobre la tarjeta escaneada (derecha)

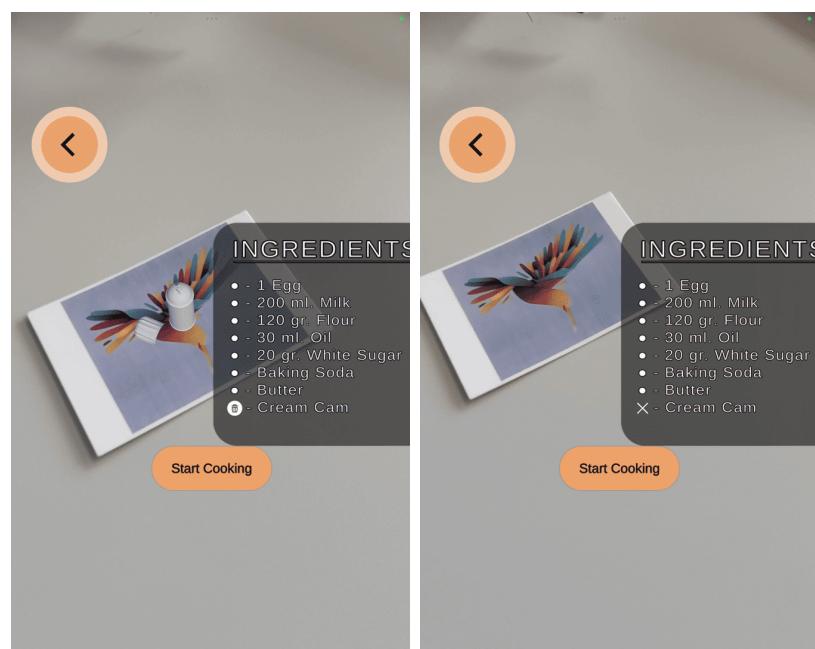


Figura G.3: Ingrediente opcional mostrado (izquierda) y eliminado (derecha)

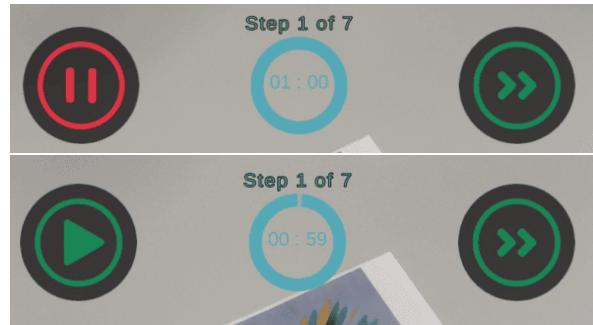


Figura G.4: Temporizador con los botones *pause*, *play* y *nextStep*

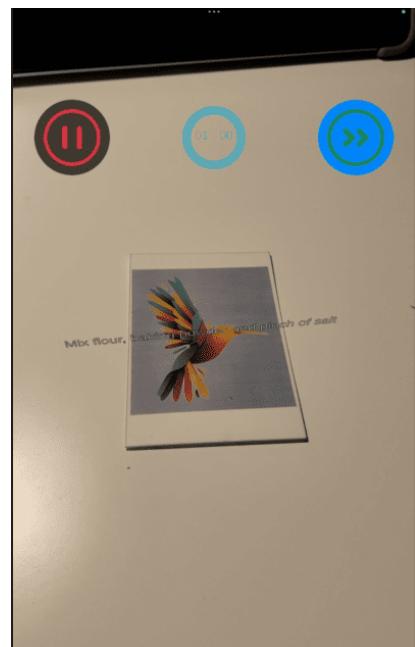


Figura G.5: Temporizador con la tarjeta donde se describe el paso actual