



UNIVERSIDAD AUTÓNOMA DE QUERÉTARO
FACULTAD DE INGENIERÍA



ANIMACIÓN DIGITAL
Y MEDIOS INTERACTIVOS

UNIVERSIDAD AUTÓNOMA DE QUERÉTARO
FACULTAD DE INGENIERÍA



APLICACIÓN MÓVIL INMERSIVA E INTERACTIVA
GEOESPACIAL EN REALIDAD AUMENTADA.
"TERANOVUS"

TESIS INDIVIDUAL PROYECTUAL
PARA OBTENER EL GRADO DE
LICENCIATURA EN ANIMACIÓN DIGITAL Y MEDIOS
INTERACTIVOS

PRESENTA:

JESÚS ANTONIO BARRIOS AGUIRRE

DIRECTOR:

I.C. DIEGO HERNÁNDEZ REYES

SECRETARIO:

M. EN A. LUIS SÁNCHEZ LARA

VOCAL:

DRA. ADRIANA ROJAS MOLINA

SANTIAGO DE QUERÉTARO, QRO

12 de febrero, 2024.





Dirección General de Bibliotecas y Servicios Digitales
de Información



Aplicación móvil inmersiva e interactiva geoespacial
en realidad aumentada: TeraNovus

por

Jesús Antonio Barrios Aguirre

se distribuye bajo una [Licencia Creative Commons
Atribución-NoComercial-SinDerivadas 4.0 Internacional](#).

Clave RI: IGLIN-242502



RESUMEN

Se propone "TeraNovus", como una aplicación inmersiva e interactiva geoespacial en realidad aumentada, aprovechando el entorno real en conjunto con las tecnologías modernas, localizando estructuras y/o edificios, así como puntos de interés e información en el campus "Centro Universitario" de la Universidad Autónoma de Querétaro.

Esta aplicación hace uso de la geolocalización, datos descargados de la nube, la realidad aumentada, una función que es posible con la realidad aumentada, los "image Target" u objetivos AR, esto en concordancia el mundo real, para guiar y ayudar al usuario a tener conocimiento no solo de las edificaciones, estructuras y puntos de interés del campus, sino también a conocer proyectos de la comunidad universitaria que se desarrollan dentro de los espacios físicos mencionados, además, que podrá ser una guía que le informe sobre los contactos a los que podrá acudir en busca de ayuda e información que requiera de cada oficina o dirección educativa instalada en estas áreas.

Con esta aplicación los usuarios podrán adentrarse en el mundo de realidad aumentada dentro del campus "Centro Universitario" teniendo la oportunidad de conocer y ser guiados hacia las edificaciones, estructuras y puntos de interés, fomentado así el conocimiento y la cultura e introduciendo a la universidad a las tecnologías modernas, entrando al mundo de la realidad aumentada desde la practicidad de un simple toque de su teléfono celular.



Agradecimientos especiales:

Agradezco mucho a las personas que me ayudaron con sus retroalimentaciones, consejos y apoyo, a los amigos que generé en esta carrera, sobre todo a mi director de tesis, Diego, quién con su experiencia y ayuda en conjunto con las enseñanzas que obtuve en esta carrera y el trabajo, logré y conseguí lanzar esta aplicación en mi tiempo reto propuesto. Gracias a estas relaciones conseguí tener empleos incluso antes de terminar la carrera, así como una vida estudiantil muy divertida.

A mis familiares por su apoyo y sobre todo a mi señora madre, quien, con su gran esfuerzo como madre soltera, me apoyo en esta decisión, quien aún con sus dificultades de salud y tiempo siempre estuvo ahí en mis momentos de necesidad, quién con su máximo esfuerzo y apoyo logró disminuir una discapacidad con mucho tiempo y esfuerzo para yo tener una vida tranquila y "normal" en el que, según doctores, no debería tener ni capacidad de hablar ni de terminar una licenciatura.

Gracias madre, por llevarme a todos los lugares que debía ir, por traerme mis cosas que siempre olvidaba, por enseñarme muchas cosas, por las deliciosas comidas, por tu esfuerzo constante para mejorar mi vida, por darme la oportunidad de tener educación completa y prepararme para un futuro y por tenerme paciencia.

-Gracias a todos-



ÍNDICE

<i>Resumen</i>	II
<i>Índice</i>	IV
<i>Tabla de ilustraciones</i>	VIII
<i>I. Introducción</i>	- 11 -
<i>II. Justificación y Descripción del problema</i>	- 14 -
<i>III. Objetivos</i>	- 16 -
3.1 Objetivo general	- 16 -
3.2 Objetivos específicos	- 16 -
3.3 Fases y metas	- 16 -
3.4 Actividades	- 17 -
<i>IV. Alcance y resultados esperados</i>	- 19 -
4.1 Alcance del proyecto	- 19 -
4.2 Resultados esperados	- 19 -
<i>V. Marco teórico</i>	- 20 -
5.1 Marco conceptual	- 20 -
5.1.1 Realidad aumentada (AR)	- 20 -
5.1.2 Realidad aumentada basada en marcadores	- 20 -
5.1.3 Unity 3D	- 20 -
5.1.4 Modelado Low-Poly y High-Poly	- 20 -
5.1.5 Sistema de geolocalización	- 21 -
5.1.6 Anclajes geoespaciales	- 21 -
5.1.7 Desarrollo de aplicación móvil	- 21 -
5.1.8 Servidor en la nube	- 22 -
5.1.9 Microservicios	- 22 -
5.1.10 Interfaz de programación de aplicaciones (API)	- 22 -
5.1.11 Amazon Web Services (AWS)	- 22 -
5.1.12 ARCore	- 23 -
5.1.13 AR Foundation	- 23 -
5.1.14 Campus: Centro Universitario (CU)	- 23 -
5.1.15 Solicitudes Web REST API	- 23 -
5.2 Fundamentación del proyecto	- 24 -
5.3 Requerimientos preliminares	- 25 -
5.3.1 Teléfonos inteligentes	- 25 -
5.3.2 Software Unity	- 25 -
5.3.3 La existencia del campus Centro Universitario	- 25 -



5.3.4	Paquete de plataformas de Google: ARCore y sistema de geolocalización asistida por Google Maps y Google Earth	- 25 -
5.3.5	Internet	- 26 -
5.3.6	Datos e información de la Universidad Autónoma de Querétaro	- 26 -
5.3.7	Objetivos AR o "image targets" colocados en las distintas oficinas e instalaciones de la Universidad Autónoma De Querétaro	- 26 -
5.3.8	Servicio de la nube (AWS de Amazon)	- 26 -
VI.	Metodología	- 27 -
6.1	Preproducción	- 27 -
6.1.1	SCRUM y KANBAN	- 27 -
6.1.2	Recopilación de datos	- 27 -
6.1.3	Datos de puntos de interés	- 28 -
6.1.4	Datos de coordenadas	- 28 -
6.1.5	Datos de trabajos realizados por la comunidad	- 28 -
6.1.6	Datos de directorios institucionales	- 28 -
6.2	Creación y animación de gráficos tridimensionales	- 29 -
6.2.1	Modelado grafico en Low Poly	- 29 -
6.2.2	Generación de UV del grafico	- 29 -
6.2.3	Generación de materiales y/o texturas del grafico	- 29 -
6.2.4	Exportación del grafico en formato FBX	- 29 -
6.3	Creación de objetivos AR.	- 30 -
6.4	Subida de datos a la nube	- 30 -
6.4.1	Formato JSON	- 30 -
6.4.2	Formato prefab	- 31 -
6.4.3	Formato PNG o JPG	- 31 -
6.5	Revisión de datos y funcionalidades de la aplicación	- 31 -
VII.	Recursos	- 32 -
7.1	Equipo	- 32 -
7.1.1	Laptop DELL G7 7790	- 32 -
7.1.2	Teléfono inteligente Xiaomi Redmi Note 9s	- 32 -
7.1.3	Teléfono inteligente Xiaomi 11T Pro	- 32 -
7.2	Herramientas	- 32 -
7.2.1	GitHub: Sistema de control de versiones	- 32 -
7.2.2	Unity: Motor gráfico	- 33 -
7.2.3	Visual Studio Community: Entorno de desarrollo	- 33 -
7.2.4	Herramientas Android SDK	- 33 -
7.2.5	Geospatial VPS Google	- 33 -
7.2.6	Google Earth	- 33 -
7.2.7	Postman	- 33 -

7.2.8	S3 de Amazon Web Services	- 34 -
7.2.9	Figma	- 34 -
7.2.10	Adobe Illustrator	- 34 -
7.2.11	Blender	- 34 -
7.3	Información	- 34 -
7.3.1	Información de puntos de interés	- 34 -
7.3.2	Coordenadas de puntos de interés	- 35 -
7.3.3	Proyectos y trabajos realizados por la comunidad	- 35 -
7.3.4	Directorios y contactos institucionales de docentes o administrativos	- 35 -
VIII.	Desarrollo del trabajo	- 36 -
8.1	Generación, inicialización y preparación del proyecto.	- 36 -
8.1.1	Trello	- 36 -
8.1.2	GitHub	- 39 -
8.1.3	Unity	- 40 -
8.1.4	API Google	- 43 -
8.1.5	AWS: S3	- 45 -
8.2	Desarrollo del diseño gráfico	- 47 -
8.2.1	Diagrama de las funciones básicas	- 48 -
8.2.2	Diseño del apartado gráfico: Wireframe y diseño básico	- 50 -
8.2.3	Diseño del apartado gráfico: Color y logotipo	- 56 -
8.2.4	Diseño del apartado gráfico: Diseño final de la aplicación.	- 59 -
8.2.5	Diseño del apartado gráfico: Generación de las pantallas en Unity	- 62 -
8.3	Objetos en una coordenada global dentro del sistema AR	- 68 -
8.3.1	Lógica de generación de objetos en coordenadas globales	- 69 -
8.3.2	Información de los puntos de interés a generar en sistema AR	- 73 -
8.3.3	Visualización de la información del elemento	- 76 -
8.3.4	Recopilación de puntos de interés	- 82 -
8.3.5	Corrección: funcionalidad UI, números de extensión	- 86 -
8.3.6	Visualización y dirección del punto seleccionado	- 88 -
8.3.7	Información de los trabajos a generar en el sistema AR	- 89 -
8.3.8	Iconos a mostrar en los puntos de interés	- 92 -
8.3.9	Creación de objetos 3D para los puntos de interés	- 94 -
8.3.10	Agregado de objetos a la lógica de generación de objetos	- 98 -
8.4	Objetivos AR o "Image Target"	- 100 -
8.4.1	Creación base de Objetivos AR	- 100 -
8.4.2	Diseño de objetivos AR	- 103 -
8.4.3	Lógica de Objetivos AR en Unity	- 105 -
8.4.4	Agregado de las imágenes a la librería de imágenes	- 106 -
8.4.5	Corrección: Problema de carga en runtime del LibraryImageTarget de Unity	- 107 -



8.4.6	Testeo de Objetivos AR y agregado de botones interactivos	_____	- 108 -
8.5	Agregado de las funcionalidades en la nube (AWS)	_____	- 110 -
8.5.1	Descarga y obtención: Datos a través del JSON desde la nube	_____	- 111 -
8.5.2	Descarga y obtención: Objetos 3D a través de la nube	_____	- 114 -
8.5.3	Descarga y obtención: Iconos de cada punto de interés	_____	- 116 -
8.5.4	Corrección: Calidad de las imágenes de los iconos de cada punto de interés	_____	- 119 -
8.6	Subida de la aplicación en la Play Store	_____	- 120 -
8.6.1	Creación y configuración de los ajustes de Play Store	_____	- 120 -
8.6.2	Configuración del proyecto de Unity y exportación de la aplicación	__	- 125 -
8.6.3	Subida de aplicación a Google Console	_____	- 129 -
IX.	<i>Conclusiones y recomendaciones</i>	_____	- 134 -
X.	<i>Bibliografías</i>	_____	- 138 -

TABLA DE ILUSTRACIONES

8.1.1.1: EQUIPO DE TRABAJO CREADO EN TRELLO	- 37 -
8.1.1.2: LISTAS CREADAS EN EL TABLERO DE TRELLO	- 37 -
8.1.1.3: TARJETAS CREADAS EN LAS LISTAS DEL TABLERO EN TRELLO	- 39 -
8.1.2.1: AJUSTES BÁSICOS DE LA CREACIÓN DEL REPOSITORIO EN GITHUB	- 40 -
8.1.3.1: TEMPLATE "AR" DE UNITY	- 41 -
8.1.3.2: PAQUETES INSTALADOS EN EL PROYECTO DE UNITY DESDE EL PANEL "PACKAGE MANAGER"	- 41 -
8.1.3.3: PANEL "BUILD SETTINGS" DE UNITY	- 42 -
8.1.3.4: AJUSTE DE API LEVEL DE ANDROID SOPORTADOS	- 42 -
8.1.3.5: PAQUETE "ARCORE EXTENSIONS" INSTALADO EN EL PROYECTO DE UNITY	- 42 -
8.1.3.6: AJUSTES DE "ARCORE EXTENSIONS CONFIG" DISPONIBLES	- 43 -
8.1.3.7: AJUSTES COMPONENTE SCRIPT "ARCORE EXTENSIONS"	- 43 -
8.1.4.1: PROYECTO "TERANOVUS" EN LA CONSOLA DE GOOGLE CLOUD	- 44 -
8.1.4.2: DETALLES DEL API "ARCORE API"	- 44 -
8.1.4.3: PANTALLA DE CLAVES API DEL PROYECTO	- 44 -
8.1.4.4: AJUSTES DE "ARCORE EXTENSIONS" EN "PROJECT SETTINGS" DENTRO DE UNITY ...	- 44 -
8.1.4.5: COMMIT INICIAL EN GITHUB DESKTOP	- 45 -
8.1.5.1: PÁGINA INICIO DE LA CONSOLA AWS	- 46 -
8.1.5.2: SERVIDORES AWS PARA EL SERVICIO S3 DISPONIBLES (2022)	- 46 -
8.1.5.3: PÁGINA DE INICIO DE LA BUCKET DE "TERANOVUS_UNITY"	- 46 -
8.2.1.1: GLOSARIO DIAGRAMA DE ACCIONES BÁSICAS DEL USUARIO EN LA APLICACIÓN	- 48 -
8.2.1.2: DIAGRAMA DE ACCIONES BÁSICAS DEL USUARIO EN LA APLICACIÓN	- 48 -
8.2.1.3: DIAGRAMA DE EVENTOS DEL SISTEMA AR HABILITADO Y DESHABILITADO	- 50 -
8.2.2.1: OPCIONES DE FORMATOS DE PANTALLAS VERTICALES POR DEFECTO EN FIGMA (2022)- 51 -	
8.2.2.2: DISEÑO DE LA PANTALLA INICIAL (LOGIN) Y SUS POSIBLES PANELES DE DIÁLOGOS-	52 -
8.2.2.3: INFORMACIÓN DE ESTUDIANTE A TRAVÉS DEL PORTAL DE LA UNIVERSIDAD AUTÓNOMA DE QUERÉTARO	- 52 -
8.2.2.4: DISEÑO DE LA PANTALLA DE MENÚ PRINCIPAL (MAIN MENU) Y PANEL DE AJUSTES .-	53
-	
8.2.2.5: DISEÑO DE LA PANTALLA DE INFORMACIÓN (INFORMATION)	- 54 -
8.2.2.6: DISEÑO DE LOS PANELES DE INFORMACIÓN EXTRA	- 54 -
8.2.2.7: DISEÑO DE PANTALLA Y PANELES DE LA VISTA AR (ARVIEW)	- 56 -
8.2.3.1: JOE HALLOCK'S "COLOUR ASSIGNMENTS"	- 57 -
8.2.3.2: PALETA DE COLORES	- 58 -
8.2.3.3: DISEÑO DEL LOGO DE LA APLICACIÓN	- 58 -
8.2.4.1: DISEÑO FINAL PANTALLA INICIAL	- 59 -
8.2.4.2: DISEÑO FINAL PANTALLA MENÚ PRINCIPAL	- 60 -
8.2.4.3: DISEÑO FINAL PANTALLA INFORMACIÓN	- 61 -
8.2.4.4: DISEÑO FINAL PANTALLA VISTA AR	- 62 -
8.2.5.1: AJUSTES COMPONENTE "CANVAS SCALER" EN UNITY	- 62 -

8.2.5.2: UI GENERADO EN UNITY	- 63 -
8.2.5.3: UI DE PANTALLAS DE CARGA (IZQUIERDA) Y TRANSICIÓN (DERECHA)	- 64 -
8.2.5.4: UI PANTALLA MENÚ PRINCIPAL	- 64 -
8.2.5.5: DIÁLOGO DE INCOMPATIBILIDAD CON LAS FUNCIONES AR	- 65 -
8.2.5.6: UI PANTALLA INFORMACIÓN Y PANEL INFORMACIÓN EXTRA	- 66 -
8.2.5.7: FILTRO POR "FACULTAD" (IZQUIERDA) Y POR "TIPO" (DERECHA)	- 66 -
8.2.5.8: UI PANTALLA VISTA AR	- 67 -
8.3.1.1: COMPONENTE "AREARTHMANAGER"	- 69 -
8.3.1.2: CHEQUEO DEL ESTATUS DEL "TRACKING"	- 70 -
8.3.1.3: COORDENADAS EN WGS84 DEL EDIFICIO B DE INGENIERÍA	- 70 -
8.3.1.4: GEOGRAPHICLIB, "ONLINE GEOID CALCULATIONS"	- 71 -
8.3.1.5: REQUEST A LA API ELEVATION EN POSTMAN	- 72 -
8.3.1.6: RESULTADO REQUEST DE LA API ELEVATION	- 72 -
8.3.1.7: OBJETO GENERADO EN COORDENADAS DEL EDIFICIO B DE INGENIERÍA.....	- 73 -
8.3.2.1: ESTRUCTURA JSON DEL ELEMENTO "EDIFICIO B"	- 75 -
8.3.2.2: CLASES CON VARIABLES PARA LA LECTURA DEL JSON	- 76 -
8.3.3.1: VISIBILIDAD DEL ELEMENTO EN FORMATO CERCANO	- 77 -
8.3.3.2: VISIBILIDAD DEL ELEMENTO EN FORMATO LEJANO.....	- 77 -
8.3.3.3: NUEVO FORMATO DE VISUALIZACIÓN DE DATOS.....	- 78 -
8.3.3.4: MUESTRA DE LA VISIBILIDAD DE LOS ELEMENTOS UI 2D EN UN ESPACIO 3D	- 78 -
8.3.3.5: MUESTRA DE LA VISIBILIDAD CORREGIDA DE LOS ELEMENTOS UI 2D	- 79 -
8.3.3.6: COMPONENTE "CANVAS RAYCASTER".....	- 79 -
8.3.3.7: FUNCIONALIDAD UI: DETALLES.	- 80 -
8.3.3.8: FUNCIONALIDAD UI: CORREO ELECTRÓNICO	- 80 -
8.3.3.9: FUNCIONALIDAD UI: TELÉFONO	- 81 -
8.3.3.10: FUNCIONALIDAD UI: UBICACIÓN	- 82 -
8.3.4.1: UBICACIONES GUARDADAS EN GOOGLE MAPS	- 83 -
8.3.4.2: DATOS DE ELEMENTOS AGREGADOS AL EXCEL	- 84 -
8.3.4.3: INFORMACIÓN DE LOS ELEMENTOS GUARDADOS EN UN ARCHIVO JSON	- 85 -
8.3.4.4: OBJETOS GENERADOS MEDIANTE LA LISTA DADA DENTRO DEL ARCHIVO JSON	- 85 -
8.3.4.5: ELEMENTOS GENERADOS A PARTIR DE LA LISTA DADA EN ARCHIVO JSON	- 86 -
8.3.5.1: ERROR DE FUNCIONALIDAD DE TELÉFONO, MARCAR CON EL NÚMERO COMPLETO MANDA DIRECTAMENTE AL BUZÓN	- 87 -
8.3.5.2: CORRECCIÓN DE FUNCIONALIDAD DE TELÉFONO	- 88 -
8.3.6.1: FUNCIONALIDAD "MOSTRAR", VISUALIZACIÓN DE LAS GUÍAS VISUALES	- 89 -
8.3.7.1: NUEVA SINTAXIS JSON.....	- 92 -
8.3.8.1: FOTO SIN DISEÑO CIRCULAR	- 93 -
8.3.8.1: FOTO CON DISEÑO ICONO CIRCULAR	- 93 -
8.3.9.1: DROP DE LIBRERÍA BASE	- 95 -
8.3.9.2: BOXDROP DE INGENIERÍA	- 95 -
8.3.9.3: BOXDROP DE INGENIERÍA CON DROP DE LIBRERÍA BASE	- 96 -
8.3.9.4: UV DE OBJETO 3D SIN MODIFICAR.....	- 96 -
8.3.9.5: UV DE OBJETO 3D MODIFICADO	- 97 -
8.3.9.6: DROP DE LIBRERÍA CON MATERIAL AGREGADO	- 97 -



8.3.9.7: BOXDROP DE INGENIERÍA CON DROP DE LIBRERÍA	- 98 -
8.3.10.1: ACTUALIZACIÓN DE JSON DE ICONOS, BOXDROPS Y DROPS	- 99 -
8.3.10.2: ICONO, DROP Y BOXDROPS DE SECRETARIA ACADEMICA DE INGENIERÍA	- 99 -
8.4.1.1: IMAGEN "QR" DE REFERENCIA	- 101 -
8.4.2.1: PRIMERA PARTE DEL DISEÑO DE OBJETIVO AR	- 104 -
8.4.2.2: DISEÑO FINAL DEL OBJETIVO AR	- 104 -
8.4.3.1: OBJETO XR REFERENCE IMAGE LIBRARY	- 105 -
8.4.4.1: LIBRERÍA DE IMÁGENES	- 106 -
8.4.6.1: DISEÑO OBJETIVO AR DIGITAL	- 108 -
8.4.6.2: VISTA REAL DEL OBJETIVO EN OFICINAS DE RECTORÍA	- 109 -
8.4.6.3: VISTA DIGITAL DEL OBJETIVO AR EN LAS OFICINAS DE RECTORÍA	- 109 -
8.5.1.1: ARCHIVO JSON CARGADO EN EL SISTEMA S3	- 111 -
8.5.1.2: INFORMACIÓN DENTRO DEL ARCHIVO JSON EN EL NAVEGADOR WEB	- 112 -
8.5.1.3: MECÁNICA BASE PARA LA RECOGIDA DE INFORMACIÓN A TRAVÉS DE LA NUBE	- 113 -
8.5.1.4: CLASE UTILIZADA PARA LA SERIALIZACIÓN DEL JSON	- 113 -
8.5.1.5: SCRIPTABLEOBJECT QUE GUARDA LOS DATOS DEL JSON YA SERIALIZADO	- 114 -
8.5.2.1: VISUALIZACIÓN DE LOS ADDRESSABLES CREADOS DE LOS BOXDROPS Y LOS DROPS ...	-
115 -	
8.5.2.2: ARCHIVOS BUNDLES DE LOS BOXDROPS Y DROPS	- 115 -
8.5.2.3: BUNDLES SUBIDOS AL SERVIDOR S3	- 116 -
8.5.2.4: CONFIGURACIÓN DE CONEXIÓN SERVIDOR/APLICACIÓN PARA LA DESCARGA DE LOS BUNDLES	- 116 -
8.5.3.1: IMAGENES DE UBICACIONES SUBIDAS AL SERVIDOR S3	- 117 -
8.5.3.2: IMÁGENES DE TRABAJOS SUBIDOS AL SERVIDOR S3	- 117 -
8.5.3.3: MUESTRA DE IMAGEN DE TRABAJO A TRAVÉS DEL LINK DESCARGABLE	- 118 -
8.5.3.4: LINK DESCARGABLE DE LA IMAGEN EN EL JSON	- 118 -
8.6.1.1: BOTÓN "CREAR APLICACIÓN" EN GOOGLE CONSOLE	- 120 -
8.6.1.2: CREACIÓN DE APLICACIÓN, DETALLES	- 121 -
8.6.1.3: CONFIGURACIÓN: INFORMACIÓN DE LA APLICACIÓN	- 122 -
8.6.1.4: POLÍTICAS DE PRIVACIDAD	- 122 -
8.6.1.5: CLASIFICACIÓN DEL CONTENIDO: CUESTIONARIO	- 123 -
8.6.2.1: VERSIÓN DE EDITOR DE UNITY 3D Y SUS MÓDULOS	- 125 -
8.6.2.2: MÓDULO ANDROID DE UNITY	- 126 -
8.6.2.3: BUILD SETTINGS	- 126 -
8.6.2.4: PROJECT SETTINGS: AJUSTES BÁSICOS	- 127 -
8.6.2.5: PROJECT SETTINGS: AJUSTES AVANZADOS	- 127 -
8.6.2.6: "SPLASH IMAGE"	- 128 -
8.6.2.7: BUILD SETTING: BUILD APP BUNDLE	- 128 -
8.6.3.1: GOOGLE CONSOLE: PRUEBA ABIERTA	- 130 -
8.6.3.2: GOOGLE CONSOLE: APP BUNDLES	- 130 -
8.6.3.3: GOOGLE CONSOLE: RESUMEN VERSIÓN	- 131 -
8.6.3.4: GOOGLE CONSOLE: BOTÓN REVISIONES	- 131 -
8.6.3.5: ACEPTACIÓN DE APLICACIÓN REVISADA EN GOOGLE CONSOLE	- 132 -
8.6.3.6: FICHA EN LA PLAY STORE	- 133 -



9.1.1 CANCHAS DE BÁSQUETBOL, FACULTAD DE QUÍMICA	- 134 -
9.1.2 VISTA SATELITAL, CANCHAS DE BÁSQUETBOL, FACULTAD DE QUÍMICA	- 135 -
9.1.3 VISTA STREET VIEW PHARMAUAQ	- 135 -
9.1.4 VISTA SATELITAL DEL EDIFICIO BIOTECNOLÓGICO	- 135 -

I. INTRODUCCIÓN

Los teléfonos inteligentes tienen sus orígenes desde la década de 1990, y se popularizan rápidamente en el transcurso de la década de los 2010, cuando los teléfonos inteligentes comienzan a superar en venta a los teléfonos convencionales (Fundación Wikimedia Inc., 2022).

Congruentemente con esto, el Instituto federal de Comunicaciones (IFT, 2020) declara el 17 de febrero de 2020 en su página de internet que en México hay 80.6 millones de usuarios de internet y 86.5 millones de usuarios de teléfonos celulares, mientras en la página de The Competitive Intelligence Unit (The CIU, 2021) declara:

"Al cierre de la primera mitad de 2021 (2T-2021), se contabilizaron 127.7 millones de líneas móviles en activo en el país. Tomando en consideración el Censo Nacional de Población y Vivienda 2020, realizado por el INEGI, en México existen 126 millones de habitantes, lo que significa que, en 2021, por primera vez, alcanzamos la tan anhelada teledensidad 100%."

Lo que sin duda marca el auge de la comunicación a través de la telefonía celular, que ha migrado a los denominados teléfonos inteligentes.

Mientras el inicio de la herramienta de geolocalización tiene más tiempo, considerando dos hechos importantes, el inicio de esta herramienta en el 57' con el primer satélite artificial y el 8 de diciembre de 1993, siendo el día en el cual se permite el uso del GPS como uso civil (SINC, 2013). El inicio de esta herramienta en la sociedad como aplicación web fue hasta el 8 de febrero de 2005, siendo el año en que se creó *Google Maps* (Google, LLC, 2020). Esta tecnología está implementada en aplicaciones populares como *Rappi*, *UberEats*, *Waze*, entre otras.

En razón que desde el año 2005 esta herramienta tiene su auge al ser considerada como una herramienta de marketing y "delivery" o servicio de reparto con la llegada de las aplicaciones de pedido y compras a domicilio, pero también de movilidad gracias a los diversos juegos como *Pokémon GO* o aplicaciones de seguimiento de ejercicio y/o salud, ambos factores, marketing

y movilidad, son justificantes para llevar a la universidad a la vanguardia en materia de utilización de herramientas tecnológicas modernas.

En el año 2010, durante el foro de la reforma universitaria, el entonces secretario académico de la Universidad Autónoma de Querétaro, declaró que la modernización de la legislación universitaria debía contener criterios de equidad, entendiéndose con esto que se abría la puerta a realizar modificaciones a la legislación universitaria, a fin de dar paso a la equidad, no solo de género si no inclusiva, a aquellos alumnos con capacidades diferentes, siendo en 2012 el año en el cual se implementan por el entonces rector, acciones hacia la discapacidad, entendiéndose que desde ese momento la universidad mantiene un compromiso de inclusividad y de velar por la equidad de oportunidades para todos los alumnos. Y hasta enero de 2013 se firma un convenio entre la Universidad Autónoma de Querétaro, la Comisión Estatal de Derechos Humanos de Querétaro, la Coordinación de Accesibilidad y Desarrollo para las Personas con Discapacidad del Municipio de Querétaro y la Universidad Tecnológica de Santa Catarina (UTSC) en la cual la Universidad reafirma su compromiso de constituirse en una institución incluyente. (PIEE, 2022)

En fecha 17 de febrero de 2021, la actual rectora Dra. Margarita de Jesús Teresa García Gasca, emite su tercer informe de labores en el cual se puede leer que en el año 2020 - 2021 que acudían a nuestra universidad al menos 3,776 estudiantes foráneos, que la universidad contaba en ese periodo con 23 campus o planteles en los 18 municipios del estado (UAQ, 3er Informe / Teresa García Gasca, 2021), comunidad a la que sin duda forma parte del objetivo meta que va dirigido el actual proyecto que pretendo realizar, puesto que la universidad cuenta con un extenso territorio con muchas estructuras lo que le llega a ser complicado y difícil el conocimiento de todas estas estructuras.

De la propia "Misión" de la universidad que se lee en su portal de internet, se desprende el compromiso con la sociedad queretana y con el "*desarrollo integral de Querétaro, México y con el reconocimiento a nivel Internacional, así como su compromiso con la investigación, como institución científica, para generar conocimientos con alto sentido humanista, compromiso y responsabilidad social*" (UAQ, Misión, 2022).

Derivado de estos antecedentes, es claro que la elaboración de este proyecto tiene definida la necesidad y justificación para su elaboración, dado que su



entorno no solo es como herramienta tecnológica, sino como una herramienta de cumplimiento de objetivos, de interacción con la sociedad y de inclusión.

II. JUSTIFICACIÓN Y DESCRIPCIÓN DEL PROBLEMA

La justificación del presente proyecto se hace basándose en las siguientes consideraciones:

Es perceptible para todos que en la última década la forma de comunicación entre los universitarios y la sociedad en general ha cambiado, introduciendo en nuestras vidas diarias la comunicación digital interactiva, de ahí que las redes sociales como *Facebook*, *Instagram* y *TikTok*, entre otras han tenido tanto auge y uso. Y si bien, habrá quienes puedan nombrar un sinnúmero de inconvenientes que estas producen, como ya establecí anteriormente, gracias a que han proyectado la comunicación digital interactiva a la sociedad incluso globalmente, este hecho abre un campo laboral para los estudiantes de la carrera profesional que curso, "*Animación Digital y medios interactivos*". Por lo que, sin duda, a través de este proyecto de aplicación geoespacial, se estará permitiendo visualizar lo que los estudiantes de esta carrera somos capaces de realizar, permitiendo con ello promocionar esta carrera profesional la cual está por entregar a la sociedad queretana en su primera generación.

En conjunto con lo anterior de que en nuestra universidad concurren un gran número de estudiantes "foráneos" tal y como lo establece el 3er informe de la rectoría, de la *Dra. Margarita de Jesús Teresa García Gasca*, en el cual se puede leer que en el año 2020 - 2021 acudían a nuestra universidad al menos 3,776 estudiantes foráneos y que en algún momento de la carrera o durante el curso de bachillerato o universidad de esos estudiantes deberán acudir al campus central de "Centro Universitario" por algún trámite, lo cual a veces se vuelve complicado por la gran extensión y múltiples edificios con los que cuenta el campus central, en razón de ello, esta herramienta interactiva permitirá a los estudiantes locales y foráneos, así como visitantes poco familiarizados con las instalaciones del campus central, ubicar más fácilmente el lugar al que deben acudir.

Considerando lo anterior, es de establecerse que una de las ocupaciones de nuestra máxima casa de estudios ha sido la inclusión para estudiantes con capacidades diferentes, comunidad a la cual yo pertenezco en el área de "*hipoacúsicos*", a quienes se nos da más la comunicación gráfica o escrita, sin



embargo, cierto es que no soy el único con esta condición, existen además de universitarios como yo o con distintas capacidades diferentes, universitarios con un cierto grado o nivel de autismo a quienes se les dificulta interactuar de forma espontánea con los demás estudiantes, y muchas veces no les es tan fácil solicitar ayuda para poder saber la ubicación de algún edificio o lugar al que quieren acudir, pretendo que esta herramienta interactiva pueda serles de ayuda y sentir seguridad y/o comodidad al usarla.

La universidad siempre ha sido del interés de la sociedad queretana, quien ve en ella un reflejo de familiaridad y de pertenencia, y es claro que siempre buscará sentirse cercana, por lo cual esta herramienta geoespacial dará esa oportunidad de acercamiento, dando la oportunidad de poder conocer la distribución y ubicación de edificios icónicos de nuestra alma mater a la sociedad.

No pasa por alto que la universidad autónoma de Querétaro es una universidad de vanguardia con innumerables logros, que creo son necesarios promocionar y si en un momento dado, en el futuro, esta herramienta interactiva logra enlazar el edificio a una breve descripción de los proyectos de la comunidad estudiantil que se realizan en cada edificio o localización, considero que sería una promoción que visibilizaría el trabajo de nuestra máxima casa de estudios queretana.

Por estas consideraciones, es que he decidido elaborar el presente proyecto.

III. OBJETIVOS

3.1 Objetivo general

Este proyecto tiene como objetivo general, el crear una herramienta inmersiva al campus universitario "Centro Universitario" sencilla y funcional, misma que contribuya a la modernidad digital, a la movilidad, a la inclusión, al cumplimiento de la misión de la universidad, a la interacción y el desarrollo integral con la sociedad queretana.

3.2 Objetivos específicos

- a. Con esta herramienta tecnológica se pretende modernizar la forma de interactuar de los universitarios y su forma de concebir la universidad desde un punto de vista interactivo.
- b. En virtud de los grandes pasos de la universidad hacia la educación inclusiva, uno de los objetivos de este proyecto es coadyuvar a propiciar esa inclusión, convirtiendo esta aplicación en una forma de contribuir a esa inclusión.
- c. Si bien, la universidad siempre ha sido un precedente en la sociedad queretana, por el impulso que el alma mater otorga al desarrollo integral de la misma, esta herramienta permitirá que la sociedad pueda interactuar en los sitios de interés desde la comodidad de su teléfono inteligente.
- d. Ofrecer a la comunidad estudiantil una mejor y más ágil forma de comunicación con las oficinas administrativas.

3.3 Fases y metas

- A. Creación del proyecto y configuraciones del proyecto.
- B. Recopilación de datos.
- C. Creación de prototipado gráfico de la aplicación.

- D. Desarrollo de funcionalidades de la aplicación.
- E. Alojamiento de la aplicación en la Play Store.

3.4 Actividades

- i. Generación del tablero con las actividades a través de la plataforma 'Trello' en formato Kanban.
- ii. Creación del proyecto en Unity y la configuración las APIs de AWS y Google.
- iii. Recopilación de datos: Puntos de interés y coordenadas.
- iv. Recopilación de datos: Trabajos hechos por la comunidad.
- v. Recopilación de datos: Directorios y contactos institucionales.
- vi. Creación del diseño de interfaz gráfica de la aplicación en Figma.
- vii. Creación de gráficos: gráficos vectoriales para la interfaz gráfica de la aplicación en y agregados al proyecto de Unity.
- viii. Creación de gráficos: gráficos tridimensionales necesarios para el proyecto y reemplazarlos por los gráficos primitivos.
- ix. Creación de gráficos: Objetivos AR.
- x. Implementación de las funcionalidades de la interfaz gráfica (UI) así como también las pantallas de la aplicación.
- xi. Desarrollo de funcionalidad: generación de gráficos primitivos tridimensionales en los puntos de interés.
- xii. Desarrollo de funcionalidad: Mostrar ubicación y dirección desde la posición del usuario al punto de interés.



- xiii. Desarrollo de funcionalidad: Agregado de trabajos en los puntos de interés.
- xiv. Desarrollo de funcionalidad: Generación de librería de objetivos AR y la detección de las mismas.
- xv. Desarrollo de funcionalidad: Visualización de datos del contacto según el objetivo 2D.
- xvi. Desarrollo de funcionalidad: Subida de datos a la nube e implementación de las funcionalidades principales de la aplicación a través de los datos de la nube.
- xvii. Generación de la aplicación y sus datos para el respectivo alojamiento de la aplicación en la tienda Play Store de Google.



IV. ALCANCE Y RESULTADOS ESPERADOS

4.1 Alcance del proyecto

Debido al tiempo que pongo como marca y reto para desarrollar esta aplicación con lo aprendido en la carrera, así como en las experiencias laborales, para esta aplicación se planea tener gráficos básicos y publicada en la Play Store como acceso anticipado, funcional con al menos 15 puntos de interés y con al menos 10 trabajos desarrollados por la comunidad universitaria, así como también 10 contactos con sus respectivos objetivos AR, ubicadas entre distintas ubicaciones relevantes dentro del campus CU.

4.2 Resultados esperados

La aplicación desarrollada en este proyecto, será una aplicación que utilice las tecnologías nuevas y disponibles de la realidad aumentada, con la que podamos guiar a usuarios entre su punto de ubicación y el punto de ubicación de interés seleccionado, así como también tener información desplegada en los más de 10 distintos puntos de interés, ubicados a los alrededores de la Facultad de Ingeniería y Rectoría, así como también información desplegada a través de los objetivos AR ubicados en varios puntos.

V. MARCO TEÓRICO

5.1 Marco conceptual

5.1.1 Realidad aumentada (AR)

La realidad aumentada, conocido por las siglas AR, es un entorno interactivo y superpuesto al mundo real que es posible, gracias y principalmente, a través de elementos digitales tanto visuales y de sonidos, así como también de otros estímulos sensoriales de distintas tecnologías como la inteligencia artificial, procesamiento de datos, utilización de cámaras y también de sensores como los giroscopios o acelerómetros que permiten la mezcla de contenido generado por computadora con el mundo real posible en 3D, en tiempo real e interactiva. El AR incorpora varias características, entre ellas la combinación del mundo físico con un entorno virtual, la interacción en tiempo real dentro del entorno virtual superpuesto con el físico y la identificación de objetos e imágenes digitales y reales. (Microsoft, 2022)

5.1.2 Realidad aumentada basada en marcadores

Este tipo de realidad aumentada se crea utilizando el reconocimiento de imágenes para identificar objetos e imágenes a la vista de las cámaras del dispositivo en las aplicaciones AR. Estos objetos sirven como puntos de referencia para determinar una posición u orientación de la cámara u objeto. Este objeto a la vista se compara con los datos del banco de información previamente generada hasta encontrar una coincidencia. (Microsoft, 2022)

5.1.3 Unity 3D

Unity 3D, es uno de los motores de videojuegos más utilizados en mercado como Unreal y Godot, en estos motores de videojuegos se pueden crear videojuegos 2D o 3D para distintas plataformas, todo gracias a las herramientas que tiene, al editor visual y a la programación basada en C#.

5.1.4 Modelado Low-Poly y High-Poly

Los modelos 3D, al estar conformado por polígonos o triángulos en su expresión resumida y más pequeña, pueden llegar a tener ciertos niveles de

detalles, el Low-Poly busca simplificar la figura o forma del modelo, reduciendo el número de triángulos al contrario del High-Poly donde se crean modelos más completos y detallados.

5.1.5 Sistema de geolocalización

El sistema de geolocalización determina la posición de cualquier objeto, persona o vehículo en un sistema de coordenadas determinado, con un margen de error de solo unos pocos metros en el espacio geográfico, esta información se obtiene a través de los dispositivos móviles o de sobremesa y satélites para identificar o describir la ubicación física real. Hoy en día es lo suficientemente precisa para obtener una dirección en el mapa, en vez de obtener solo las coordenadas de latitud, longitud y altitud. Esta ubicación se obtiene a través de las torres de telefonía móvil, aunque esta no suele ser precisa, también se puede obtener a través de las conexiones WIFI mediante la dirección IP de un router y a través de la localización por el sistema de posicionamiento global (GPS) que se logra mediante los satélites ubicados alrededor de la tierra. (Google, LLC, 2015)

5.1.6 Anclajes geoespaciales

Las anclas geoespaciales son anclas que permiten colocar contenidos 2D o 3D en cualquier coordenada latitud, longitud y altitud, estas coordenadas son independientes de la ubicación virtual del software o del usuario debido a que estas anclas se colocan dentro de las coordenadas del mundo. (Google, LLC, 2022)

5.1.7 Desarrollo de aplicación móvil

Se diseñan aplicaciones para ejecutarse en un dispositivo móvil que se desarrollan a través distintas plataformas, según el sistema operativo al que se requiera desarrollar, según el sistema operativo al que se requiera desarrollar son las herramientas a utilizar como los Kits de desarrollo de software (SDK), librerías, bibliotecas o programas de desarrollo, estas aplicaciones van de la mano de dos partes importantes, el diseño de la interfaz y el desarrollo software en uno de los diversos lenguajes de programación. (Microsoft, 2022)

5.1.8 Servidor en la nube

Estos servidores son servidores virtuales de los servidores físicos, estos se ejecutan en un entorno virtual, lo que permite a una cantidad de usuarios ilimitados acceder a los datos, archivos y aplicaciones alojados a través de una red como lo es la internet. Estos servidores no tienen limitaciones en cuanto a cantidad de potencia informática, por lo que es fácil actualizar datos o el espacio para admitir más usuarios, también los hace seguros frente a sobrecarga de usuarios o peticiones, así como de malas modificaciones de datos o de algún problema a nivel de software y los hace más asequibles gracias a que tienen menos gastos de hardware y energéticos. Estos servidores suelen ser públicos, privados o híbridos. (Microsoft, 2022)

5.1.9 Microservicios

Con las arquitecturas monolíticas, los procesos están estrechamente asociados y se ejecutan como un solo servicio, limitando la experimentación, ya que si una se ve más demandada, los otros servicios tendrán dificultades para funcionar correctamente, lo que hace más difícil el tener control sobre estos servicios, sin embargo, los microservicios están siendo ejecutadas de forma independiente esta comunicación sucede a través de API ligeras, el microservicio permite además actualizar, escalar e implementar de forma independiente para satisfacer la demanda de funciones de una aplicación. (Amazon, 2022)

5.1.10 Interfaz de programación de aplicaciones (API)

Estos contratos de servicio denominados interfaz son mecanismos que permiten a dos componentes de software comunicarse entre sí mediante un conjunto de definiciones y protocolos a modo de solicitudes y respuestas de cliente a servidor. (Amazon, 2022)

5.1.11 Amazon Web Services (AWS)

Es la plataforma en la nube que ofrece Amazon, donde tiene más de 200 servicios integrales de centros de datos a nivel global como el cómputo,

almacenamiento, base de datos, aprendizaje automático, inteligencia artificial, análisis de datos. (Amazon, 2022)

5.1.12 ARCore

Es la plataforma proporcionada por *Google* para desarrollar aplicaciones inmersivas en realidad aumentada, esta plataforma cuenta con diferentes APIs. ARCore permite al teléfono inteligente detectar el entorno, comprender e interactuar con el mundo y la información. Esta plataforma utiliza tres capacidades claves para integrar el contenido digital en el real, según lo que la cámara y los sensores de los dispositivos móviles detecten, como el seguimiento de movimiento, la comprensión del entorno y la estimación de luz. (Google, LLC, 2022)

5.1.13 AR Foundation

Esta plataforma es proporcionada por *Unity*, permitiendo así trabajar y desarrollar proyectos en realidad aumentada para las distintas plataformas existentes, esta herramienta es utilizada por los desarrolladores y se utiliza en conjunto con ARCore y *Unity*, ya que por sí solo no implementa ninguna característica de inmersión aumentada. (Unity Technologies, 2022)

5.1.14 Campus: Centro Universitario (CU)

La Universidad Autónoma de Querétaro (UAQ), del estado de Querétaro, contiene más de 10 distintos planteles de nivel bachillerato, más de 10 campus, más de 10 facultades, entre estos campus está el campus Centro Universitario, ubicado en el municipio de Querétaro, que ha estado en funcionamiento desde 1960, siendo el campus central de la universidad y uno de los campus más desarrollados con una mayor cantidad de estructuras, facultades, carreras, docentes y estudiantes. (UAQ, Historia, 2022).

5.1.15 Solicitudes Web REST API

La funcionalidad base es el poder agregar varias funciones o integraciones de terceros comunicándose con diferentes aplicaciones, sitios web o servidores privados, estas solicitudes organizan los datos y el cómo se accede a ellos e incluso pueden agregarse ciertas restricciones a estas para intercambiar información de forma segura a través de Internet



5.2 Fundamentación del proyecto

El presente proyecto se sustenta en la necesidad de crear una herramienta de geolocalización en realidad aumentada que permita la internación dentro del campus universitario denominado "Centro Universitario", herramienta que sin duda contribuirá al cumplimiento de la universidad autónoma de Querétaro, a razón de que permitirá primeramente ingresar a la universidad en la utilización de tecnologías modernas y en auge, en beneficio de los propios universitarios, visitantes y foráneos así como la movilidad de los mismos, lo que sin duda con tecnologías como aparatos de comunicación convencionales como los teléfonos de casa, o consolas de juegos o computadoras de casa no sería posible.

Aunado a lo anterior, esta herramienta contribuirá en visibilizar los logros de la comunidad universitaria, esto, al poder mostrar los puntos de interés y proyectos que se desarrollan en esos puntos, lo que dará realce académico y modernidad, aumentando el prestigio de nuestra alma mater no solo de forma local y sino también a nivel nacional e inclusive internacional, considerando que con un solo toque de los teléfonos inteligentes y a través del internet se está al alcance de difundir información a todos los rincones del mundo; en razón de que esta aplicación no solo difunde información y puntos de interés, sino que además el usuario podrá interactuar con ellos, lo que sin duda moderniza estos servicios.

5.3 Requerimientos preliminares

El presente proyecto requiere de los siguientes factores condicionantes:

5.3.1 Teléfonos inteligentes

En razón de la utilización de los sensores de giroscopio, acelerómetro, ubicación GPS y WIFI para el correcto funcionamiento de la aplicación, así como el dispositivo para alojar la aplicación misma.

5.3.2 Software Unity

Unity, es una plataforma de desarrollo de juegos y aplicaciones multiplataforma que tiene una licencia gratuita bajo ciertas condiciones, esta plataforma es necesaria para el desarrollo y la creación de la aplicación con la utilización en conjunto del AR Foundation, AWS y las APIs de *Google*.

5.3.3 La existencia del campus Centro Universitario

La ubicación del campus como sus diferentes puntos de interés, así como la existencia de la misma, es totalmente necesaria para el correcto funcionamiento de la aplicación, puesto que requiere de la ubicación para generar los puntos de interés y que sean mostrados al usuario que se encuentre en esa ubicación.

5.3.4 Paquete de plataformas de Google: ARCore y sistema de geolocalización asistida por Google Maps y Google Earth

Los paquetes de *Google* son necesarios para el funcionamiento de la aplicación, puesto que *Google* ofrece la plataforma ARCore que es la responsable de manipular los datos del entorno virtual con el mundo en la aplicación y la Geolocalización de Google Maps y Google Earth ayuda a la misma a tener los puntos de interés posicionados en el espacio de realidad aumentada.

5.3.5 Internet

El internet es completamente necesario para la comunicación entre el cliente (aplicación) y los servidores (API, microservicios, descarga de datos), puesto que sin ellas la aplicación no podrá funcionar correctamente.

5.3.6 Datos e información de la Universidad Autónoma de Querétaro

Que la Universidad Autónoma de Querétaro, a través de sus áreas respectivas, desee otorgar información respecto de puntos de interés, proyectos que desea difundir e información de contactos, directorios institucionales y horarios del personal de la universidad, de modo que esta información sea precisa, correcta y de libre acceso, puesto que sin esta información la aplicación no podrá compartir datos verídicos y precisos para la guía al usuario.

5.3.7 Objetivos AR o "image targets" colocados en las distintas oficinas e instalaciones de la Universidad Autónoma De Querétaro

Estos objetivos AR permiten tener un mejor control de qué información mostrar y que no mostrar al usuario al momento de enfocar con la cámara del dispositivo el objetivo del contacto, sin ellas, la aplicación se hará más lenta debido a los constantes procesos extras que tendría esta aplicación.

5.3.8 Servicio de la nube (AWS de Amazon)

AWS es la plataforma utilizada para la gestión de datos, en específico el servicio S3 que ofrece, donde se puede almacenar datos y acceder a mismos a través solicitudes del cliente según lo que necesite, lo que permite a la aplicación ser más ligera en datos y procesos para los dispositivos móviles, además de ayudar a modificar solo la base datos de la aplicación desde el servicio S3, sin necesidad modificar y de lanzar una nueva versión de la aplicación para que lo descarguen los usuarios.

VI. METODOLOGÍA

6.1 Reproducción

Este proyecto presenta distintos procesos de desarrollo, puesto que, por un lado, está el desarrollo visual y el desarrollo de funcionalidades de la aplicación móvil, además de la publicación de la misma, también se encuentra de generación de gráficos tridimensionales que se mostrarán en el entorno virtual, por último, el apartado de la obtención, generación y gestión de metadatos necesarias para la aplicación en la nube.

6.1.1 SCRUM y KANBAN

Por ello se plantea el uso de una metodología especial, evitando en gran medida de lo posible, retrasos o problemas que puedan surgir, reduciendo la dificultad al organizar las tareas y agilizando el proceso. Utilizando una de las metodologías ágiles de desarrollo de software, que es también una de las más utilizadas a nivel mundial por empresas que desarrollan software por excelencia. En un estudio realizado por PMI indica que el 71% de las organizaciones han aumentado su competitividad mediante la agilidad. (PMI's Pulse of the Profession, 2018)

Esta metodología es utilizada por su alta flexibilidad y agilidad en la que cada ciclo de desarrollo es más corto y rápido, lo que permite adaptar el trabajo a las condiciones del proyecto, esto según se van agregando las nuevas funcionalidades a la aplicación. Dentro de esta metodología, se utilizará el trabajo por Scrum y Kanban. Donde el trabajo se divide en porciones mínimas y se organizan en un tablero visual dividido en tareas pendientes, en curso y finalizadas, mientras se hacen en por bloques en tiempos cortos y fijos. (Santander Universidades, 2020)

6.1.2 Recopilación de datos

Debido a que algunos datos son privados y una parte de estos se podrían obtener en la página de directorios institucionales de la universidad, sin embargo, estos datos no están actualizados o hay directorios y contactos que no se encuentran ahí, por lo tanto no es 100% confiable de utilizar por el momento (UAQ, Directorio Institucional, 2022), ya que esta aplicación planea dar datos precisos, por lo que estos datos se deberán recopilar en su mayoría

de forma manual, ya sea de forma física en las instalaciones de la universidad o en las distintas páginas web para el caso de los contactos de los trabajos o proyectos.

6.1.3 Datos de puntos de interés

Esta recopilación de datos se hará de manera manual y selectiva en el campus Centro Universitario, juntando todos los puntos de interés y exposición que tenga en la Facultad de Ingeniería y sus alrededores y clasificándolos por edificios (bibliotecas, laboratorios, auditorios, centros de investigaciones, oficinas administrativas, centros de medias como TVUAQ o RADIOUAQ), esculturas y/o construcciones ubicadas en los distintos puntos.

6.1.4 Datos de coordenadas

Estos datos serán recopilados de manera visual al mismo tiempo la de obtención de puntos de interés, primero se utilizará Google Earth para obtener la coordenada de la ubicación en físico por el GPS del teléfono inteligente y guardando esta ubicación geográfica como marcador o *waypoints* en la misma aplicación para mantener un guardado digital de estos datos

6.1.5 Datos de trabajos realizados por la comunidad

Estos datos serán obtenidos de diversas fuentes, puesto que la gran mayoría de estos se exponen en las distintas páginas de *Facebook*, *Twitter* e *Instagram* de las carreras y facultades de la universidad. Por lo que se deberá hacer una recopilación y selección de trabajos recientes y variados junto a la información necesaria como el nombre del trabajo, fotografía del proyecto, nombre del creador e información del trabajo y contacto en caso de ser necesario.

6.1.6 Datos de directorios institucionales

Estos datos serán obtenidos por distintos medios, de manera física, en las diversas oficinas, de manera digital, mediante las páginas web de la universidad y de documentos recientes. Se clasificará la información por secretarías (secretaría académica, servicios escolares, etc.), espacios administrativos (centros de investigación, bibliotecas, etc.), personal de la universidad (docentes, coordinadores de las distintas carreras). Esta

información deberá contener nombre del contacto, correo institucional, número de extensión y ubicación del contacto (Oficina o edificio).

6.2 Creación y animación de gráficos tridimensionales

Utilizando el software Blender se crearón los gráficos tridimensionales de la aplicación con la metodología implementada en los estudios de videojuegos de la siguiente forma:

6.2.1 Modelado grafico en Low Poly

Modelado de objetos necesarios para el proyecto con densidad y detalle baja, se busca una menor cantidad de polígonos para tener un uso bajo de recursos y optimizar el procesamiento de estos en la aplicación dentro del dispositivo móvil.

6.2.2 Generación de UV del grafico

Estas UV son necesarias para la correcta visualización del material y textura del gráfico en el entorno digital.

6.2.3 Generación de materiales y/o texturas del grafico

Gracias a la herramienta 'Texture paint' de Blender, se pueden generar texturas, "dibujando" sobre las UV generadas del modelo.

6.2.4 Exportación del grafico en formato FBX

Este formato permite exportar solo los datos que se ocupen del modelo por lo que permite tener un modelo, con un bajo espacio de almacenamiento y que además tiene una gran compatibilidad con Unity. (Unity Technologies, 2022)

6.3 Creación de objetivos AR.

La creación de estas imágenes debe ser especial, puesto que estas imágenes deben cumplir con ciertas condiciones en medida de lo mejor posible (Wikitude, 2020):

- A. Ser de una dimensión cuadrada (1:1) para un mejor resultado.
- B. Deben tener contrastes altos para mejorar la fácil detección.
 - i. Imágenes con áreas de detalle distribuidas para mejorar el rastreo.
 - ii. Deben tener un menor espacio o volumen de un solo color para ayudar a la fácil detección.
 - iii. Deben tener muchas áreas de alto contraste, estructuras texturizadas para un mejor rastreo.
 - iv. No deben contener grandes áreas de textos, puesto que le hace difícil la detección.
 - v. No utilizar patrones repetitivos, puesto que el rastreo será más difícil.

Estas imágenes pueden ser revisadas por inteligencias artificiales o desde la propia herramienta de AR Foundation.

6.4 Subida de datos a la nube

Los datos deben ser subidos al servicio S3 de Amazon Web Services, estos datos son de diversos formatos:

6.4.1 Formato JSON

Este formato parecido a un bloc de notas, pero con cierta estructura, es utilizado para guardar los datos recopilados para su fácil compresión y comunicación de datos entre servidor y clientes.



6.4.2 Formato prefab

Este formato es especial y específico de Unity, donde se guardan los ajustes o la configuración de los objetos listos para utilizar en Unity. Estos contendrán el objeto ya sea 2D o 3D, así como las configuraciones necesarias del espacio 3D, componentes configurados y scripts que puedan utilizar, esto para la correcta función en la aplicación.

6.4.3 Formato PNG o JPG

Este formato se utilizará para mostrar al usuario imágenes en los puntos de interés, así como guardar la librería de objetivos AR.

6.5 Revisión de datos y funcionalidades de la aplicación

Debido a la funcionalidad principal del proyecto en la realidad aumentada, estas pruebas deben ser revisadas en físico en las distintas ubicaciones de interés para el correcto funcionamiento de los datos, gráficos tridimensionales y optimización de la aplicación en los distintos teléfonos inteligentes, uno con especificaciones altas y otro con especificaciones bajas, ambas con compatibilidad de ARCore.

VII. RECURSOS

7.1 Equipo

7.1.1 Laptop DELL G7 7790

Esta laptop se usa para la utilización de las herramientas necesarias para el desarrollo de la aplicación, así como la generación de la aplicación o la build de la aplicación. Con sistema operativo Windows 10, procesador Intel Core i5-10300H a 2.50Ghz, GPU NVIDIA GeForce RTX 2060 de 6GB y 16 GB de RAM.

7.1.2 Teléfono inteligente Xiaomi Redmi Note 9s

Se utiliza como dispositivo de pruebas de gama media, lanzado al mercado en 2020 que cuenta con un procesador Qualcomm 720G octa-core de 2.3GHz, GPU Adreno 618, sistema operativo Android Pie (9), 4 GB de memoria RAM y 64 GB de memoria interna, con cámara principal de 48 megapíxeles y frontal de 16 megapíxeles

7.1.3 Teléfono inteligente Xiaomi 11T Pro

Se utiliza como dispositivo de pruebas de gama media-alta, lanzado al mercado en 2021, cuenta con un procesador Qualcomm 888 octa-core de 2.84GHz, GPU Adreno 660, sistema operativo Android Pie 11, 12 GB de memoria RAM y 256 GB de memoria interna, con cámara principal de 108 megapíxeles y frontal de 16 megapíxeles.

7.2 Herramientas

7.2.1 GitHub: Sistema de control de versiones

Utilizado para alojar el proyecto en la nube a modo de backup en caso de perder datos en la memoria local del equipo de trabajo

7.2.2 Unity: Motor gráfico

Software utilizado como plataforma para el desarrollo de la aplicación móvil, la compilación final y de pruebas de la misma en conjunto con demás herramientas necesarias.

7.2.3 Visual Studio Community: Entorno de desarrollo

Esta herramienta se utiliza como entorno de desarrollo en su versión 2022 que permite trabajar con distintos lenguajes de programación, entre ellos C#.

7.2.4 Herramientas Android SDK

Estas herramientas y bibliotecas de desarrollo ayudan en el desarrollo de software para Android, estas herramientas permiten generar aplicaciones compiladas compatibles para esta plataforma móvil, viene incluido con Android Studio que es el entorno de desarrollo integrado (IDE) oficial de Google para el sistema operativo Android

7.2.5 Geospatial VPS Google

Esta herramienta se utiliza para alojar en los distintos puntos de interés objetos o información necesaria

7.2.6 Google Earth

Esta herramienta es utilizada como medio de obtención de coordenadas en el sistema WGS84, puesto que este sistema de coordenadas es necesario para disminuir el margen de error en metros entre el objeto digital y el punto de interés en la realidad.

7.2.7 Postman

Al usar almacenamientos en la nube es necesario hacer pruebas de solicitudes web REST API (GET, POST) con la finalidad del correcto funcionamiento de los datos, por ello se utiliza esta herramienta para estas pruebas.

7.2.8 S3 de Amazon Web Services

Este servicio es parte de los servicios de almacenamiento de AWS y es necesario para el alojamiento de datos y objetos a la nube, de modo que al requerir cambiar datos de la aplicación solo se modifique la información y no la aplicación en sí.

7.2.9 Figma

Es utilizado como herramienta en el diseño de prototipado de la interfaz gráfica del proyecto

7.2.10 Adobe Illustrator

Editor de gráficos vectorial, esta herramienta es utilizada para la creación de gráficos 2D vectoriales, como el apartado gráfico de la aplicación y objetivos AR para el proyecto

7.2.11 Blender

Software dedicado al modelo, animación y creación de gráficos tridimensionales. Esta herramienta será utilizada para la generación de objetos 3D con texturas necesarios para el proyecto.

7.3 Información

7.3.1 Información de puntos de interés

No están todos los datos en una base de datos y esta información no es privada, Se deberá hacer una selección y un registro manual de varios puntos de interés en la Facultad de Ingeniera y alrededores. Esta información es necesaria para mostrar objetos virtuales en la aplicación según el punto de interés.

7.3.2 Coordenadas de puntos de interés

No hay datos existentes en alguna base de datos sobre la mayoría de las coordenadas y estas coordenadas no son privadas, se pueden obtener a través de sistemas de navegación o de visualización de globo terráqueo virtual como Google Earth, por lo que se tendrá que hacer un registro de estas coordenadas según los puntos de interés seleccionados. Estos datos se deben tener como coordenadas de latitud, altitud y longitud en el sistema WGS84. Son importantes para la correcta visualización de los objetos virtuales en el espacio real.

7.3.3 Proyectos y trabajos realizados por la comunidad

Esta información se encuentra incompleta entre los distintos sitios web de la Universidad, no es de carácter privado, por lo que se podría hacer una investigación y recopilación para la selección manual de estos proyectos hechos por la comunidad universitaria, estos datos deben tener, nombre del proyecto, nombre del creador del proyecto, información de la misma, imagen de logo o del producto. Esta información es necesaria para efectos de la presentación de proyectos en el entorno virtual de la aplicación.

7.3.4 Directorios y contactos institucionales de docentes o administrativos

Existen datos de docentes, no es privada, las oficinas e instalaciones son públicas y estos datos están por partes en los distintos sitios web de la Universidad y existe un documento de nombres de profesores de la Facultad de Ingeniería con los respectivos correos institucionales que se comparte al inicio de los semestres en formato PDF por lo que se podría utilizar esa información en parte, pues solo faltaría obtener los datos de ubicación en los cubículos y/o números de extensión. Esta información es necesaria para una de las funciones de la aplicación

VIII. DESARROLLO DEL TRABAJO

8.1 Generación, inicialización y preparación del proyecto.

Crear proyectos requiere de un proceso de preproducción, producción y postproducción en el que gran parte del tiempo no se ven los procesos de gestión del proyecto que permiten guiar el proyecto a lo largo de su creación hasta su "release" de forma organizado. Es importante la creación, generación y configuración de los espacios necesarios para el proyecto, tanto para su guardado en nube, para no perder el proyecto y que pueda ser compartido fácilmente, como las extensiones necesarias para el proyecto y la correcta función de la aplicación, así como también del espacio de seguimiento de tareas y actividades.

Con ello se crea un espacio en Trello para mostrar las actividades en un modelo básico de Kanban con las listas necesarias para la gestión optimizada y organizada del proyecto. Se generan todas las tarjetas con las actividades necesarias para el proyecto.

Se genera un repositorio de GitHub con las configuraciones necesarias donde se almacenará el proyecto de Unity, proyecto que estará basado en realidad virtual con la SDK necesaria para ARCore Extensions.

Además de generar el espacio en el servidor S3 de AWS con la configuración necesaria para poder descargar los assets necesarios para la aplicación.

8.1.1 Trello

Trello es una herramienta visual basada en tableros donde puedes almacenar en "tarjetas" información o datos, esta herramienta visual permite a un equipo de trabajo gestionar cualquier tipo de proyecto y su flujo de trabajo, supervisando y manejando las tarjetas como tareas. Generalmente en un proyecto de gran magnitud con una metodología de Scrum existe una persona encargada de supervisar y manejar las tareas, así como de distribuir estas tareas entre los distintos departamentos según el cliente y el CEO o lo que el proyecto necesite. Y en cada reunión se fija un bloque de tareas por hacer y las tareas que se lograron terminar, dando reflexiones o alimentaciones sobre los triunfos o fracasos en las tareas.

Por motivos de interfaz de Trello, se debe crear una organización para almacenar "Tableros" en el que nos permitirá crear nuestro tablero con listas en base a la metodología Kanban.

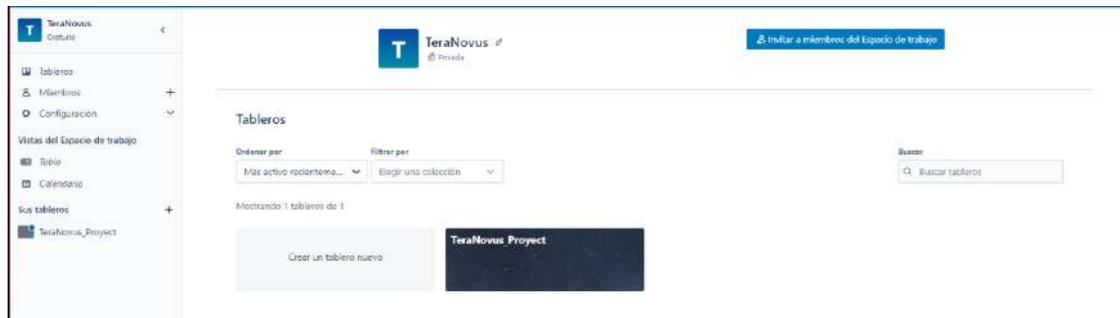


Imagen I: 8.1.1.1: Equipo de trabajo creado en Trello

Una vez creado el tablero, se procede a crear las listas correspondientes, un modelo básico de tablero Kanban, donde el flujo es continuo, es utilizar listas para las tareas pendientes por hacer, tareas que se están implementando y tareas finalizadas. Pero debido a la complejidad y funcionalidad del proyecto y del limitado personal de trabajo, se usarán varios tableros más desarrollados para un mejor seguimiento y orden visual de las tareas.

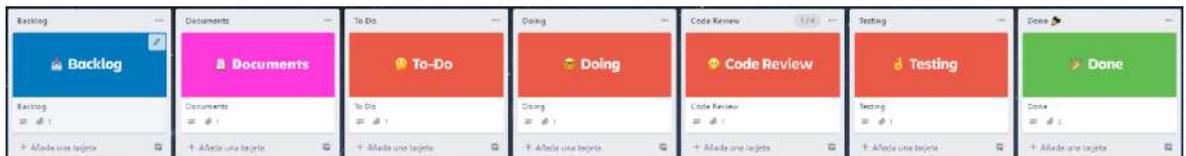


Imagen II: 8.1.1.2: Listas creadas en el tablero de Trello

De modo que "Backlog" queda para aquellas tareas "extras" que podrían agregarse o que harían falta por agregar al proyecto según haga falta por algún imprevisto como un "bug" o "fallo" encontrado en el transcurso del proyecto, estos se ordenan por prioridad en orden descendiente.

"Documents" está destinado para el guardado de la información del proyecto, como documentos, archivos Excel con los datos recopilados, imágenes o links. Esto generalmente se tiene en un espacio en la nube manejado por un equipo detrás, por lo que se crea esta lista para mantener controlada la información lo mejor posible en un mismo punto.

- I. "To-Do" está destinado a tareas de primera prioridad que faltan por hacer y que son sumamente importantes para el desarrollo de la aplicación.
- II. "Doing" está destinado para las tareas que se encuentran en proceso de implementación. En una metodología Scrum se escogen tareas o bloques de tareas por hacer en cada cierto periodo de tiempo.
- III. "Code Review" queda destinado para las tareas que se han terminado de implementar y que hace falta revisar su funcionalidad completa. Esto debido a que en equipos de trabajos generalmente el jefe del departamento es quien revisa estas tareas y el mismo autoriza colocar la tarea en "Doing" o "Testing" a "Done".
- IV. "Testing" debido a la dimensión y funcionalidades de la aplicación, es necesario hacer varias pruebas de la aplicación en el exterior. Por lo que se deja para las tareas que aún falta por probar su correcta funcionalidad.
- V. "Done" está destinado a las tareas que han sido terminadas.

Con estas listas creadas se procede a crear las tareas según las actividades marcadas del proyecto, estas tareas serán divididas en su mínima expresión de acción posible para una mejor organización.

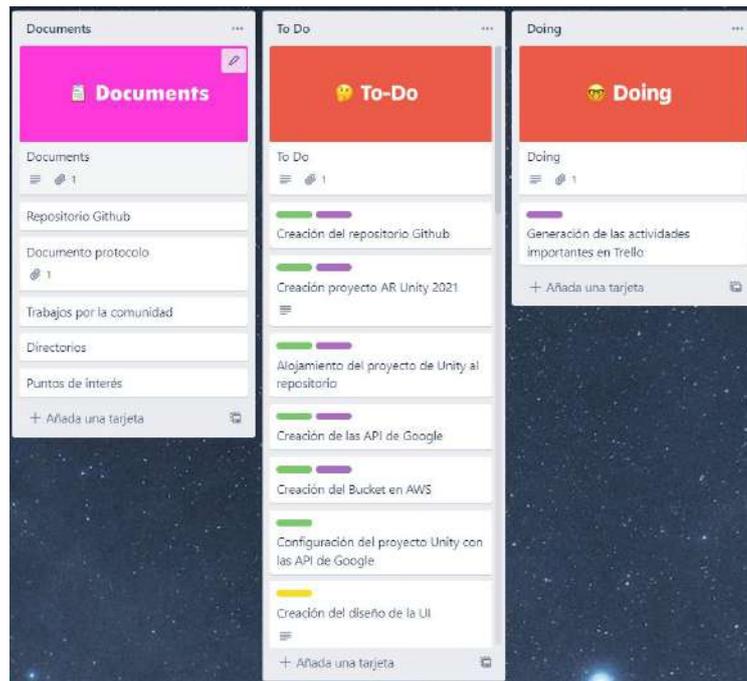


Imagen III: 8.1.1.3: Tarjetas creadas en las listas del tablero en Trello

8.1.2 GitHub

GitHub es una herramienta que permite alojar el código de las aplicaciones o herramientas de modo que no solo puedas descargar la aplicación, sino que además tiene la capacidad para leer y colaborar con el desarrollo de la aplicación, estos espacios donde se aloja el código de la aplicación se llaman "repositorios". GitHub utiliza el sistema de control de versiones en donde el código y las modificaciones del proyecto se ordenan en versiones nuevas, teniendo así copias de cada una de las versiones de la aplicación. Esta herramienta cuenta con dos versiones de interfaz, la versión web y la versión para desktop, en este caso se intentará crear el repositorio desde la versión desktop.

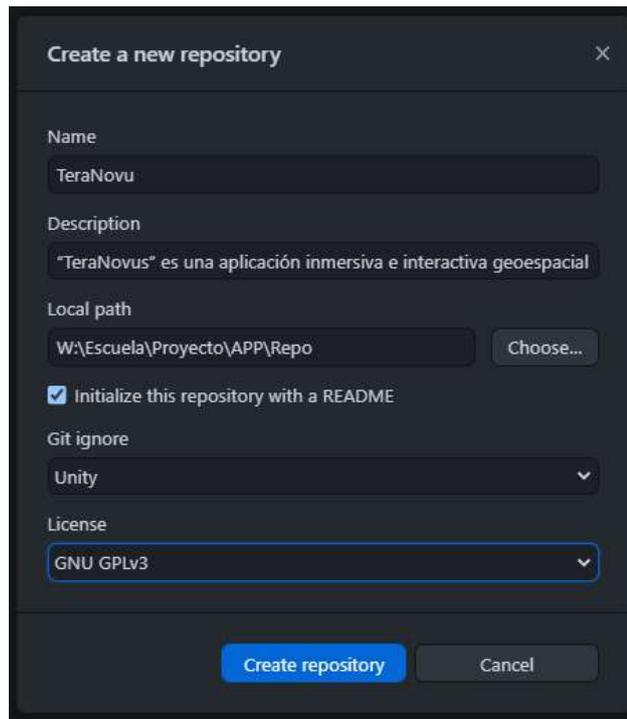


Imagen IV: 8.1.2.1: Ajustes básicos de la creación del repositorio en Github

Para su creación requiere de un nombre y como opción permite agregar un "Git ignore" que permite ignorar archivos no importantes como los archivos ".meta" que genera Unity donde guarda metadatos de cada archivo de modo que no se suben a la nube ahorrando espacio y tamaño de descarga y subida de datos.

8.1.3 Unity

Unity es un motor gráfico de videojuegos que permite desarrollar videojuegos o aplicaciones en gráficos 2D, 2.5D o 3D. Actualmente, Unity tiene proyectos básicos predeterminados o templates donde se le agregan a los proyectos extensiones que ayudan a desarrollar un tipo de proyecto en específico, en este caso se opta por el predeterminado de AR:

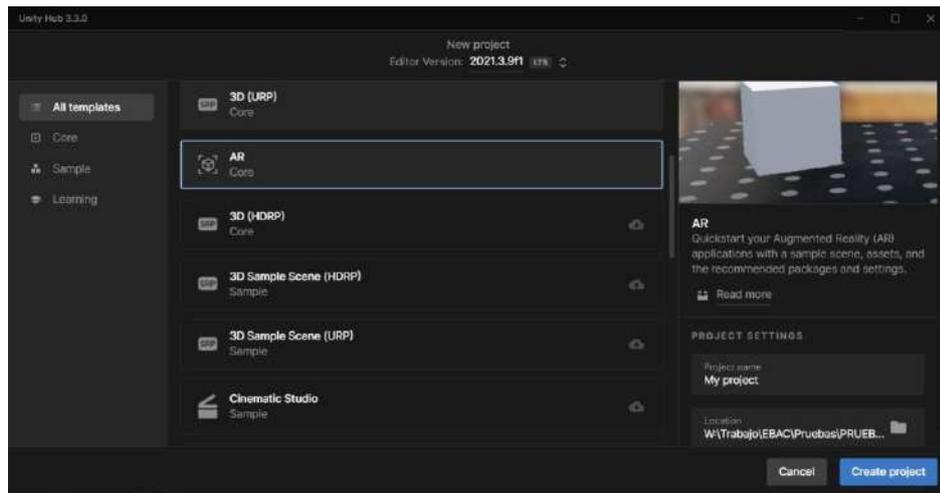


Imagen V: 8.1.3.1: template "AR" de Unity

Este tipo de proyecto permite tener los módulos o extensiones necesarios para la utilización de la realidad aumentada, así como los módulos correspondientes de AR Foundation y ARKit:

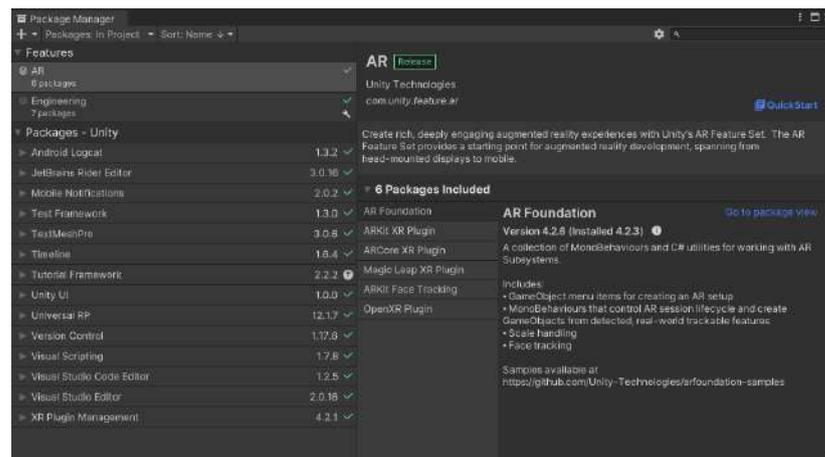


Imagen VI: 8.1.3.2: Paquetes instalados en el proyecto de Unity desde el panel "Package Manager"

Con estos cambios, ahora se procede configurar el proyecto para el desarrollo en la plataforma de Android, así como también la arquitectura y las versiones de Android, el cual es actualmente hasta la fecha (2022-2) las versiones de Android 7 (Nougat o API Level 24) en adelante son compatibles con ARCore.

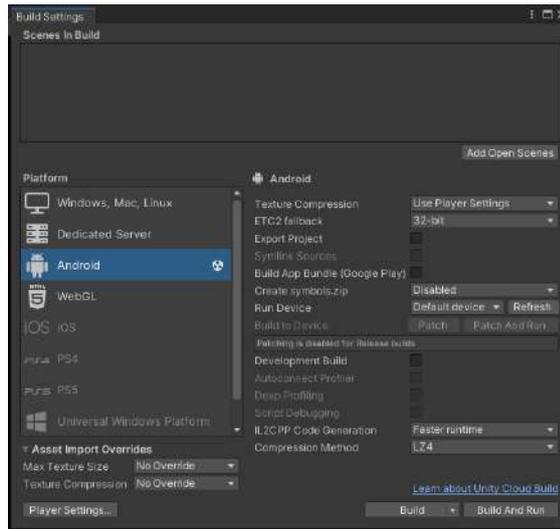


Imagen VII: 8.1.3.3: Panel "Build Settings" de Unity

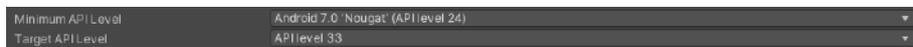


Imagen VIII: 8.1.3.4: Ajuste de API Level de Android soportados

Para poder utilizar funcionalidades extras de la extensión de Google ARCore, es necesario tener el paquete "ARCore Extensions" y configurar el paquete "ARCore Extensions" para el uso de "Geospatial" que ofrece Google:

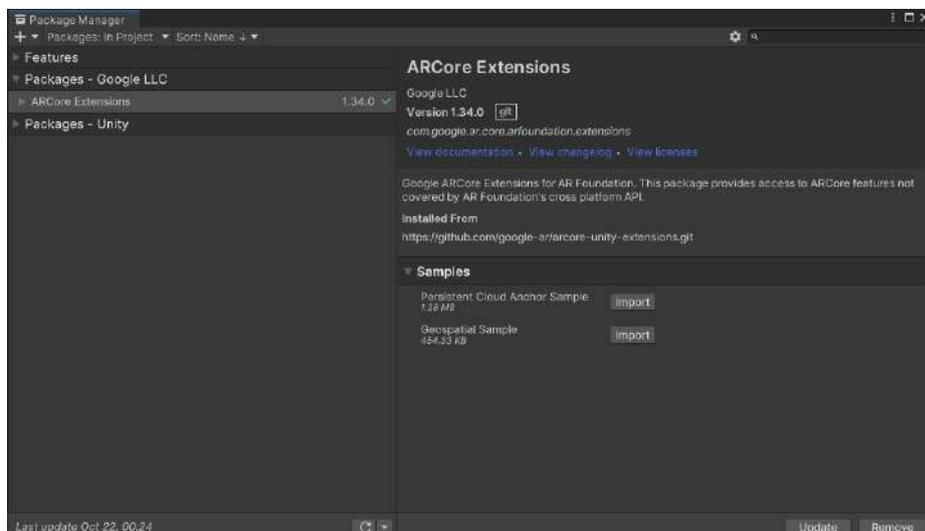


Imagen IX: 8.1.3.5: Paquete "ARCore Extensions" instalado en el proyecto de Unity

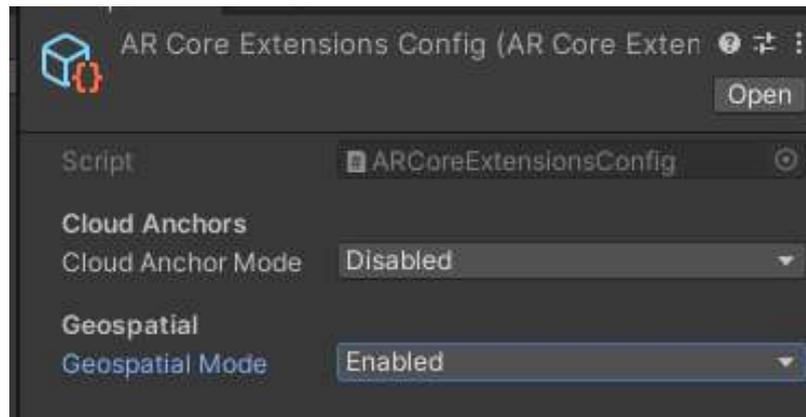


Imagen X: 8.1.3.6: Ajustes de "ARCore Extensions Config" disponibles

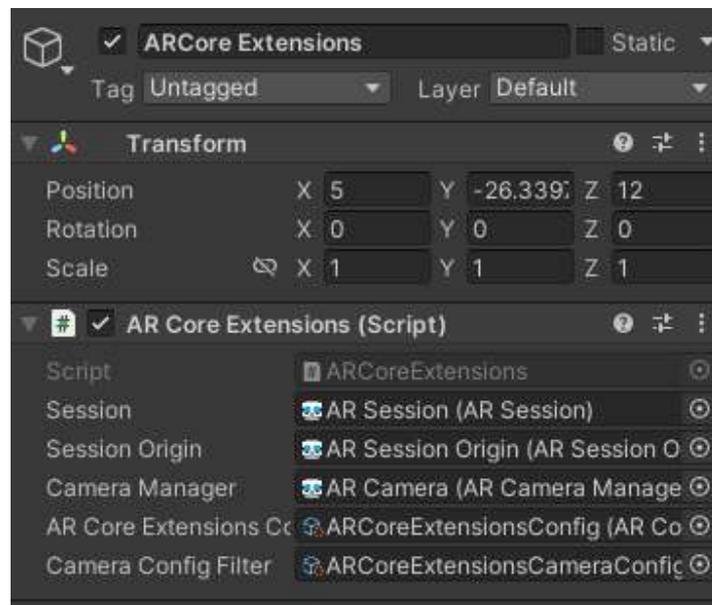


Imagen XI: 8.1.3.7: Ajustes componente script "ARCore Extensions"

8.1.4 API Google

Para utilizar la API correspondiente de Google para el correcto funcionamiento del proyecto y la aplicación, es necesario habilitar la API "ARCore" de Google en un proyecto dentro de la consola de desarrollador de Google y agregar la API que proporciona Google al proyecto de Unity.

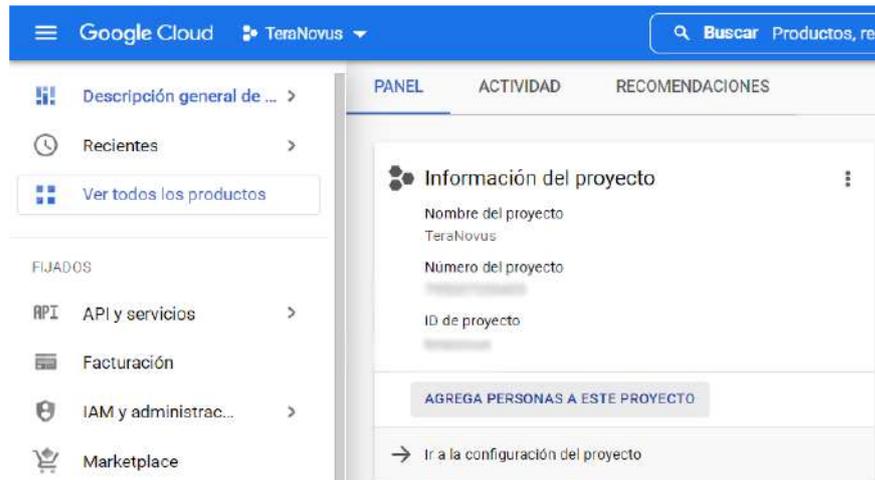


Imagen XII: 8.1.4.1: Proyecto "TeraNovus" en la consola de Google Cloud



Imagen XIII: 8.1.4.2: Detalles del API "ARCore API"

Claves de API



Imagen XIV: 8.1.4.3: Pantalla de Claves API del proyecto.

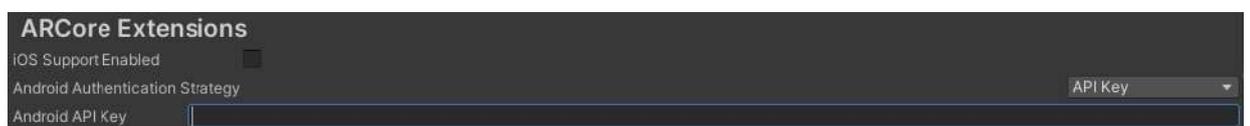


Imagen XV: 8.1.4.4: Ajustes de "ARCore Extensions" en "Project Settings" dentro de Unity

Con esto queda configurado el proyecto de Unity, por lo que se sube a la nube (GitHub) y se procede a crear el espacio respectivo en la S3 de AWS:

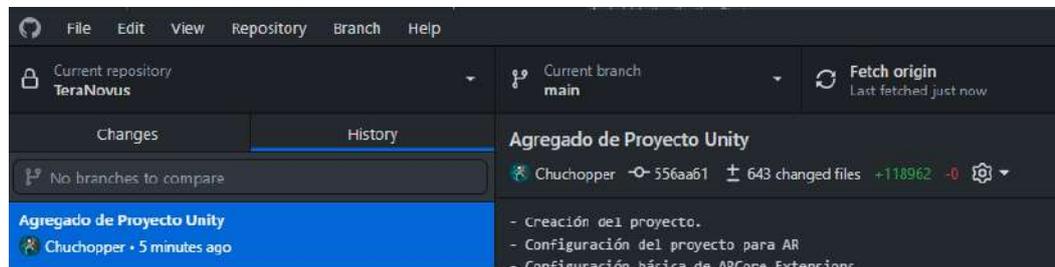


Imagen XVI: 8.1.4.5: Commit inicial en Github Desktop

8.1.5 AWS: S3

AWS es un sistema varios servicios en la nube que ofrece Amazon, uno de estos servicios es "S3" que es un servicio de almacenamiento de objetos que ofrece escalabilidad, disponibilidad y seguridad de datos a diversas aplicaciones como datos de sitios web, aplicaciones móviles, copias de seguridad, dispositivos IoT y análisis de big data. Para esta aplicación será necesario para el almacenamiento de los gráficos 3D, información de puntos de interés, contactos institucionales y etc.

AWS tiene presenta distintas capas como el Desarrollador, Empresarial y la capa Gratuita que nos ofrece 5 GB de almacenamiento en S3 en la clase de almacenamiento S3 Standard con 20 000 solicitudes GET, 2000 solicitudes PUT, POST y 100 GB de transferencia de datos de salida al mes.

Para ingresar a S3 es necesario tener una cuenta de AWS y entrar a la consola de AWS para acceder a S3.

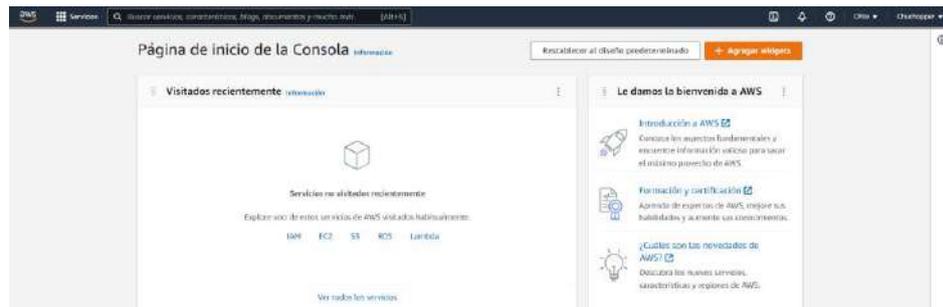


Imagen XVII: 8.1.5.1: Página inicio de la Consola AWS

S3 funciona a modo de "buckets" que estos a su vez funcionan como "unidades" de almacenamiento en la nube, estos se alojan en un servidor de AWS situados en algunos puntos del mundo, los más cercanos a México actualmente (2022) son los servidores de Estados Unidos:



Imagen XVIII: 8.1.5.2: Servidores AWS para el servicio S3 disponibles (2022)

Una vez configurado y creado el bucket para la aplicación en S3, se dejan las carpetas preparadas:



Imagen XIX: 8.1.5.3: Página de Inicio de la bucket de "TeraNovus_Unity"

8.2 Desarrollo del diseño gráfico

El diseño gráfico de la aplicación es uno de los puntos importantes de una aplicación, pues a través de ella podemos guiar al usuario a usar las funcionalidades de una aplicación de forma visual e intuitiva. Sin embargo, para generar un diseño completo de forma en el menor tiempo y de forma optimizada se desarrolla de ante mano un diagrama con las acciones y/o eventos que ocurrirán de forma básica en la aplicación. Esto permitirá generar las pantallas y paneles de aquellos procesos expuestos en el diagrama, además de darle un orden más adecuado de cómo serán las acciones y/o eventos que se generen con el resultado de la interacción del usuario y las funcionalidades de la aplicación.

Con la generación de un diagrama de todos los procesos de la aplicación se procede a generar el "borrador" visual de las pantallas que contendrá esta aplicación, así como su posible distribución de los botones o elementos UI interactivos y no interactivos por el usuario, sus posibles formas y ubicaciones de cada elemento visual en las pantallas.

Aunado a esto se hace una breve investigación de los colores en el mercado según la psicología del color, obteniendo una paleta de colores basándonos en los colores verde y azul, y se consigue generar logos "abstractos" en distintas versiones, ya sea únicamente con el logo o en mezcla con el nombre de la aplicación.

En conjunto con el diagrama, el wireframe, el color y el logo de la aplicación, se recrea con los colores y diseños definidos en las pantallas, así como también los paneles definidos previamente con la finalidad de generar estas pantallas en distintas escenas dentro de Unity con las funcionales necesarias así como los elementos UI interactivos y no interactivos, además de los paneles necesarios como el panel de diálogos, panel de carga y el panel de transición que permitirán mostrar al usuario mensajes o/y ocultar procesos que ocurren mientras se cargan o mueven datos.

8.2.1 Diagrama de las funciones básicas

Antes de diseñar el apartado visual de la aplicación se hace un diagrama de las funciones, así como el proceso de acciones que tendrá la aplicación, para esto se utilizó "Drawio" una herramienta que ayuda a generar diagramas.

Sobre el diagrama generado se han colocado distintos elementos e identificadores de las funciones básicas, además de los procesos según las etapas de la aplicación.

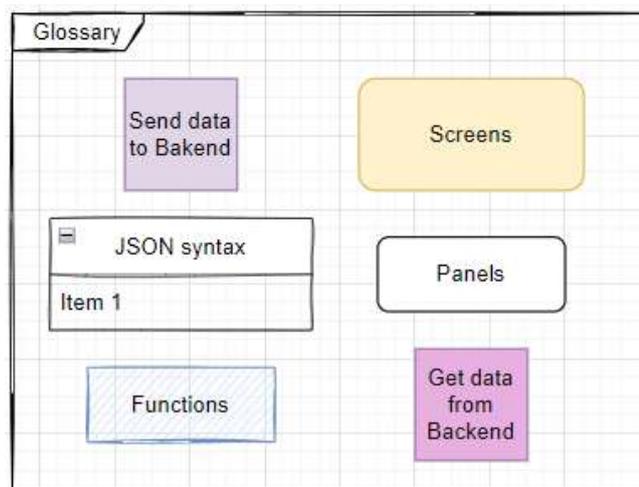


Imagen XX: 8.2.1.1: Glosario diagrama de acciones básicas del usuario en la aplicación

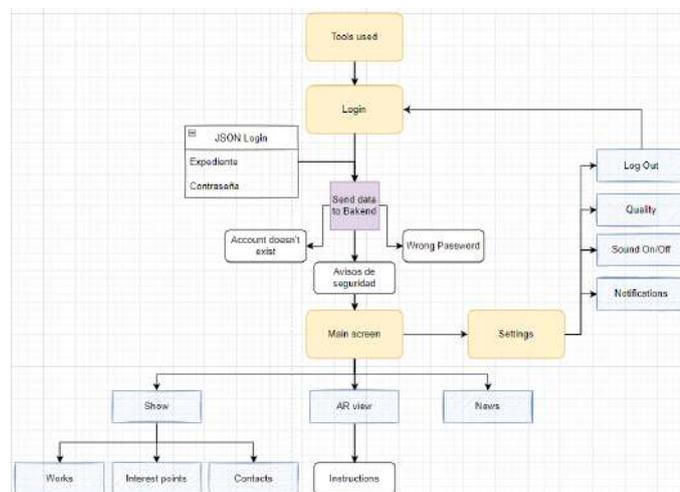


Imagen XXI: 8.2.1.2: Diagrama de acciones básicas del usuario en la aplicación

Estas indicaciones comienzan con el apartado de las pantallas "splash" o las pantallas de inicio, que son la primera pantalla visible para el usuario cuando se inicia la aplicación como una pantalla de bienvenida. Por defecto, Unity coloca una pantalla de inicio cuando se trabaja con la licencia básica y se puede omitir si se tiene una licencia "pro", sin embargo, además de esta pantalla se deben agregar el uso de las APIs de Google.

Tras estas pantallas se procede a la pantalla de "Login" en el cual en un futuro se podría utilizar esta pantalla para que el usuario pueda utilizar funciones extras para la comunidad estudiantil que podrían agregarse en un futuro, claro, también se prevé el acceso a la aplicación como invitado para usuarios que no pertenezcan a la comunidad estudiantil.

Según la información colocada por el usuario podría mostrarse paneles que indiquen al usuario si tiene la información incorrecta o si la cuenta no existe, en caso de ser correcto se redirige a la pantalla principal.

La pantalla principal está destinada para dar al usuario la posibilidad de elegir entre distintas opciones:

- I. "Ver trabajos" / "Ver puntos de interés" / "Ver directorios": donde el usuario podrá ver la información que necesite sin necesidad del uso del sistema AR.
- II. "Vista AR": donde el usuario podrá entrar al sistema AR que le permita utilizar las funciones básicas de la aplicación, esto después de mostrar una serie de instrucciones.
- III. Como posibilidad futura, "Noticias": Donde el usuario podrá ver noticias respecto a la aplicación o sobre la Universidad.

La aplicación tendrá la posibilidad de funcionar dentro del sistema AR como fuera de este para incluir y permitir a usuarios que no tengan teléfonos inteligentes que puedan soportar esta tecnología AR y poder acceder a estas funcionalidades de la aplicación fuera del sistema AR, por lo que se crea un diagrama con las funciones que ocurren durante el sistema AR habilitado o deshabilitado:

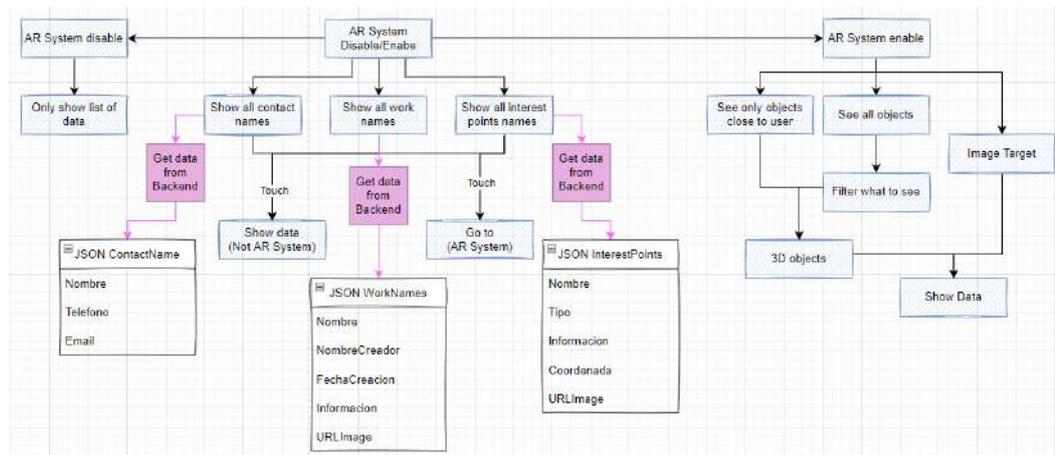


Imagen XXII: 8.2.1.3: Diagrama de eventos del sistema AR habilitado y deshabilitado

En este diagrama creado se especifica como serán estas funciones según el sistema elegido, en caso de que el usuario haya elegido las opciones de solo ver la información, al usuario se le permitirá ver los contenidos de trabajos, puntos de interés y el directorio.

En el caso de elegir el sistema AR, el usuario podrá elegir si ver solo objetos cercanos al usuario, o ver todos los iconos de los puntos de interés e incluso filtrar los puntos de interés por tipo. Según la decisión del usuario será la información que se muestre. Además de tener sistema de información integrado.

En ambos casos se podrán ver estas listas de información donde al estar habilitado el sistema AR y elegir una de las opciones de estas listas podrá ser dirigido el usuario hasta ese punto de interés desde la ubicación en el que este se encuentre y al estar deshabilitado solo podrá ver la información estas opciones de las listas.

8.2.2 Diseño del apartado gráfico: Wireframe y diseño básico

Con el diagrama de las funciones ya se tiene una noción básica de qué apartados gráficos, pantallas o elementos de la interfaz gráfica se procede a crear el diseño en Figma.

Figma es una herramienta visual de gráficos vectoriales que permite generar o diseñar prototipos, wireframes o interfaces gráficas de páginas web o

aplicaciones móviles. Esta herramienta ya tiene tamaños de "pantalla" por defecto:

Phone	
iPhone 14	390 × 844
iPhone 14 Pro	393 × 852
iPhone 14 Plus	428 × 926
iPhone 14 Pro Max	430 × 932
iPhone 13 Pro Max	428 × 926
iPhone 13 / 13 Pro	390 × 844
iPhone 13 mini	375 × 812
iPhone SE	320 × 568
iPhone 8 Plus	414 × 736
iPhone 8	375 × 667
Android Small	360 × 640
Android Large	360 × 800

Imagen XXIII: 8.2.2.1: Opciones de formatos de pantallas verticales por defecto en Figma (2022)

Lo que permite tener una medida más correcta, en este proyecto se optara por una resolución "Android Large" de 360 * 800 como base, ya que esta resolución es una resolución de los teléfonos inteligentes modernos actualmente más utilizada, puesto que en estos últimos años se están viendo tendencias de pantallas alargadas en marcas como Samsung, Motorola, LG, Xiaomi, Huawei o Sony, utilizando un aspecto de 20:9 debido a la llegada del "Notch" (Camara integrada directamente en la pantalla) de la mano de Apple. En este caso, 360 * 800 resulta de un aspecto de 20:9.

Con esta base se comienza a diseñar las pantallas tal como se definió en el diagrama en un formato de parecido al "Wireframe" donde no solo se diseñará el acomodo u organización de los elementos visuales que conformaran la interfaz gráfica, sino que además se definirá el diseño básico de estos elementos visuales, claro que al ser un "wireframe" por ahora no se definen paletas de colores o el logotipo necesario para la aplicación.

Se plantea esta base para la interfaz de la pantalla de "Login", pantalla en la que el usuario iniciara sesión como miembro de la comunidad universitaria

(Docentes, administrativos o estudiantes) o como invitado, tal como en los sistemas del "portal UAQ" o en el Wi-Fi que proporciona la universidad, estas pantallas se agregan con los posibles mensajes al usuario según el error que se pueda encontrar. A este también se le agrega debajo un aviso del uso de las API de Google.



Imagen XXIV: 8.2.2.2: Diseño de la pantalla Inicial (login) y sus posibles paneles de diálogos

La pantalla del menú principal, debe contar con el acceso "intuitivo" a las funcionalidades de la aplicación así como el acceso rápido a los ajustes de la aplicación en cuanto al sistema AR. Se plantea dejar un espacio para el icono de perfil del usuario y del nombre del usuario tal como se puede ver en el apartado de información del sistema del portal UAQ:



Imagen XXV: 8.2.2.3: Información de estudiante a través del portal de la Universidad Autónoma de Querétaro



Imagen XXVI: 8.2.2.4: Diseño de la pantalla de menú principal (Main Menu) y panel de ajustes

Se agregará un icono que permita, además del nombre, conocer a qué función está accediendo o a qué función desea acceder el usuario de forma rápida e intuitiva. En los ajustes se agregará la posibilidad a futuro de cambiar la calidad de los gráficos 3D, el sonido de la aplicación, así como la posibilidad de recibir noticias igual a futuro en forma de notificaciones. Además de un botón para salir de la cuenta, permitiendo al usuario poder ingresar con la cuenta deseada.

En cuanto a las pantallas de "información" donde el usuario podrá ver el directorio, las ubicaciones, así como los trabajos de la comunidad, estos contarán con la misma base. Esta base será una lista de elementos de contactos, ubicaciones o trabajos y de una sección para poder filtrar los elementos por letra del abecedario, por facultad o por tipo.

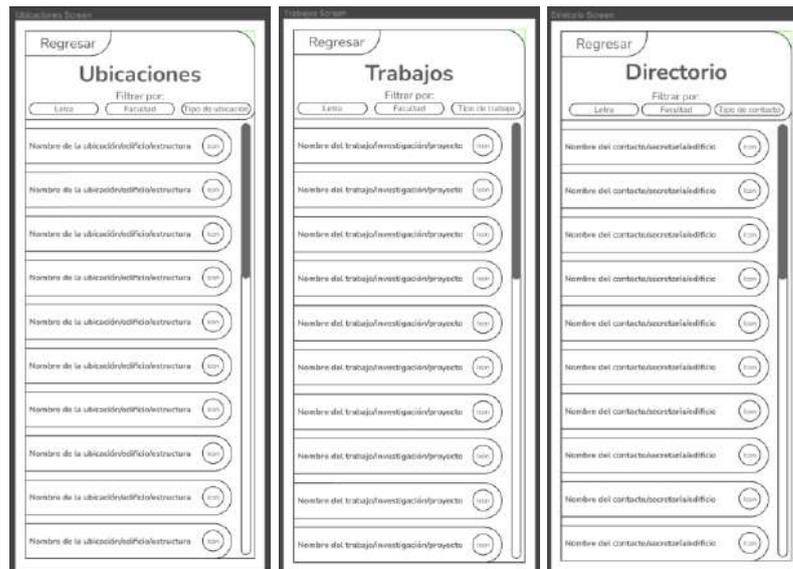


Imagen XXVII: 8.2.2.5: Diseño de la pantalla de información (Information)

También se crea un panel dentro de la pantalla de "información" para mostrar los detalles del elemento seleccionado por el usuario, la información que se mostrará dependerá del tipo de elemento (contacto, ubicación o trabajo), además de la información que este elemento contenga.



Imagen XXVIII: 8.2.2.6: Diseño de los paneles de información extra

Dentro de esta información, habrá elementos que serán "interactivos" como es el caso de "Teléfono - extensión" que al tocar el elemento dirigirá al usuario a su aplicación de llamadas que el usuario tenga instalado en su dispositivo móvil, o del "Correo electrónico" que al tocarlo se le dirigirá al usuario a su aplicación de correo e incluso tocar el elemento "Ubicación" fuera del sistema AR le dirigirá a su aplicación de mapas que el usuario tenga instalada.

En el caso de la pantalla que contiene las funcionalidades principales en el sistema AR, esta pantalla será "minimalista" donde se contara con 3 botones en la parte inferior a modo de "Navigation bar" que presenta Android. Estos 3 botones tienen las siguientes funcionalidades:

- I. "Regresar": permite al usuario salir del sistema AR y volver al menú principal.
- II. "Ir a": permite al usuario elegir una ubicación a la que el usuario desee ir, pudiendo elegir entre las ubicaciones, trabajos y contactos que contengan una ubicación en las instalaciones de la universidad.
- III. "Ver solo": Le permite al usuario elegir entre las distintas opciones de visualización de datos en la pantalla, teniendo de elecciones la posibilidad de ver todos los elementos, lo que podría hacer más lento la aplicación si el dispositivo móvil sobre el que se ejecuta la aplicación no es lo suficiente potente, también puede elegir ver solo los cercanos al usuario o filtrar si solo desea ver uno o todos los tipos de elementos (ubicaciones y trabajos).

La pantalla "Ir a" tiene la misma funcionalidad que las pantallas de información, a diferencia de que al ser seleccionado un elemento este indicara un botón "mostrar" en la parte inferior del panel que permite ver la ubicación en pantalla y ser guiado desde el punto en que se encuentre el usuario hasta al punto de ubicación seleccionado.



Imagen XXIX: 8.2.2.7: Diseño de pantalla y paneles de la vista AR (ARView)

8.2.3 Diseño del apartado gráfico: Color y logotipo

El color es una parte fundamental del proyecto, puesto que es el color afecta al comportamiento humano según la "psicología del color" y se espera que el usuario tenga paz, tranquilidad y comodidad al utilizar esta aplicación.

Motivo por el cual se hace una investigación sobre los colores en el mercado y la imagen, donde se buscan entre los distintos colores, los colores que puedan generar paz, tranquilidad y comodidad y que a su vez estos colores puedan ser aplicados bajo una regla de armonía de color, entre estos colores, el verde y el azul tienen significados positivos lo más cercanos a los deseados:

- I. Verde: naturaleza, salud, paz, orgánico, equilibrio, restauración, conciencia, armonía.
- II. Azul: Inteligencia, confianza, seguridad, serenidad, comunicación, eficiencia, lógica, reflexión, calma.

(Canva, s.f.)

Sin contar que estos colores cumplen con la base de los colores de la tierra, además de ser el color azul el color representativo de la universidad.

Además, según la página encontrada y utilizada para la investigación, contiene una gráfica y comenta:

"Las personas toman una decisión dentro de los 90 segundos de sus interacciones iniciales con personas o productos. Alrededor del 62 al 90 por ciento de la evaluación se basa únicamente en los colores. Por tanto, el uso prudente de los colores puede contribuir no sólo a diferenciar los productos de los competidores, sino también influir en los estados de ánimos y sentimiento" (Singh, 2006)

Por ende, cuando un consumidor se acerca a un producto o servicio que tiene su color favorito, la mayor parte de la decisión de compra ya está hecha.

COLORES FAVORITOS ENTRE HOMBRES Y MUJERES



Imagen XXX: 8.2.3.1: Joe Hallock's "Colour Assignments"

Y según la gráfica, los colores azul y verde son también dos de los mejores colores que logran una mayor persuasión en hombres y mujeres. Mediante varias investigaciones se definen estos tonos azules y verdes para la base del color de la aplicación y el logo.

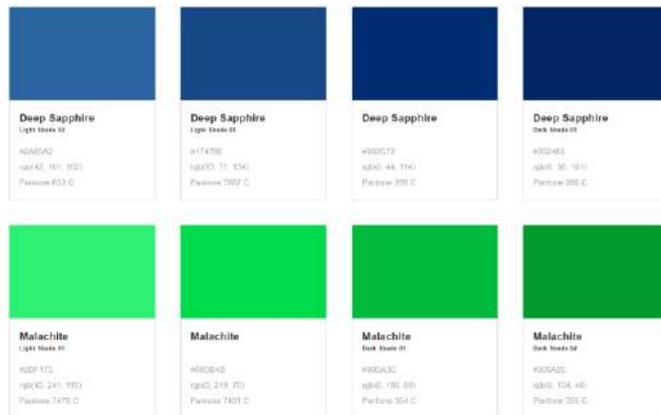


Imagen XXXI: 8.2.3.2: Paleta de colores

Para el diseño del logo se basó en el nombre "TeraNovus" que deriva del latín, "Tierra" y "Nueva" debido a que se explora la tierra de una forma nueva. Utilizando de base una tierra como icono de la aplicación utilizando estos tonos.



Imagen XXXII: 8.2.3.3: Diseño del logo de la aplicación

Se mantiene un icono del mundo partido por la mitad, donde una mitad muestra el mundo y la otra un mundo "tecnológico" puesto que la realidad aumentada utiliza el mismo espacio real para superponer objetos y medios digitales.

8.2.4 Diseño del apartado gráfico: Diseño final de la aplicación.

Utilizando de base el wireframe o diseño básico generado en conjunto con el color y logotipo, se generan las pantallas con el diseño final.

Dejando en la parte inferior la política de privacidad pedida por Google al utilizar las APIs proporcionados por Google LCC, donde el usuario al tocar las partes del texto subrayado se le enviara al link proporcionado por Google. Se mantiene además en esta pantalla contrastes o acentuación en zonas donde se necesita que el usuario vea de primera mano o se concentre más en ellas, como es el caso de los inputs de texto "Expediente" o "Contraseña" incluso el botón "Aceptar" y "X" para cerrar panel de diálogo expuestos en verde, además de los dos botones de inicio "Entrar" y "Entrar como invitado" en un marco llamativo.



Imagen XXXIII: 8.2.4.1: Diseño final pantalla inicial

En cuanto al menú principal, se mantiene en los botones el texto e imágenes referentes a la función del botón, además de un texto para el nombre de la persona acreedora del expediente con el que se ingrese junto a un espacio círculo para colocar la foto de la misma.

Por la finalidad y funcionalidad principal de la aplicación, se agrega en grande el botón "Vista AR" y al inicio, dejando los demás botones debajo de este,

cerrando con "Noticias" con un botón grande para mantener una simetría, estos botones están englobados en un marco para delimitar el área interactiva dándoles más enfoque en un fondo degradado negro.



Imagen XXXIV: 8.2.4.2: Diseño final pantalla menú principal

En cuanto a las pantallas de información (Directorio, ubicaciones, trabajos) se mantiene iconos con referencia al tipo del elemento.



Imagen XXXV: 8.2.4.3: Diseño final pantalla información

Para la pantalla del sistema AR, los botones o cuadros de diálogo que se mostraran cuando el usuario se encuentre cerca de un elemento o lea un objetivo AR, se mantienen en color azul fuerte, puesto que los tonos verdes y el azul más claro podrá hacer que el usuario tenga dificultades de diferenciar la el entorno de los botones, puesto que la universidad utiliza tonos azules de claros a medios en los edificios y al menos en el campus C.U. está rodeado en gran parte de zonas verdes y por su buen mantenimiento se mantiene gran parte del año verde, agregando que el cielo es azul gran parte del tiempo.



Imagen XXXVI: 8.2.4.4: Diseño final pantalla vista AR

8.2.5 Diseño del apartado gráfico: Generación de las pantallas en Unity

Para generar estas pantallas en Unity, primero es necesario generar una escena por cada pantalla y después es importante definir el tamaño base que tendrá de pantalla, así como su tipo de escalado o adaptación a las distintas pantallas y el orden de escalado.



Imagen XXXVII: 8.2.5.1: Ajustes componente "Canvas Scaler" en Unity

Tras esto se comienza a colocar en las pantallas los iconos, texturas, imágenes necesarias según el tipo de elemento UI que se requiera. Se agregan elementos UI de tipo: botón como activadores de eventos; de tipo "inputField"

como guardado de caracteres dados por el usuario; de tipo desplega- bles para mostrar distintas opciones al usuario.



Imagen XXXVIII: 8.2.5.2: UI generado en Unity

Es necesario las pantallas extras de carga o de transición que permitan ocultar visualmente al usuario los procesos de los eventos de descarga y carga de contenido dentro de la aplicación. Además de estas pantallas, es necesario agregar una comprobación de compatibilidad del dispositivo del usuario con los sistemas AR, desactivando las funciones AR de la aplicación en caso del que dispositivo no sea completamente compatible con el sistema AR.



Imagen XXXIX: 8.2.5.3: UI de pantallas de carga (izquierda) y transición (derecha)

Si se verifica el usuario o no para entrar, el nombre e icono del usuario se mostrará, dándole la bienvenida. El panel "Ajustes" se configura en la misma escena, este permitirá al usuario cambiar los ajustes de la aplicación, además de permitirle regresar a la pantalla de inicio de sesión



Imagen XL: 8.2.5.4: UI pantalla menú principal

Si el dispositivo no es compatible con el sistema AR entonces el botón "Vista AR" avisará al usuario que no podrá utilizar esta función debido a la incompatibilidad con el sistema AR, pero de igual manera se permitirá al usuario ver los directorios, ubicaciones y trabajos mostrando una pequeña transición dejada para las cargas de datos y para un cambio visual más suave al ojo.



Imagen XLI: 8.2.5.5: Diálogo de incompatibilidad con las funciones AR

Según el botón elegido ("Directorio", "Ubicaciones", "Trabajos") Se dirigirá al usuario a la pantalla "Información" que le mostrara la información pedida por el usuario, donde se mostrara el título de la información pedida y al cargar la información de todos los elementos se agregaran los botones pertinentes con su información que permitirán cargar los datos al panel de "información extra", que mostrara (si es que tiene la información): nombre, teléfono, correo electrónico, horario, detalles del elemento y la ubicación.

Estos datos pueden ser interactivos, incluso cuando no se tenga activado el sistema AR, cuando se interactúa con estos datos: Teléfono, permitirá enviar al usuario a su aplicación de llamadas que el dispositivo tenga instalado con el número de teléfono seleccionado; Correo electrónico, permitirá enviar al usuario a su aplicación de correo instalado en su dispositivo con el correo seleccionado; y Ubicación, permitirá enviar al usuario a la aplicación "Google Maps" si este se encuentra instalado en su dispositivo con la dirección del punto seleccionado.



Imagen XLII: 8.2.5.6: UI pantalla Información y panel información extra

Al ser una aplicación que puede contener una gran cantidad de información si se agregaran todos los puntos de interés de C.U. se mantiene un filtro que permita decidir por facultad y el tipo de punto (estructura, edificios, salones, persona, trabajo).



Imagen XLIII: 8.2.5.7: Filtro por "Facultad" (Izquierda) y por "Tipo" (Derecha)

Por último se genera la pantalla de "VistaAR" con la misma base de la pantalla "información" si se selecciona el botón "Ir a", con la diferencia de mostrar todos los elementos disponibles, además de tener un botón "Mostrar" cuando se está en el panel "información extra".



Imagen XLIV: 8.2.5.8: UI pantalla Vista AR

8.3 Objetos en una coordenada global dentro del sistema

AR

Los objetos que se mostraran en una coordenada específica global a través de esta aplicación es la mecánica principal de este proyecto, ya que a través del sistema AR se mostraran estos, estos objetos que se mostraran serán objetos 3D que permita identificar el tipo de punto en el que se encuentran, además de un nombre identificador del lugar y la información que este tendrá, de modo que se podrán ver varios puntos conforme se camina o desplaza por el espacio real.

Estos objetos son generados mediante unas coordenadas WGS84 en longitud y latitud, mientras en altitud serán obtenidos mediante otro tipo de coordenadas. Las coordenadas del usuario serán leídas tras conocer el estatus que Google reciba y mande del dispositivo y según los datos disponibles que pueda mandar este dispositivo, si el usuario está cerca de las coordenadas de una ubicación, entonces los objetos se mostrarán en el dispositivo.

Según los estados del estatus que Google dará sobre el estado del dispositivo, será el resultado de las acciones y se comenzara a determinar la posición del dispositivo en el mundo para luego tener una posición en las coordenadas globales virtuales y ser posicionados virtualmente dentro de la aplicación, tras ello se colocarán los objetos según la lista de coordenadas disponibles en un archivo JSON.

Dichos objetos mostrarán una información propia del lugar como su nombre del lugar, horario, correo de contacto, teléfono del lugar o de contacto, detalles importantes del lugar o descripción del lugar, además de un agregado visual que es el modelo 3D característico del lugar y de la facultad en la que se encuentra.

8.3.1 Lógica de generación de objetos en coordenadas globales

Para el funcionamiento de la aplicación en AR con objetos en coordenadas globales es necesario primero comprobar si el modo geoespacial es compatible, para ello es necesario acceder al "AREarthManager", script que contiene o provee localización en coordenadas relativas a la tierra, que además contiene una función para saber si es compatible o no con esta herramienta el dispositivo e incluso saber si se ha podido activar la sesión correctamente.

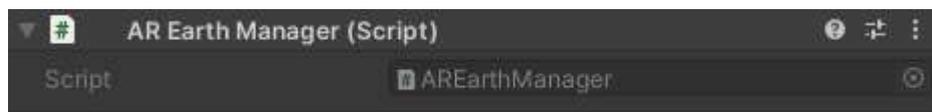


Imagen XLV: 8.3.1.1: Componente "AREarthManager"

Tras saber el estatus de "GeospatialMode" que nos dice la compatibilidad con esta función, se puede continuar los eventos necesarios: en caso de que sea "incompatible" ya sea porque el mismo dispositivo sea adecuado o no para la realidad aumentada, también se deberá a la capacidad de los sensores, se le avisara al usuario; en caso de ser compatible se continúa con los siguientes procesos necesarios.

Sin embargo, hay un estado más que puede tener este estatus, el estatus desconocido (Unknown), este estatus puede ocurrir de varias formas, entre ellas, porque se sigue determinando la compatibilidad o por la falta de internet, por tener desactivado el GPS en el dispositivo, por la alta latencia de conexión entre la red y el servicio de Google y entre otros motivos, por lo que, si el estatus cae sobre este resultado, se reintentará conocer la compatibilidad del servicio.

Este servicio utilizará una combinación del sistema de posicionamiento virtual (VPS) y el GPS del dispositivo para determinar poses geoespaciales, por lo que es completamente necesario tener el GPS activado, al menos en el proceso de generación de los objetos, el propio servicio determinara como utilizar estos dos sistemas, según sea conveniente, ya sea que no exista en el ambiente construcciones u obstáculos para que el sistema VPS no funcione correctamente o que el GPS no esté ofreciendo una alta precisión. Sin

embargo, la precisión proporcionada por estos dos sistemas puede variar de hasta alrededor de 5 metros.

Para poder colocar objetos en el espacio 3D digital, es necesario confirmar una vez más dentro del "AREarthManager" si el estado del "tracking" está en funcionamiento. Puesto que es necesario hacer una lectura del entorno que permita conocer los planos del suelo, edificios, paredes o algún otro obstáculo.

```
if (earthManager.EarthTrackingState == TrackingState.Tracking) ...  
else if (earthManager.EarthTrackingState == TrackingState.None) ...
```

Imagen XLVI: 8.3.1.2: Chequeo del estatus del "Tracking"

Si este estatus es correcto, entonces se requiere generar un "anchor" o ancla, que contendrá la ubicación o posición 3D relativa a la tierra. Este proceso requiere que las coordenadas de las ubicaciones globales de latitud y longitud estén especificadas en el sistema de referencia de coordenadas WGS 84 geodésicas (World Geodetic System 1984). La limitación de este servicio es colocar anclas en el polo norte, donde será incompatible.

Para esto, la misma documentación que ofrece Google sobre este servicio, en el que ofrece un instructivo sobre cómo obtener estas coordenadas, habla sobre Google Maps, herramienta que utiliza este mismo sistema de coordenadas WGS84.

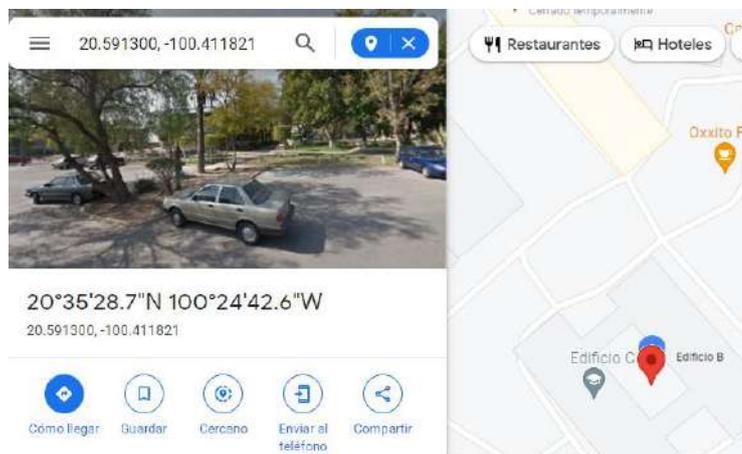


Imagen XLVII: 8.3.1.3: Coordenadas en WGS84 del edificio B de Ingeniería

No obstante, este método solo será para obtener la latitud y longitud de una ubicación global, todavía queda obtener la altitud de la coordenada. Para esto, la documentación de Google nos ofrece y recomienda una página para obtener la altitud, página que utiliza GeoidEval utility, el cual permite calcular y obtener la altitud según la longitud y latitud en el sistema WGS84 dadas en distintos sistemas de coordenadas globales entre ellas la EGM96, por lo que aún queda pasar la altitud en el sistema de coordenadas EGM96 a WGS84.

Online geoid calculations using the [GeoidEval](#) utility

Position (ex. «16.78 -3.01», «16d46'33"N 3d0.6'W»):

Select action:

Geoid height:

```
lat lon =  
geoid heights (m)  
EGM2008 =  
EGM96 =  
EGM84 =
```

Imagen XLVIII: 8.3.1.4: GeographicLib, "Online geoid calculations"

Pero Google también ofrece una API para conocer la altura de una ubicación, "Elevation Maps API", dicha coordenada será obtenida en el formato WGS84. Para la utilización de esta API, se requiere habilitarla para obtener la key y hacer un request a un endpoint o "link", enviando de parámetros la key y las coordenadas de longitud y latitud. Para estos request será utilizado la herramienta "Postman".



Imagen LI: 8.3.1.7: Objeto generado en coordenadas del edificio B de Ingeniería

8.3.2 Información de los puntos de interés a generar en sistema AR

Estos puntos de interés a generar deberán mostrar una información adicional, además de que estos puntos deberán ser obtenidos de una base de datos, por lo que se deberá generar una estructura JSON que permita compartir información de servidor a cliente, esta estructura deberá contener toda la información necesaria a mostrar a lo largo de la aplicación.

- I. "nameEle": este atributo contendrá el nombre del elemento.
- II. "typeEle": atributo que contendrá el tipo de elemento, es decir, si este elemento será un directorio, una ubicación o un trabajo.
- III. "facultad": atributo con la información acerca de a que facultad se encuentra dicho elemento, agregando además de las facultades disponibles en C.U, una nueva clase, "Universitario", para aquellos

edificios, contactos o trabajos que no sean de una facultad en especial.

- IV. "classEle": lo que podrá ayudar a filtrar de mejor manera los elementos por: "Estructura", para construcciones que no sean un edificio, por ejemplo, la esfinge de Sayuri que se encuentra frente a la biblioteca central; "Edificios", para aquellas construcciones que no sean salones; "Salones"; "Persona", para aquellos directorios que puedan contener una ubicación en la universidad; "Trabajo", para aquellos elementos que sean trabajos.

- V. "basicdata": atributo que contendrá toda la información (si es que la tiene) que será mostrada en los paneles de "información extra":
 - a. "teléfono": atributo para guardar el teléfono del elemento.
 - b. "email": atributo que guardara el correo electrónico del elemento.
 - c. "horario": atributo que guardara el horario de atención.
 - d. "detalles": atributo que guardara información adicional del elemento.
 - e. "ubicación": atributo que indicara una ubicación del elemento.

- VI. "worldpoint": atributo que contendrá las coordenadas en los respectivos sistemas globales del elemento:
 - a) "have_ar": atributo para mostrar si dicho elemento será mostrado en AR.
 - b) "latitude": atributo que contendrá la latitud en el sistema WGS84.

- c) "longitude": atributo que contendrá la longitud en el sistema WGS84.
- d) "altitude": atributo que contendrá la altitud en el sistema EGM94.

VII. "urldata": atributo con los links necesarios para el elemento:

- a) "urlgps": este atributo tendrá el link la ubicación del elemento en Google Maps, para aquellos elementos que se muestren en las pantallas de información sin el sistema AR.

Dicha estructura deberá ser preservada en cada elemento.

```
"nameEle": "Edificio B",  
"typeEle": "Ubicaciones",  
"facultad": "Ingeniería",  
"classEle": "Salones",  
"basicdata": {  
  "telefono": "",  
  "email": "",  
  "horario": "",  
  "detalles": "Edificio dedicado a los salones de clase para alumnos, cuenta con un baño en la planta baja.",  
  "ubicacion": "Se encuentra ubicado a un costado del laboratorio de automatización y a un costado del estacionamiento de ingeniería."  
},  
"worldpoint": {  
  "have_ar": true,  
  "latitude": 20.5913733015913,  
  "longitude": -100.411874106252,  
  "altitude": 1820.20432519075  
},  
"urldata": {  
  "urlgps": "https://goo.gl/maps/JECHkpTijf8uouRU6",  
}
```

Imagen LII: 8.3.2.1: Estructura JSON del elemento "Edificio B"

Para que Unity pueda leer un archivo JSON, se deberá tener una las clases con las variables en el mismo nivel de acceso y de tipo de variable que tenga la estructura JSON, es decir, un DTO que es un patrón de diseño que consiste en crear un objeto con una serie de atributos que puedan ser enviados o recibidos del servidor, estos atributos deben ser serializables.

```
[System.Serializable]
1 referencia
public class UrlData
{
    public string urlgps;
}

[System.Serializable]
1 referencia
public class WorldPoint
{
    public bool have_ar;
    public double latitude;
    public double longitude;
    public double altitude;
}

[System.Serializable]
1 referencia
public class BasicData
{
    public string telefono;
    public string email;
    public string horario;
    public string detalles;
    public string ubicacion;
}

[System.Serializable]
7 referencias
public class Element
{
    public string nameEle;
    public string typeEle;
    public string facultad;
    public string classEle;
    public BasicData basicdata;
    public WorldPoint worldpoint;
    public UrlData urldata;
}

[System.Serializable]
4 referencias
public class Elements
{
    public Element[] elements;
}
```

Imagen LIII: 8.3.2.2: Clases con variables para la lectura del JSON

Tras esto, existe una librería que permite la lectura de la información de un JSON como la clase "Elements" previamente creada, "JsonUtility.FromJson<>", esto permite al final tener una lista con toda la información de los elementos que contenga el JSON. Información que servirá para mostrar los elementos correctamente a lo largo de las funcionalidades de la aplicación.

8.3.3 Visualización de la información del elemento

Puesto que los objetos generados deberán mostrar información adicional a los usuarios, estos objetos deberán tener elementos UI, además de un objeto 3D. Por motivos de la generación de objetos, estos objetos deberán mostrarse en dos formatos, a lo lejos deberá ser visible solo una imagen de la ubicación y el nombre, pero si el usuario se encuentra cerca, deberá mostrar el objeto 3D y los elementos UI respectivos para mostrar la información adicional.



Imagen LIV: 8.3.3.1: Visibilidad del elemento en formato cercano



Imagen LV: 8.3.3.2: Visibilidad del elemento en formato lejano

Además, permitirá ahorrar recursos del dispositivo gracias a que no tendrá que renderizar todos los modelos 3D presentes, más que aquellos en los que el usuario se encuentre cerca, evitando también la sobrecarga de elementos visuales en la pantalla del usuario.

Sin embargo, se ha optado por cambiar la visualización de estos elementos UI para mantener una visualización más limpia e intuitiva, además de permitir ocultar estos elementos UI si el usuario solo desea interactuar con el modelo 3D.



Imagen LVI: 8.3.3.3: Nuevo formato de visualización de datos

Solo que al ser una aplicación que tiene un sistema en realidad aumentada, el usuario puede moverse libremente hasta donde el espacio real le permite, lo que hace que los elementos UI puedan perderse de vista en ciertos ángulos debido a que son elementos 2D en un espacio 3D.

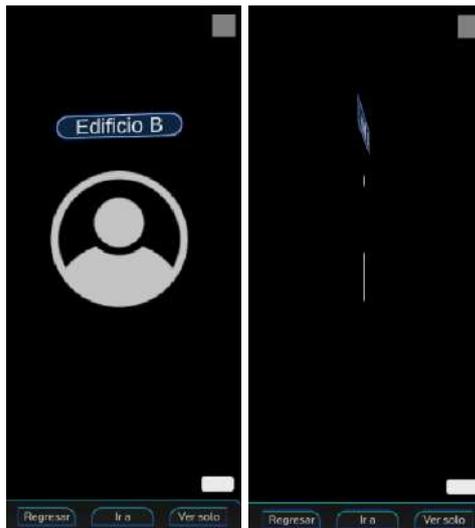


Imagen LVII: 8.3.3.4: Muestra de la visibilidad de los elementos UI 2D en un espacio 3D

Para esto, se propuso la solución de que estos elementos UI estén siempre "mirando" o roten siguiendo al usuario, permitiendo, así, evitar que los elementos UI se pierdan y siempre estén rotando según la posición del usuario.



Imagen LVIII: 8.3.3.5: Muestra de la visibilidad corregida de los elementos UI 2D

No obstante, esta solución generó un problema más, puesto que estos elementos 2D tienen dos caras, la cara frontal que siempre se le será permitida la interacción al usuario y la cara trasera que no se le será permitida la interacción al usuario. Y estos elementos UI que siguen al usuario terminan dando la cara trasera al usuario, evitando el poder interactuar con estos elementos. Pero gracias a una opción dentro del componente "Canvas raycast", es posible permitir estas interacciones.

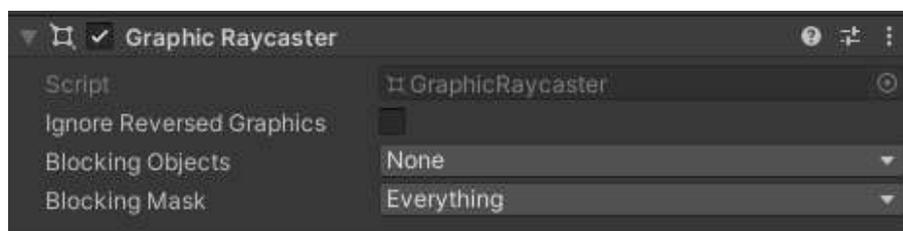


Imagen LIX: 8.3.3.6: Componente "Canvas Raycaster"

Por último, se agregan las funcionalidades a estos elementos UI, empezando por mostrarse solo si el elemento contiene la información necesaria:

- I. Detalles: este elemento UI mostrará la información dentro del atributo "detalles" además del horario dentro del atributo "horario".
- II. Correo electrónico: este elemento interactivo al ser tocado enviará al usuario a su aplicación de correo seleccionado, con un correo nuevo, apareciendo el correo electrónico del contacto sobre el campo del "Destinatario".
- III. Teléfono: Enviará al usuario a su aplicación de llamadas seleccionado con el teléfono del contacto y la extensión que usa la universidad.

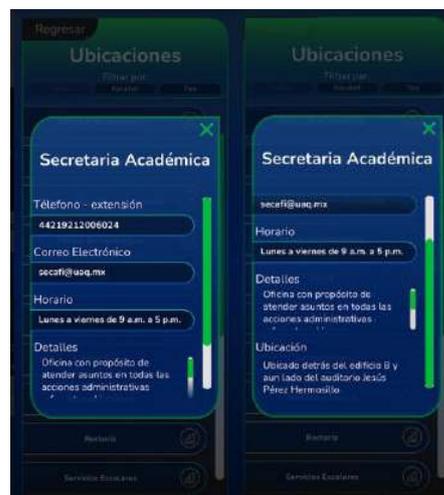


Imagen LX: 8.3.3.7: Funcionalidad UI: Detalles.



Imagen LXI: 8.3.3.8: Funcionalidad UI: Correo electrónico



Imagen LXII: 8.3.3.9: Funcionalidad UI: Teléfono

Estas funcionalidades se aplican también a los elementos fuera del sistema AR, además de las otras funcionalidades necesarias:

- I. Nombre: este elemento mostrará la información dentro del atributo "nameEle" del elemento.
- II. Horario: este elemento UI mostrará la información dentro del atributo "horario" que contenga el elemento.
- III. Ubicación: este elemento será interactivo, al ser tocado se le enviará al usuario a la ubicación del elemento dentro del Google Maps, si es que el usuario lo tiene instalado.

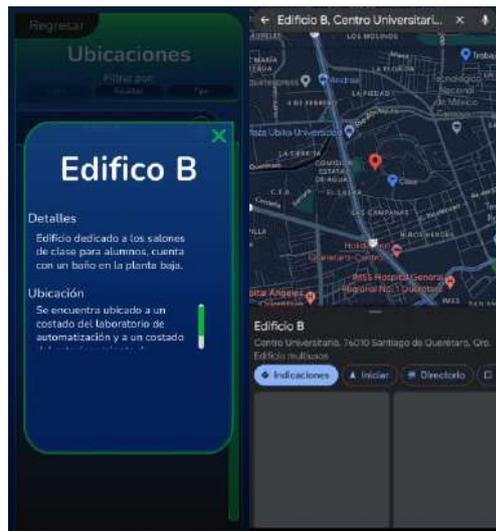


Imagen LXIII: 8.3.3.10: Funcionalidad UI: Ubicación

8.3.4 Recopilación de puntos de interés

Para recopilar la información, se precisó de una investigación de distintos puntos que los alumnos utilicen o que sean de mayor interés e importancia, para después investigar toda la información necesaria, si es que la tiene y agregarla a un documento de Excel, con las siguientes celdas: Nombre, Tipo de elemento, Facultad, Clase, Teléfono, Correo, Horario, Detalles, Ubicación, Punto AR, Latitud, Longitud, Altitud, GPS url.

Para las coordenadas se utilizó Google Maps para la obtención de las coordenadas de longitud y latitud, mientras que para la coordenada de la altitud se utilizó la API Elevation de Google a través de Postman.

En Google Maps se guardan los puntos de cada ubicación en una lista nueva, estos puntos se añadieron en la misma ubicación física que marca el dispositivo.



Imagen LXIV: 8.3.4.1: Ubicaciones guardadas en Google Maps

Tras tener estos puntos, se obtienen de cada uno la longitud y altitud, para posteriormente usar la API Elevation y obtener la altitud desde Postman.

Las ubicaciones elegidas son:

- I. Edificios de clases de las facultades: Puesto que los alumnos pasan mayor tiempo en clases dentro de estos edificios.
- II. Secretaria Académica de cada facultad: Ya que todos los problemas que puedan ocurrir, se tienen solución en estas instalaciones.
- III. Auditorios: Para los auditorios en general, los auditorios han sido una de las direcciones que más llegan a preguntar los visitantes externos, además de estudiantes.
- IV. CEDIT: Al ser una instalación en el que puedes obtener herramientas para elaboración de productos, además de ser un área muy concurrida, era necesario ser agregado.

Al tener esta base de datos, se agregan a un JSON con la estructura básica propuesta anteriormente, para que la aplicación pueda generar todos los elementos con la información obtenida y agregada a cada elemento del JSON.

```
AnalizomastButtons.json - X
Esquema: <No se seleccionó ningún esquema>
244  "altitude": 1828.24365234375
245  },
246  "urldata": {
247    "urlgps": "https://goo.gl/maps/H36XytVfC5ignC3X7"
248  }
249  },
250  },
251  "nameEle": "Auditorio Lic. Fernando Díaz Ramírez",
252  "typeEle": "Ubicaciones",
253  "facultad": "Universitario",
254  "classEle": "Edificios",
255  "basicdata": {
256    "telefono": "",
257    "email": "",
258    "horario": "",
259    "detalles": "El Auditorio Lic. Fernando Díaz Ramírez está disp",
260    "ubicacion": "Ubicado a un costado de Rectoría."
261  },
262  "worldpoint": {
263    "have_ar": true,
264    "latitude": 20.591553246455,
265    "longitude": -100.40972894347,
266    "altitude": 1827.20184980468
267  },
268  "urldata": {
269    "urlgps": "https://goo.gl/maps/AvRigVhqvb5Gdg3q8"
270  }
}
```

Imagen LXVI: 8.3.4.3: Información de los elementos guardados en un archivo JSON

Al tener este archivo JSON con la información de todos los puntos, ahora se podrán generar los elementos en AR y en las pantallas de información fuera del sistema AR con su información.

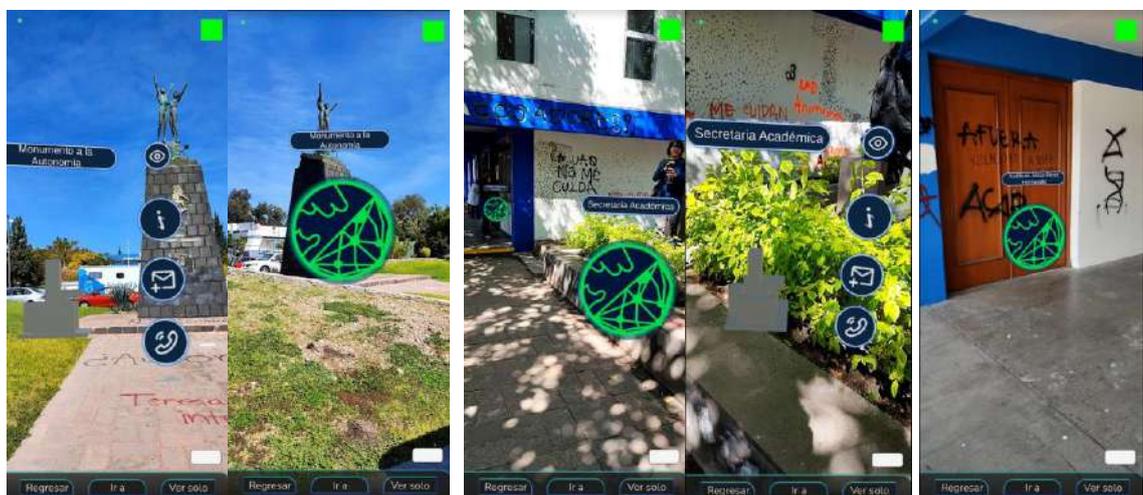


Imagen LXVII: 8.3.4.4: Objetos generados mediante la lista dada dentro del archivo JSON



Imagen LXVIII: 8.3.4.5: Elementos generados a partir de la lista dada en archivo JSON

8.3.5 Corrección: funcionalidad UI, números de extensión

La universidad cuenta con números telefónicos en sus distintas oficinas e instalaciones, estos números telefónicos cuentan de dos partes importantes, el número base que toda la universidad tiene más una extensión. Debido al funcionamiento del número de la universidad donde primero se llama con este número base y tras un cierto tiempo, el conmutador dará un mensaje de bienvenida, es en ese momento es en el que puedes ingresar el número de extensión.

Dicho funcionamiento hace que la funcionalidad del contacto telefónico, que ofrece el número base más el número de extensión en el mismo contacto, hace que fracase y no se realice correctamente la llamada, dando así una marcación errónea.

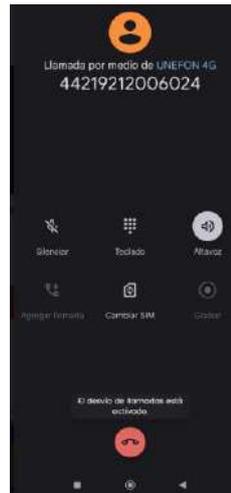


Imagen LXIX: 8.3.5.1: Error de funcionalidad de teléfono, marcar con el número completo manda directamente al buzón

A pesar de este funcionamiento, los teléfonos inteligentes tienen opciones para saltar u omitir estos enlaces al conmutador mediante dos símbolos admitidos en la marcación, estos símbolos están clasificados como símbolos de espera:

- I. Coma (,): Agregar el número base de la línea principal, el símbolo coma seguido de los números de la extensión, permite ingresar al conmutador o la línea principal para después ingresar a la extensión.
- II. Punto y coma (;): Mientras el símbolo coma ingresa a la línea principal, el uso de este símbolo después del número base de la línea principal y antes de los números de la extensión, "omite" la entrada al conmutador e ingresa directamente al número de extensión.

Con todo esto, es posible agregar el símbolo punto y coma entre estos números, solo que este agregado deberá ser en el momento en que el usuario toca los botones de teléfonos, de lo contrario, la visibilidad de estos números, deberá conservarse.

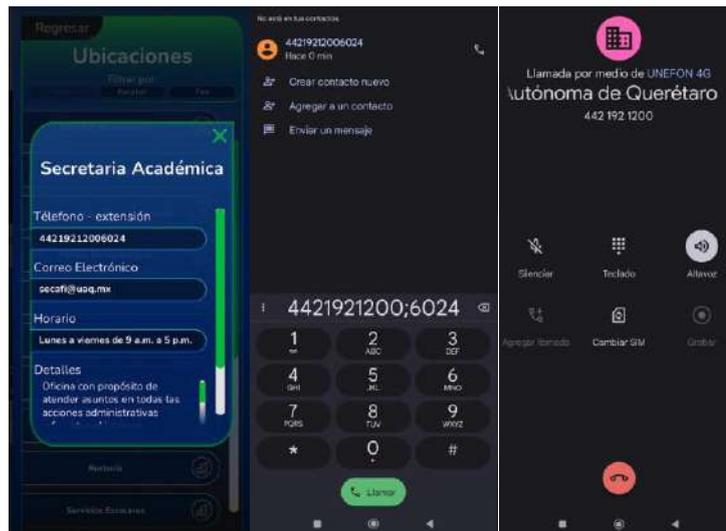


Imagen LXX: 8.3.5.2: Corrección de funcionalidad de teléfono

8.3.6 Visualización y dirección del punto seleccionado

Es necesario tener todos los objetos generados en el entorno virtual con las coordenadas relativas a la tierra, todo esto para poder mostrar al usuario el objeto al que este desee ir. Sin embargo, es importante dar ayudas visuales al usuario que permitan guiarlo al destino de forma más amena e intuitiva. Con ello se opta por dos opciones, una línea desde la posición del usuario a la posición del destino y una flecha que indique la dirección hacia el objeto seleccionado o de destino.

La línea será una guía de forma directa para el usuario hasta la ubicación de destino, aun con esto, la línea se puede llegar a perder de vista o salir del campo visual que ofrece el dispositivo debido a la rotación del usuario. Por ello la flecha guiará al usuario mostrando la dirección con la rotación en caso de que esto suceda, evitando que el usuario pierda la guía o la dirección del objeto, de esta forma podemos acompañar al usuario en todo momento.

Claro, una ayuda visual que indique a cuantos metros se encuentra el objeto o posición de destino podría ayudar al usuario a conocer con antelación que tan cerca o que tan lejos puede estar este del lugar de destino, además de mostrar en el icono del objeto una imagen del lugar, de modo que el usuario pueda conocer como es el edificio al que tiene que llegar.



Imagen LXXI: 8.3.6.1: Funcionalidad "Mostrar", visualización de las guías visuales

Para mostrar la distancia entre un punto y la posición del usuario se aplica una fórmula básica de la geometría, el cual, es necesario tener la posición de ambos puntos para posteriormente construir un vector de distancia entre estos dos puntos, luego transformar estos valores resultantes en metros y por último mostrar al usuario la información con solo una cifra de precisión. Estas líneas visuales deberán desaparecer en cuanto el usuario tenga una determinada cercanía al objetivo de destino. De modo que evita una sobreposición de varios elementos sobre el objetivo. Evidentemente, se debe agregar una funcionalidad para quitar las guías manualmente por si el usuario no desea verlas más o desea elegir otra ubicación.

8.3.7 Información de los trabajos a generar en el sistema AR

Puesto que la universidad presenta un gran desempeño en proyectos tanto sociales como tecnológicos por parte de su comunidad tanto estudiantil como de docentes e investigadores, se hizo una investigación a lo largo de las distintas páginas de las distintas facultades aledañas entre la facultad de ingeniería y rectoría además de las páginas oficiales de universidad tanto en Twitter, Facebook, Instagram y las propias.

A pesar de que estas facultades tienen una gran variedad de logros por parte de sus estudiantes, docentes o investigadores en varios concursos,

competencias o excelencias académicas así como también proyectos sociales internos o externos, gran parte de este contenido se consideró como "noticias" a excepción de aquellos proyectos de estas facultades que hayan dado un "cambio", ayuda o mejora a la universidad o hayan dejado una huella en el desarrollo, innovación tecnológico/a o en la vida cotidiana para la salud, seguridad e integridad o para la sociedad tanto interna como externa.

Esto debido a la poca información y visibilidad de los proyectos, los cuales muchos de ellos solo están listados en apartados escondidos conociendo solo el nombre y el creador del proyecto y su fecha, proyectos que además son de varios años anteriores, entre 2017 y 2019, de modo que de esta forma se pueda mantener una mejor divulgación o difusión de estos proyectos. También en un futuro se puede cooperar con las coordinaciones de difusión, divulgación científica o con las coordinaciones de investigación e innovación e incluso con los programas de divulgación para dar una mayor promoción a estos proyectos además de los logros.

Entre los proyectos recientes elegidos están:

- I. "App y sensor preventivos de daño renal" por Cristián Flores y Andrea Parrales. Por su innovación y cambio para la salud.
- I. "Caminamos juntas" aplicación por Priscila Peralta Martínez. Esta aplicación creada para evidenciar el acoso que ha tenido una escasa presencia en la sociedad y es una innovación para la seguridad social.
- II. "Dispensador automático de agua" proyecto que hizo un cambio a la facultada de ingeniería y que afirma el compromiso por el cuidado ambiental, este esfuerzo por distintas entidades logra un cambio social y un cambio al espacio de la universidad.

Además de proyectos recientes también han sido elegidos proyectos que han sido generados gracias al apoyo de las distintas facultades como de los distintos espacios y tecnologías que ofrece la universidad y que han dado un cambio y presencia a la universidad en distintos aspectos:

- I. "MUAQ, Lácteos de Verdad", este proyecto genera varios productos lácteos gracias al conjunto de las distintas operaciones en los distintos campus.
- II. "APREHENDERE TVUAQ (CITE)", el programa "Aprehendere" en un programa de TVUAQ educativo, dicho proyecto utiliza en parte alumnos de distintas facultades y carreras para generar estos contenidos audiovisuales, para ser subidos en el canal TVUAQ los domingos por la mañana.
- III. "Ingenautas TVUAQ", "Ingenautas" es un programa de divulgación científica, dicho programa tiene de objetivo mostrar a la población los avances de la ciencia y sus aplicaciones, en este programa participan varias carreras.
- IV. Entre otros productos y servicios que ofrece la universidad: "AGUAQ", "FarmaUAQ", USC o Unidad de Servicios Clínicos, LECRIMA y "Flavis"

De estos trabajos han sido tomados varios datos además de los necesarios descritos en los de tipo de ubicaciones:

- I. Fundador o creador del proyecto.
- II. Link o URL del proyecto (si es que la tiene), en el caso de "Caminamos juntas" se agregó el enlace de descarga de la aplicación.
- III. Imágenes de estos proyectos.

De modo que se agrega a la base de datos esta información y se modifica la estructura JSON, agregando una lista de elementos "workdata" con los siguientes elementos:

- I. "fundador": donde se guardará la información del fundador o creador.
- II. "urlimage": URL que será obtenido del banco de imágenes en el S3 de AWS.
- III. "textworkurl": texto que se mostrara en lugar del "urlwork".

- IV. "urlwork": esta información puede contenerla un proyecto o trabajo si tiene una página o link a su proyecto para utilizarse, en caso de tenerla se agrega a este campo.

```
"nameEle": "Caminamos juntas",
"typeEle": "Trabajos",
"facultad": "Ingeniería",
"classEle": "Trabajo",
"basicdata": {
  "telefono": "",
  "email": "",
  "horario": "",
  "detalles": "Evidenciar el acoso callejero en el diario andar por la ciudad es la meta principal de 'Cami",
  "ubicacion": ""
},
"worldpoint": {
  "have_ar": true,
  "latitude": 20.5917181068879,
  "longitude": -100.411282686299,
  "altitude": 1829.06799316406,
  "urlgps": "",
  "objectid": "",
  "iconid": ""
},
"workdata": {
  "fundador": "Priscila Peralta Martinez",
  "urlimage": "",
  "urlwork": "https://play.google.com/store/apps/details?id=com.jaav.reporteacosoapp&hl=es_MX&gl=US&pli=1"
}
```

Imagen LXXII: 8.3.7.1: Nueva sintaxis JSON

8.3.8 Iconos a mostrar en los puntos de interés

Otra guía importante a mostrar al usuario y que es sumamente importante sobre todo para las personas que sean más del tipo visual para guiarse o para personas que no sepan cómo es el lugar al que van, es el dar a conocer como es el edificio o espacio al que el usuario desea llegar de modo que el usuario pueda ubicar a una determinada distancia hacia donde está su destino.

Para ello es necesario tomar o mostrar imágenes recientes de estos espacios o al menos imágenes que tengan un gran parecido con la actualidad. Para ello se tomarán imágenes y se les dará un formato "circular" para mostrarse en el modo "lejos" de cada elemento generado en el espacio virtual.



Imagen LXXIII: 8.3.8.1: Foto sin diseño circular



Imagen LXXIV: 8.3.8.1: Foto con diseño icono circular

Sin embargo, dar estos iconos en tiempo real a los objetos será complejo, en primer lugar porque no debería hacer esperar por mucho tiempo al usuario cuando estos objetos se descarguen y se generen, segundo, la descarga de objetos por internet no siempre puede resultar en un éxito, puede generar fallos por diversos motivos, por lo que se propone la idea de que al ser generados en el espacio virtual estos por su cuenta descarguen la imagen o imágenes que necesita para su correcto funcionamiento en las distintas funcionalidades y si esta descarga tiene un problema o termina en un fallo por algún motivo, se agregará una imagen por defecto incluso antes de que se

descargue la imagen, de modo que se puede continuar con la generación de objetos y además descargar de forma asincrónica las imágenes y así no se tiene que esperar a que se generen todos los objetos en el espacio virtual y que se descarguen todas las imágenes para que el usuario pueda disfrutar su sesión.

8.3.9 Creación de objetos 3D para los puntos de interés

Cada punto de interés tendrá un modelo u objeto 3D a mostrar cuando este se encuentre en su modo "cercano", dicho objeto será interactivo mediante los gestos básicos del usuario en la pantalla (rotación y escalado).

Estos modelos serán creados en la herramienta Blender. Estos serán creados desde 0 basándose en el concepto que le caracterice a cada punto de interés, estos no podrán contener detalles pequeños debido a que el usuario no siempre se ubicara muy cerca del objeto y que este no tendrá la capacidad de escalarse en gran medida. De modo que se deberán crear en base al concepto manteniendo detalles medianos a grandes.

Agregando a esto, estos modelos deberán tener un distintivo que permita al usuario saber en qué facultad se encuentra permitiendo conocer qué tipo de ubicación será y en que facultad se encuentra, debido al poco tiempo y la cantidad de personas especializadas en el proyecto se opta por reducir estos tiempos mediante una separación en dos conceptos distintos, la parte distintiva del lugar y la parte distintiva de la facultad, es decir, del "Drop" y el "BoxDrop" respectivamente, el "Drop" será el objeto 3D que permita reconocer si es un edificio, una secretaria académica o servicio escolares, un auditorio y etc., mientras el "BoxDrop" será el objeto 3D que "englobará" al "Drop". Generando así los BoxDrops para las distintas facultades con algo característico de su logo o de la facultad:

- I. Ingeniería: Un átomo con los electrones en forma de símbolos numéricos.
- II. Química: El benceno como base.
- III. Contaduría: El Yin Yang, el concepto de la dualidad de las cosas: cargo y abono.

IV. Psicología: La mano con el símbolo PSI.

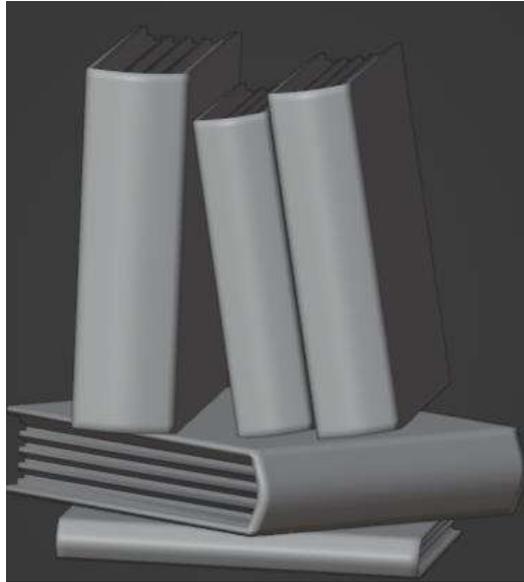


Imagen LXXV: 8.3.9.1: Drop de librería base

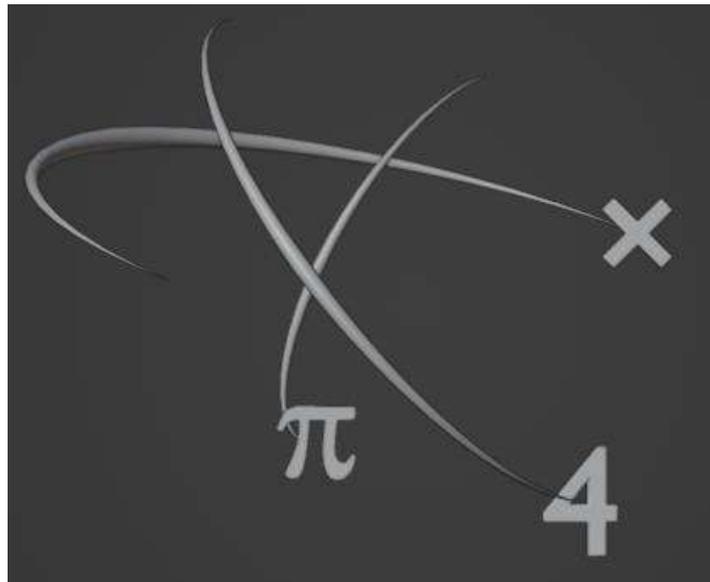


Imagen LXXVI: 8.3.9.2: BoxDrop de Ingeniería

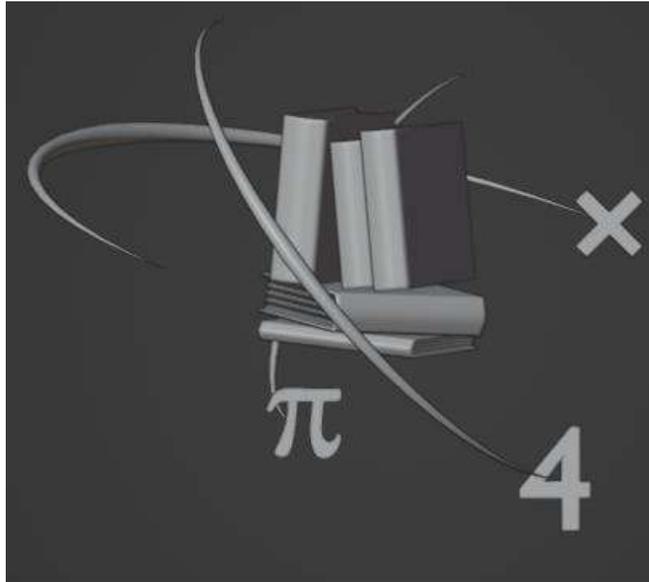


Imagen LXXVII: 8.3.9.3: BoxDrop de Ingeniería con Drop de Librería base

Tras generar estos modelos, las UV deberán ser creados o modificados de estos mismos para la correcta visualización del material, textura o shader. Si estos no están correctamente generados es posible que el apartado visual de estos objetos tenga errores visuales y no se muestre correctamente e incluso tenga partes que no sean visibles si es que el shader agrega transparencia al objeto o se iluminen y tengan brillos mal "cortados" o donde no deberían tenerlos.

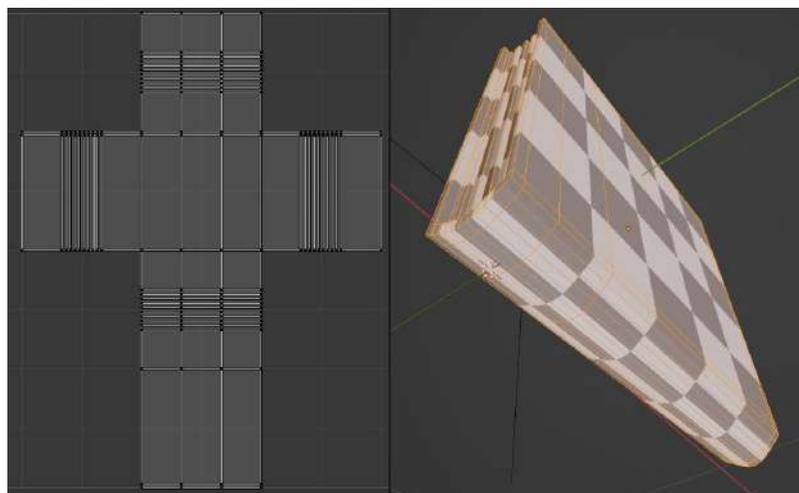


Imagen LXXVIII: 8.3.9.4: UV de objeto 3D sin modificar

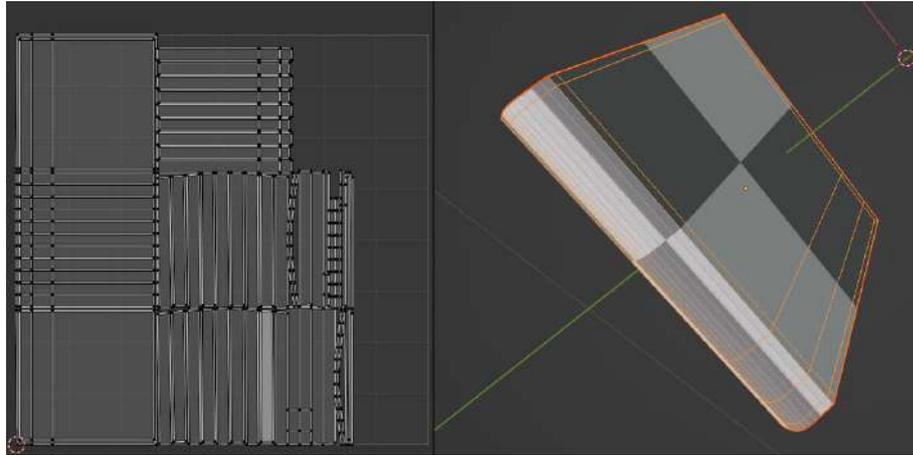


Imagen LXXIX: 8.3.9.5: UV de objeto 3D modificado

Una vez que estos objetos 3D estén creados y tengan sus UV correctas es necesario darles un acabado visual, dándoles color, brillos o transparencias donde lo necesite.



Imagen LXXX: 8.3.9.6: Drop de Librería con material agregado

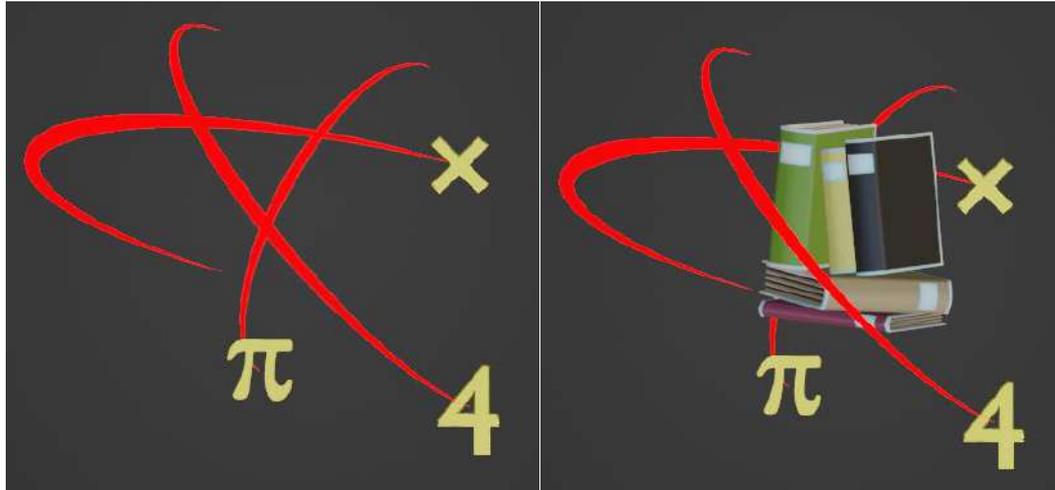


Imagen LXXXI: 8.3.9.7: BoxDrop de Ingeniería con Drop de Librería

8.3.10 Agregado de objetos a la lógica de generación de objetos

Con los objetos 3D y los iconos de cada punto de interés hechos se agregan a Unity y se les da el nombre a los BoxDrop según los valores o parámetros existentes dentro del campo "facultad" en el JSON, mientras a los Drops se les da un nombre especial según el tipo en el campo "objectid".

Los iconos tendrán un nombre abreviado del nombre del punto de interés y será agregado en cada campo "iconid" de cada punto de interés, para que puedan ser descargados próximamente.

```
"nameEle": "Biblioteca Central",  
"typeEle": "Ubicaciones",  
"facultad": "Universitario",  
"classEle": "Edificios",  
"basicdata": {  
  "telefono": "3515",  
  "email": "",  
  "horario": "Lunes a viernes de 8 a.m. a 3:30 p.m.",  
  "detalles": "Biblioteca general del Campus Universitario.",  
  "ubicacion": "Ubicado a un costado de Rectoría, detrás de la efige Sayuri."  
},  
"worldpoint": {  
  "have_ar": true,  
  "latitude": 20.591830536222655,  
  "longitude": -100.41052276289552,  
  "altitude": 1829.989379882812,  
  "urlgps": "https://goo.gl/maps/khRwsW1VvKwJ8CtPv7",  
  "objectid": "Biblioteca",  
  "iconid": "Biblio_Central"  
},  
"workdata": {  
  "fundador": "",  
  "urlimage": "",  
  "textworkurl": "",  
  "urlwork": ""  
}
```

Imagen LXXXII: 8.3.10.1: Actualización de JSON de iconos, BoxDrops y Drops

Tras esto, cuando se generen los objetos de los puntos de ubicación, en la vista lejana se podrán ver los nuevos cambios, que son los iconos de cada punto, además de los objetos 3D mostrando la facultad con el BoxDrops y el tipo de ubicación con el drop, todo esto en funcionamiento con las demás mecánicas y lógicas implementadas de visualización del nombre, correo, información extra, teléfono y la dirección y distancia.



Imagen LXXXIII: 8.3.10.2: Icono, Drop y BoxDrops de secretaria Academica de Ingeniería

8.4 Objetivos AR o "Image Target"

Las tecnologías avanzan con el tiempo y con ellas también se mejoran las herramientas y se pueden llegar a utilizar en diversos dispositivos móviles aunque no cuenten con una gran capacidad de procesamiento o rendimiento como el caso de los dispositivos de sobremesa, sumando inteligencias artificiales junto a los sensores es posible tener nuevos campos en el sistema AR en los dispositivos móviles como los objetivos AR o "image target" que permiten tener un seguimiento de su posición y rotación del objetivo dentro del sistema AR en un entorno 3D a través de una vista 2D proporcionada por las cámaras.

Con estos objetivos AR se pretende que puedan ser colocadas alrededor de ciertas zonas, oficinas o edificios para que puedan ser utilizados a través de la aplicación con la finalidad de mostrar ciertos contenidos de información según la disponibilidad del contacto.

8.4.1 Creación base de Objetivos AR

Los objetivos AR son objetos que pueden ser usados en la realidad que pueden mostrar elementos digitales sobre ellos una vez que sean detectados, todo esto es posible gracias al reconocimiento de imagen y al seguimiento de objetivo.

Estos objetivos pueden ser objetos materiales tridimensionales o bidimensionales, en el caso del proyecto se usarán objetos bidimensionales y para que estos objetivos AR puedan ser completamente compatibles con el reconocimiento de imagen y el seguimiento de objetivo es necesario que tengan cierto patrón "único" que puedan ser reconocidos fácilmente y que puedan ser distinguidos de cualquier otro.

Es ideal que un objetivo AR cumpla por lo menos en gran medida los siguientes principales atributos:

- I. Ricos en detalles u objetivos con gran variedad de elementos.
- II. Objetivo bien iluminado o que tengan un buen contraste que permita distinguir todos los elementos.

- III. Patrones no repetitivos, evitando la simetría y las áreas sin características.

Claro, hay otras medidas importantes a tener en cuenta como:

- I. Tener elementos con detalles o características distinguibles.
- II. Tener una distribución de los elementos a lo largo del objetivo.
- III. Tener elementos con formas lo más orgánicas posible.
- IV. No tener motivos repetidos, incluso si cada uno de estos están girados o escalados de diferente forma.
- V. No es necesario tener una forma rectangular de la base del objetivo.

El crear objetivos AR con estas características puede hacer que el seguimiento y el reconocimiento funcione correctamente, por ello, se ha optado por diseños con una estructura de códigos QR.



Imagen LXXXIV: 8.4.1.1: Imagen "QR" de referencia

Estos diseños "QR" son más sencillos a largo plazo de hacer y estos cumplen con gran parte de las características necesarias:

- I. El nivel de detalle puede variar desde un bajo nivel hasta un alto nivel de detalle.
- II. Sus elementos son elementos distinguibles.

- III. Sus elementos estan muy bien distribuidos.
- IV. El contraste entre las zonas oscuras y brillantes no jugara tanto en contra de estos elementos al ser elementos blanco y negro.
- V. A pesar de presentar una base geométrica sencilla sobre un cuadrado, estos no generan un patrón repetitivo, evita la simetría y en gran parte los espacios vacíos.
- VI. La unión de varios cuadrados negros sobre la superficie blanca no crea figuras inorgánicas del todo.

Y por supuesto, aunque tenga figuras inorgánicas no ocupan gran parte del objetivo al menos no en un 65% - 80% del espacio, esto dependiendo la densidad y el tamaño de los cuadros de gran proporción, aunque si se desea, se pueden evitar esos cuadros de gran proporción y esto podría ser casi un 100% de espacio con figuras más "orgánicas".

Para generar estos códigos QR, existen generadores de estos códigos QR de varios tipos en la internet que podrían ser utilizados, por lo que lo hace más sencillos de usar y crear en caso de que se necesite generar varios objetivos, lo que llega a ser rápido, flexible y fácil de modificar en un futuro, además de que puede reducir costos de tiempo y fondos para generarlos.

Con lo anterior explicado queda definir el diseño final de los objetivos, el cual será dividido en dos partes, la primera parte que es código QR que hará en gran parte de objetivo AR para el seguimiento y reconocimiento, siendo la segunda parte lo que será la carátula o diseño que tendrá este objetivo que defina que es un objetivo de la aplicación "TeraNovus", esto para que el usuario pueda identificarlo, así como de un distintivo entre cada objetivo que permita ayudar a que el encargado de ponerlos en los lugares correspondientes no se confunda de objetivo o se coloquen en lugares equivocados.

8.4.2 Diseño de objetivos AR

Con lo anterior explicado queda definir el diseño final de los objetivos, el cual será dividido en dos partes, la primera parte que es código QR que hará en gran parte de objetivo AR para el seguimiento y reconocimiento dentro de la aplicación y la segunda parte que será la carátula o diseño visual que tendrá este objetivo para definir que es un objetivo de la aplicación "TeraNovus" para que el usuario pueda identificarlo en los distintos espacios, así como de un distintivo entre cada objetivo que permita ayudar a que el encargado de ponerlos en los lugares correspondientes no los confunda y se coloquen erróneamente.

Ahora, en lo correspondiente de la parte del código QR se han propuesto las siguientes características y cambios:

- I. El código QR no tendrá esos cuadros grandes para tener más proporción de figuras "orgánicas" más asimétricas frente a las inorgánicas más simétricas.
- II. Se basará sobre el fondo negro y cuadros blancos, lo que permitirá diferencia la parte del objetivo del papel blanco, claro, respetando los colores marcados para el proyecto, el color negro (000000) a un color más grisáceo (2E2E2E).
- III. La densidad cuadros será bajo, más que nada, porque en un código QR puede ir desde módulos bajos de 16x16 hasta la versión 10 que tiene módulos de 177x177, entre más módulos más información puede tener, pero al solo ser un diseño que funcione excelente con el reconocimiento y el seguimiento no es necesario almacenar información, por lo tanto, se puede ir con módulos o densidades más bajas.
- IV. También se está pensando que no todos los celulares tendrán gran capacidad de procesamiento, si ya es "pesado" el AR, la lectura de objetivos lo aumenta, por lo que aumentar la densidad de elementos en el código QR aumentara también la capacidad de procesamiento necesario debido a que aumenta los puntos para el seguimiento y reconocimiento o de "tracking".

Con esta base de cambios y características necesarias se obtiene un diseño en la primera parte lo más fiel posible



Imagen LXXXV: 8.4.2.1: Primera parte del diseño de Objetivo AR

En cuanto a la segunda parte del diseño, que necesita un distintivo entre cada objetivo y que se reconozca como parte de la aplicación, se agrega un logo sencillo y no tan llamativo, además de agregar el abreviado del nombre del contacto en una esquina:



Bib_Central

Imagen LXXXVI: 8.4.2.2: Diseño final del objetivo AR

Para colocarse en uno de los puntos de contacto, se recortaría desde la parte circular hacia dentro, dejando así la oportunidad de conocer qué objetivo era y donde debería ir en una sola palabra antes de ser colocado.

8.4.3 Lógica de Objetivos AR en Unity

Los objetivos funcionan gracias a la funcionalidad extra que tiene "AR Foundation" que es la posibilidad de utilizar una base de "imágenes" que servirán como librería de imágenes de referencia, para que al detectar en cámara alguna imagen parecida a cualquiera que las que esté en la librería de imágenes avise de este suceso.

Para ello es necesario crear un objeto en Unity de AR Foundation que tenga y guarde las imágenes de referencia, es decir, la librería de imágenes "XR Reference Image Library" para agregar todos los objetivos AR

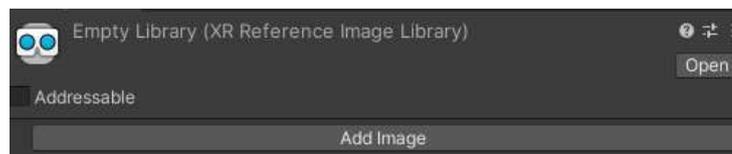


Imagen LXXXVII: 8.4.3.1: Objeto XR Reference Image Library

Esta librería es la que se usará mediante un script que pregunte a través de un sistema de eventos si una imagen de la librería ha sido vista a través de la cámara, esto ya lo tiene el sistema AR Foundation y es la funcionalidad de AR Tracked Images, es decir, el evento a revisar sería el "ARTrackedImagesChangedEventArgs". Esta funcionalidad suelta una serie de eventos según los estados que sucedan al detectar una imagen: cuando se encuentra una imagen dentro de la librería por primera vez, así como cuando este se encuentra aún en el campo de visión de la cámara y también al momento de salir dicha imagen del campo de visión de la cámara, siendo estos dos últimos eventos más importantes. Estos estados vienen de un parámetro del argumento que viene con el evento.

Dentro de estos estados es necesario hacer una comparación entre objetos para saber cuál imagen dentro de todas las disponibles en la librería de imágenes se ha detectado, esto se hará gracias a que los objetos de la lista de información de objetos tendrán un nombre dentro de un parámetro "objectId"

en los contactos del JSON igual al nombre de las imágenes que se encuentra en la librería de imágenes, si estos son iguales entonces dentro de ese objeto en el JSON se buscará la información y se mostrará alrededor de la imagen a través de la aplicación.

Claro, esta visualización de datos será dado con un diseño visual superpuesto a la imagen, por lo que es necesario utilizar y dejar una configuración de la funcionalidad que permita el rastreo en posición y rotación de la imagen para que la superposición del diseño digital funcione bien sin importar la posición y rotación de la imagen o del usuario mismo.

8.4.4 Agregado de las imágenes a la librería de imágenes

Para el agregado de las imágenes a la librería de imágenes de referencia de Unity (XR Reference Image Library), primero es necesario e importante que la forma y el diseño sean estrictamente el mismo que el que se usara en la realidad.

Además de esto, es importante mantener también el tamaño de la imagen entre la realidad y la digital para que el rastreo funcione correctamente, sin olvidar el punto de que sus nombres deben ser exactamente el mismo que utilizara el campo o atributo "objectId" en el JSON para que estos puedan obtener la información que necesitarían al momento de mostrarla.

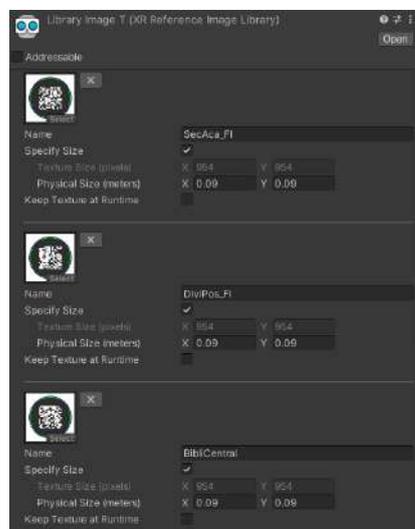


Imagen LXXXVIII: 8.4.4.1: Librería de imágenes

Hecho lo anterior, las imágenes que contenga esta librería serán utilizadas para comparar, encontrar que objeto se está detectando en cámara y mostrar la información de ese objeto. Por lo que agregar nuevas imágenes, solo será necesario agregar la imagen en la librería y los datos en el JSON de ese contacto que tenga objetivo AR, ambos con su mismo respectivo nombre en el "objectid".

8.4.5 Corrección: Problema de carga en runtime del LibraryImageTarget de Unity

La librería tiene un problema al momento de ser cargando durante el proceso activo de la aplicación o mientras la aplicación se está ejecutando, es decir, en "runtime".

La librería puede aceptar el cambio de esta en runtime, pero no será posible por medios simples, sino a través de métodos más complejos como la lectura de la imagen casi a nivel de bits para ser transferidos como textura 2D a la librería en el que al final deberá ser actualizado. Sin embargo, no es conveniente como tal, debido a que la imagen primero debería ser descargada, después deberá ser leído y se actualizaría la librería, todo esto mientras se descargan las demás imágenes y la información de las demás funcionalidades de la aplicación, esto en etapas tempranas de la aplicación no tiene muchos problemas, pero con el tiempo cuando se tenga más de 50 imágenes es hacer 50 veces el mismo proceso, esto cada vez que se inicie el sistema AR de la aplicación.

En un inicio se propuso la idea de utilizar otro sistema de Unity que es el "AddressableAssets" que permite crear "paquetes" de objetos de Unity que puedan ser descargados y ser agregados al juego o la aplicación en un momento más adelante que sea necesario, permitiendo reducir el peso de la aplicación y la posibilidad de descargar el objeto de la librería de referencia ya actualizada, siendo esta librería ya precargada con la nueva información al momento de ser requerida y descargada.

Sin embargo, esta última propuesta fue puesta a prueba y no tuvo éxito debido a que Unity no detecta muy bien esta nueva librería descargada, pues al momento de utilizarla el objeto arroja el número de imágenes que existen dentro de la librería, pero no la información ni el nombre de las imágenes, es

decir, el objeto guarda una lista "vacía" de elementos con el mismo número de objetos que tenía.

Por lo tanto, se ha propuesto temporalmente que el agregado de nuevas imágenes de referencia a la librería sea actualizado a los usuarios a través de actualizaciones de aplicación, con esto, agregando también la información de estos objetos en el JSON en el momento que se lance la actualización con las nuevas imágenes de referencia. Dejando así la librería precargada y evitando problemas de datos y de carga de información.

8.4.6 Testeo de Objetivos AR y agregado de botones interactivos

Al tener la librería precargada en la aplicación y la información correcta en el archivo JSON, al momento de detectar una imagen dentro de la librería, mostrará cierta información del contacto, siempre y cuando dicho contacto en su parámetro "objetid" dentro del JSON sea exactamente el mismo que el nombre de la imagen.

Esta información será mostrada con un diseño sencillo e intuitivo dentro de la aplicación, este diseño tendría el logo de la aplicación en un diseño 3D junto a una serie de botones interactivos:



Imagen LXXXIX: 8.4.6.1: Diseño objetivo AR digital

Con esto el usuario puede saber el nombre del contacto en la parte arriba, así como el horario usual del contacto en su oficina en la parte de abajo o si lo requiere un correo y teléfono de contacto ubicado a los lados, todo esto en

caso de que dicho usuario haya llegado a la oficina del contacto y este no se encuentre por algún motivo.



Imagen XC: 8.4.6.2: Vista real del objetivo en oficinas de Rectoría



Imagen XCI: 8.4.6.3: Vista digital del objetivo AR en las oficinas de Rectoría

8.5 Agregado de las funcionalidades en la nube (AWS)

Las funciones de la nube en las aplicaciones y videojuegos permiten mejorar varios aspectos de estos, como la optimización a la hora de actualizar, añadir o borrar datos extras de la aplicación, así como también el fácil manejo de datos de los usuarios a través de la red, el guardado de estos datos como una copia de seguridad, la optimización y reducción del peso de una aplicación o videojuego e incluso la corrección de errores que pueda tener una aplicación de forma rápida y sencilla.

Ya que se intenta que la aplicación sea fácil de manejar para los usuarios y que la información proporcionada por la aplicación hacia el usuario es susceptible a cambios constantes en el tiempo, ya sea porque el contacto ya no trabaja en las horas designadas o se cambió de lugar, de correo o teléfono e incluso porque el lugar ya haya sido renovado, modificado o removido para nuevos espacios administrativos o escolares, es necesario tener una base de datos que pueda ser modificada en línea a través de la nube para que la aplicación en cada inicio descargue dicha información y se le pueda mostrar al usuario una información siempre actualizada sin necesidad de hacer actualizaciones a la aplicación.

De este modo, es posible tener una base de datos avanzada que pueda ser gestionada con un sistema de gestión de bases de datos en tablas como las aplicaciones "MySQL" y otras. O es posible tener una base de datos simplificada en una estructura JSON que pueda ser leído y modificado con una herramienta sencilla de lectura de texto.

Debido al tiempo de producción hacer una base de datos simplificada a través de un formato JSON hará un avance más rápido para crear las mecánicas de conexión entre servidor/usuario a través de este formato, aunque el otro sistema de base de datos es casi el mismo proceso es requerido hacer una serie de procesos extras, sin embargo, la aplicación solo necesita una lectura rápida de la base de datos, lo cual no necesariamente necesita modificar, añadir o eliminar algún dato desde la aplicación más que solo leer y descargar los datos, lo que le hace menos requerido tener que hacer esos procesos extras innecesarios al menos en una etapa temprana de la aplicación, pero aun así mientras se crean estas mecánicas de conexión entre servidor/usuario se

dejará la base necesaria para el caso en el que un futuro pueda requerirlo, por lo mientras se decide ir por este método simplificado.

El servidor al que el usuario y la aplicación se conectara será a través de los servicios que ofrece AWS de Amazon, dentro de estos servicios se encuentra el servicio S3 que permite acceder a archivos colocados en un espacio de almacenamientos siempre y cuando estos accesos sean permitidos.

Dentro de este servicio se alojarán las imágenes de las ubicaciones y de los trabajos, así como también los objetos 3D de los Drops y BoxDrops creados, además de la información recopilada en la base de datos.

Cabe recalcar que el servicio AWS que ofrece Amazon, puede cambiar al leer este documento, puesto que este servicio se encuentra siempre en constante cambio, dejando así una diferencia visual en la interfaz o de funcionalidades mostradas parcial o totalmente distintas a las mostradas en este documento.

8.5.1 Descarga y obtención: Datos a través del JSON desde la nube

Una vez que se tiene el espacio de almacenamiento dentro del servicio S3 de Amazon en AWS, es posible cargar sobre él cualquier archivo, siempre que no se pase del límite establecido, en este caso el archivo cargado es un archivo JSON que contiene toda la información de los contactos, ubicaciones y trabajos con un peso menor de 1 MB (más de 50 KB).



Imagen XCII: 8.5.1.1: Archivo JSON cargado en el sistema S3

```
{
  "elements": [
    {
      "nameEle": "Edificio B (FI)",
      "typeEle": "Ubicaciones",
      "facultad": "Ingeniería",
      "classEle": "Salones",
      "basicdata": {
        "telefono": "",
        "email": "",
        "horario": "",
        "detalles": "Edificio dedicado a los salones de clase para alumnos, cuenta con un baño en la planta baja.",
        "ubicacion": "Se encuentra ubicado a un costado del laboratorio de automatización y a un costado del estacionamiento de ingeniería."
      },
      "worldpoint": {
        "have_ar": true,
        "latitude": 20.591260128378725,
        "longitude": -100.41169670901303,
        "altitude": 1826.069580078125,
        "urlgps": "https://goo.gl/maps/JEChkpPTjf8uouRU6",
        "objectid": "Clase",
        "iconid": ""
      },
      "workdata": {
        "fundador": "",
        "urlimage": "",
        "textworkurl": "",
        "urlwork": ""
      }
    }
  ]
}
```

Imagen XCIII: 8.5.1.2: Información dentro del archivo JSON en el navegador web

Estos datos se descargarán en Unity a través de la funcionalidad que ofrece este mismo, es decir, "UnityWebRequest" con esta funcionalidad es posible la descarga de distintos archivos como archivos de textos, imágenes o audios y entre otras cosas.

La descargar consistirá en enviar una petición GET request a un enlace específico del archivo JSON obtenido de S3, en el que se debe esperar a que se complete la solicitud y ver si esta petición ha fallado al conectar de modo que si falla se tiene que avisar en donde ha sido el fallo al usuario, en caso de no haber un error podemos obtener el resultado en forma de texto en bruto, este formato en bruto aún no es legible completamente por C#, para ello es necesario hacer una serialización del texto en bruto al formato JSON, esto es posible lograrlo a través de una serie de librerías como "Newtonsoft.JSON" o de la clase "JsonUtility" que ya contiene Unity por defecto.

```
async Task GET_DownloadJson(string urlink)
{
    UnityWebRequest requestText = UnityWebRequest.Get(urlink);
    requestText.SendWebRequest();
    while(!requestText.isDone)
    {
        await Task.Yield();
    }

    if (requestText.result == UnityWebRequest.Result.ConnectionError ||
        requestText.result == UnityWebRequest.Result.ProtocolError ||
        requestText.result == UnityWebRequest.Result.DataProcessingError)
    {
        Debug.Log(requestText.error);
    }
    else
    {
        //Look Data
        //Create class T to read Json
        TypeStore<Elements> JsonReadable = new TypeStore<Elements>();

        //Read Json
        string tempData = requestText.downloadHandler.text;
```

Imagen XCIV: 8.5.1.3: Mecánica base para la recogida de información a través de la nube

Sin embargo, es necesario también crear una clase con las variables existentes en la sintaxis del JSON para que el texto en bruto pueda ser leíble, basándonos en la clase creada con todos los parámetros como si de un valor de una variable se tratase.

```
[System.Serializable]
1 referencia
public class BasicData
{
    public string telefono;
    public string email;
    public string horario;
    public string detalles;
    public string ubicacion;
}

[System.Serializable]
14 referencias
public class Element
{
    public string nameEle;
    public string typeEle;
    public string facultad;
    public string classEle;
    public BasicData basicdata;
    public WorldPoint worldpoint;
    public Workdata workdata;
}
```

Imagen XCV: 8.5.1.4: Clase utilizada para la serialización del JSON

Con estas dos bases ya se puede formar una variable del tipo "Element" en forma de lista para almacenar todos los datos en esa misma variable y que pueda ser utilizado en todos y cada una de las funciones de la aplicación, esto

se aplicara al momento de iniciar sesión, es decir, en la pantalla principal, para que el usuario pueda utilizar esta información en las diferentes pantallas de información (Trabajos, Ubicaciones, Contactos, AR)

No obstante, esta información se necesita ser pasada entre diferentes escenas de Unity y para no perder dicha información descargada y preparada, es necesario hacer algunas contramedidas, en este caso se hará uso de un ScriptableObject que permite guardar información incluso una vez cargado y esta información cargada puede ser guardada incluso aún después de haber cerrado la aplicación, lo que permitirá ver esta información en caso de que el usuario no tenga internet, siempre y cuando ya haya descargado esta información previamente.

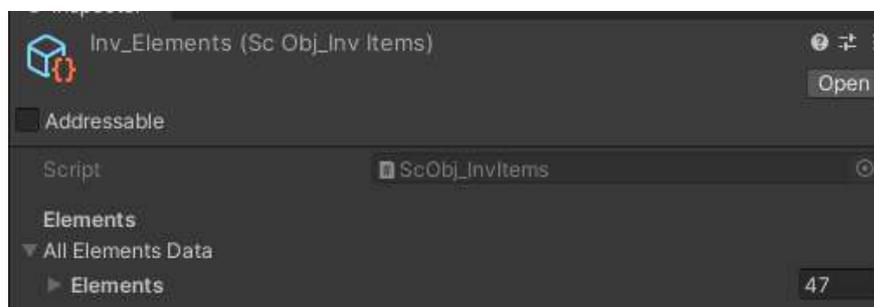


Imagen XCVI: 8.5.1.5: ScriptableObject que guarda los datos del JSON ya serializado

8.5.2 Descarga y obtención: Objetos 3D a través de la nube

Los objetos 3D al ser prefabs, que básicamente son objetos con una configuración específica creada dentro de Unity, será necesario el uso del "Addressables" de Unity para crear archivos bundles que contiene información del objeto en una forma "comprimida" que al ser leída por Unity "recrea" en cierto modo el objeto con la configuración, todo esto con la finalidad de que el prefab pueda ser utilizable por Unity y que estos no estén haciendo peso extra a la aplicación.

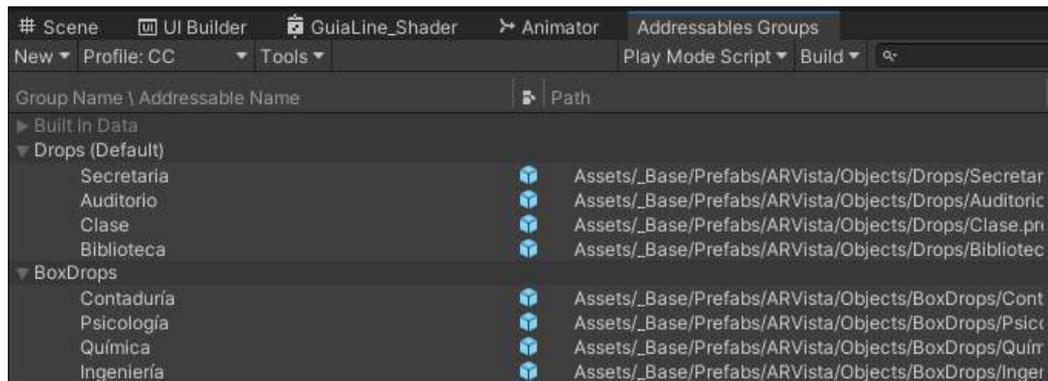


Imagen XCVII: 8.5.2.1: Visualización de los addressables creados de los BoxDrops y los Drops

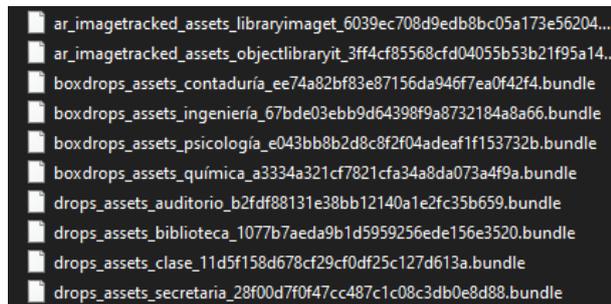


Imagen XCVIII: 8.5.2.2: Archivos bundles de los BoxDrops y Drops

Estos archivos bundles son los que serán subidos al servidor S3 para que sean descargados y utilizados en el momento en que se requieran, estos objetos pueden ser modificados sin necesidad de actualizar la aplicación, solo es necesario configurar el JSON para los nuevos Drops y BoxDrops que se puedan agregar, modificar o remover en un futuro. A pesar de ello, si es necesario que la aplicación tenga una conexión con la base de datos del servidor remoto, de lo contrario el sistema no encontrara los archivos necesarios y las funcionalidades básicas AR de la aplicación no funcionarían correctamente.

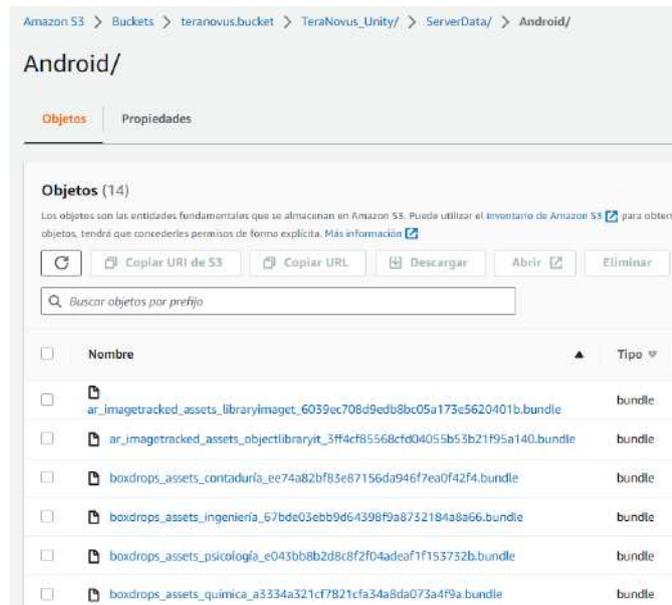


Imagen XCIX: 8.5.2.3: bundles subidos al servidor S3

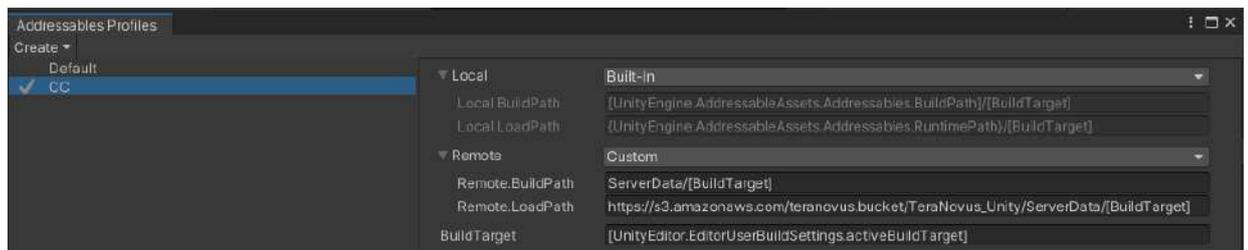


Imagen C: 8.5.2.4: Configuración de conexión servidor/aplicación para la descarga de los bundles

8.5.3 Descarga y obtención: Iconos de cada punto de interés

Las imágenes pueden ser descargadas uno a uno mediante un link con un request de tipo GET siendo el resultado de este la imagen, debido a que es una imagen que ha sido descargada en el tiempo de ejecución y no precargada en la aplicación es necesario tomar ciertas medidas, pues estas imágenes pueden borrarse de la aplicación al salir de la pantalla o escena en la que fueron descargados, se pueden mantener guardándolos en un ScriptableObject, sin embargo, el único lugar en el que serán utilizados será en el sistema AR, fuera de ella solo se utilizara la información descargada en el JSON.

Estas imágenes serán subidas al servidor S3 y el enlace obtenible de cada una de ellas será colocado en el archivo JSON dentro de cada objeto del que sea la imagen justo en el parámetro "urlimage" para los trabajos o dentro el parámetro "iconid" para las ubicaciones, estas imágenes aplican para los trabajos y las ubicaciones. Estas imágenes serán descargadas asincrónicamente en orden desde arriba hacia abajo según la lista de la base de datos del JSON, una vez descargados serán reemplazados en los elementos UI de las ubicaciones y trabajos que tienen de base como imagen el logo de la aplicación

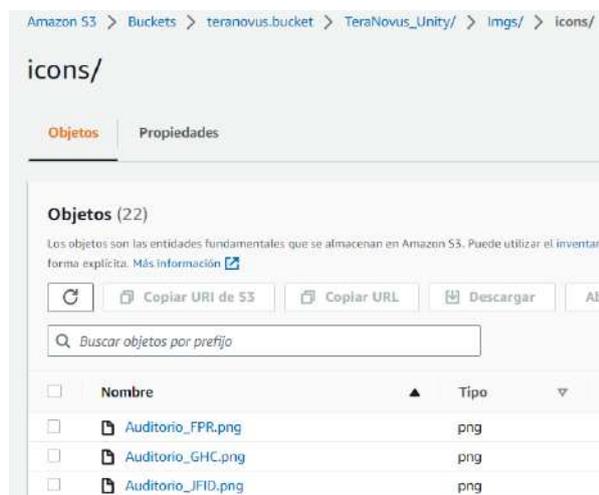


Imagen CI: 8.5.3.1: Imágenes de ubicaciones subidas al servidor S3

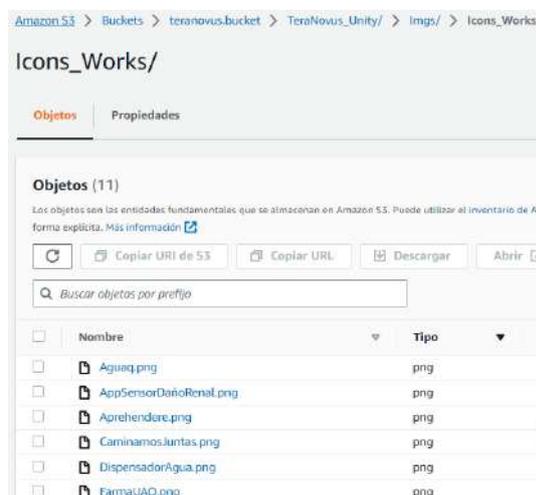


Imagen CII: 8.5.3.2: Imágenes de trabajos subidas al servidor S3

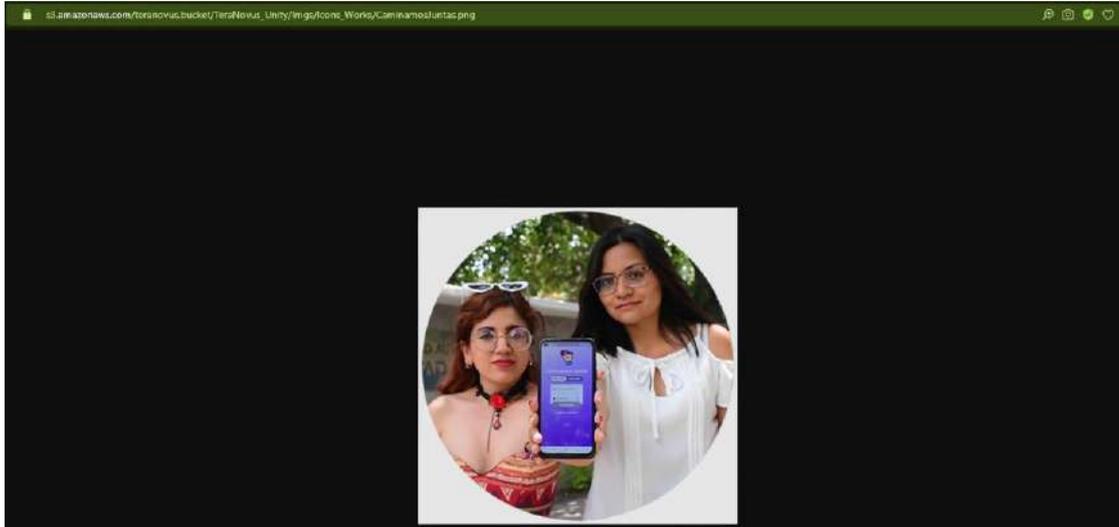


Imagen CIII: 8.5.3.3: Muestra de imagen de trabajo a través del link descargable

```
{
  "elements": [
    {
      "nameEle": "Edificio B (FI)",
      "typeEle": "Ubicaciones",
      "facultad": "Ingeniería",
      "classEle": "Salones",
      "basicdata": {
        "telefono": "",
        "email": "",
        "horario": ""
      },
      "detalles": "Edificio dedicado a los salones de clase para alumnos, cuenta con un baño en la planta baja.",
      "ubicacion": "Se encuentra ubicado a un costado del laboratorio de automatización y a un costado del estacionamiento de ingeniería."
    },
    {
      "worldpoint": {
        "have_ar": true,
        "latitude": 20.591260128378725,
        "longitude": -100.41169670901303,
        "altitude": 1826.069580078125,
        "urlgps": "https://goo.gl/maps/JEChkpPTjf8uouRU6",
        "objectid": "Clase",
        "iconid": "https://s3.amazonaws.com/teranovus.bucket/TeraNovus_Unity/Imgs/icons/EdiB_FI.png"
      },
      "workdata": {
        "fundador": "",
        "urlimage": "",
        "textworkurl": "",
        "urlwork": ""
      }
    }
  ]
}
```

Imagen CIV: 8.5.3.4: Link descargable de la imagen en el JSON

8.5.4 Corrección: Calidad de las imágenes de los iconos de cada punto de interés

Tras una comprobación de las imágenes a través del internet que la misma casa de estudio ofrece para estudiantes e invitados, se ha notado que la velocidad de descarga no es lo suficientemente bueno y menos si dicha velocidad de descarga debe ser dividida entre varias descargas de imágenes y objetos puestas en la lista o la cola de descarga.

Lo que hace que la descarga de las imágenes vaya con algo de lentitud y se termine de colocar todas las imágenes en un aproximado de 2 a 5 minutos según el sitio en el que se utilice.

La propuesta inicial sería reducir el peso de estas imágenes reduciendo el tamaño, sin embargo, estas imágenes perderían calidad y será muy notorio en cuanto el usuario se encuentre más cerca de estas imágenes, por lo que se hizo la propuesta de reducir las imágenes tomadas en una resolución de casi 2K (2048x1080) a la mitad dejándolas en una resolución 1080p.

Aun con esta modificación agregada, la carga de imágenes en la aplicación era lenta, aunque el peso total de las imágenes había sido reducido de 100 MB a casi 60 MB, aún tardaba más allá de los 3 minutos. Y por ello se ha propuesto reducir no la resolución de estas imágenes sino la calidad en profundidad de bits por color de 24 bits a 12 bits y cambiando el formato de compresión a una más apropiada para la web, de este modo se ha reducido el peso total a un aproximado de 15 MB, esta optimización puede descargar todo en un aproximado de 2 minutos o menos, lo cual ya es algo "más" aceptable a los 5 minutos iniciales, sin embargo, reducir más a partir de este punto es perder la calidad de estas imágenes.

8.6 Subida de la aplicación en la Play Store

La Play Store, es una plataforma de distribución digital de aplicaciones móviles para dispositivos con base de sistema Android desarrollada por Google, este ofrece un servicio de gestión y administración para desarrolladores que quieran alojar sus aplicaciones o videojuegos en Play Store, este servicio, Google Console, permite gestionar las distintas versiones por regiones, por tipo de salidas (prueba o producción), precios dentro y de la aplicación misma, códigos promocionales, suscripciones, rendimientos de la aplicación y las estadísticas de la misma.

Cabe recalcar que el servicio de la consola de la Play Store que Google, puede cambiar al leer este documento, puesto que este servicio se encuentra siempre en constante cambio, dejando así una diferencia visual en la interfaz o de funcionalidades mostradas parcial o totalmente distintas a las mostradas en este documento.

8.6.1 Creación y configuración de los ajustes de Play Store

Para alojar una aplicación en la Play Store es necesario compartir o proporcionar todo tipo de información de la aplicación que permita a ellos revisar la aplicación y comprobar que cumpla con las normas de Estado Unidos sin importar desde que país se aloja la aplicación.

Lo primero es entrar en el servicio de Google Console y registrarse como desarrollador, al hacerlo es necesario pagar una cuota de inscripción de 25 dólares, esta cuota es única, no es tratada como una suscripción o cuota mensual o anual, al inscribirse como desarrollador ya podrás alojar aplicaciones y juegos sin más requisitos.



Imagen CV: 8.6.1.1: Botón "Crear aplicación" en Google Console

Al momento de crear una aplicación en la Google Console, debes registrar el nombre de la aplicación, además del lenguaje base y definir las características de esta aplicación, ya sea si es un juego o una aplicación, así como también si es gratis o tiene un costo descargarlo y por último una aceptación de declaraciones de la política del programa para desarrolladores y la exportación de EE.UU. para definir que la aplicación estará sujeta a las leyes de Estados Unidos.

Crear aplicación

Detalles de la aplicación

Nombre de la aplicación	<input type="text"/>
	<small>Este es el nombre que tendrá tu aplicación en Google Play. 0/90</small>
Idioma predeterminado	<input type="text" value="Inglés (Estados Unidos) - en-US"/>
Aplicación o juego	<small>Puedes cambiar esta opción más tarde en Configuración de la tienda</small> <input type="radio"/> Aplicación <input type="radio"/> Juego
Gratis o de pago	<small>Puedes editar esta información más tarde en la página de aplicación de pago</small> <input type="radio"/> Gratis <input type="radio"/> De pago

Imagen CVI: 8.6.1.2: Creación de aplicación, detalles

Tras crear el espacio de la aplicación es necesario llenar toda la información de la aplicación, Google Console tiene 11 puntos de información (2023-1).

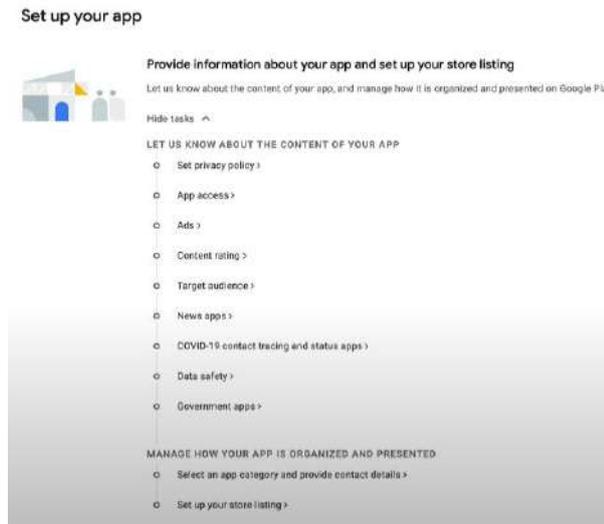


Imagen CVII: 8.6.1.3: Configuración: Información de la aplicación

- I. La política de privacidad se puede generar mediante un servicio de FireBaseApp "App Privacy Policy Generator" que, al dar información de contacto, además de seleccionar que servicios son utilizados, te generará un texto de la política de privacidad. Esto se utiliza y se agrega a una página creada en Sites, un servicio de Google. Esta página se agrega a la información necesaria.

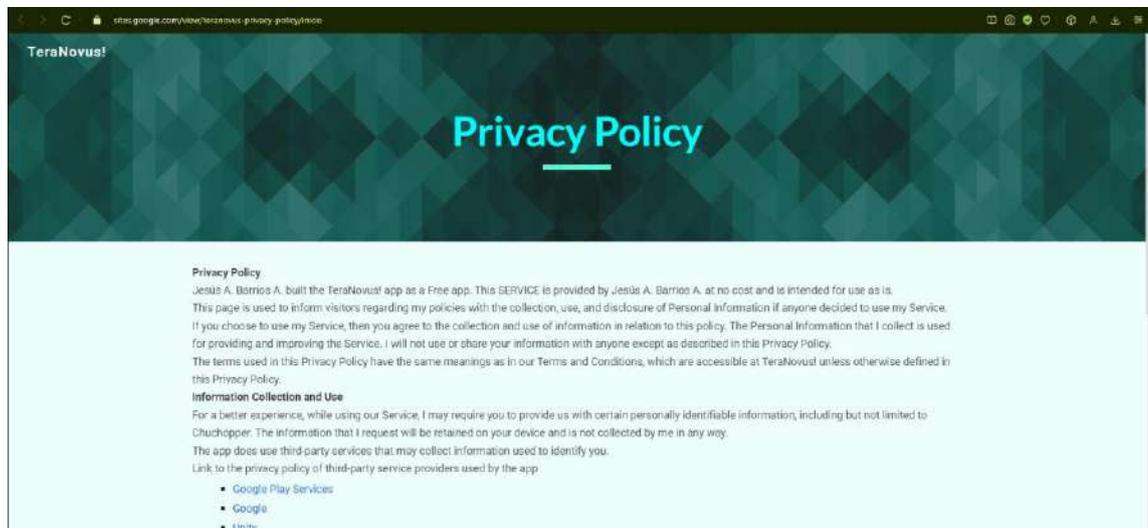


Imagen CVIII: 8.6.1.4: Políticas de privacidad

- II. En el acceso a la aplicación se define si esta aplicación tiene restricciones a las funcionalidades de la aplicación a través de un acceso de usuario. En el caso de la aplicación puedes entrar como un usuario invitado, por lo tanto, no cuenta con una restricción.
- III. El apartado de anuncios es para definir si la aplicación utiliza anuncios o muestra anuncios, esta aplicación por el momento no tiene anuncios.
- IV. En la clasificación del contenido, es necesario contestar un cuestionario que permitirá generar una clasificación internacional a tu aplicación, en el cuestionario se agrega un contacto de correo para que ellos se puedan poner en contacto contigo si es necesario, se selecciona primero la categoría, luego se define si el contenido de la aplicación está sujeto a clasificación, es decir, si tiene contenido ofensivo o contenido maduro, también se define si el usuario es conectado con una comunidad de la aplicación o si este tendrá una comunicación dentro de la aplicación con otros usuarios así como si la aplicación promociona o vende productos, si es un navegador web o si la aplicación es informativa o educativa. Al terminar el cuestionario, se da una clasificación de la aplicación en las distintas clasificaciones de contenido como la ESRB.

Imagen CIX: 8.6.1.5: Clasificación del contenido: cuestionario

- V. En el apartado de la audiencia objetivo, se define la edad objetivo de la aplicación, ya que se definió la aplicación como una "aplicación informativa o educativa" puedes definir si la aplicación puede ser revisada por profesores que revisan el contenido educativo, puesto que esta aplicación no tiene contenido educativo se deniega esa revisión.
- VI. El apartado de aplicación de noticias es para ayudar a definir si la aplicación es una aplicación que contiene contenido informativo de tipo noticias. Por el momento no tiene noticias y solo se ha colocado como un posible extra en la aplicación.
- VII. El apartado de aplicación de rastreo de contactos y de estado de Covid-19, define si la aplicación es una aplicación que recopila ubicaciones y contactos para mostrar la densidad de personas en ciertas ubicaciones o si este mostrará los estados del Covid-19, en este caso no es ese tipo de aplicación.
- VIII. En el apartado de la seguridad de datos, es necesario definir qué tipo de información recolecta la aplicación, debido a la utilización de varios servicios de Google, hay datos que están siendo recolectados y estos los dice Google en la página del servicio de ARCore, en este se comenta que utiliza datos como los datos de la ubicación precisa y la cámara además de datos como el ID del usuario y del dispositivo así como el diagnóstico del rendimiento del dispositivo sobre la aplicación, el uso de la API y la actividad de la aplicación para ayudar a mejorar las características implementadas en ARCore.
- IX. El último apartado del tipo de contenido que tiene la aplicación es definir si la aplicación está hecha por una institución gubernamental.
- X. El primer apartado de la sección de la configuración de como se muestra la aplicación en la tienda de Play Store es seleccionar una categoría y etiquetas que definan a tu aplicación, es este caso ha

sido seleccionado la categoría de "Mapas y navegación" así como también las etiquetas de:

- a) Mapas y navegación
- b) Herramientas
- c) Productividad

- XI. El último apartado de Ficha de Play Store se colocan las imágenes de muestra, así como videos demos y las imágenes de logo y banner, este apartado es el más importante visualmente, ya que es la página de la aplicación que los usuarios verán a través de Play Store. Para ello se han tomado capturas de pantalla de los funcionamientos de la aplicación y se ha subido un video mostrando las funcionalidades sin AR y con la funcionalidad AR.

8.6.2 Configuración del proyecto de Unity y exportación de la aplicación

El software Unity 3D que es un motor de videojuegos gratuito siempre que cumplas las condiciones, es capaz de exportar un juego o aplicación a numerosas plataformas, algunas de estas plataformas hay que cumplir con algunas condiciones y hace que ya no sean gratuitas las licencias, sin embargo, el exportar la aplicación a PC y a teléfonos móviles Android es gratuito.

Para exportar un juego o aplicación para teléfonos móviles es necesario tener el módulo de Android en el editor de Unity en el que se trabaje dicha aplicación, este módulo instalara el SDK de Android, además de otras herramientas necesarias para hacer posible la conversión de la aplicación al sistema Android.

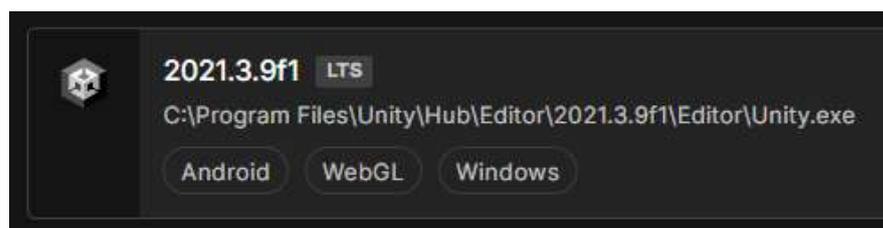


Imagen CX: 8.6.2.1: Versión de editor de Unity 3D y sus módulos

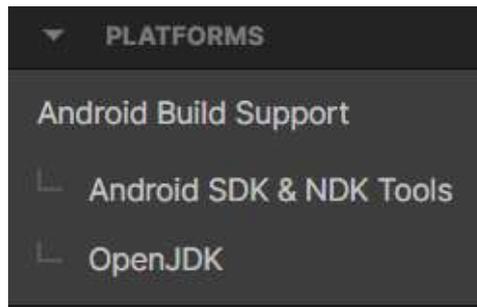


Imagen CXI: 8.6.2.2: Módulo Android de Unity

Una vez que se tenga este módulo instalado en el editor de Unity, es necesario cambiar o tener ciertas configuraciones necesarias para hacer una excelente exportación de la aplicación. Empezando por el panel de "Build Settings", en él se deben agregar todas las escenas que necesitemos en la aplicación final, además de tener la plataforma seleccionada de "Android":

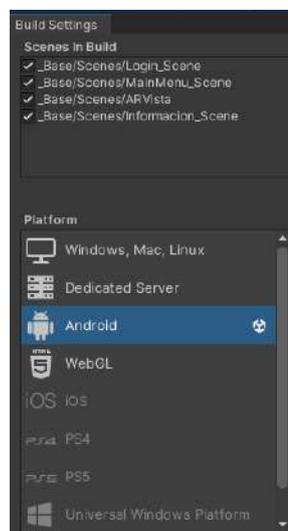


Imagen CXII: 8.6.2.3: Build Settings

Teniendo estas bases completadas, es necesario cambiar los ajustes básicos y algunos avanzados en el panel "Project Settings", ajustes básicos como el icono de la aplicación, el nombre del producto que aparecerá en la parte inferior del icono de la aplicación:

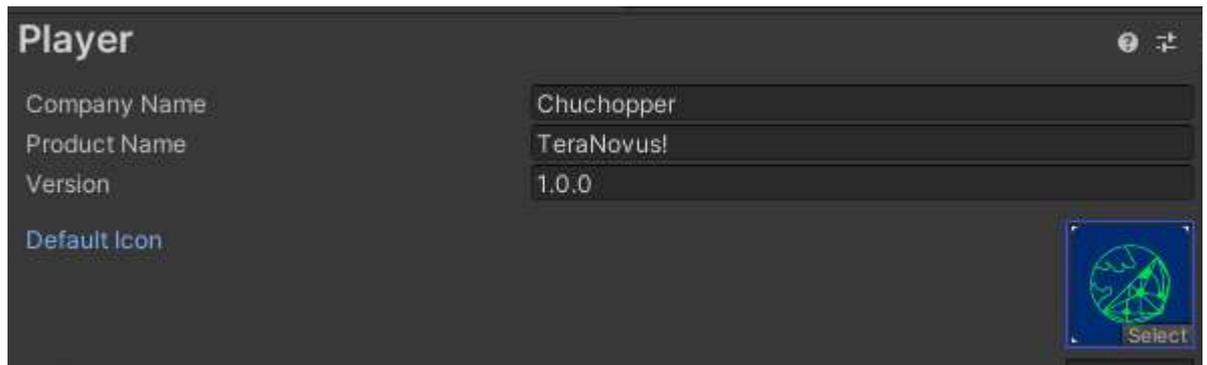


Imagen CXIII: 8.6.2.4: Project Settings: ajustes básicos

Mientras que los ajustes avanzados están divididos entre tres secciones:

- I. "Icon": ajustes referentes a los distintos tamaños en px (píxeles) para los diversos tamaños de muestra de la aplicación.
- II. "Resolution and Presentation": ajustes que definen la resolución, la orientación, el modo pantalla completa o si es posible modificar el tamaño de la aplicación.
- III. "Splash Image": Es la "intro" de marca de agua que Unity agrega en licencias gratuitas, también puede ser agregado un logo.

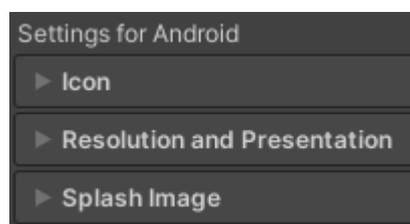


Imagen CXIV: 8.6.2.5: Project Settings: ajustes avanzados

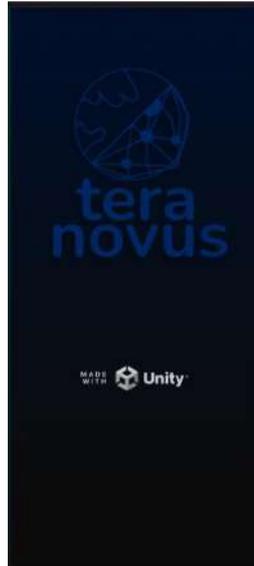


Imagen CXV: 8.6.2.6: "Splash image"

Una vez agregado los iconos de distintos tamaños en píxeles y haber cambiado varios ajustes, es necesario crear una Build especial para Google, es decir, una Build app bundle y ya se puede tener la Build de la aplicación o la aplicación exportada para subir a Google Console.

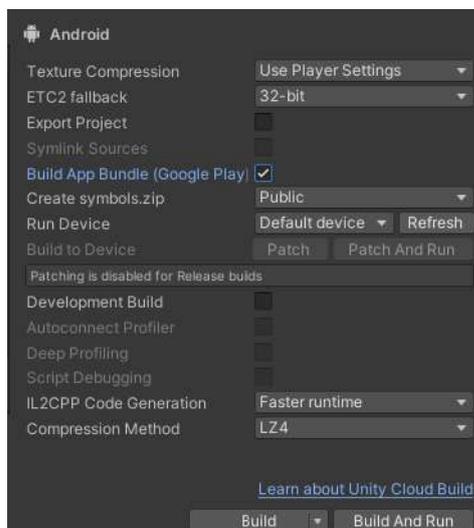


Imagen CXVI: 8.6.2.7: Build setting: Build App Bundle

8.6.3 Subida de aplicación a Google Console

Para subir archivos a la Google Console es necesario primero saber qué tipo de salida será, si la aplicación será lanzada primero con una salida de prueba (testers) o si esta saldrá oficialmente. En este caso, el proyecto que contenía cierto tiempo corto y un corto personal, se tiene una aplicación de prueba que muestre todas las funcionalidades principales de la aplicación, por ello se considerara como una salida de prueba, entre estas salidas de prueba disponibles en Google Console están:

- I. Prueba abierta: aplicación disponible para tester, cualquier usuario de la Play Store puede acceder.
- II. Prueba cerrada (Alpha): aplicación (preliminar) disponible para un grupo de testers específicos a través de canales.
- III. Prueba interna: aplicación disponible para un grupo más reducido de testers específicos.
- IV. Registro previo: publica la ficha de la aplicación, pero aún no es descargable, el usuario interesado puede hacer un registro previo para recibir una notificación para descargar la aplicación.

Para que la aplicación pueda ser utilizada y probada por distintas personas no específicas, se necesita hacer una prueba abierta, debido a eso, hay una serie de ajustes extras que necesitan hacerse.

Prueba abierta

Telefonos, Tablets, Chrome OS

Crear nueva versión

Crea y gestiona versiones de pruebas abiertas para que tu aplicación esté disponible para los testers. Cualquier usuario podrá unirse a las pruebas en Google Play. [Más información](#)

Configurar canal de prueba abierta

Inactivo

Completadas: 1 de 4

CONFIGURA TU CANAL

Seleccionar países

Seleccionar testers

CREA Y LANZA UNA VERSIÓN

Crear un nuevo lanzamiento

Revisa y confirma la versión

Envía la versión a Google para que se revise

Versiones

Países o regiones

Testers

Versiones

No hay versiones

Crear nueva versión

Imagen CXVII: 8.6.3.1: Google Console: Prueba abierta

- I. Administrar países en los que estará la aplicación, ya que la aplicación tiene funcionalidades importantes dentro de México, no es necesario que este fuera de este.
- II. Las pruebas van gestionadas por versiones, por lo que es necesario crear una nueva versión donde agreguemos ahí la Build de la aplicación obtenida por Unity 3D, lo que se agregará será el archivo ".aab" primero, el ".aab" contiene el juego. Después de subir el archivo ".aab" se soltará cierta información, donde se podrá agregar el archivo ".zip" de "Symbols" que contiene la información de debugs que permite a Google tener mejor información cuando la aplicación entre en un estado de "crash"

App bundles

Arrastra aquí los app bundles que quieras subir

[Subir](#) [Añadir de la biblioteca](#)

Tipo de archivo	Versión	Niveles de API	SDK objetivo	Diseños de pantalla	ABIs	Funciones obligatorias
App bundle	1 (1.0.0)	24 y posterior	33	4	2	8

Imagen CXVIII: 8.6.3.2: Google Console: App Bundles

Una vez que se suban estos archivos a Google Console, obtienes un resumen general o detallado que tiene toda la información agregada y explica todos los ajustes que has hecho a la aplicación, también muestra cuántos dispositivos Android son compatibles con la aplicación. Una vez que se llegue a este paso es momento de subir los cambios para su revisión.



Imagen CXIX: 8.6.3.3: Google Console: Resumen versión

Imagen CXX: 8.6.3.4: Google Console: Botón revisiones

Google hará una revisión en ciertos dispositivos Android, donde además de pruebas de seguridad, serán pruebas que determine si la app cumple con todos los requisitos incluso de accesibilidad como el tener un total de 48px en la altura y la anchura en un elemento UI interactivo.



En caso de tener algún problema en algún aspecto de la aplicación Google te lo notificará en el apartado de "Informe previo al lanzamiento" donde se dividirá en cuatro secciones:

- I. Estabilidad
- II. Rendimiento
- III. Accesibilidad
- IV. Seguridad y confianza

Que al mismo tiempo serán divididos en advertencia o peligro.

En cuanto se termine de revisar, que puede tardar un aproximado máximo de 7 días, Google avisara por correo que ya está listo la aplicación alojada.

✔ Disponible para un número limitado de testers · 1 código de versión

Mostrar resumen ▾ Promocionar versión ▾

Imagen CXXI: 8.6.3.5: Aceptación de aplicación revisada en Google Console

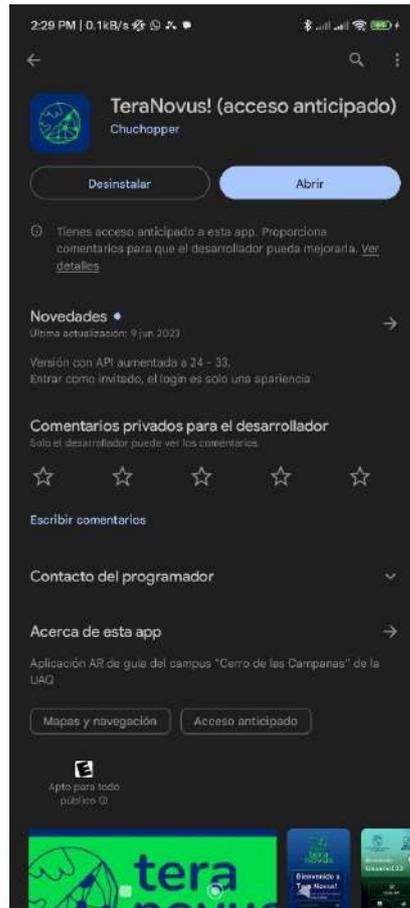


Imagen CXXII: 8.6.3.6: Ficha en la Play Store

IX. CONCLUSIONES Y RECOMENDACIONES

El servicio de ARCore, solo es compatible con ciertos celulares, sin embargo, esta lista seguirá creciendo conforme nuevos celulares lleguen al mercado, dejando así la posibilidad de que en un futuro todos tengan acceso a este servicio. "<https://developers.google.com/ar/devices>" (Google, 2023)

Además, el servicio de VPS, dentro del AR Foundation de Google, necesita ciertos datos, al menos para tener un mejor detalle del entorno tridimensional del espacio y así saber ubicar mejor al usuario y a los objetos. Estos datos no son importantes del todo, pero debemos considerarlos dado que estos datos permiten posicionar mejor visualmente los objetos en el mundo real a través de la aplicación según los relieves, paredes, obstáculos y los diferentes niveles de piso que tenga el espacio real.

Debiendo considerar asimismo que esta aplicación que guarda datos como la ubicación e imágenes para Google, datos que sin duda ayudarán a mejorar en gran medida el servicio de VPS en el entorno del campus de la universidad justo en las zonas que no existen estos datos hoy en día, por lo que la utilización a través de una gran cantidad usuarios permitirá corregir esta limitante actual, aunque durante las etapas de pruebas se presentaron algunas dificultades por esta situación he de concluir que esta no fue mayor, solo se agudizó un poco al querer reconocer los puntos de ubicación con mayor precisión, toda vez que estos no están centradas o ubicadas en donde deberían de estar por un rango de error aproximado de entre 30 cm a 3 metros.



Imagen CXXIII: 9.1.1 Canchas de básquetbol, facultad de Química



Imagen CXXIV: 9.1.2 Vista satelital, canchas de básquetbol, facultad de Química



Imagen CXXV: 9.1.3 Vista Street View PharmaUAQ



Imagen CXXVI: 9.1.4 Vista satelital del edificio Biotecnológico

Sin embargo, algo que generalmente sucede dependiendo del sistema o celular que se utilice incluso arreglando este problema, seguirá existiendo un rango de 1 a 2 metros de error como mínimo según la cantidad y calidad de los sensores GPS, esto debido a la comunicación entre los sensores GPS de los teléfonos inteligentes y de los satélites GPS.

Para finalizar he de decir que mi investigación es un buen avance y el precursor de una APP que puede emplearse en beneficio de los propios estudiantes de la universidad sobre todos de los foráneos, así como de las personas que tiene que acudir al campus y que no están inmersas o familiarizadas con el mismo, pudiendo ser una herramienta que facilite sus tareas y actividades, asimismo sería útil para la propia universidad a fin de promocionar a sus estudiantes y a la sociedad en general puntos de interés que den a conocer los logros y actividades de la misma; por lo cual sin duda su creación esta más que justificada por lo beneficiosa que pudiera llegar a ser.

Las recomendaciones para futuros cambios a este proyecto en base a retroalimentaciones, consejos e ideas que se fueron recopilado a lo largo del desarrollo, es que esta aplicación pueda dar una guía del paso a paso para los tramites , diciéndole a alumno o usuario que pasos son necesarios y en qué orden deberían hacerse para completar ese trámite que el usuario desea realizar.

Claro, este cambio es sumamente importante y largo, puesto que se deberá dirigir a alumno a los diferentes puntos importantes donde este debería hacer su trámite, guardando el progreso de este trámite en la aplicación y estos puntos dependerá de la facultad a la que pertenezca, agregando así nuevos puntos a la aplicación. Sin considerar que esta es la información que más actualización debe tener, pues un paso que no esté agregado o que esté mal informado puede dificultar al usuario hacer su trámite.

Además, es recomendable cambiar la base de datos creada en texto a una base de datos relacional, de este modo se puede modificar de forma más rápida la base de datos, en vez de cambiar el archivo con los datos modificados.

De ser posible, sería bueno que esta aplicación mostrará los datos que se muestran en el portal de la universidad como las calificaciones, materias que actualmente se está cursando y la lista de materias.



Como extra, se puede utilizar la herramienta ARCore más a fondo para crear juegos que requieran de dos estudiantes como mínimo, como el ajedrez, dominó, etc.

Finalizando con una mejora de detalles visuales como efectos de color en la interfaz, VFX a los modelos, guía y acciones en el sistema AR y mejora a la interfaz visual.

Estos cambios pueden ayudar a esta aplicación a tener una mejor base con funcionalidades diversas que pueden ayudar al usuario.

X. BIBLIOGRAFÍAS

- Amazon. (2022). *¿Qué es AWS?* Recuperado el 25 de Septiembre de 2022, de AWS:
<https://aws.amazon.com/es/what-is-aws/>
- Amazon. (2022). *¿Qué es una API?* Recuperado el 25 de Septiembre de 2022, de AWS:
<https://aws.amazon.com/es/what-is/api/>
- Amazon. (2022). *Microservicios*. Recuperado el 25 de Septiembre de 2022, de AWS:
<https://aws.amazon.com/es/microservices/>
- Canva. (s.f.). *La psicología del color: el significado de los colores y cómo aplicarlos a tu marca*.
Obtenido de Canva: https://www.canva.com/es_mx/aprende/psicologia-del-color/
- Fundación Wikimedia Inc. (12 de Septiembre de 2022). *Teléfono inteligente*. Recuperado el 25 de Septiembre de 2022, de Wikipedia:
https://es.wikipedia.org/wiki/Teléfono_inteligente
- Google. (2023). *ARCore supported devices*. Obtenido de ARCore supported devices:
<https://developers.google.com/ar/devices>
- Google, LLC. (3 de Junio de 2015). *Sistemas de geolocalización*. Recuperado el 25 de Septiembre de 2022, de Google Maps:
https://www.google.com/maps/d/embed?mid=13kxZTfA8HyYkBbNgwPsFtBvQbEU&hl=en_US&ehbc=2E312F
- Google, LLC. (6 de Febrero de 2020). *A look back at 15 years of mapping the world*.
Recuperado el 25 de Septiembre de 2022, de Blog Google:
<https://www.blog.google/products/maps/look-back-15-years-mapping-world/>
- Google, LLC. (2022). *ARCore: Anclajes geoespaciales*. Recuperado el 25 de Septiembre de 2022, de Google Developers: <https://developers.google.com/ar/develop/cloud-anchors?hl=es-419>
- Google, LLC. (2022). *Descripción general de ARCore y entornos de desarrollo compatibles*.
Recuperado el 25 de Septiembre de 2022, de Google Developers:
<https://developers.google.com/ar/develop>
- IFT, I. F. (17 de Febrero de 2020). *En México hay 80.6 millones de usuarios de internet y 86.5 millones de usuarios de teléfonos celulares: ENDUTIH 2019. 17 de febrero*. Recuperado el Septiembre de 25 de 2022, de Instituto Federal de Telecomunicaciones:
<https://www.ift.org.mx/comunicacion-y-medios/comunicados-ift/es/en-mexico-hay-806-millones-de-usuarios-de-internet-y-865-millones-de-usuarios-de-telefonos-celulares>
- Microsoft. (2022). *¿Qué es el desarrollo de aplicaciones móviles?* Recuperado el 25 de Septiembre de 2022, de Microsoft Azure: <https://azure.microsoft.com/es->

es/resources/cloud-computing-dictionary/what-is-mobile-app-development/#definition

Microsoft. (2022). *¿Qué es la realidad aumentada o AR?* Recuperado el 25 de Septiembre de 2022, de Microsoft Dynamics 365: <https://dynamics.microsoft.com/es-es/mixed-reality/guides/what-is-augmented-reality-ar/>

Microsoft. (2022). *¿Qué es un servidor en la nube?* Recuperado el 25 de Septiembre de 2022, de Microsoft Azure: <https://azure.microsoft.com/es-es/resources/cloud-computing-dictionary/what-is-a-cloud-server/#what-is-a-cloud-server>

PIEE, P. d. (2022). *Programa de Inclusión y Equidad Educativa - PIEE - "La UAQ somos todos" (tríptico)*. Recuperado el 25 de Septiembre de 2022, de Dirección de Planeación y Gestión Institucional UAQ: <https://planeacion.uaq.mx/docs/piee/Folleto%20-%20PIEE%20UAQ.pdf>

PMI's Pulse of the Profession. (2018). *Success in Disruptive Times*. PMI, Project Management Institute. Recuperado el 25 de Septiembre de 2022, de <https://www.pmi.org/-/media/pmi/documents/public/pdf/learning/thought-leadership/pulse/pulse-of-the-profession-2018.pdf>

Santander Universidades. (21 de Diciembre de 2020). *Metodologías de desarrollo de software: ¿qué son?* Recuperado el 26 de Septiembre de 2022, de Santander Becas: <https://www.becas-santander.com/es/blog/metodologias-desarrollo-software.html#:~:text=Las%20metodologías%20de%20desarrollo%20de%20software%20son%20un%20conjunto%20de,diseñar%20soluciones%20de%20software%20informático.>

SINC. (3 de 12 de 2013). *GPS, la tecnología de localización que empezó con la carrera espacial*. Recuperado el 25 de Septiembre de 2022, de SINC: [https://www.agenciasinc.es/Visual/Ilustraciones/GPS-la-tecnologia-de-localizacion-que-empezo-con-la-carrera-espacial#:~:text=La%20historia%20del%20GPS%20\(Global,por%20su%20señal%20de%20radio.](https://www.agenciasinc.es/Visual/Ilustraciones/GPS-la-tecnologia-de-localizacion-que-empezo-con-la-carrera-espacial#:~:text=La%20historia%20del%20GPS%20(Global,por%20su%20señal%20de%20radio.)

Singh, S. (2006). *Impact of color on marketing*. Winnipeg, Canada: Emerald Group Publishing Limited.

The CIU, T. C. (11 de Octubre de 2021). *Mercado de Smartphones en México al 2T-2021: Reconfiguración Competitiva a la Vista*. Recuperado el 25 de Septiembre de 2022, de The CIU: <https://www.theciu.com/publicaciones-2/2021/10/11/mercado-de-smartphones-en-mxico-al-2t-2021-reconfiguracin-competitiva-a-la-vista>

UAQ, U. A. (2021). 3er Informe / Teresa García Gasca. *Informes de Rectoría*, 18.

UAQ, U. A. (2022). *Directorio Institucional*. Recuperado el 26 de Septiembre de 2022, de Universidad Autónoma de Querétaro: <https://directorio.uaq.mx>



- UAQ, U. A. (2022). *Historia*. Recuperado el 25 de Septiembre de 2022, de Universidad Autónoma de Querétaro: <https://www.uaq.mx/index.php/conocenos/sobre-la/historia>
- UAQ, U. A. (2022). *Misión*. Recuperado el 25 de Septiembre de 2022, de Universidad Autónoma de Querétaro: <https://www.uaq.mx/index.php/conocenos/sobre-la/mision>
- Unity Technologies. (2022). *About AR Foundation*. Recuperado el 25 de Septiembre de 2022, de Unity Docs.
- Unity Technologies. (2022). *Formatos 3D*. Recuperado el 26 de Septiembre de 2022, de Unity Documentation: <https://docs.unity3d.com/es/530/Manual/3D-formats.html>
- Wikipedia. (2022). *Metaverso*. Recuperado el 25 de Septiembre de 2022, de Wikipedia: <https://es.wikipedia.org/wiki/Metaverso>
- Wikitude. (12 de Abril de 2020). *Image Recognition & Tracking: best practices and target guidelines*. Recuperado el 26 de Septiembre de 2022, de Wikitude see more.: Image Recognition & Tracking: best practices and target guidelines