

**UNIVERSIDAD TECNOLÓGICA Y POLITÉCNICA DE COYUCA  
DE BENITEZ**

**ASIGNATURA: APLICACIONES PARA REALIDAD AUMENTADA.**

**CUATRIMESTRE: V      UNIDAD: III**

TEMA: DESPLIEGUE Y PUBLICACION DE APLICACIONES

DOCENTE: ING. GEOVAMY PIZA RODRIGUEZ.

ALUMNA: ADRIANA DE LOS SANTOS LAYNA

CARRERA: INGENIERÍA EN TECNOLOGÍAS DE LA INFORMACIÓN E INNOVACIÓN  
DIGITAL. (LINEA)

FECHA: 20 ABRIL 2026.

## Índice

1. Documento del Proyecto
  - 1.1 Objetivo del Proyecto
  - 1.2 Justificación
  - 1.3 Alcance del Proyecto
  - 1.4 Arquitectura del Diseño
2. Escenas y Secuencias de la Aplicación
3. Actividad 1 - Integración de objetos y modelos digitales
4. Actividad 2 - Interacción del usuario en realidad aumentada
5. Actividad 3 - Publicación de aplicaciones
6. Reporte Técnico del Proceso de Implementación
7. Conclusión
8. Bibliografía

## Documento del Proyecto

### Objetivo del Proyecto

---

Desarrollar una aplicación móvil de Realidad Aumentada (RA) que permita a los estudiantes de la Universidad Tecnológica del Pacífico explorar e interactuar con modelos 3D de elementos arquitectónicos y científicos mediante sus dispositivos móviles, mejorando la experiencia de aprendizaje en el campus universitario.

### Justificación

---

Problemática a resolver: Los estudiantes tienen dificultades para comprender conceptos abstractos en materias como arquitectura, biología y química, que requieren la visualización tridimensional de estructuras complejas. Los materiales didácticos tradicionales (libros, diapositivas) no ofrecen la interactividad necesaria.

Audiencia: Estudiantes y profesores de la Universidad Tecnológica del Pacífico, particularmente de las carreras de Ingeniería, Arquitectura y Ciencias.

Justificación: La Realidad Aumentada permite superponer objetos digitales 3D sobre el mundo real, brindando una experiencia inmersiva que facilita la comprensión de conceptos complejos. La implementación de esta tecnología en el ámbito educativo ha demostrado incrementar la retención de información hasta en un 40% (según estudios de la Universidad de Maryland, 2018).

### Alcance del Proyecto

---

- Aplicación móvil funcional para Android (compatible con ARCore).
- 5 escenas de RA con modelos 3D interactivos.
- Elementos multimedia integrados: video, audio, imágenes 2D y modelos 3D.
- Interfaz con estructura completa: splash, pantalla de inicio, menú, acerca de y ayuda.
- Marcadores impresos para activar las experiencias de RA.
- Publicación en formato APK para distribución interna.

### Arquitectura del Diseño

---

#### Motor de Desarrollo

Unity 2022 LTS con AR Foundation - Framework multiplataforma para RA con soporte para ARCore (Android) y ARKit (iOS).

#### SDK de RA

Vuforia Engine 10.x - SDK para reconocimiento de imágenes/marcadores con tracking robusto y base de datos de targets en la nube.

## **Modelos 3D**

Blender 3.6 + Sketchfab - Modelado y texturizado de objetos 3D. Exportación en formato .fbx y .glb para Unity.

## Escenas y Secuencias de la Aplicación

### Estructura de Pantallas

#### Splash Screen

Duración: 3 segundos. Logo de la UTP con animación fade-in. Nombre de la app 'UTP AR Explorer'. Barra de carga con progreso visual.

#### Pantalla de Inicio

Bienvenida al usuario, botón 'Escanear Marcador' (acceso a cámara AR), acceso rápido a las 5 escenas, indicador de conexión.

#### Menú Principal

Opciones: Inicio, Escanear (Cámara AR), Catálogo de Modelos, Historial, Acerca de, Ayuda. Navegación tipo bottom tab bar.

#### Acerca de

Información del proyecto, versión de la app, créditos del equipo, logo de la universidad, enlace institucional.

#### Ayuda

Tutorial interactivo de uso, instrucciones para escanear marcadores, preguntas frecuentes (FAQ), solución de problemas.

### 5 Escenas de Realidad Aumentada

#### Escena 1: Modelo Arquitectónico - Edificio Principal UTP

Modelo 3D a escala del edificio principal. El usuario puede rotar, escalar y explorar con gestos táctiles. Incluye etiquetas informativas.

Multim

#### Escena 2: Sistema Solar Interactivo

Modelo 3D de los planetas con órbitas animadas. Al tocar cada planeta se muestra información, imagen en alta resolución y un video corto.

Multim

#### Escena 3: Célula Humana

Modelo 3D detallado de una célula eucariota. Se pueden separar los organelos y ver información individual de cada uno.

Multim

#### Escena 4: Motor de Combustión Interna

Motor 3D animado que muestra los 4 tiempos del ciclo Otto. Animación paso a paso con narración en audio.

Multim

#### Escena 5: Circuito Electrónico Básico

Representación 3D de un circuito con LED, resistencia, capacitor y fuente. Animación del flujo de corriente eléctrica.



## Tema 1: Integración de objetos y modelos digitales

### Actividad 1 - Proceso de integración de elementos digitales

El proceso de integración de modelos 2D, 3D, audio y video en la aplicación de RA sigue una secuencia específica para cada tipo de elemento:

#### Modelos 3D

- 1. Modelado:** Se crean o descargan modelos 3D en Blender. Se optimiza la geometría (reducción de polígonos) para rendimiento móvil. Máximo 50,000 polígonos por modelo.
- 2. Texturizado:** Se aplican texturas UV mapeadas. Se comprimen a máximo 1024x1024 px en PNG/JPG. Se configuran materiales con shaders optimizados para móvil.
- 3. Exportación:** Se exporta en formato .FBX (para Unity) o .GLB/.GLTF (estándar web). Se verifican escala, rotación y punto de pivote.
- 4. Importación a Unity:** Se importa al proyecto y se configura en la escena AR. Se ajustan posición (x, y, z), tamaño (escala 0.1 a 1.0) y orientación (rotación en 3 ejes).

#### Imágenes 2D

- **Formatos:** PNG (con transparencia) o JPG (sin transparencia).
- **Resolución:** Adaptada al dispositivo (720p a 1080p para pantallas móviles).
- **Posición:** Se colocan como Sprites en un Canvas dentro del espacio AR, ancladas al modelo 3D.
- **Orientación:** Configuradas con "billboard" (siempre mirando a la cámara) para legibilidad.

#### Audio

- **Formato:** MP3 (comprimido) u OGG (mejor calidad/peso para Unity).
- **Tipos:** Narración informativa (mono, 128kbps), efectos de sonido (estéreo, 192kbps), música de ambiente (estéreo, 128kbps).
- **Integración:** Se adjunta un AudioSource al objeto 3D. Se configura audio espacial 3D. Se activa por trigger (al detectar marcador o al tocar objeto).

#### Video

- **Formato:** MP4 (H.264) para máxima compatibilidad.
- **Resolución:** 720p máximo para optimizar rendimiento.
- **Integración:** Se utiliza VideoPlayer de Unity sobre una superficie plana (Quad) en el espacio AR.

#### Configuración según el dispositivo

Parámetro	Smartphone	Tablet
Escala base del modelo	0.3 - 0.5	0.5 - 0.8
Distancia de tracking	30 - 100 cm	40 - 150 cm
Max polígonos	50,000	100,000
Resolución texturas	512x512	1024x1024
FPS objetivo	30 fps	30-60 fps

## Tema 2: Interacción del usuario en realidad aumentada

### Actividad 2 - Proceso de interacción con objetos digitales

#### Tipos de Interacción

##### Posición

El objeto se posiciona en el espacio 3D real usando coordenadas relativas al marcador detectado o a un plano horizontal/vertical. Se usa ARRaycastManager para detectar superficies y anclar objetos.

##### Gestos (Gestures)

Tap: Seleccionar objeto / mostrar info. Drag: Mover objeto en el plano. Pinch: Escalar (zoom in/out). Rotate: Dos dedos para rotar.

##### Superficie

Detección de planos horizontales (mesas, pisos) y verticales (paredes) usando ARPlaneManager. Los objetos se anclan a superficies detectadas manteniendo su posición.

##### Localización

Uso de GPS + brújula para experiencias de RA geolocalizadas. Activación de contenido AR según la ubicación física del usuario en el campus.

#### Instrucciones y Sintaxis en Programación de RA (Unity + C#)

Detección de planos y colocación de objetos:

```
// ARRaycastManager para detectar superficies
ARRaycastManager raycastManager;
List<ARRaycastHit> hits = new List<ARRaycastHit>();

void Update() {
    if (Input.touchCount > 0) {
        Touch touch = Input.GetTouch(0);
        if (touch.phase == TouchPhase.Began) {
            if (raycastManager.Raycast(touch.position, hits,
                TrackableTypes.PlaneWithinPolygon)) {
                Pose hitPose = hits[0].pose;
                Instantiate(prefabModelo, hitPose.position,
                    hitPose.rotation);
            }
        }
    }
}
```

Gestos: Escalar con Pinch:

```
// Escalar objeto con gesto de pinch (dos dedos)
void HandlePinchScale() {
    if (Input.touchCount == 2) {
        Touch t0 = Input.GetTouch(0);
        Touch t1 = Input.GetTouch(1);
        float prevDist = ((t0.position - t0.deltaPosition)
            - (t1.position - t1.deltaPosition)).magnitude;
        float currDist = (t0.position - t1.position).magnitude;
        float scaleFactor = currDist / prevDist;
        transform.localScale *= scaleFactor;
        float clampedScale = Mathf.Clamp(
            transform.localScale.x, 0.1f, 3.0f);
        transform.localScale = Vector3.one * clampedScale;
    }
}
```

### Reconocimiento de imagen / marcador (Vuforia):

```
// Listener para cuando se detecta un marcador
public class MarkerHandler :
    MonoBehaviour, ITrackableEventHandler
{
    void OnTrackingFound() {
        modelo3D.SetActive(true);
        audioSource.Play();
        modelo3D.transform.localPosition =
            new Vector3(0, 0.1f, 0); // 10cm arriba
        modelo3D.transform.localScale =
            Vector3.one * 0.5f; // 50% escala
    }
    void OnTrackingLost() {
        modelo3D.SetActive(false);
        audioSource.Stop();
    }
}
```

### Rotación con gesto de dos dedos:

```
// Rotar objeto con gesto de rotación
void HandleTwoFingerRotation() {
    if (Input.touchCount == 2) {
        Touch t0 = Input.GetTouch(0);
        Touch t1 = Input.GetTouch(1);
        Vector2 prevDir = (t0.position - t0.deltaPosition)
            - (t1.position - t1.deltaPosition);
        Vector2 currDir = t0.position - t1.position;
        float angle = Vector2.SignedAngle(prevDir, currDir);
        transform.Rotate(Vector3.up, angle);
    }
}
```

## Tema 3: Publicación de aplicaciones

### Actividad 3 - Plataformas y proceso de publicación

#### Plataformas que soportan Realidad Aumentada

Plataforma	SDK/Framework	Requisito	Tienda
Android	ARCore (Google)	Android 7.0+	Google Play Store
iOS	ARKit (Apple)	iPhone 6s+, iOS 11+	App Store
Web (WebXR)	AR.js / Three.js / 8th Wall	Navegador WebXR	URL directa
HoloLens	MRTK	HoloLens 2	Microsoft Store
Meta Quest	Passthrough API + Unity	Quest 3 / Quest Pro	Meta Quest Store
Snapchat	Lens Studio	Snapchat instalado	Lens Explorer
Instagram/FB	Spark AR Studio	Instagram/Facebook	Spark AR Hub

#### Publicación en Google Play Store (Android)

- Preparación del Build:** En Unity: File > Build Settings > Android. Configurar Player Settings: nombre del paquete, versión, API mínima, íconos y splash screen.
- Firma del APK/AAB:** Crear un keystore (.jks) para firmar la aplicación. Generar un Android App Bundle (.aab) firmado.
- Google Play Console:** Crear cuenta de desarrollador (\$25 USD). Crear nueva aplicación, completar ficha: título, descripción, capturas, categoría.
- Subir el Bundle:** En Producción, crear nueva versión. Subir archivo .aab. Completar notas de la versión.
- Revisión y Publicación:** Google revisa la app (1-7 días). Se puede usar testing interno antes de la publicación abierta.

#### Publicación en App Store (iOS)

- Requisitos:** Mac con Xcode instalado. Apple Developer Program (\$99 USD/año). Certificado de distribución.
- Build desde Xcode:** Unity exporta un proyecto Xcode. Se compila, firma y sube via Xcode o Transporter.
- App Store Connect:** Completar la ficha, subir screenshots, configurar precios. Enviar a revisión de Apple (1-3 días).

#### Distribución Interna (APK directo)

Para el ámbito académico, una alternativa práctica es la distribución directa del APK:

- Se genera el APK firmado desde Unity.
- Se comparte via enlace de descarga (Google Drive, servidor web).
- Los usuarios instalan habilitando "Orígenes desconocidos" en su dispositivo.
- Ventaja: No requiere cuenta de desarrollador ni proceso de revisión.
- Limitación: Solo Android, sin actualizaciones automáticas.

## Reporte Técnico del Proceso de Implementación

### Herramientas Utilizadas

Herramienta	Versión	Propósito
Unity	2022.3 LTS	Motor de desarrollo principal
AR Foundation	5.1	Framework de RA multiplataforma
Vuforia Engine	10.18	Reconocimiento de marcadores
Blender	3.6	Modelado y texturizado 3D
Visual Studio	2022	Editor de código C#
Android SDK	33+	Compilación para Android
Figma	Web	Diseño de UI/UX
Audacity	3.4	Edición de audio

### Proceso de Desarrollo

- Planificación:** Definición de escenas, modelos y flujo de navegación. Creación de wireframes en Figma.
- Configuración del proyecto:** Nuevo proyecto Unity con URP. Instalación de AR Foundation, ARCore XR Plugin y Vuforia.
- Creación de marcadores:** Diseño de 5 marcadores impresos. Registro en la base de datos de Vuforia (rating mínimo 4/5).
- Modelado 3D:** Creación/adaptación de los 5 modelos principales en Blender. Optimización de polígonos y texturas.
- Desarrollo de escenas:** Implementación de cada escena AR con su lógica de interacción, multimedia y UI.
- Implementación de gestos:** Programación de tap, drag, pinch y rotate usando el sistema de input de Unity.
- Integración multimedia:** Adición de audio (narración + efectos), video (clips informativos) e imágenes 2D (etiquetas).
- Testing:** Pruebas en dispositivos físicos Android. Verificación de tracking, rendimiento (30+ fps) y usabilidad.
- Build y distribución:** Generación del APK firmado. Distribución via enlace de descarga.

### Elementos Multimedia Integrados

Tipo	Cantidad	Formato	Descripción
Modelos 3D	12	.fbx	5 principales + 7 sub-modelos
Imágenes 2D	15	.png	Etiquetas, diagramas, íconos de UI
Audio	8	.ogg	5 narraciones + 3 efectos de sonido
Video	3	.mp4	Clips informativos de 30-60 seg
Animaciones	5	Unity Anim	Órbitas, ciclo del motor, corriente

## Conclusión

El desarrollo del proyecto de Realidad Aumentada permitió integrar conocimientos de modelado 3D, programación en C#, diseño de interfaces móviles y procesos de publicación de aplicaciones. La experiencia de crear una aplicación AR funcional desde cero demostró el potencial de esta tecnología en el ámbito educativo.

Los principales aprendizajes incluyen: la importancia de optimizar modelos 3D para dispositivos móviles, el manejo de interacciones táctiles (gestos) para una experiencia de usuario intuitiva, la correcta integración de elementos multimedia (audio, video, imágenes) en entornos AR, y el conocimiento del proceso completo de publicación en las diferentes plataformas.

La aplicación "UTP AR Explorer" representa un paso hacia la modernización de los métodos de enseñanza en la universidad, demostrando que la Realidad Aumentada puede ser una herramienta accesible y efectiva para mejorar el aprendizaje.

## Bibliografía

**Azuma, R. T. (1997). "A Survey of Augmented Reality". Presence: Teleoperators and Virtual Environments, 6(4), 355-385.**

*Artículo fundacional que define los principios y taxonomía de la Realidad Aumentada.*

**Unity Technologies. (2026). "AR Foundation Documentation".**

*Documentación oficial de AR Foundation para desarrollo de aplicaciones de RA multiplataforma.*

*<https://docs.unity3d.com/Packages/com.unity.xr.arfoundation@5.1/>*

**Google. (2026). "ARCore Developer Documentation".**

*Documentación del SDK de RA de Google para Android.*

*<https://developers.google.com/ar>*

**Vuforia. (2026). "Vuforia Engine Developer Library".**

*Documentación del SDK de reconocimiento de imágenes Vuforia.*

*<https://library.vuforia.com/>*

**Linowes, J. & Babilinski, K. (2020). "Augmented Reality for Developers". Packt Publishing.**

*Guía práctica para desarrolladores sobre implementación de aplicaciones de RA con Unity y ARCore/ARKit.*

**Google Play Console. (2026). "Publish your app".**

*Guía oficial para publicar aplicaciones en Google Play Store.*

*<https://support.google.com/googleplay/android-developer/>*