



Benemérita Universidad Autónoma de Puebla.

Facultad de Ciencias de la Computación.

**Recorrido del Centro Histórico de la Ciudad de Puebla con
Realidad Aumentada.**

T E S I S

Para obtener el título de:

Licenciado en Ingeniería en Ciencias de la Computación.

PRESENTA:

SALVADOR ARTURO RAMIREZ ESCAMILLA.

ASESORES DE TESIS:

DR. JUAN MANUEL GONZÁLEZ CALLEROS.

DRA. JOSEFINA GUERRERO GARCÍA.

Diciembre, 2024.

DEDICATORIA / AGRADECIMIENTOS.

Dedico este trabajo a mi madre Rosaura, porque siempre ha creído en mí, incluso cuando yo mismo he dudado de mis capacidades, gracias por siempre haberme guiado por el camino correcto y a pesar de tantos tropiezos, siempre me ayudó a regresar al camino correcto, que a pesar de las dificultades nunca se dejó vencer y me mostró que lo difícil no es caer, si no levantarse; te doy gracias por todo lo que has hecho por mí y te aseguro que, si no fuera por todo eso, jamás hubiera logrado nada.

Hoy puedo decir que, gracias a todo ese apoyo, a todas esas llamadas que me motivaban a seguir en la universidad, cuando creí que todo iba a salir mal, que el camino que elegí era el incorrecto, cuando me sentía solo en una ciudad que no conocía, gracias a todo eso puedo decir que valió la pena.

Agradezco al Ing. Camilo F. Isaías Vaquier Gonzales y a la enfermera Anita Alvarado Escamilla, por haberme recibido en su casa y haberme cobijado durante el tiempo que fuera necesario, por los consejos, por las tardes de risas y buenos momentos que pasé en mi corta etapa viviendo con ellos, por siempre estar al pendiente y recibiendo ánimo de ellos, porque sin nada de eso no hubiera conseguido nada de esto.

También quiero agradecer al Doctor Juan Manuel González Calleros, quien no me dejó tirar por la borda este trabajo y siempre estuvo apoyándome y guiándome por el camino correcto para la culminación de este, por los recursos que me proporcionó y por siempre estar disponible para escuchar las ideas que le fui presentando durante todo el proceso de este trabajo.

De igual manera agradezco a quienes considero mis mejores amigas, Psic. Marisol Patricio Sánchez y L.R.I. Yadira Franco Santos que me estuvieron apoyando en todo momento, en el proceso de experimentación de este proyecto, gracias por apoyarme con sus consejos, consiguiendo gente para las pruebas y escuchándome cuando sentía que no podía terminar con esto, ayudándome en los momentos más pesados de este proceso, por las charlas y los ratos en los que salíamos a dar la vuelta para despejarnos y por jamás dejar nuestra amistad a la deriva a pesar de las dificultades que hemos atravesado, gracias porque a pesar del tiempo y la distancia siempre han estado para mí.

A las personas que se animaron a realizar los experimentos conmigo, gracias por el tiempo que me dedicaron, por los comentarios que aportaron y por las ideas que me fueron proporcionando durante esas tardes en las que llevamos a cabo las pruebas de la aplicación, también gracias por confiar en el desarrollo que realizó un servidor y por soportar las miradas de las personas que nos rodeaban sin entender el objetivo de nuestro recorrido, mientras veían que enfocaban sus pantallas sin entender que no solo enfocaban contenido, sino que enfocaban los sueños y aspiraciones de un servidor.

Agradezco a mi pariente el médico Alejandro, que estuvo acompañándome siempre en todo el camino de la universidad, cuando nos cansábamos o nos agotábamos de la vida universitaria y estuvo ahí para escucharme, para platicar, siempre pensando en proyectos en los que pudiésemos combinar nuestras carreras y por siempre tenderme una mano cuando lo necesite.

A mis familiares, que, si bien no los he nombrado, saben que les viviré siempre agradecido, por estar al pendiente de mi camino en este logro, por apoyarme en todas las dificultades que se me fueron presentando y por siempre estar dispuestos a brindarme ese apoyo en el momento en el que lo necesitara.

Por último, pero no menos importante agradezco a Dios, por darme la capacidad y motivación para realizar y superar cada obstáculo que se me ha ido presentando a lo largo de mi vida, porque sin importar cuan perdido me sienta, sé que siempre está a mi lado y eso me ha ayudado a no rendirme nunca.

Gracias.

RESUMEN

“La cultura es uno de los principales motivos por los cuales se inicia un viaje, ya que ofrece al individuo satisfacciones perdurables que rebasan los límites de la simple diversión” [1], el día 11 de diciembre de 1987, el centro histórico de la ciudad de Puebla fue declarado Patrimonio Cultural de la Humanidad por la UNESCO (Organización de las naciones unidas para la ciencia y la cultura). Actualmente, los turistas visitan el centro histórico de Puebla, aunque se llevan una gran impresión por lo bello de los edificios, muchas personas desconocen realmente la historia que contiene el centro histórico, ya que por lo general los turistas no contratan o no realizan sus viajes de la mano de un guía de turistas, y no pueden tener acceso fácil a la información cultural del centro histórico.

Para aumentar el turismo, algunos lugares y gobiernos utilizan algunas estrategias como crear rutas centradas en temas específicos, como la gastronomía, la historia, la arquitectura o el arte. También eventos culturales, festivales de música, ferias gastronómicas o exposiciones temáticas puede atraer visitantes interesados en experiencias únicas. Los festivales anuales, como carnavales o fiestas locales, pueden convertir una ciudad en un destino turístico recurrente. Algunos más utilizan tecnologías como la realidad aumentada (RA) o la realidad virtual (RV) para enriquecer la experiencia turística. Por ejemplo, en sitios históricos, los visitantes podrían ver representaciones de épocas pasadas mediante RA.

“Cada 10 o 15 años, hay una nueva plataforma informática importante. Y ahora estamos empezando a prepararnos para las plataformas del mañana. Con mucho, la plataforma del futuro más atractiva gira en torno a la visión virtual, o la modificación de lo que vemos para crear experiencias aumentadas e inmersivas” [2]; es por ello por lo que debemos apoyarnos de estas nuevas tecnologías, así como adaptar estas experiencias a las tecnologías de fácil acceso para las personas, como lo son los dispositivos móviles.

“La realidad aumentada es una nueva ventana a través de la cual se puede ver el mundo enriquecido” [3], es decir que, la realidad aumentada permite mostrar objetos digitales combinados con el entorno real y físico a través de un dispositivo electrónico con el fin de modificar la percepción física del usuario por medio de la cual se le puede proporcionar bastante información al usuario, sin necesitar de acceder a libros o consultar la información previa, además de que se

vuelve una manera llamativa para que las personas se interesen en conocer ciertos lugares importantes, en este caso del centro histórico de la ciudad de Puebla.

“La realidad aumentada es una prometedora tecnología, ya presente en muchas aulas, que puede ayudar a mejorar el proceso de enseñanza-aprendizaje” [4]. La realidad aumentada es y será una herramienta muy importante y eficaz para apoyar la enseñanza en ciertas áreas como la educación, la arquitectura, el diseño, etc. En esta ocasión aplicaremos la realidad aumentada en el turismo, ya que mediante su aplicación se tratará de motivar a los turistas para que visiten distintos puntos importantes para el centro histórico de la ciudad de Puebla al proporcionar datos interesantes para conocer y obtener más información cultural.

La combinación de diferentes estrategias puede hacer que una ciudad sea más atractiva para los turistas, generando ingresos económicos y fortaleciendo la identidad local. Al crear experiencias auténticas, mejorar la infraestructura, y fomentar una promoción digital efectiva, las ciudades pueden atraer tanto a turistas nacionales como internacionales, contribuyendo al crecimiento sostenible del turismo local. Este trabajo de tesis propone el desarrollo de una aplicación creada con Unity que considera el uso de RA como estrategia para atraer la atención de las personas, pun una aplicación creada con Unity.

“Entre los recursos que se han utilizado para la consecución de esta nueva orientación de los espacios museísticos, las nuevas Tecnologías de la Información y Comunicación (TIC), se presentan como un importante filón por el grado de cotidianidad que presentan para la sociedad actual, ofreciendo importantes resultados en combinación con las técnicas museográficas tradicionales.” [5], gracias a esto nos podemos dar cuenta que la implementación de la realidad aumentada para el descubrimiento de nueva información es una opción favorable, ya que todo se vuelve más digerible e interesante con una aplicación que nos dé la oportunidad de tener acceso a la información de manera rápida, además de ser una forma interesante e innovadora que podría llamar la atención a la gente y podría crear curiosidad para que no solo personas de otros estados o países conozcan el centro histórico de la ciudad de Puebla, sino también invitar y lograr que las mismas personas del estado se interesen por lugares con gran riqueza cultural que contiene el centro de nuestra ciudad.

ÍNDICE.

RESUMEN.....	3
ÍNDICE.....	5
ÍNDICE DE IMÁGENES.....	8
ÍNDICE DE GRÁFICAS.....	13
ÍNDICE DE TABLAS.....	13
CAPÍTULO 1. INTRODUCCIÓN.....	14
1.1 MOTIVACIÓN Y JUSTIFICACIÓN.....	14
1.2. OBJETIVOS.....	15
1.2.1 <i>Objetivo general.</i>	15
1.2.2.- <i>Objetivos específicos.</i>	15
1.3. MAPA DE LECTURA DE LA TESIS.	16
CAPÍTULO 2. ESTADO DEL ARTE.....	17
2.1.- ¿QUÉ ES LA REALIDAD AUMENTADA?	18
2.2.- LA REALIDAD AUMENTADA EN EL SECTOR TURÍSTICO.....	20
2.3.- DISPOSITIVOS PARA UTILIZAR REALIDAD AUMENTADA.....	26
2.6.- HERRAMIENTAS PARA CREAR REALIDAD AUMENTADA.....	31
2.4.- HERRAMIENTAS DE MODELADO.....	34
2.5.- MOTORES DE VIDEOJUEGOS.....	37
CAPÍTULO 3. DISEÑO DE LA APLICACIÓN.....	41
3.1.- DIAGRAMA DE COMPONENTES.....	41
3.2.- DIAGRAMA DE SECUENCIA.....	45
3.3.-DISEÑO DEL EXPERIMENTO Y APLICANDO SUS.....	49
3.4.- ESCENARIOS DEL EXPERIMENTO.....	56
CAPÍTULO 4. SOLUCIÓN PARA EL RECORRIDO CON REALIDAD AUMENTADA.....	59
4.1.- INTRODUCCIÓN.....	59
4.2.- DEFINICIONES NECESARIAS.....	59
4.2.1.- <i>Elementos y componentes de Unity.</i>	59
4.2.2.- <i>Definiciones de instrucciones dentro de los scripts.</i>	61
4.2.3.- <i>Definiciones ajenas a Unity.</i>	63

4.3.- CONOCIENDO UNITY.....	64
4.4.- CREACIÓN DEL PROYECTO.....	66
4.5.-VOLVIENDO LA INTERFAZ RESPONSIVA.....	68
4.6.- TÉRMINOS Y CONDICIONES.....	69
4.7.- REGISTRAR USUARIO.....	74
4.7.1.- <i>Envío de correo electrónico.</i>	80
4.8.- INTEGRACIÓN DE BASES DE DATOS.....	82
4.8.1 <i>Arquitectura.</i>	82
4.8.2 <i>Archivos PHP</i>	83
4.8.3 <i>Scripts para la ejecución desde Unity.</i>	86
4.8.4 <i>Agregando el GameObject y trabajando en la interfaz.</i>	92
4.9.- INICIAR SESIÓN.....	94
4.9.1- <i>Actualizando la contraseña en la base de datos.</i>	96
4.9.2.- <i>Archivo PHP de inicio de sesión.</i>	97
4.9.3.- <i>Tratando la información que devolvió el archivo PHP.</i>	98
4.10.- CREACIÓN DEL MENÚ DE LA APLICACIÓN.....	99
4.10.1.- <i>Botón “cerrar sesión”.</i>	101
4.10.2.- <i>Botón “Créditos”.</i>	101
4.10.3.- <i>Botón “Negocios Disponibles”.</i>	102
4.10.4.- <i>Botón “Perfil”.</i>	103
4.10.4.1.- <i>Actualizando el correo electrónico.</i>	104
4.10.4.2.- <i>Visualización de los datos del perfil.</i>	106
4.10.5.- <i>Generador de códigos QR.</i>	107
4.11.- COLOCACIÓN DE CONTENIDO VIRTUAL CON LIGHTSHIP.....	111
4.11.1.- <i>¿Qué es lightship?</i>	112
4.11.3.- <i>Descarga de Lightship ARDK 3.0</i>	112
4.11.4.- <i>Importando los recursos en Unity.</i>	115
4.11.5.- <i>Configurando el proyecto.</i>	119
4.11.6.- <i>Empezando a posicionar contenido.</i>	124
4.12.- PROPONIENDO Y ACTIVANDO WAYSPOTS.....	129
4.12.1.- <i>Proponiendo un Wayspot.</i>	130
4.12.2.- <i>Activando un Wayspot.</i>	138
4.13.- DESARROLLANDO LA TRIVIA.....	142
4.14.- GENERACIÓN EL APK PARA LA INSTALACIÓN EN ANDROID.....	153
CAPÍTULO 5. RESULTADOS Y CONCLUSIONES.....	156

5.1.- RECORRIDO REALIZADO.	156
5.2.- ENCUESTAS SUS.	160
5.3.- RESULTADOS DE LA ENCUESTA PROPIA.	163
5.4.- RECOMENDACIONES.	165
5.5.- CONCLUSIONES.	168
5.6.- TRABAJO FUTURO.	169
ANEXOS.	171
GLOSARIO.	171
BIBLIOGRAFÍA	198

ÍNDICE DE IMÁGENES.

IMAGEN 1.- ESQUEMA SIMPLE DE LO QUE ES REALIDAD AUMENTADA.	19
IMAGEN 2.- INTERFAZ DE LA APLICACIÓN “VIEWRANGER”	24
IMAGEN 3.- INTERFAZ DE LA APLICACIÓN “MOUNTAINS AR”	25
IMAGEN 4.- INTERFAZ DE LA APLICACIÓN “ARTLENS 2.0”	25
IMAGEN 5.- INTERFAZ DE LA APLICACIÓN “NIGHT-SKY”	26
IMAGEN 6.- HOLOLENS.	27
IMAGEN 7. - VISTA QUE SE TIENE CON LAS HOLOLENS.....	27
IMAGEN 8. - INFORMACIÓN MOSTRADA A TRAVÉS DE LOS HOLOLENS	27
IMAGEN 9.-META GLASES QUEST 3.	28
IMAGEN 10.- INFORMACIÓN MOSTRADA A TRAVÉS DE LAS META GLASES 2.	28
IMAGEN 11.- CASCO DAQRI.....	29
IMAGEN 12.- VISIÓN PRO DE APPLE QUE SALDRÁN EN 2024	30
IMAGEN 13.- VISTA QUE SE PLANTEA TENDRÁN LAS VISIÓN PRO	30
IMAGEN 14.- LOGO DE VUFORIA.....	32
IMAGEN 15.- LOGO DE ARCORE	32
IMAGEN 16.-LOGOTIPO DE LIGHTSHIP	33
IMAGEN 17.- 8TH WALL.....	33
IMAGEN 18.- INTERFAZ DE GOOGLE SCKETCHU.....	35
IMAGEN 19.- INTERFAZ INICIAL DE BLENDER.....	35
IMAGEN 20.- INTERFAZ DE MAYA.....	36
IMAGEN 21.- INTERFAZ DE AUTODESK 3DS MAX.....	36
IMAGEN 22.- INTERFAZ DE CINEMA 4D.	37
IMAGEN 23.- INTERFAZ DE UNITY.	38
IMAGEN 24.- INTERFAZ DE UNREAL ENGINE.....	39
IMAGEN 25.- INTERFAZ DE GODOT ENGINE.....	39
IMAGEN 26.- INTERFAZ DE GAME MAKER.	40
IMAGEN 27.-DIAGRAMA DE COMPONENTES DE LA APLICACIÓN PROPUESTA.	43
IMAGEN 28 .-DIAGRAMA DE SECUENCIA PARTE 1.....	46
IMAGEN 29 .- DIAGRAMA DE COMPONENTES PARTE 2.	47
IMAGEN 30.- DIAGRAMA DE SECUENCIA PARTE 3.....	48
IMAGEN 31.- ENCUESTA SUS.	51
IMAGEN 32.- SIMULACIÓN DE RESPUESTAS A LAS PREGUNTAS DE LA ENCUESTA SUS.....	52
IMAGEN 33.- OPERACIONES A LAS PREGUNTAS IMPARES DE LA ENCUESTA SUS.....	53
IMAGEN 34.- CÁLCULO DEL VALOR RESULTANTE PARA LAS PREGUNTAS PARES DE LA ENCUESTA SUS.	54

IMAGEN 35.- RESULTADO DE LA SUMA DE LOS VALORES CALCULADOS EN LOS PASOS ANTERIORES DEL INSTRUMENTO.....	55
IMAGEN 36.- VISTA DE LOS LUGARES A SER VISITADOS EN EL RECORRIDO DE ESTE EXPERIMENTO.....	57
IMAGEN 37.-FORMAS DE CREAR LOS ELEMENTOS.....	61
IMAGEN 38.-COMPONENTES BÁSICOS DEL ENTORNO DE UNITY.	64
IMAGEN 39.- PANEL PRINCIPAL DE UNITY HUB.....	67
IMAGEN 40.- PANEL DE CREACIÓN DE PROYECTO.	68
IMAGEN 41.-CANVAS CREADO CON UN PANEL Y UN COMPONENTE IMAGE.	69
IMAGEN 42.-CONVIERTIENDO UNA IMAGEN RECIÉN IMPORTADA A “SPRITE 2D”.....	71
IMAGEN 43.-REFERENCIANDO LOS ELEMENTOS A LOS COMPONENTES INDICADOS.	72
IMAGEN 44.-SCRIPT CON EL QUE TRABAJAMOS EN LA ESCENA DE TÉRMINOS Y CONDICIONES.	73
IMAGEN 45.-SOLICITANDO LOS COMPONENTES Y VERIFICANDO QUE NO SE TENGA LA SESIÓN INICIADA.	76
IMAGEN 46.-FUNCIÓN QUE PERMITE REGISTRAR USUARIO Y CHECA QUE LOS CAMPOS ESTÉN LLENOS.	77
IMAGEN 47.-FUNCIÓN QUE CONECTARA CON EL SCRIPT DE LAS UNITYWEBSERVICE PARA INSERTAR LOS DATOS.....	77
IMAGEN 48.-INTERFAZ DE REGISTRO DE USUARIO.....	79
IMAGEN 49.-CÓDIGOS DE ÉXITO Y ERROR DE TODOS LOS ARCHIVOS PHP.....	79
IMAGEN 50.- ARCHIVO PHP PARA ENVIAR CORREOS DESDE EL SERVIDOR.....	81
IMAGEN 51.-CUERPOCORREO.PHP, QUE CONTIENE ELEMENTOS PARA EL ENVÍO DE CORREO.	81
IMAGEN 52.- ARQUITECTURA DE LAS UNITYWEBSERVICES.....	83
IMAGEN 53.- ARCHIVO “HEADER.PHP”.....	84
IMAGEN 54.- ARCHIVO PHP EN EL CUAL GENERAREMOS LA CONSULTA PARA INSERTAR LOS DATOS.....	85
IMAGEN 55.-CONSULTA “SELECT” QUE NOS AYUDA A VER SI LOS DATOS INGRESADOS YA ESTÁN EN LA BASE DE DATOS.	85
IMAGEN 56.-CONSULTA “INSERT” LA CUAL NOS AYUDA A REGISTRAR LOS DATOS EN LA BASE DE DATOS.....	86
IMAGEN 57.- VARIABLES Y LIBRERÍAS DE LA CLASE Y EL ARCHIVO “SERVIDOR”.....	87
IMAGEN 58.- VARIABLES DE LA CLASE “SERVICIO”.....	87
IMAGEN 59.- VARIABLES DE LA CLASE “RESPUESTA”.....	88
IMAGEN 60.- MÉTODO “CONSUMIRSERVICIO”.....	90
IMAGEN 61.-ARCHIVO “REGISTUSER.CS” EL QUE CONTIENE Y UTILIZA EL ARCHIVO “SERVIDOR.CS”.....	90
IMAGEN 62.-RUTA PARA LA CREACIÓN DEL COMPONENTE “. ASSET” PARA NUESTROS SERVICIOS.....	92
IMAGEN 63.-COMPONENTE “SERVIDOR.ASSET” CON EL CUAL CONTROLAREMOS LOS SERVICIOS QUE IREMOS EJECUTANDO.	93
IMAGEN 64.-INTERFAZ DE REGISTRO DE USUARIO, CON LOS COMPONENTES SEÑALADOS Y REFERENCIADOS.	94
IMAGEN 65.- INTERFAZ DE INICIO DE SESIÓN.....	94
IMAGEN 66.-PANEL PARA REESTABLECER LA CONTRASEÑA EN LA ESCENA DE INICIO DE SESIÓN.	95
IMAGEN 67.- ARCHIVO PHP PARA ACTUALIZAR EL PASSWORD POR MEDIO DE UNA CONSULTA “UPDATE”.....	96
IMAGEN 68.- ARCHIVO “LOGINNECAXATRAVEL.PHP” PARA INICIAR SESIÓN EN LA APLICACIÓN.	97

IMAGEN 69.- MÉTODO DE INICIAR SESIÓN DE LA ESCENA LOGIN SEÑALANDO LA INSTRUCCIÓN DONDE TRATAMOS LOS DATOS DEL JSON QUE LLEGA DESDE EL ARCHIVO PHP.....	98
IMAGEN 70.- CLASE USUARIODB A LA CUAL LE ASIGNAREMOS LOS DATOS DEL JSON.	99
IMAGEN 71.- PANEL QUE PRETENDE INDICARLE AL USUARIO QUE PRESTE MUCHA ATENCIÓN A LAS INSTRUCCIONES DE USO.	100
IMAGEN 72.- INTERFAZ DEL MENÚ, MOSTRANDO LOS BOTONES QUE CONTIENE.	100
IMAGEN 73.- MÉTODO DE CERRAR SESIÓN, EL CUAL UTILIZA LA INSTRUCCIÓN PARA ELIMINAR LAS KEYS.	101
IMAGEN 74.- ESCENA DE CRÉDITOS.	102
IMAGEN 75.- PANEL DE INDICACIONES PARA EL CANJEO DE PRODUCTOS EN LOS NEGOCIOS DISPONIBLES.	102
IMAGEN 76.- INTERFAZ DE NEGOCIOS AFILIADOS.	103
IMAGEN 77.- INTERFAZ DE LOS DATOS DEL PERFIL.	104
IMAGEN 78.- SCRIPT DE “ACTUALIZARCORREO.CS” EN EL QUE VEMOS COMO SE REALIZA LA ACTUALIZACIÓN DEL CORREO.....	105
IMAGEN 79.- ARCHIVO PHP QUE REALIZA LA ACTUALIZACIÓN DEL CORREO ELECTRÓNICO.	106
IMAGEN 80.- ARCHIVO PHP EN EL QUE EXTRAEMOS LOS DATOS DE LA BASE DE DATOS.....	107
IMAGEN 81.- PRIMERA PARTE DEL SCRIPT QUE SERÁ EL ENCARGADO DE GENERAR EL CÓDIGO QR.	109
IMAGEN 82.- SEGUNDA PARTE DEL SCRIPT QUE SERÁ EL ENCARGADO DE GENERAR EL CÓDIGO QR.	110
IMAGEN 83.- INTERFAZ DE GENERA QR Y UN EJEMPLO DE UN CÓDIGO YA CREADO CON EL MENSAJE DE UN USUARIO REGISTRADO.	111
IMAGEN 84.-PÁGINA PRINCIPAL DE LA CUENTA DE LIGHTSHIP, DE DONDE DESCARGAREMOS TODOS LOS RECURSOS.	112
IMAGEN 85.-DOCUMENTACIÓN PARA LA INSTALACIÓN DE LIGHTSHIP EN UNITY DE DIFERENTES MODOS	113
IMAGEN 86.-REPOSITORIO DEL VÍNCULO I.	114
IMAGEN 87.-REPOSITORIO DEL VÍNCULO II.	114
IMAGEN 88.-OPCIONES SELECCIONADAS PARA DESPLEGAR EL “PACKAGE MANAGER”.	115
IMAGEN 89.-PACKAGE MANAGER, SELECCIONANDO LA OPCIÓN PARA IMPORTAR LOS RECURSOS DESDE TARBALL.	116
IMAGEN 90.-IMPORTANDO “ADRK-UPM-3.4.0” A UNITY DESDE TARBALL.	116
IMAGEN 91.-IMPORTANDO “SHAREDAR-UPM-3.4.0”.	117
IMAGEN 92.-CREANDO LA “KEY” PARA EL PROYECTO.	117
IMAGEN 93.-COPIANDO LA “KEY” SELECCIONANDO LA OPCIÓN QUE SE MUESTRA EN LA IMAGEN.	118
IMAGEN 94.-MENÚ PARA PROPORCIONAR LA CLAVE DEL PROYECTO, ASÍ COMO LOS ERRORES QUE SE NOS MUESTRAN POR NO HABER AGREGADO LA CLAVE ANTES.	118
IMAGEN 95.-APARTADO EN EL QUE AGREGAREMOS LA CLAVE DEL PROYECTO.	119
IMAGEN 96.-SELECCIONANDO LA OPCIÓN DE “BUILD SETTINGS”.	120
IMAGEN 97.-VENTANA DE “BUILD SETTINGS” EN DONDE CAMBIAREMOS LA PLATAFORMA A “ANDROID”.	120
IMAGEN 98.-PRIMERAS CONFIGURACIONES DEL PROYECTO.....	121
IMAGEN 99.-CAMBIANDO LAS CONFIGURACIONES DEL NIVEL MÍNIMO DE ANDROID, ASÍ COMO EL SCRIPTING, API COMPATIBILITY Y TARGET ARCHITECTURES.	122
IMAGEN 100.-SELECCIONANDO LAS CONFIGURACIONES DEL MANEJADOR DE LIGHTSHIP PARA IOS.....	123

IMAGEN 101.-CONFIGURACIONES DEL MANEJADOR DE LIGHTSHIP PARA ANDROID.....	123
IMAGEN 102.-MENÚ PARA AGREGAR LOS COMPONENTES PARA COLOCAR EL CONTENIDO.....	124
IMAGEN 103.-AGREGANDO “AR LOCATION MANAGER” AL COMPONENTE XR ORIGIN.....	125
IMAGEN 104.-VENTANA DE GEOSPATIAL BROWSER, CON LA OPCIÓN “NOT ACTIVATED” DESMARCADA, PARA QUE SOLO SE MUESTREN LOS WAYSPTS QUE SE TIENEN ACTIVADOS EN DICHA ZONA.	125
IMAGEN 105.-DESCARGANDO EL WAYSPOT PARA COLOCAR EL CONTENIDO.....	126
IMAGEN 106.- WAYSPOT IMPORTADO EN UNITY, CONVERTIDO EN LOS 2 ARCHIVOS AUTOMÁTICAMENTE.....	127
IMAGEN 107.-REFERENCIANDO EL WASPOT AL COMPONENTE QUE NOS PERMITIRÁ COLOCAR EL CONTENIDO.....	127
IMAGEN 108.-COLOCANDO EL CONTENIDO EN LA APLICACIÓN QUE CREAMOS.	128
IMAGEN 109.-RESULTADOS DE LA APLICACIÓN QUE CREAMOS, MOSTRANDO COMO SE VE EL CONTENIDO QUE COLOCAMOS.....	129
IMAGEN 110.-APARTADO DESDE DONDE PODEMOS DESCARGAR LA APLICACIÓN PARA REALIZAR LOS ESCANEOS, ASÍ COMO LOS HORARIOS EN LOS QUE NOS SOLICITAN QUE REALICEMOS LOS MISMOS.....	130
IMAGEN 111.- OPCIÓN QUE SELECCIONAREMOS PARA INICIAR LA PROPUESTA DEL WAYSPOT.....	130
IMAGEN 112.- SELECCIONANDO LA UBICACIÓN DEL WAYSPOT QUE ESTAMOS SUGIRIENDO.....	131
IMAGEN 113.- CONDICIONES PARA QUE SE LLEVE A CABO LA ACTIVACIÓN DE LOS WAYSPOTS.....	132
IMAGEN 114.- INDICACIONES Y ADVERTENCIAS PARA LA ACTIVACIÓN.....	132
IMAGEN 115.- REAFIRMANDO LA UBICACIÓN DE NUESTRA PROPUESTA.....	133
IMAGEN 116.- INDICACIONES PARA TOMAR LA FOTO DE LA PROPUESTA.....	133
IMAGEN 117.- SUBIENDO LA FOTO DE NUESTRA PROPUESTA.....	134
IMAGEN 118.- EMPEZAMOS INGRESANDO LA DESCRIPCIÓN DE LA UBICACIÓN.....	135
IMAGEN 119.- SELECCIONANDO LA CATEGORÍA DE LAS PROPUESTAS.....	135
IMAGEN 120.- CONFIRMANDO LA INFORMACIÓN QUE PROPORCIONAMOS.....	136
IMAGEN 121.- EXPLICANDO LA IMPORTANCIA DE LA ACTIVACIÓN.....	136
IMAGEN 122.- NUESTRA SOLICITUD ENVIADA, PARA LA ACTIVACIÓN DEL WAYSPOT.....	137
IMAGEN 123.- SELECCIÓN EN DONDE PODREMOS VER EL ESTATUS DE NUESTRA SOLICITUD.....	137
IMAGEN 124.- MENÚ QUE NOS PERMITIRÁ REALIZAR EL ESCaneo DEL WAYSPOT.....	139
IMAGEN 125.- MALLA QUE RODEA EL ENTORNO QUE ESTAMOS MAPEANDO.....	139
IMAGEN 126.- IMAGEN QUE MUESTRA EL PROCESO DE CÓMO SE GENERA EL CONTENIDO 3D PARA GENERAR EL ARCHIVO 3D QUE SE DESCARGA EN LA MISMA PÁGINA.....	140
IMAGEN 127.- SE NOS MUESTRA LA CALIDAD DEL CONTENIDO QUE SE GENERÓ.....	141
IMAGEN 128.-OPCIONES PARA SUBIR EL ARCHIVO QUE SE GENERÓ.....	141
IMAGEN 129.-SCRIPTS “OPTION.CS” Y “QUESTION.CS” LOS CUALES NOS PERMITIRÁN CREAR Y MANEJAR NUESTRAS PREGUNTAS.....	143
IMAGEN 130.- PREFAB DE LA PREGUNTA, EL CUAL CLONAREMOS Y MODIFICAREMOS PARA EL BANCO DE PREGUNTAS.....	143
IMAGEN 131.- “QUIZDB.CS” DONDE CREAREMOS LA LISTA DE PREGUNTAS PARA EL BANCO.....	144
IMAGEN 132.- FUNCIONES PARA QUE SE ENVÍEN LAS PREGUNTAS DE MANERA ALEATORIA.....	145

IMAGEN 133.- GAMEOBJECT QUE FUNCIONARA COMO NUESTRO BANCO DE PREGUNTAS.	145
IMAGEN 134.- ARCHIVO “OPTIONBUTTONS.CS”	147
IMAGEN 135.- PRIMERAS CONFIGURACIONES DE LA INTERFAZ DE LA TRIVIA.....	147
IMAGEN 136.- CONFIGURACIONES DE LOS ELEMENTOS DE PREGUNTAS Y RESPUESTAS, ENCERRANDO EL NOMBRE DEL ELEMENTO DEL MISMO COLOR QUE SI RESPECTIVA VENTANA INSPECTOR.....	148
IMAGEN 137.- SCRIPT “QUIZUI.CS” QUE SE ENCARGARA DEL MANEJO DE LA INTERFAZ DE LA TRIVIA.....	149
IMAGEN 138.- ELEMENTOS CON LOS QUE TRABAJARA EL CONTROLADOR DE LA TRIVIA.....	150
IMAGEN 139.- FUNCIONES: START, UPDATE, NEXTQUESTION Y GIVEANSEWER DEL CONTROLADOR DE TRIVIA.	151
IMAGEN 140.- CORRUTINA QUE DETERMINARA EL COMPORTAMIENTO DE LA TRIVIA.	151
IMAGEN 141.- INTERFAZ DE LA TRIVIA JUNTO CON EL PANEL DE FIN DE LA TRIVIA.....	152
IMAGEN 142.-VENTANA DE “BUILD SETTINGS” DESDE DONDE GENERAREMOS EL APK.	153
IMAGEN 143.- PRIMERAS VISTAS DEL CONTENIDO QUE SE MUESTRA EN EL MONUMENTO AL CORONEL JUAN GALINDO.	156
IMAGEN 144.- SEGUNDA PARTE DEL CONTENIDO QUE SE MOSTRÓ EN EL MONUMENTO AL CORONEL JUAN GALINDO.	157
IMAGEN 145.- SEGUNDA ESCENA, MONUMENTO DE LA TURBINA ELÉCTRICA.	157
IMAGEN 146.- SEGUNDA MUESTRA DEL CONTENIDO DE LA ESCENA DE LA TURBINA ELÉCTRICA Y LA PLACA POR LOS 100 AÑOS DE LYFC. .	158
IMAGEN 147.- CONTENIDO DE LA ESCENA DE LA PLAZA CÍVICA, TERCER PUNTO DEL RECORRIDO.	158
IMAGEN 148.- SEGUNDA PARTE DEL CONTENIDO DEL TERCER PUNTO.....	159
IMAGEN 149.- CONTENIDO DEL ÚLTIMO PUNTO DEL RECORRIDO.	159

ÍNDICE DE GRÁFICAS.

GRÁFICA 1.- GRÁFICA DE PROMEDIOS DE LAS PERSONAS QUE EVALUARON NUESTRA APLICACIÓN CON EL MÉTODO SUS.	162
GRÁFICA 2.- GRÁFICA DE LAS RESPUESTAS DE LA ENCUESTA QUE CREAMOS NOSOTROS, QUE MUESTRA QUE TAN INNOVADORA LES PARECIÓ LA APLICACIÓN A LOS USUARIOS ENCUESTADOS.	164
GRÁFICA 3.- GRÁFICA DE LOS DISPOSITIVOS UTILIZADOS PARA LOS RECORRIDOS DEL EXPERIMENTO.	165

ÍNDICE DE TABLAS.

TABLA 1 COMPARACIÓN DE APLICACIONES QUE UTILIZAN RA	21
TABLA 3.- TABLA QUE MUESTRA COMO RESPONDIERON LOS USUARIOS LA ENCUESTA SUS.	161
TABLA 4.- TABLA DE LOS RESULTADOS DE LA EVALUACIÓN DESPUÉS DE LAS OPERACIONES SUGERIDAS POR SUS.	161
TABLA 5.- AQUÍ MOSTRAMOS LAS RESPUESTAS QUE NOS DIERON A LA PREGUNTA 11, SOLO 3 PERSONAS NOS HICIERON RECOMENDACIONES.....	162
TABLA 6.- RECOMENDACIONES QUE NOS DIERON EN EL FORMATO QUE SE LLENÓ DESPUÉS DEL RECORRIDO.	167

Capítulo 1. Introducción.

1.1 Motivación y justificación.

Cuando algunas personas visitan el centro histórico de la ciudad de Puebla desconocen muchos de los lugares que son interesantes y están llenos de información cultural, quizás esto se debe a que no cuentan o contratan un(a) guía que les pueda dar recorridos guiados. Además, que, no cuentan con una aplicación que indique una ruta recomendada o brinde información de un punto turístico de manera atractiva, ni que busque la manera de promover la visita a ciertas zonas importantes del centro histórico de la ciudad.

Muchas ciudades en México tienen una gran riqueza cultural, sin embargo, son pocas las personas que saben sobre estos lugares. Algunas veces, los turistas hacen una investigación previa del lugar a visitar para saber qué lugares visitar, sin embargo, no todos los lugares son recomendados en las redes sociales o páginas web de los gobiernos de esas ciudades; esto debido, muchas veces, a lo turístico que pueden llegar a ser o a la facilidad de ingreso a ellos. En el centro histórico existen diferentes opciones para realizar los recorridos, sin embargo, ninguna de las opciones consultadas cuenta con una aplicación o dinámica como la que proponemos en este trabajo.

Hay que considerar que algunos de los sitios históricos no son de fácil acceso, ya sea por la ubicación física o las instalaciones de infraestructura. Por ejemplo, el uso de escaleras no es de fácil acceso para personas en sillas de ruedas, que tienen problemas para caminar debido a la edad o alguna lesión en las piernas; incluso debemos considerar a las personas con alguna discapacidad visual. Es por ello por lo que en este trabajo se considera el uso de Realidad Aumentada (RA) que es una tecnología que permite la superposición de elementos virtuales sobre nuestra visión de la realidad [6]. Actualmente, la RA está tomando bastante relevancia y se están encontrando cada vez más aplicaciones a situaciones de la vida cotidiana, en el turismo no solo mejora la experiencia del viajero, sino que también ofrece beneficios prácticos para la preservación, promoción y accesibilidad de los destinos turísticos.

La RA permite superponer información digital sobre el entorno físico, proporcionando a los turistas datos históricos, culturales o arquitectónicos en tiempo real. Esto enriquece la visita a museos, monumentos y lugares históricos al ofrecer una comprensión más profunda del lugar. También permite a los visitantes interactuar con su entorno de formas novedosas. Por ejemplo, los turistas pueden ver recreaciones en 3D de cómo lucían ciertos sitios en el pasado o explorar

contenido multimedia que no sería accesible de otra manera, como videos, audios, y animaciones. Sin olvidar que las experiencias turísticas pueden ser personalizadas según los intereses y necesidades del visitante. Por ejemplo, un turista puede recibir recomendaciones personalizadas basadas en sus preferencias previas o seleccionar información en diferentes idiomas. Asimismo, la RA puede hacer que el turismo sea más accesible para personas con discapacidades. Por ejemplo, puede proporcionar guías virtuales en lenguaje de señas o descripciones de audio para personas con discapacidades visuales; también puede ayudar a reducir el impacto físico sobre sitios históricos o naturales. En lugar de permitir el acceso físico a áreas delicadas, la RA puede ofrecer una experiencia virtual que permite a los visitantes explorar estos lugares sin causar daño. A través de aplicaciones de RA, los turistas pueden aprender sobre la historia, la cultura y la importancia ambiental de los lugares que visitan, fomentando una mayor apreciación y respeto por el patrimonio cultural y natural.

1.2. Objetivos.

1.2.1 Objetivo general.

Desarrollar una aplicación con realidad aumentada del centro histórico de Puebla, utilizando estándares internacionales de diseño interactivo, con el fin de ofrecer a la ciudadanía una experiencia inmersiva y educativa que permita conocer y apreciar la historia de la ciudad desde una perspectiva innovadora.

1.2.2. Objetivos específicos.

- Identificar y mapear los puntos de interés históricos del centro histórico de Puebla, que serán integrados en la aplicación, asegurando la relevancia y precisión de la información proporcionada.
- Diseñar la arquitectura de software del sistema de realidad aumentada del centro histórico de la ciudad de Puebla, utilizando notaciones estándares apropiadas al problema a resolver, para visualizar cómo se implementará el sistema y facilitar la comunicación de nuestra propuesta con otros stakeholders.
- Desarrollar el software del sistema de realidad aumentada del centro histórico de la ciudad de Puebla, utilizando patrones de diseño alineados a la arquitectura diseñada, para probar la solución en un contexto de la vida real, asegurando su funcionalidad.

- Evaluar la propuesta de solución, usando métricas de usabilidad y pruebas funcionales para determinar la calidad del proyecto propuesto y realizar ajustes que optimicen la experiencia del usuario.

1.3. Mapa de lectura de la tesis.

En el capítulo 2 “*Estado del arte*” se describe el concepto de realidad aumentada. Posteriormente, se muestra una investigación de las aplicaciones existentes y que son similares a esta propuesta de aplicación y los escenarios en donde se está investigando y desarrollando aplicaciones que utilizan la realidad aumentada para beneficiar varios sectores artísticos.

En el capítulo 3 “*Componentes de la aplicación*” se exponen los elementos de ingeniería de software los cuales se utilizaron para realizar la aplicación y la forma en la que se desarrolló, también se plantean los escenarios del experimento que se realizó para evaluar la aplicación y la herramienta que se utilizó para que los usuarios la evalúen.

El capítulo 4 “*Solución para el recorrido con realidad aumentada*” se presenta el desarrollo detallado de la aplicación propuesta, se indica como se fue implementando cada sección de la aplicación y se dan algunas recomendaciones que sirvan de inspiración para que el lector pueda comenzar a utilizar la herramienta que se eligió. También, se mencionan definiciones para que el lector se familiarice con los términos utilizados a la hora de llevar a cabo el desarrollo en Unity y se explica cómo utilizar la herramienta que nos ayudó a trabajar la realidad aumentada explicando procesos desde como implementar la herramienta, hasta como desarrollar los recursos que necesitamos con la ayuda de esa herramienta.

El capítulo 5 “*Resultados y conclusiones*” se presentan los resultados obtenidos de los experimentos y un ejemplo del contenido de la solución, mostramos los comentarios recibidos y las “calificaciones” que los evaluadores asignaron a nuestra aplicación. Se informa también de las futuras implementaciones en la aplicación.

Capítulo 2. Estado del arte.

“Las ciudades con riqueza histórica y patrimonial representan una excelente opción de destino cultural que ofrecen al turista un amplio abanico de atractivos y experiencias muy diversas” [7] y es obligación de los nativos de dichas ciudades o estados dar a conocer el patrimonio que se posee, para que la gente de otros países, estados e incluso los mismos habitantes se den cuenta del tesoro histórico con el que cuenta el centro; además, el turismo genera una economía favorable para los residentes de cada lugar.

La ciudad de Puebla “fue fundada en el siglo XVI con el fin de instituir una ciudad para españoles en un lugar intermedio entre la Ciudad de México y el puerto de Veracruz” [7]. “En 1987, el Centro Histórico de Puebla fue inscrito en la lista del Patrimonio Cultural de la Humanidad de la Organización de las Naciones Unidas para la Educación, la Ciencia y la Cultura (UNESCO)” [7]; esto demuestra que la ciudad de Puebla posee un vasto patrimonio cultural inmaterial y su centro histórico es un lugar de gran relevancia, con numerosos sitios de interés que atraen tanto a turistas como a los propios ciudadanos.

La importancia de dar a conocer estos puntos radica en la posibilidad de incrementar el flujo de visitantes en el centro histórico de Puebla. Con el desarrollo y uso de esta aplicación, se busca incentivar el consumo en los locales cercanos a los puntos turísticos promovidos mediante la realidad aumentada. Diversos proyectos muestran cómo la realidad aumentada puede resaltar tanto el patrimonio tangible (como edificios y monumentos) como el intangible (como historias, tradiciones y mitología) en distintas ciudades alrededor del mundo. La RA no solo mejora la experiencia del visitante, sino que también contribuye a la educación, la sensibilización cultural y la preservación del patrimonio cultural de manera innovadora y accesible. Un ejemplo es Hidden Florence que guía a los usuarios a través de rutas de la ciudad basadas en personajes históricos como un aprendiz de artista del Renacimiento (<https://hiddenflorence.org/>).

“La realidad aumentada es una tecnología madura que utiliza el mundo real como sustrato y lo amplía superponiendo información generada por ordenador” [8]. Esta tecnología nos permitirá mostrar al usuario información, que de otro modo, no estaría fácilmente accesible, ya que no todos los visitantes están acompañados por un guía turístico o por alguien que conozca la historia detrás de ciertos puntos clave. A través de la aplicación, se podrán guiar a los usuarios a museos y narrar las historias de los edificios icónicos del centro histórico de Puebla. Además, la digitalización

representa una oportunidad para la preservación del patrimonio cultural mundial, que enfrenta amenazas tanto naturales como provocadas por acciones humanas [8].

Además, la realidad aumentada es una tecnología emergente, de la que se encuentran diversas aplicaciones en distintos campos: la educación, la mercadotecnia y estudios recientes se centran en la innovación de museos con realidad aumentada.

2.1.- ¿Qué es la realidad aumentada?

“La Realidad Aumentada (RA) adquiere presencia en el mundo científico a principios de los años 1990 cuando la tecnología basada en: a) ordenadores de procesamiento rápido, b) técnicas de renderizado de gráficos en tiempo real, y c) sistemas de seguimiento de precisión portables, permiten implementar la combinación de imágenes generadas por el ordenador sobre la visión del mundo real que tiene el usuario” [9]; desde entonces se empezó a experimentar con la realidad aumentada y la misma empezó a adquirir gran fuerza en la comunidad de desarrolladores y artistas digitales, ya que esta permite obtener información en tiempo real en un punto específico y entender cierta información que podría resultar llamativa para el usuario.

La RA funciona de forma simple, ya que la tecnología actúa como una lente a través de la cual se observa el mundo físico (pueblo, lugares y cosas). La gran capacidad de esta lente es la de superponer sobre el entorno físico información digital relevante con el contexto en el que se encuentra la persona que está “mirando”. Esta información generalmente se encuentra en la nube, es decir, en la red [9], además de que esta información puede estar contenida también en el mismo dispositivo que es el encargado de mostrar el contenido que el desarrollador desee en un entorno real, este contenido puede ser texto, video, imagen, modelos 3D, etc., los cuales ayudaran a los usuarios a obtener más información sobre su entorno, información que se considere importante o necesaria de conocer, la Imagen 1 es un esquema general del concepto de realidad aumentada.



Imagen 1.- Esquema simple de lo que es realidad aumentada.

Fuente: Elaboración a partir de [10].

Acorde con [11] la RA se clasifica en diferentes niveles:

Nivel 0. Hiperenlaces en el mundo físico, este tipo de RA se considera simple o inicial y permite enlazar el mundo físico con el virtual, como los códigos QR que nos informan cuando una cámara los descifra.

Nivel 1. Realidad aumentada basada en marcadores de referencia. Estos son objetos utilizados para la observación de sistemas de imágenes, apareciendo en la imagen como puntos de referencia o de medición. En este nivel, los activadores son marcadores, figuras que, al ser escaneadas, normalmente generan un modelo 3D que se superpone sobre la imagen real. Los marcadores deben contar con un patrón único, que permite a la cámara identificar y determinar el objeto o los objetos que deben mostrarse.

Nivel 2. Realidad aumentada sin marcadores. Este nivel utiliza activadores como imágenes, objetos o ubicaciones GPS, sin necesidad de marcadores físicos. Existen aplicaciones móviles conocidas como navegadores de RA. Estas aplicaciones aprovechan el hardware de los teléfonos inteligentes (GPS, brújula y acelerómetro) para localizar y superponer información sobre puntos de interés (POIs) en el entorno. Al mover el teléfono para captar imágenes del entorno, el navegador muestra los POIs cercanos basándose en un mapa de datos.

Nivel 3. Visión aumentada. Este punto consiste en desarrollar dispositivos especiales para usar la tecnología, como ejemplo, las gafas de Google, para mostrar información disponible para los usuarios sin usar las manos, permitiendo el acceso a internet mediante órdenes de voz.

A partir de esta información, surge la opción de crear interacciones de los usuarios con el patrimonio turístico del centro histórico de Puebla, para que el usuario se interese cada vez más por conocerlo y recibir una remuneración virtual mediante puntuaciones obtenidas por encuestas, para que se incentive la visita a locales de comida cercanos al centro histórico y generar más afluencia en comercios.

2.2.- La realidad aumentada en el sector turístico.

En distintas partes del mundo se han realizado investigaciones e implementaciones de realidad aumentada en museos, para lograr que diferentes usuarios se interesen más por visitarlos y generar una mejor interacción entre el patrimonio cultural intangible y el usuario, provocando que retenga más información sobre las esculturas, pinturas y demás objetos exhibidos en estos. Ejemplo de ello es National Gallery, con sede en Londres, sacó algunas de sus obras a las calles para que pudieran visionarse mediante una aplicación de realidad aumentada. Al escanear un código QR en ciertos lugares, las obras se aparecían ante los ojos de los usuarios. O el Museo Nacional de Historia Natural de Washington D.C. en Estados Unidos donde se guardan numerosos esqueletos de animales. La realidad aumentada permite contemplar a las criaturas reales con tan solo proyectarlas desde un teléfono o tableta (<https://evergine.com/es/realidad-aumentada-museos/>). Otro ejemplo es el uso de realidad virtual implementada para el museo de las Termópilas ubicado en el sitio de la batalla original, cerca de la ciudad de Lamia en Grecia, donde se utilizaron técnicas de narración y principios de videojuegos modernos para difundir el conocimiento histórico sobre la batalla y las leyendas asociadas.[12]

Una de las ventajas encontradas es que se puede interactuar con ciertas piezas inaccesibles, ya que se arriesgan a degradarse o sufrir un daño de alguna manera, pero al usar realidad aumentada se simula una interacción con ellas e incluso se puede obtener información que explique el origen o algunos datos adicionales, además de que la información no solo se puede complementar con texto u objetos 3D, sino que también se puede sobreponer imágenes, videos, audio, etc.

También se ha implementado la realidad aumentada en algunas aplicaciones que se enfocan en distintos sectores turísticos, como senderismo, rutas definidas, etc. Estas aplicaciones no manejan un incentivo más que el de utilizar la tecnología recién desarrollada. No se encontró evidencia de una aplicación que haya sido desarrollada en México que consideré la

implementación de una remuneración por medio de cupones para ayudar a promover también a los restaurantes que decidan utilizarla.

En la **¡Error! No se encuentra el origen de la referencia.** se presentan algunas aplicaciones existentes con este esquema de apoyar el turismo.

Tabla 1 Comparación de aplicaciones que utilizan RA

Nombre	Descripción	Dispositivos	Dirigido a	Comparación con la propuesta
ViewRanger <i>Fuente: [13]</i>	Aplicación de realidad aumentada para ayudar a los caminantes a navegar con seguridad por montañas y valles.	Esta aplicación esta para iOS y Android	Personas interesadas en conocer montañas y rutas seguras para poder recorrerlas, ya se a pie, en bicicleta u otro medio, trazando una ruta predefinida por el usuario.	El objetivo de la aplicación de este trabajo es promover lugares históricos y no enfocarse en recorridos exclusivamente.
Mountains AR <i>Fuente: [14]</i>	En un software de realidad aumentada que brinda información de montañas del mundo entero, útil para quien practica senderismo, ciclismo, escalada, camping, etc.	Esta aplicación esta desarrollada solo para iOS.	Personas que deseen conocer información de las montañas existentes en el mundo.	Esta aplicación esta más enfocada al montañismo y la aplicación propuesta está enfocada a promover el patrimonio cultural del centro histórico

				de la ciudad de Puebla.
ArtLens 2.0 <i>Fuente: [15]</i>	Es una aplicación que permite crear un recorrido en el museo de Cleveland, aunque no lo visites personalmente.	Para iOS, iPad y Android.	Visitantes al museo de Cleveland de forma presencial o a distancia.	Aplicación enfocada a un museo específico, no para una zona turística.
Chicago 00 Project <i>Fuente: https://chicago00.org/</i>	Es un proyecto digital galardonado desarrollado, diseñado y administrado por John Russick (ex vicepresidente sénior del Museo de Historia de Chicago) y Geoffrey Alan Rhodes.	El objetivo del proyecto era producir y publicar una serie de experiencias multimedia interactivas, inmersivas y específicas del sitio diseñadas para mostrar el archivo de películas, fotografías y sonidos del Museo de Historia de Chicago y compartir las historias de Chicago de formas nuevas y únicas.	Interesados en ver el archivo cinematográfico y sonoro del museo y el compartir las historias de Chicago. También para personas interesadas en conocer la historia que el museo ha preservado con el paso de los años.	El proyecto es similar en cuanto al recorrido por la zona específica, pero se limita a solo mostrar la información y contenido de dicho recorrido, no implementa una interacción con el usuario y tampoco se plantean recompensas.
Remembering 7th Street	Un proyecto de la Escuela de	No se indica para qué	Interesados en conocer el oeste	Es un videojuego que

<p>Fuente: https://projects.journalism.berkeley.edu/7thstreet-archive/</p>	<p>Periodismo y el Departamento de Arquitectura de UC Berkeley para recrear West Oakland en las décadas de 1940 y 1950, cuando era una comunidad próspera repleta de clubes de blues y jazz.</p>	<p>dispositivos fue desarrollada, es un juego de gráficos 3D simples, por tanto, se asume que puede usarse en cualquier computadora.</p>	<p>de Oakland posterior a la Segunda Guerra Mundial. Quienes quieran saber sobre historia del blues y jazz.</p>	<p>permite ver archivos sobre una calle representativa para los amantes del blues y jazz, no promueve la visita al lugar.</p>
<p>Onirix Fuente: https://www.onirix.com/es/</p>	<p>Página web que permite a los usuarios crear experiencias 3D para que puedan publicar contenido y hacer más atractivos sus negocios.</p>	<p>Al ser una página web, esta puede ser compilada por cualquier dispositivo que tenga acceso a un internet y una cámara.</p>	<p>Negocios y personas que pretendan crear experiencias inmersivas 3D de manera simple y accesible para todos sus usuarios.</p>	<p>Es una alternativa de desarrollo, se incluye por su fácil acceso y forma de utilizar, sin embargo, solo se puede ver el contenido accediendo a la página, ya que no permite generar un instalador.</p>
<p>Night sky Fuente: [16]</p>	<p>Aplicación de referencia para observar las estrellas, donde el usuario puede explorar una representación virtual del cielo nocturno para identificar estrellas, planetas,</p>	<p>iOS, tvOS y watchOS</p>	<p>Interesados en la astrología.</p>	<p>Es totalmente diferente, pero se menciona por su alternativa por utilizar la realidad aumentada para que el usuario comprenda las</p>



Imagen 3.- Interfaz de la aplicación "Mountains AR"

Fuente: [14]



Imagen 4.- Interfaz de la aplicación "ArtLens 2.0"

Fuente: [15].



Imagen 5.- Interfaz de la aplicación "night-sky"

Fuente: [16].

2.3.- Dispositivos para utilizar Realidad Aumentada.

Durante la revisión de literatura se encontró que la mayoría de las aplicaciones están desarrolladas para dispositivos móviles y esto responde a la necesidad de hacer la tecnología más accesible, portátil y masivamente adoptada. Existe distintos dispositivos como lo son:

- ❖ Microsoft HoloLens: Gafas creadas por Microsoft para manejar RA prácticas y cómodas para el usuario que no requieren cables; dentro de sus funcionalidades se contemplan hologramas articulados de seguimiento de la mano, toque, agarre y movimiento, de manera que se sienten naturales. HoloLens 2 se adapta a las manos y por eso los hologramas responden como objetos reales, se puede habilitar por voz, hace seguimiento de ojos para adaptar los hologramas en tiempo real; traza un mapa perfecto del entorno físico y deja que el contenido digital se ancle a objetos o superficies dondequiera que se esté. [17]



Imagen 6.- Hololens.

Fuente: [17].



Imagen 7. - Vista que se tiene con las HoloLens

Fuente: <https://www.microsoft.com/es-mx/hololens/industry-education>

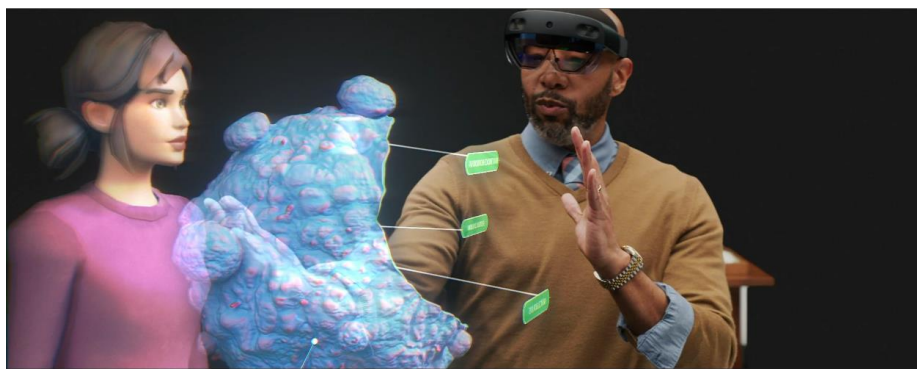


Imagen 8. - información mostrada a través de los Hololens

Fuente: <https://www.microsoft.com/es-mx/hololens/industry-education>

- ❖ **Meta Glasses:** Son gafas desarrolladas por Meta (Empresa de Facebook) permiten al usuario interactuar con los 10 dedos de la mano, 2 de las versiones permiten el trabajo con

la RA, los cuales son el Meta Quest 3 y el Meta Quest Pro. Estas gafas pueden ser conectadas a un PC (por cable o por WiFi) y jugar a juegos VR (realidad virtual), ver películas en 3D, animaciones VR y experiencias inmersivas e incluso explorar metaversos: Rec Room, VRchat, Horizon Worlds.[18]



Imagen 9.-Meta gafas Quest 3.

Fuente: <https://www.meta.com/quest/quest-3/>.



Imagen 10.- Información mostrada a través de las Meta gafas 2.

Fuente: <https://www.youtube.com/watch?v=iEiWVBN7Jhk>

- ❖ El casco DAQRI, conecta a los trabajadores con el entorno de trabajo dando información útil en tiempo real para acelerar y simplificar el trabajo. Cuenta con una serie de sensores que registran e interactúan en 360 grados con lo que sucede en el exterior y una visión “térmica” que simplifica la realización de algunas tareas específicas: desde la reconstrucción de todas las características de un determinado accesorio hasta el control de

un determinado elemento, como una válvula. El trabajador puede evaluar, como si estuviera equipado con rayos X, tubos y maquinaria verificando el funcionamiento correcto. El Daqri Smart Helmet, crea un mapa completo de todo el entorno: desde las estructuras que se están completando hasta las que se acaban de esbozar. Es un casco que puede funcionar sin conexión Wifi, Bluetooth o GPS; aprovechando al máximo el poder de la realidad aumentada y la tecnología 4D. El casco permite recibir y realizar videollamadas con la unidad operativa para tener instrucciones en tiempo real y asignación de actividades. <https://www.tworeality.com/gafas-virtuales/daqri-smart-helmet/>



Imagen 11.- Casco DAQRI

Fuente: <https://www.tworeality.com/gafas-virtuales/daqri-smart-helmet/>

- ❖ **Visión Pro:** Son las gafas creadas por Apple para el uso de la realidad mixta, utiliza visionOS, que tiene una interfaz tridimensional. El modo predeterminado del dispositivo es la realidad aumentada. Una corona en la parte superior del dispositivo puede ajustar la cantidad de fondo virtual que ocupa el campo de visión del usuario, hasta lograr una inmersión virtual completa (realidad virtual). Vision Pro tiene una pantalla orientada hacia afuera llamada EyeSight, que muestra una reproducción de los ojos del usuario y refleja el grado en el que el usuario está inmerso. Sus ojos aparecen atenuados cuando están en realidad aumentada y oscurecidos cuando están en realidad virtual completa, para indicar a los demás la conciencia del entorno del usuario. Cuando alguien se acerca o entabla una conversación, EyeSight muestra los ojos del usuario normalmente, incluso en realidad virtual completa, y la otra persona es visible para el usuario. El Apple Vision Pro tiene una pantalla frontal de vidrio laminado, un marco de aluminio cubierto por un cojín flexible y

una banda para la cabeza ajustable. El marco contiene cinco sensores, seis micrófonos y 12 cámaras. (https://es.wikipedia.org/wiki/Apple_Vision_Pro)



Imagen 12.- Visión pro de apple que saldrán en 2024

Fuente: [19].

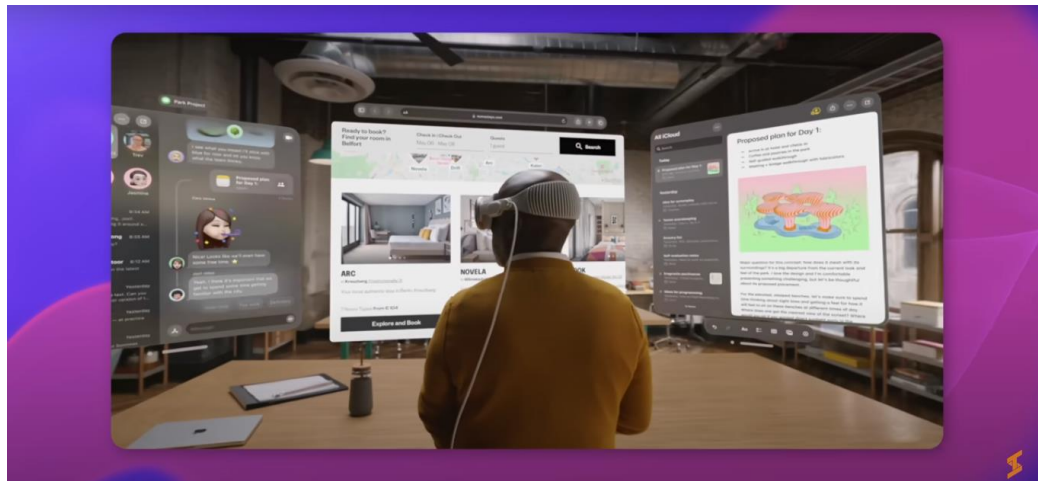


Imagen 13.- Vista que se plantea tendrán las Visión Pro

Fuente: [19].

- ❖ **Smart Phone:** es un dispositivo móvil que combina las funciones de un teléfono y de una computadora de bolsillo, es práctico y actualmente la mayoría de las personas cuentan con uno, ya que hay de precios muy accesibles con precios bajos con distintos sistemas operativos, siendo los más predominantes en el mercado IOs y Android.

Como se observa en este pequeño análisis realizado, hay dispositivos que permiten al usuario interactuar con la realidad aumentada, pero la mayoría son costosos para realizar actividades ocasionales. Y ya que en la actualidad la mayoría de las personas tiene acceso a un teléfono inteligente (Smart Phone), la opción más viable es trabajar la aplicación para estos dispositivos.

El fácil acceso a un dispositivo móvil por parte del usuario motiva a desarrollar la aplicación para dichos dispositivos, con base en los 2 sistemas operativos móviles que dominan el mercado: IOS y Android, pero, en el caso de Android, se incluye el uso en tabletas.

“La tecnología móvil sigue siendo una de las soluciones más accesibles y prometedoras para la visualización de la realidad aumentada orientada al usuario de la CH” [8], con esta afirmación se refuerza la selección de esta para este trabajo.

Además, se considera el uso de las siguientes herramientas:

- Herramientas de modelado.
- Motores de videojuegos.
- Herramientas para la creación de realidad aumentada.

2.6.- Herramientas para crear realidad aumentada.

Para crear realidad aumentada existen herramientas que facilitan su creación, con algunas herramientas tras pasos sencillos y clics, algunos elementos mencionados funcionan como detonantes para mostrar el contenido en el dispositivo deseado, algunos detonantes podrían ser los códigos QR, marcadores, etc.

En este proyecto nos enfocaremos en utilizar la funcionalidad de VPS que facilita mucho el mostrar contenido como elementos 3D, textos, videos, etc. Algunas herramientas son:

- Vuforia: es una plataforma de desarrollo de aplicaciones de realidad aumentada y realidad mixta, multiplataforma con seguimiento robusto y rendimiento en una variedad de hardware. Admite muchos dispositivos de terceros y dispositivos de realidad virtual con cámaras orientadas hacia atrás y se puede utilizar en cualquier dispositivo que cuente con una cámara para probar juegos y aplicaciones. Su sdk se puede descargar para distintas plataformas como Unity, Android, iOS, etc.



Imagen 14.- Logo de vuforia

Fuente: <https://github.com/PTCInc/vuforia-engine>

- **ARCore:** Es una plataforma desarrollada por Google que permite al desarrollador crear experiencias de realidad aumentada, esta herramienta nació en 2017 como el sistema de realidad aumentada de Google y como respuesta para ser el equivalente de Apple y su ARKit. ARCore no nació de la nada, sino que relevó a Project Tango, el cual nació con el propósito de tener un método distinto usando sensores especiales para entender el entorno.



Imagen 15.- Logo de ARCore

Fuente: <https://google.fandom.com/wiki/ARCore?file=Arcore-logo.png>.

- **Lightship:** Es un conjunto de herramientas para construir el metaverso en el mundo real porque ofrece los mejores conjuntos de características de AR compartida, segmentación semántica y mapeo mundial. Este software fue desarrollado por la empresa Niantic que es la empresa

encargada de desarrollar y dar mantenimiento al videojuego Pokemon Go, cuyas dinámicas son en realidad aumentada casi en su totalidad.



Imagen 16.-Logotipo de lightship

Fuente: <https://venturebeat.com/games/niantic-offers-broader-access-to-niantic-lightship-ar-dev-tools/>

- **8ThWallet:** Esta herramienta también ha sido elaborada por niantic, tiene una buena propuesta para los desarrolladores porque tiene una tecnología híbrida que permite trasladarse fácilmente entre web y app. Se puede generar un proyecto para navegadores y se migra fácilmente a uno para Android o iPhone. Además, tiene una buena calidad y estabilidad.



Imagen 17.- 8th wall

Fuente:

https://cdn.8thwall.com/web/share/8th_Wall_Metadata_SocialCover_2h7lvb7w0dj72yytw4n0u9ckr43p6ka878511i2d7fryd4uei4svgpx.png

Las herramientas mencionadas son algunas de las grandes variedades que existen, y los programadores pueden utilizar cualquiera para crear realidad aumentada.

Aunque cada una de estas herramientas es muy poderosa y adecuada para el desarrollo de aplicaciones significativas, en este proyecto hemos optado por utilizar Unity como motor de videojuegos y Lightship como herramienta para la creación de realidad aumentada. Esta decisión se tomó tras un análisis de las opciones disponibles.

2.4.- Herramientas de modelado.

Las herramientas de modelado son fundamentales para desarrollar la realidad aumentada, porque mediante ellas se generan los recursos 3D que se mostrarán en la aplicación.

La mayoría de los recursos que existen se desarrollan en diferentes softwares de modelado, pero no existe alguno en especial para los motores de videojuegos, se pueden desarrollar los elementos en el software como:

- Google SketchUp: Cualquier software de modelado 3D, requiere un mínimo conocimiento en dibujo, cuenta con una opción de “Instructor” que al seleccionarla muestra la funcionalidad de la herramienta, los modos abreviados de acción y brinda consejos para modelar. Además de que su cinta de herramientas puede ser personalizada de acuerdo con las necesidades del usuario, asimismo se puede descargar imágenes, texturas, objetos, etc. para ser utilizados en los proyectos.

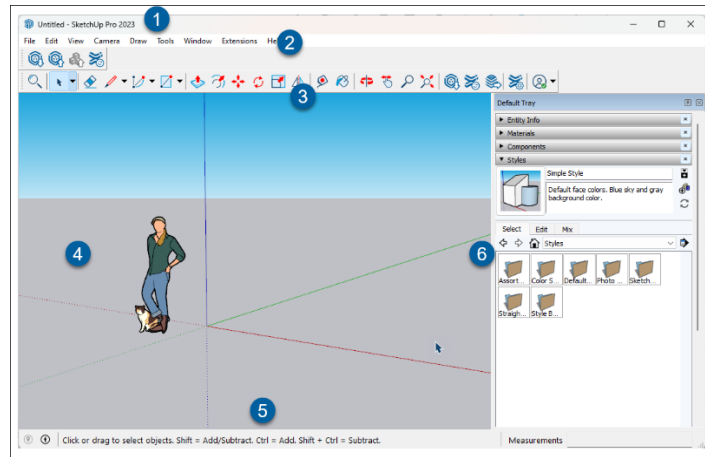


Imagen 18.- Interfaz de Google ScketchU.

Fuente: help.sketchup.com

- **Blender:** Es un conjunto de creación 3D de código abierto y gratuito. Es compatible con la totalidad del flujo de animación 3D: modelado, animación, simulación, renderizado, composición y seguimiento de movimiento, incluso edición de video y creación de juegos.

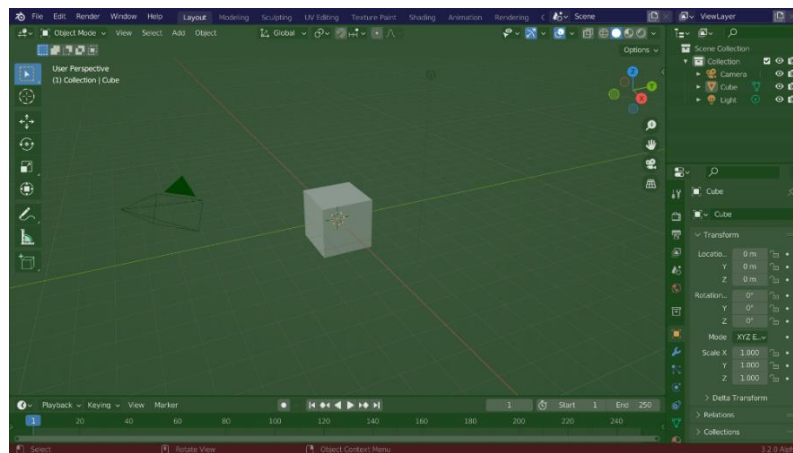


Imagen 19.- interfaz inicial de blender

Fuente: docs.blender.org

- **Maya:** Es la elección de la mayoría de los artistas 3D. Autodesk Maya es un software de renderización, simulación, modelado y animación con un conjunto de herramientas, integrado, que se utilizan para crear animaciones, entornos, gráficos de movimiento, realidad virtual y creación de personajes. Es el software más complejo y completo; esto

significa que es el más difícil de aprender y dominar, pero vale la pena invertir tiempo y presupuesto en él gracias a sus resultados profesionales.

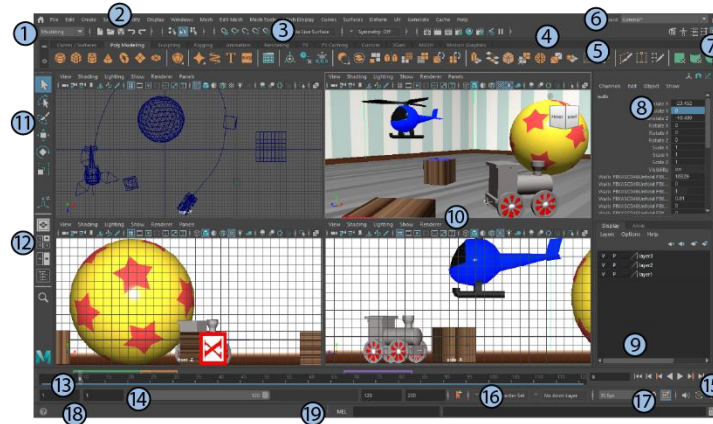


Imagen 20.- interfaz de maya.

Fuente: help.autodesk.com

- **Autodesk 3Ds Max:** Es la mejor opción para los usuarios de Windows. Al igual que su software hermano, Maya, cuenta con una robusta gama de herramientas que hacen la curva de aprendizaje igualmente compleja, pero es ideal para animadores principiantes o intermedios porque hace mejores renders que Maya.

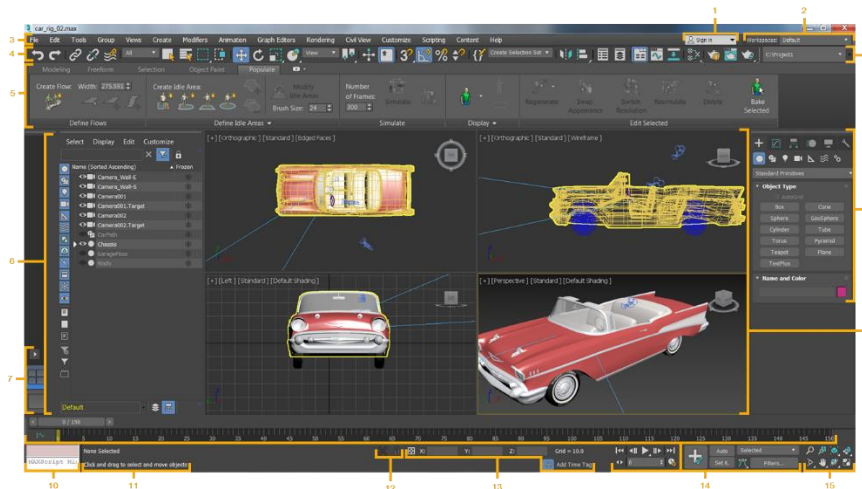


Imagen 21.- interfaz de Autodesk 3Ds max.

Fuente: help.autodesk.com

- **Cinema 4D:** Se trata de un software profesional con la que se puede crear un trabajo espectacular y también es la aplicación con la curva de aprendizaje más pequeña. Las licencias perpetuas para este software no son baratas, pero Maxon también ofrece licencias para estudiantes de corto plazo a un costo reducido. Este software es usado principalmente para hacer motion graphics. Además de que los principiantes y los profesionales experimentados, pueden aprovechar la amplia gama de herramientas y funciones de Cinema 4D.



Imagen 22.- interfaz de cinema 4d.

Fuente: www.maxon.net

Como se puede observar en la pequeña lista anterior, existen diversos softwares de animación 3D, no es necesario saber diseñar elementos 3D para realizar aplicaciones de realidad aumentada.

2.5.- Motores de Videojuegos

Otra parte importante para el desarrollo de realidad aumentada son los motores de videojuegos, los cuales nos permiten desarrollar aplicaciones móviles y generar las dinámicas que harán más llamativa la aplicación.

Existen diferentes motores de videojuegos y cada vez se desarrollan más, a continuación, se enlistan algunos ejemplos:

Unity: Es un motor de juego multiplataforma desarrollado por Unity Technologies, es un motor muy intuitivo y que ayuda a los desarrolladores a entender el entorno. Es compatible

con diferentes plataformas, innovador y con demasiados recursos para que la curva de aprendizaje sea corta. Además, soporta mapeo de relieve, reflexión, paralaje, oclusión ambiental, etc.

El hecho de poder utilizarlo en Windows, MacOS e inclusive Linux hace de Unity uno de los motores preferido para muchos, cabe mencionar que el lenguaje con el que se programan las funcionalidades es C#.

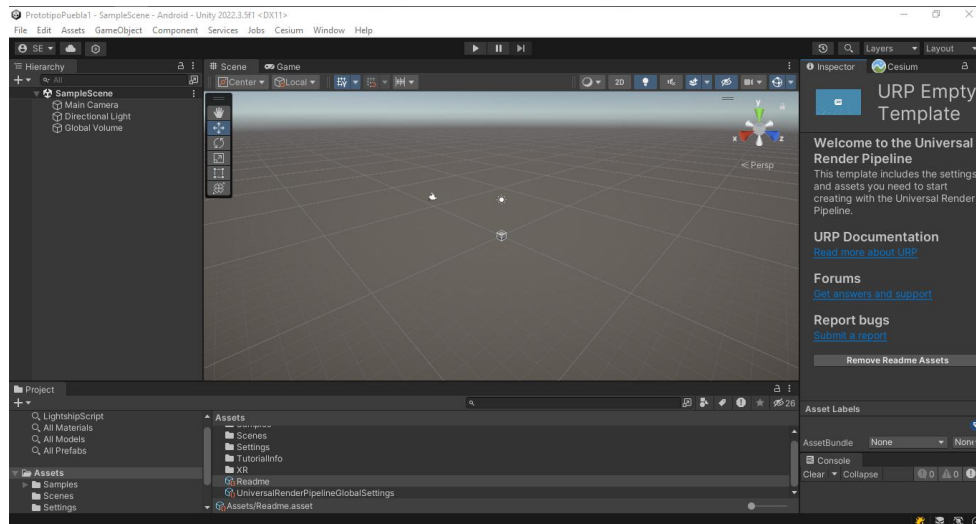


Imagen 23.- Interfaz de Unity.

Fuente: Software instalado.

- **Unreal Engine:** Es un motor que pertenece a Epic Games y sus funcionalidades se programan en el lenguaje C++. Es conocido por su potencia gráfica y su capacidad para crear entornos tridimensionales altamente realistas. Sin embargo, esta herramienta no solamente es utilizada para el desarrollo de videojuegos, sino que también se utiliza para la visualización arquitectónica, producción cinematográfica, simulación, etc. Su lenguaje principal es C++, aunque también cuenta con un sistema visual de scripting llamado Blueprints, que permite a los desarrolladores crear funcionalidades sin necesidad de escribir código.

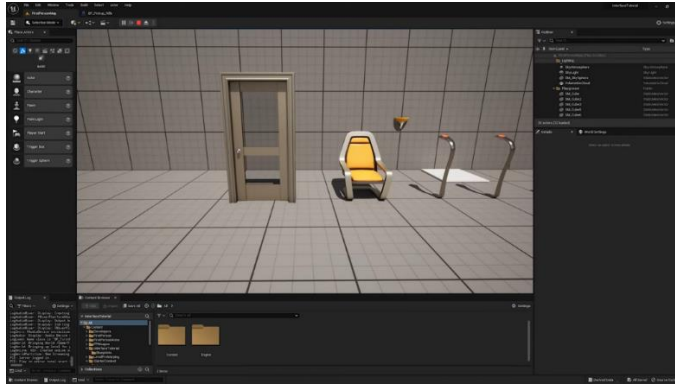


Imagen 24.- interfaz de Unreal Engine.

Fuente: <https://www.youtube.com/watch?v=wGmlKblPcw&t=1s>

- **Godot:** Es un motor de videojuegos 2D y 3D de uso general, diseñado para admitir todo tipo de proyectos, además de videojuegos se pueden crear aplicaciones para escritorio o aplicaciones móviles.

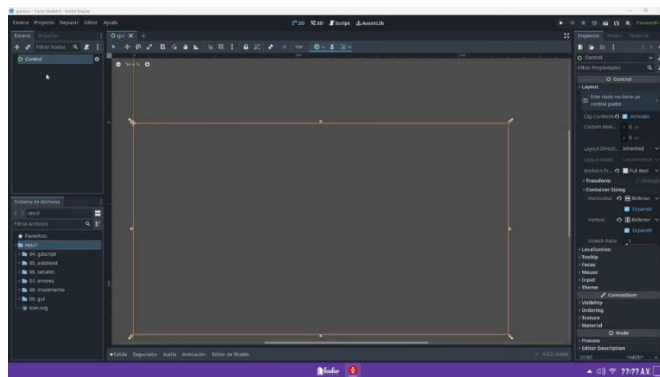


Imagen 25.- interfaz de Godot engine.

Fuente: <https://www.youtube.com/watch?v=5Pz1pvjjCfI&t=87s>

- **Open 3D Engine:** Es un motor de desarrollo de videojuegos de código abierto, creado por la Fundación Open 3D, que es una colaboración entre Amazon y la comunidad de desarrollo. Permite la creación de mundos 3D con calidad cinematográfica y simulaciones de alta fidelidad tanto en videojuegos como en aplicaciones de realidad virtual y aumentada.
- **Game Maker:** Es una plataforma basada en un lenguaje de programación interpretado y un kit de desarrollo para videojuegos, está orientado a usuarios novatos o personas con pocas nociones de programación.

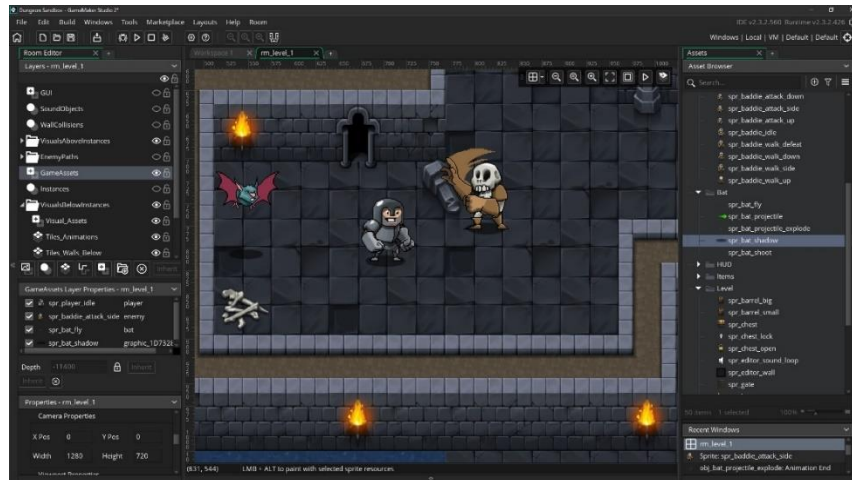


Imagen 26.- interfaz de Game Maker.

Fuente: <https://store.steampowered.com/app/1670460/GameMaker/>.

En este proyecto se utiliza Unity dada la experiencia del autor, además de que las herramientas mencionadas son compatibles con dicho motor.

Capítulo 3. Diseño de la aplicación.

El diseño del sistema es crucial en el desarrollo de software porque establece la estructura y organización del sistema antes de que comience la implementación. Se debe considerar los componentes y sus interacciones, lo cual facilita la comprensión y optimización del sistema, permitiendo la reducción de errores. Un diseño bien estructurado permite agregar o modificar funciones en el futuro con menos esfuerzo, facilitando la evolución del sistema a medida que crecen las necesidades del usuario o del negocio. Se debe considerar la usabilidad, seguridad y rendimiento durante el diseño para que el sistema sea intuitivo, confiable y eficiente para los usuarios. A continuación, se presenta el diseño de la aplicación.

3.1.- Diagrama de componentes.

Un diagrama de componentes es una representación gráfica que muestra la estructura y organización de los elementos que componen la aplicación que estamos desarrollando. Esta herramienta se utiliza en el desarrollo de software para describir la estructura de los componentes. Además, nos ayudará a planificar y documentar sistemas complejos, asegurando que todos los elementos funcionen de manera eficiente.

Como se puede observar en la Imagen 27, este es el diagrama de componentes de la aplicación propuesta, que incluye carpetas que contienen scripts, modelos 3D, sonidos, entre otros elementos.

Estos componentes son los elementos principales para el desarrollo de la aplicación, asegurando así una interacción eficiente del usuario con la interfaz. Cada elemento se comunica entre sí mediante los scripts que fueron programados de manera independiente. Además, se incluyen componentes importados de los paquetes que se instalaron al importar elementos esenciales, lo que nos permitirá integrar ciertos componentes en la aplicación. Es decir, los paneles, el texto y otros elementos que debemos manejar en la aplicación están contenidos en estos componentes.

Para clarificar el contenido de los componentes, a continuación, se presentan definiciones que permitirán comprender mejor algunos elementos de la aplicación.

- Script: Hace referencia a una secuencia de comandos o “guion” que indica al ordenador las instrucciones que debe realizar. A menudo, se compara con las instrucciones de una receta de cocina, ya que, de igual manera, se van indicando secuencialmente los pasos a seguir para conseguir el resultado deseado [20].
- Wayspots: Es un modelo 3D que genera el equipo de lighthship mediante algunos escaneos en distintos horarios y estos son para colocar contenido virtual en los puntos que fueron generados.
- Prefab: Son un tipo especial de componente que permite guardar GameObjects totalmente configurados en el Proyecto para su reutilización. Estos activos pueden compartirse entre escenas, o incluso otros proyectos sin tener que configurarse de nuevo. Esto es muy útil para los objetos que se utilizarán muchas veces, tales como plataformas [21].
- Scenes (Escenas): Son donde se trabaja con el contenido en Unity. Son activos que contienen todo o parte de un juego o aplicación. Por ejemplo, usted puede construir un juego simple en una sola escena, mientras que, para un juego más complejo, puede utilizar una escena por nivel, cada uno con sus propios entornos, personajes, obstáculos, decoraciones y UI. Puedes crear cualquier número de escenas en un proyecto [22].
- Splash Screen: Imagen que aparece en la pantalla de un ordenador, teléfono u otro dispositivo electrónico cuando se enciende por primera vez o cuando se inicia una aplicación o un programa [23].
- UnityWebRequest: Es un remplazo para el objeto WWW original de Unity. Este proporciona un sistema modular para componer peticiones HTTP y manejar respuestas HTTP. El objetivo principal del sistema de UnityWebRequest es permitirles a los juegos de Unity interactuar con backends Web modernos. También soporta características de alta demanda tal como solicitudes HTTP fragmentadas, operaciones de streaming POST/PUT y un control completo sobre encabezados HTTP y verbs [24].

El paquete de scripts está dividido en subcarpetas, cada una de las cuales contiene los scripts organizados de acuerdo con la funcionalidad para la cual han sido creados. Es evidente que

todos los componentes interactúan entre sí, ya que las escenas los usan cada uno. Los scripts activan o invocan los elementos de las otras carpetas. Cabe mencionar que los componentes como Settings, TextMesh Pro y XR. Son componentes que se incorporan al proyecto automáticamente al importar ciertos elementos para la funcionalidad de la aplicación, pero se usan en la escena.

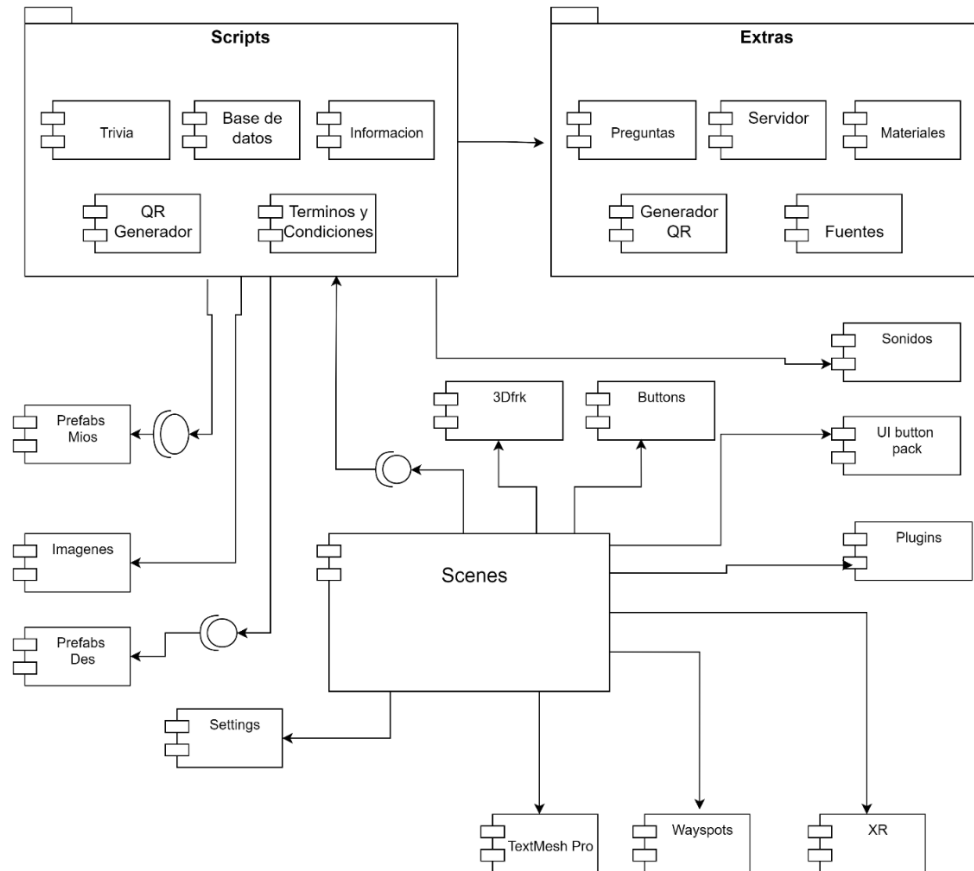


Imagen 27.-Diagrama de componentes de la aplicación propuesta.

Los componentes que están etiquetados como “prefabs Mios” son los prefabs que se fueron creando sobre la marcha para mostrar la información y el contenido que deseamos que los usuarios vean, así como los componentes que hemos ido creando para facilitar el cómo mostrar la información al usuario y “Prefabs Des” cuenta con los componentes que se mostraran con la ayuda de Lighthship, los cuales han sido descargados de páginas de internet (con licencia de uso libre) para tomar referencias y evitar trabajos de modelado complejos, ya que este no es el objetivo de este trabajo.

El componente nombrado “Preguntas” es un componente que fue creado por el desarrollador, en los cuales se contienen prefabs que contienen información básica de las preguntas, estas están compuestas por campos de texto y botones como lo son, “texto pregunta”, “respuesta 1”, “respuesta 2”, “respuesta 3”, “respuesta 4”, y un componente booleano marcado como “es correcto” el cual nos indicara si la respuesta que seleccionamos es la correcta, dichos prefabs no se encuentran en el componente denominado “Prefabs Propios” para tener un mayor control al utilizarlos, ya que deseamos tener diferentes bancos de preguntas dependiendo del sitio que se esté visitando.

El componente de imágenes contiene, esencialmente, todas las imágenes que se utilizarán en el proyecto. Por ejemplo, incluye el fondo de las interfaces que se mostrarán, la pantalla de presentación (Splash Screen), el logo que se presentará y las imágenes de los botones que utilizaremos para hacer la navegación más intuitiva para el usuario.

El componente de Wayspots incluye todos los wayspots necesarios para publicar el contenido. Estos se descargan de la página oficial de Lightship e incluyen un manifiesto que se genera en dicha página, así como un elemento 3D que sirve como referencia para ubicar el contenido que deseamos posicionar.

Como se observa, tenemos un paquete que contiene pequeños módulos. Este paquete lo hemos denominado según el uso que les daremos o el propósito de los scripts que contiene cada carpeta.

El módulo de base de datos se utiliza para realizar las solicitudes UnityWebRequest, las cuales nos permiten comunicarnos con la base de datos. Con estas solicitudes, podemos realizar el registro, el inicio de sesión, la descarga del catálogo de productos y la actualización de los puntos o cupones que los usuarios acumulan, que se almacenarán en un servidor.

Otro módulo que se encuentra dentro de los scripts es el de “información”, en el cual contendremos los scripts que controlarán la información que se mostrará al usuario durante el recorrido. Este módulo activará o desactivará el contenido que se irá presentando mediante Lightship, teniendo en cuenta que dichos elementos pueden ser controlados mediante código. Cada módulo de esta carpeta es un script organizado así según la finalidad o la actividad que realiza. Por

ejemplo, el módulo de trivia se encarga de gestionar la dinámica de la misma, mientras que el de QRGenerador es responsable de generar dicho código, entre otros.

Con estos componentes funciona la aplicación propuesta. No mencionamos los componentes que se muestran en el diagrama, ya que algunos se importan por defecto al proyecto, dependiendo de las funcionalidades que necesite.

3.2.- Diagrama de secuencia.

El diagrama de secuencia es un tipo de diagrama utilizado para representar la forma en que se comunican los objetos o componentes de un sistema mientras la aplicación se está ejecutando.

En otras palabras, el diagrama de secuencia describe el flujo de mensajes y el orden de los eventos que se ejecutan en un proceso específico. Además, contiene mensajes que pueden ser vistos por el usuario durante la ejecución y describe la lógica en tiempo real de interacción de los distintos elementos.

La **¡Error! No se encuentra el origen de la referencia.** Imagen 28 muestra el Diagrama de secuencia parte 1, que contiene los elementos: Regstuser, iniciar sesion, cambiar contraseña, revisar perfil y servidor. La Imagen 29 presenta el Diagrama de componentes parte 2 que contiene los componentes: Generador qr, actualizar correo, trivia, actualizar puntos, servidor. La Imagen 30 muestra el Diagrama de secuencia parte 3, que contiene el componente cambiar escena. Hay que tener en cuenta que cualquier clase se comunica con el usuario con un cierto set de interfaces.

En esta sección, presentaremos el diagrama de secuencia, que ilustra la interacción del usuario con la aplicación a través de las interfaces y los scripts. Mencionamos los scripts con los que consideramos que el usuario interactúa con mayor frecuencia y que son más relevantes para la aplicación. Aunque existen otros scripts que no se abordan en este momento, estos serán detallados más adelante en el capítulo 4, donde se explicará la funcionalidad de cada acción.

Primero, debes considerar que cada acción de los scripts se activa con una interfaz, que suele consistir en botones. Estos elementos no están activos en todo momento; cada botón se

encarga de desencadenar las acciones que se realizan en cada ocasión dependiendo del propósito que tenga.

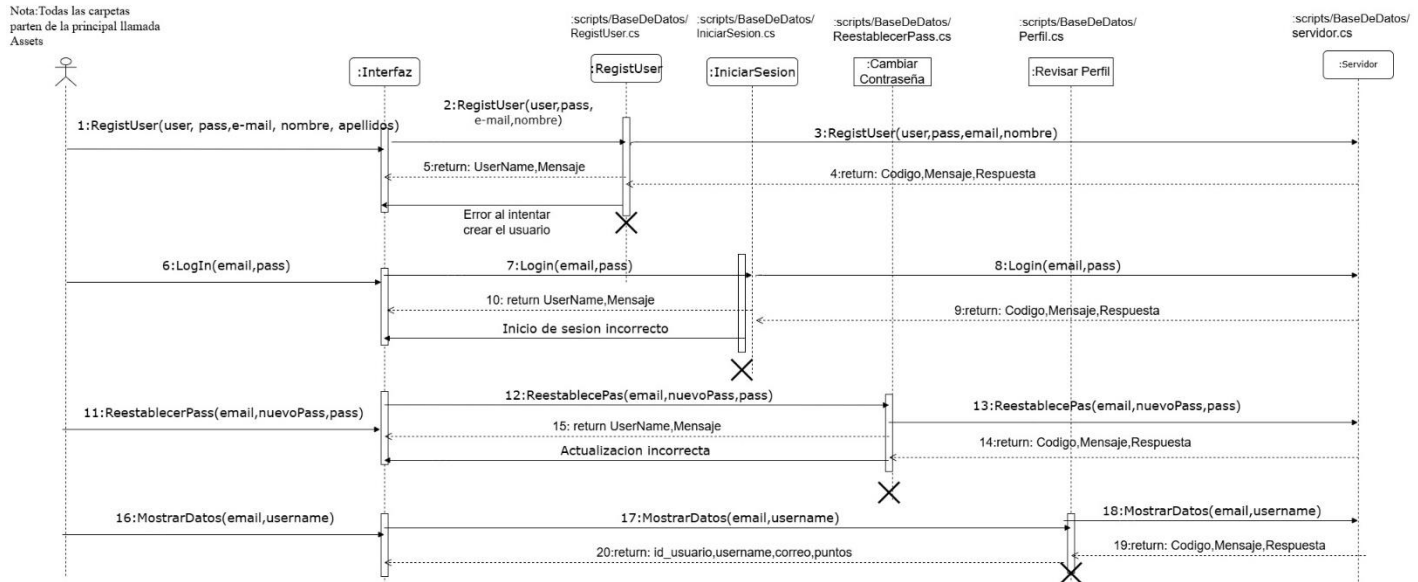


Imagen 28 .-Diagrama de secuencia parte 1.

Primero, tenemos el registro de usuario. Como se ve en el diagrama (Imagen 28; **Error! No se encuentra el origen de la referencia.**), el usuario es quien debe proporcionar los datos necesarios para realizar el registro. Estos datos serán procesados por un script que controla la escena, al que llamamos “Registrar”, se comunica con otro script denominado “Servidor”, este es el encargado de interactuar con el servidor y todos los archivos que se cargaron previamente, los cuales sirven para llevar a cabo cada consulta. El script principal de la escena envía los datos al script “Servidor” y este devuelve un código, un mensaje y una respuesta, que en este caso es una cadena de tipo JSON. El script principal procesará esta información; dependiendo del código recibido, se mostrará al usuario el mensaje correspondiente.

Nota: Todas las carpetas parten de la principal llamada Assets

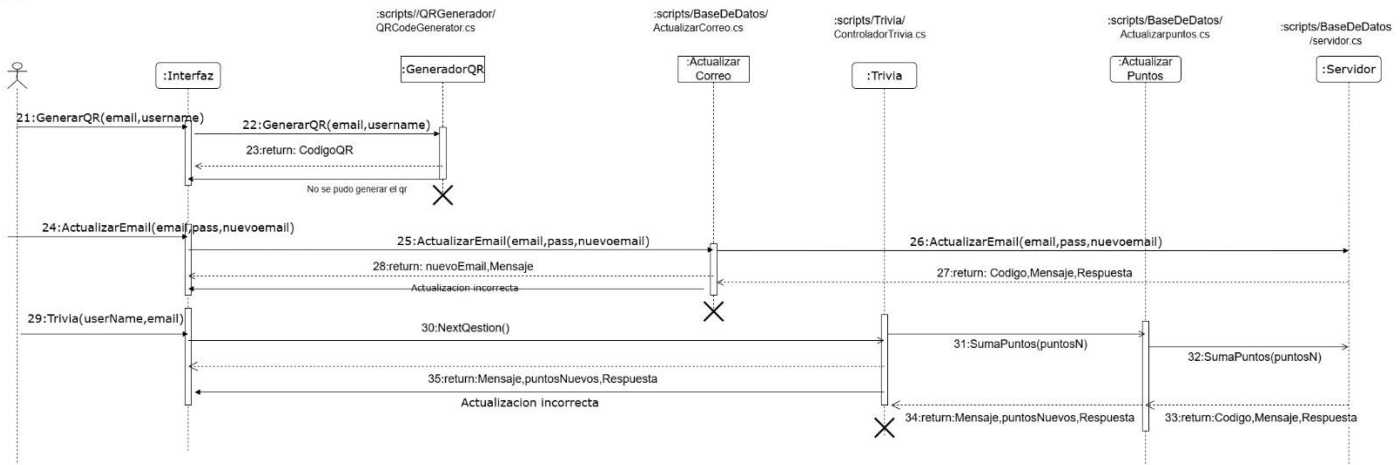


Imagen 29.- Diagrama de componentes parte 2.

En el inicio de sesión, encontramos el script principal de esta escena el cual nombramos como “IniciarSesion”. Este script es similar al registro, ya que envía los datos al script servidor, que nos ayuda a comunicarnos con el servidor donde está la base de datos y los archivos PHP que utilizamos para realizar las consultas. El servidor nos responde de manera similar, proporcionando un código, un mensaje y una respuesta. El script de iniciar sesión es responsable de procesar la respuesta según el código recibido. Este mismo procedimiento se aplica a cada archivo ilustrado en la Imagen 28 y la Imagen 29 **¡Error! No se encuentra el origen de la referencia.**, tales como:

- Cambiar contraseña.
- Revisar perfil.
- Actualizar correo.
- Actualizar puntos.

Estos scripts interactúan igual que los detallados, excepto el de actualización de puntos, ya que se ejecuta sin necesidad de interactuar con ellos mediante una interfaz. En este caso, es el script de trivia el que ejecuta la función para actualizar los puntos como se muestra en la Imagen 30.

Nota: Todas las carpetas parten de la principal llamada Assets

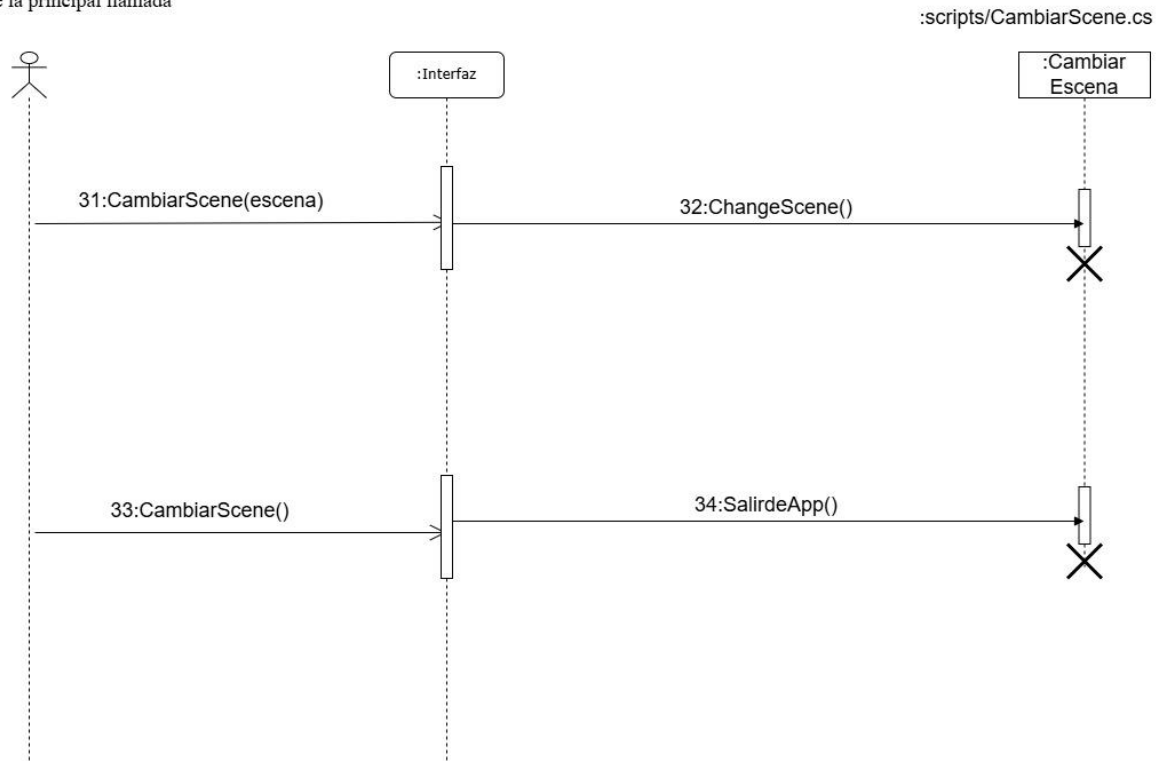


Imagen 30.- Diagrama de secuencia parte 3.

El generador de códigos QR no interactúa con la base de datos, debido a que los datos que codificamos son extraídos de unos PlayerPrefs, los cuales se detallarán más adelante.

En cuanto a los scripts de trivia, estos interactúan con el usuario al momento de seleccionar la respuesta a cada pregunta. El script se encarga de cambiar la pregunta y gestionar dos contadores. El primero suma o resta puntos; si el usuario se equivoca, se le restan 5 puntos, mientras que, si acierta, se le suman 50 puntos. El segundo contador incrementa de uno en uno hasta alcanzar un total de diez, que es la cantidad de preguntas que se realizaran.

El script para cambiar de escena se ejecuta básicamente al presionar un botón, lo que nos permite trasladarnos de una escena a otra. Es importante mencionar que, para esta funcionalidad, debemos proporcionar el nombre de la escena a la que deseamos acceder. Sin embargo, este script no devuelve ninguna respuesta. En el caso de la función “Salir de la aplicación” no le enviamos ningún dato ni recibimos una respuesta.

En conclusión, gracias a este diagrama, podemos comprender la interacción del usuario con los scripts que se crearon, asimismo, podemos observar que la respuesta que obtengamos

determinará la información que mostraremos al usuario. Pero casi toda la comunicación con el servidor es similar para cualquier operación de esta índole, excepto algunas clases que no siguen este patrón.

3.3.-Diseño del experimento y aplicando SUS.

El objetivo de este experimento es la creación de un pequeño recorrido para poner a prueba la aceptación de la aplicación cuando el desarrollo de esta finalice. La evaluación constara de una encuesta genérica, comentarios informales expresados por los usuarios y una encuesta de “Escala de usabilidad del sistema” en inglés “System Usability Scale (SUS)”.

La principal razón de su aplicación está relacionada con verificar cuan innovador es implementar el uso de la realidad aumentada en el turismo. Dicha herramienta tecnológica adquirió un gran auge gracias al metaverso y los instrumentos que se están fabricando, aunque el interés en esta tecnología crece cada vez más con el tiempo, la realidad aumentada no se está implementando en todos los campos en los cuales la herramienta tiene posibilidad.

La finalidad de utilizar el System Usability Scale (SUS), además de las encuestas tradicionales es porque SUS es una herramienta sumamente útil para la evaluación de aplicaciones. Este sistema lo creó John Brooke en 1986 y su método consiste en evaluar el producto mediante 10 preguntas. El puntaje se calcula mediante ciertas condiciones y operaciones propuestas por el propio método [25].

La escala SUS (System Usability Scale) es una herramienta eficaz para evaluar de manera general la usabilidad de un sistema, producto o servicio. Aunque es reconocida por su sencillez y capacidad para proporcionar una visión rápida de la usabilidad, la SUS no está diseñada para evaluar aspectos específicos de la usabilidad. Sin embargo, las puntuaciones obtenidas a través de la encuesta SUS pueden dar una idea general sobre las siguientes áreas:

- Eficiencia de Uso: La SUS puede ayudar a evaluar qué tan eficientemente los usuarios pueden completar tareas principales con el sistema. Una puntuación alta sugiere que los usuarios pueden realizar sus tareas de manera eficiente, mientras que una puntuación baja puede indicar problemas en la ejecución de tareas.
- Facilidad de Aprendizaje: La escala también proporciona una idea de cuán fácil es para los nuevos usuarios entender cómo usar el sistema y completar tareas sin ayuda extensiva. Las

preguntas en la SUS ayudan a medir si los usuarios sienten que necesitarían aprender muchas cosas antes de moverse dentro del sistema.

- Satisfacción del Usuario: Aunque la SUS no mide directamente la satisfacción del usuario, las puntuaciones pueden reflejar el grado general de satisfacción con la experiencia de usuario del sistema. Una puntuación alta en la SUS generalmente indica una respuesta positiva hacia la usabilidad, lo que puede correlacionarse con una mayor satisfacción.
- Cumplimiento de expectativas: La SUS puede dar una indicación de si, el sistema cumple con las expectativas de los usuarios en términos de usabilidad. Las preguntas de la escala están formuladas de manera que una puntuación alta sugiere que el sistema probablemente cumpla o exceda las expectativas de usabilidad de los usuarios [25].

La encuesta tiene 10 preguntas con respuesta que varía del 1 al 5, donde el 1 indica que no está de acuerdo con lo que propone la pregunta y el 5 indica que está totalmente de acuerdo con lo que plantea la pregunta. Después de realizar las preguntas, con las respuestas se obtiene un promedio con diferentes operaciones variando los cálculos en diferentes preguntas. Si el promedio es mayor que 68 se considera que la aplicación es funcional e intuitiva.

Básicamente, se aplicarán las siguientes 10 preguntas, las cuales podemos ver en la Imagen 31. Se muestran las preguntas que se harán los usuarios de la aplicación, solo que se tradujeron al español.

	Strongly disagree				Strongly agree
1. I think that I would like to use this system frequently	1	2	3	4	5
2. I found the system unnecessarily complex	1	2	3	4	5
3. I thought the system was easy to use	1	2	3	4	5
4. I think that I would need the support of a technical person to be able to use this system	1	2	3	4	5
5. I found the various functions in this system were well integrated	1	2	3	4	5
6. I thought there was too much inconsistency in this system	1	2	3	4	5
7. I would imagine that most people would learn to use this system very quickly	1	2	3	4	5
8. I found the system very cumbersome to use	1	2	3	4	5
9. I felt very confident using the system	1	2	3	4	5
10. I needed to learn a lot of things before I could get going with this system	1	2	3	4	5

Imagen 31.- Encuesta SUS.

Fuente: [26].

Para comenzar, el mismo método indica que si un entrevistado no puede responder a una pregunta en particular, debe marcar el punto central de la escala.

Tenemos diferentes tratamientos para las preguntas pares e impares;

- Para las preguntas 1, 3, 5, 7 y 9, se le asigna un valor que fue seleccionado entre 1 y 5, y a este valor le tenemos que restar 1.
- Para las preguntas 2, 4, 6, 8 y 10, se les asigna un valor entre 1 y 5, a dicho valor se le debe restar 5.
- Posteriormente, sumamos el resultado de esos valores y los multiplicamos por 2.5.

Una vez que obtenemos el resultado, podemos medir la usabilidad de nuestro proyecto. Es importante tener en cuenta que el valor resultante no es un porcentaje, si no un número que debe de clasificarse por percentiles. Según la investigación, una puntuación superior a 68 se considera

por encima de la media, mientras que cualquier puntuación inferior a 68 se considera por debajo de la media.

Para tener una mejor comprensión de cómo se utiliza SUS, a continuación, presentaremos un ejemplo del cálculo de las operaciones, tomando como referencia la imagen con las preguntas en inglés, aclarando que las preguntas se formularon en español.

Supongamos que una persona encuestada respondió de la siguiente manera sobre la usabilidad de una aplicación, tal como se muestra en la Imagen 32:

	Strongly disagree				Strongly agree
1. I think that I would like to use this system frequently	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
2. I found the system unnecessarily complex	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
3. I thought the system was easy to use	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
4. I think that I would need the support of a technical person to be able to use this system	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
5. I found the various functions in this system were well integrated	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
6. I thought there was too much inconsistency in this system	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
7. I would imagine that most people would learn to use this system very quickly	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
8. I found the system very cumbersome to use	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
9. I felt very confident using the system	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
10. I needed to learn a lot of things before I could get going with this system	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Imagen 32.- Simulación de respuestas a las preguntas de la encuesta SUS.

Fuente: Imagen capturada de [26] y editada por el autor.

Una vez que tengamos las respuestas, procederemos a realizar las operaciones que sugiere el instrumento. Primero, para las preguntas 1, 3, 5, 7 y 9 lo que haremos será restar 1 al resultado de la encuesta. Por ejemplo, en la simulación que realizamos, la pregunta 3 se respondió con un valor de 3. Por lo tanto, si a 3 le restamos 1, obtenemos como resultado 2, y este valor nos permitirá realizar el siguiente cálculo que sugiere el instrumento, tal como se muestra en la Imagen 33

	Strongly disagree				Strongly agree	
1. I think that I would like to use this system frequently	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	3
2. I found the system unnecessarily complex	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
3. I thought the system was easy to use	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	2
4. I think that I would need the support of a technical person to be able to use this system	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
5. I found the various functions in this system were well integrated	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	3
6. I thought there was too much inconsistency in this system	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
7. I would imagine that most people would learn to use this system very quickly	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	2
8. I found the system very cumbersome to use	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
9. I felt very confident using the system	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	4
10. I needed to learn a lot of things before I could get going with this system	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

Imagen 33.- Operaciones a las preguntas impares de la encuesta SUS.

Fuente: Imagen capturada de [26] y editada por el autor.

Después, para las preguntas 2, 4, 6, 8 y 10, restar el valor de cada pregunta a 5, para obtener el siguiente valor. Por ejemplo, si el resultado de una pregunta es 4, al restar 4 de 5 obtenemos 1, que será el valor que tomaremos en cuenta para el siguiente cálculo. Procederemos de esta manera con todas las preguntas pares, como se observa en la Imagen 34.

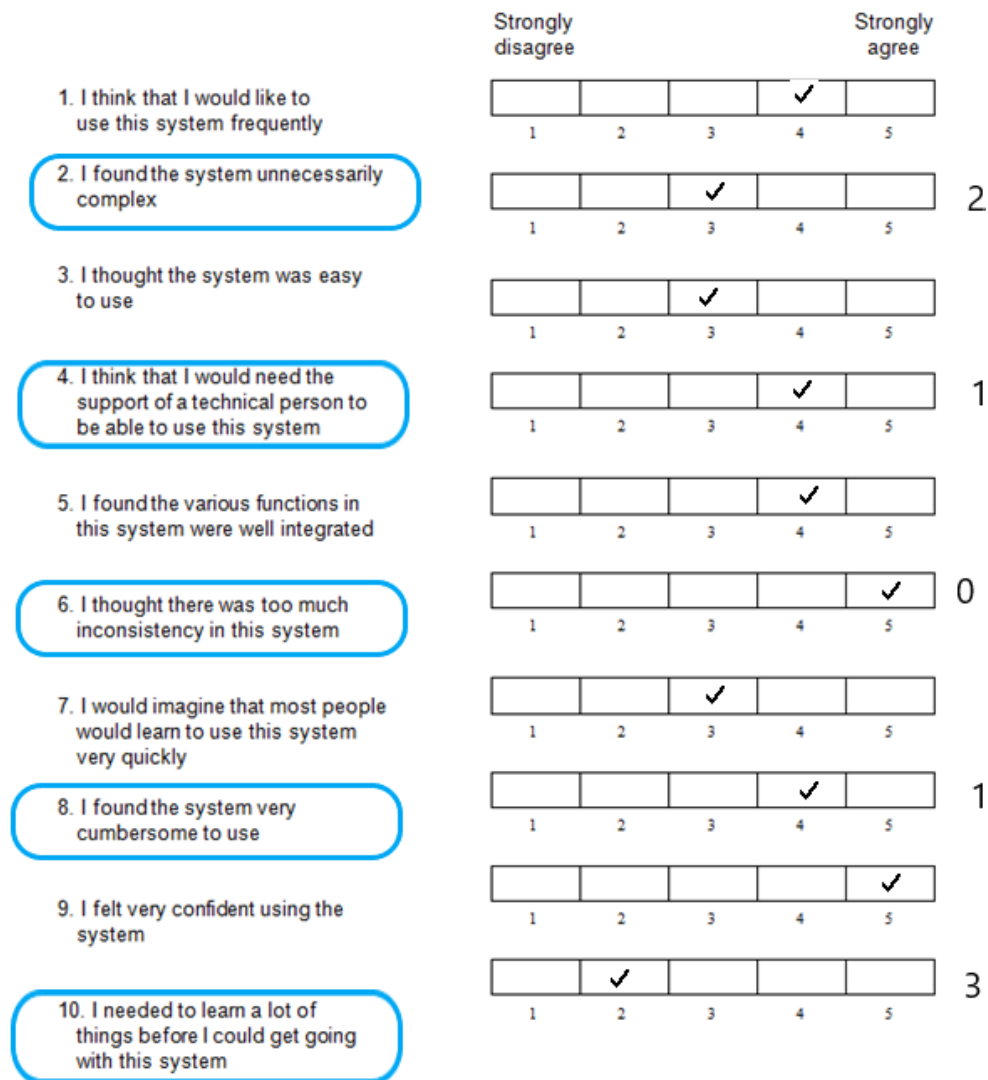
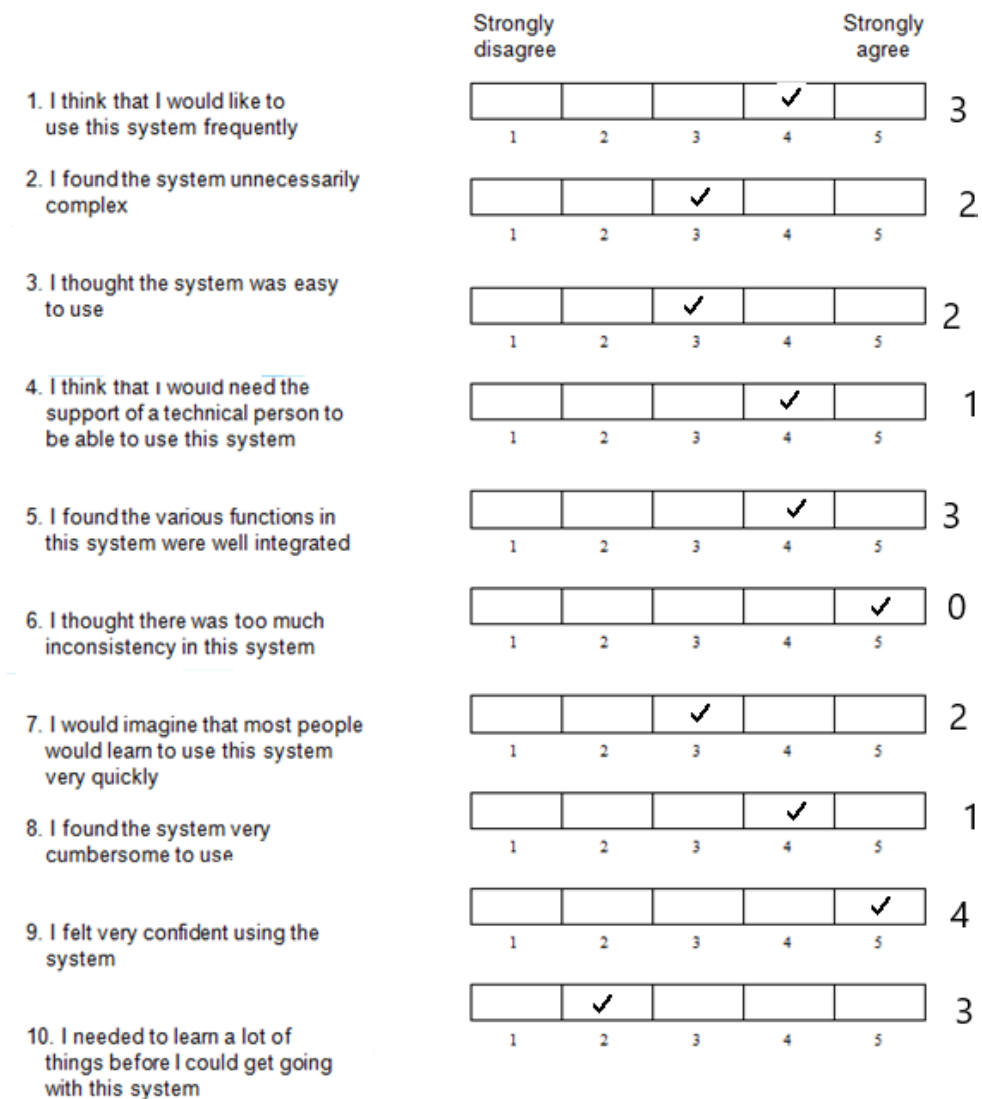


Imagen 34.- Cálculo del valor resultante para las preguntas pares de la encuesta SUS.

Fuente: Imagen capturada de [26] y editada por el autor.

Para finalizar y realizar la evaluación, sumaremos los valores obtenidos en los cálculos anteriores, para obtener una suma de todas las preguntas de la encuesta. Posteriormente multiplicaremos ese resultado por 2.5 para determinar la evaluación de la usabilidad de la aplicación. En el ejemplo que hemos planteado, el resultado de la suma es 21 como se puede observar en la Imagen 35. Si multiplicamos 21 por 2.5, obtenemos el resultado de 52.5, lo que indica un valor por debajo de la media y este valor sugiere que, para los usuarios encuestados la usabilidad de la aplicación no es tan intuitiva ni tan fácil de usar.



Suma = 21

Imagen 35.- Resultado de la suma de los valores calculados en los pasos anteriores del instrumento.

Fuente: Imagen capturada de [26] y editada por el autor.

En nuestro ejemplo, la aplicación no cumple con los estándares establecidos por el instrumento SUS, pero esto no implica que la aplicación no sea viable para su implementación; indica que debemos mejorar aspectos importantes para facilitar el uso de los usuarios. Como recomendación, se ha añadido un apartado titulado “recomendaciones”. Esto es importante, ya que puede haber aspectos que podrías pasar por alto o que consideres demasiado obvios para el usuario. Recuerda que no todos tenemos la misma lógica o comprensión de las cosas.

3.4.- Escenarios del experimento.

Para esta aplicación se plantean tres escenarios, a través de los cuales confirmaremos o desmentiremos la viabilidad de nuestro proyecto. Mediante estos escenarios, evaluaremos la usabilidad de nuestra aplicación, ya que, si alguno de ellos se presenta, podemos inferir si la aplicación es exitosa o si es necesario mejorar las deficiencias que sean identificadas. Estos escenarios han sido ideados por nosotros y son los siguientes:

- Escenario ideal.
- Escenario regular.
- Escenario desfavorable.

A continuación, presentaremos de manera detallada cómo consideraríamos estos escenarios, específicamente en relación con la forma en que cada uno de ellos se manifestaría para nosotros. La razón para plantear estos escenarios es evaluar cuál de ellos se repite al momento de realizar los experimentos y apoyados en las opiniones que nos den las personas que utilizaron nuestra aplicación podremos determinar qué tan amigable es nuestra aplicación para los usuarios finales.

El recorrido para este experimento se llevará a cabo en la zona céntrica del municipio de Juan Galindo, abarcando cuatro puntos que considerados importantes en dicha localidad, los cuales son:

- La presidencia municipal.
- El monumento al coronel Juan Galindo
- Turbina conmemorativa de los 100 años de luz y fuerza
- Plaza cívica municipal “Armada de México”

Puntos en los que se mostrará un contenido 3D, el cual fue descargado de internet, así como la información recopilada en [27], que consiste en una recopilación de antecedentes históricos sobre Juan Galindo, elaborada Sr. Cándido Garrido.

Para ilustrar el recorrido, mostraremos el lugar donde se llevó a cabo el experimento: el jardín central de Nuevo Necaxa, que pertenece al municipio de Juan Galindo. En la imagen capturada de Google en la cual marcamos con un asterisco rojo cada punto que se tomarán en cuenta para el recorrido del experimento, como se puede observar en la Imagen 36.

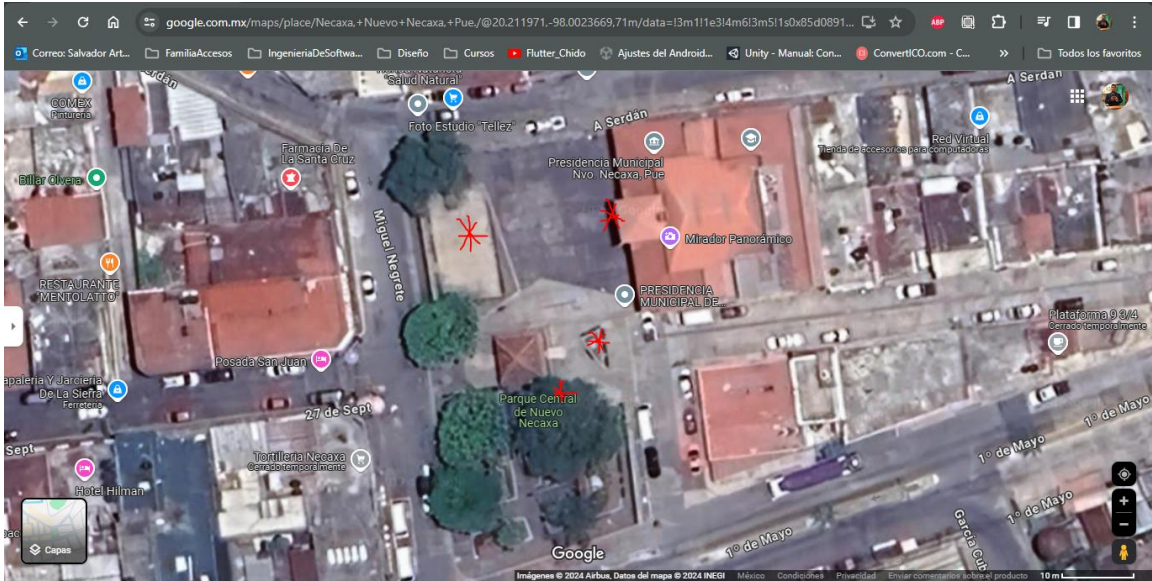


Imagen 36.- Vista de los lugares a ser visitados en el recorrido de este experimento.

Fuente: [28] y editada por el autor.

Escenario ideal:

Para considerar un experimento como exitoso, es necesario tener en cuenta diversos aspectos. Definiremos un experimento como exitoso en el que el usuario lee y acepte los términos y condiciones de manera intuitiva, de forma que no requiera un tutorial breve para completar el registro de la cuenta y el inicio de sesión de forma automática.

Otro aspecto que consideraríamos exitoso en el experimento es que el usuario pueda ubicarse rápidamente en el punto inicial del recorrido, sin necesidad de solicitar ayuda o preguntar en donde se encuentra el punto que se le está indicando, y que posteriormente continúe de manera eficiente por el camino señalado por la misma aplicación.

Consideramos exitoso que el usuario identifique las funciones del menú, que reconozca el propósito de cada acción y pueda cancelar y realizar las operaciones necesarias en la aplicación sin preguntarle a los responsables como realizar cada operación, ya que tanto apoyo necesita el usuario. Si este escenario es el que se presenta más frecuentemente, no tendríamos problemas a la hora de lanzar una versión final.

Escenario Normal:

Consideraremos como “escenario normal” el que el usuario necesite algo de ayuda para realizar ciertas tareas, como el llevar a cabo el registro o cambiar sus datos. Es importante que el usuario intuya varias funcionalidades del menú y que pueda utilizar la aplicación con poca asistencia de la persona que realiza el experimento.

El hecho de que el escenario sea considerado como “normal” es cuando las personas puedan guiarse de manera autónoma al momento de realizar el recorrido.

Escenario desfavorable:

Por último, en un escenario desfavorable, el usuario no solo necesitará ayuda para comprender el propósito de la aplicación, sino que también ignorará como utilizarla y desconocerá el propósito de sus funcionalidades.

Proponer estos escenarios tiene como objetivo establecer un punto de referencia para evaluar la aceptación de nuestra aplicación. Además, a través de este experimento podremos determinar la viabilidad de publicar nuestro proyecto, así como el nivel de apoyo que recibirá por parte de los posibles usuarios finales y ver cuan innovadora se considera nuestra aplicación.

Capítulo 4. Solución para el recorrido con realidad aumentada.

4.1.- Introducción.

En este capítulo se presenta la solución propuesta para el recorrido con realidad aumentada en el centro histórico de la ciudad de Puebla, el cual nos ayudará a incentivar el turismo y generar economía en la región. Se especificará el desarrollo de la aplicación y de cada uno de sus componentes, así como el funcionamiento del motor de videojuegos que utilizamos para llevar a cabo la aplicación. Además, se abordarán elementos y conceptos básicos que se irán adquiriendo a medida que se aprenden a desarrollar aplicaciones o videojuegos en este motor.

También complementaremos las definiciones de las secciones 1.3 y 3.1 de este documento, con el objetivo de que el lector tenga menos dificultades al leer términos como: prefab, game object, script, entre otros. Además, proporcionaremos explicaciones breves, pero precisas sobre cómo crear ciertos elementos dentro del entorno de desarrollo, así como la finalidad y la utilidad de cada uno de los elementos que hemos utilizado a lo largo de la creación de esta aplicación.

4.2.- Definiciones necesarias.

En esta sección agregaremos definiciones necesarias para entender los conceptos del capítulo, ya que hablaremos del desarrollo de la aplicación. Esta decisión se tomó porque en este capítulo encontrarás secciones en las que se te indicara que crees elementos, pero no se detallará el procedimiento para hacerlo, ya que incluir esas instrucciones provocaría que el documento se vuelva innecesariamente más extenso y tedioso de leer. Además, se proporcionarán definiciones consideradas importantes para que el lector las conozca, como las instrucciones que nos permitirán realizar ciertas tareas y las que serán de gran apoyo para el proyecto.

4.2.1.-Elementos y componentes de Unity.

- **GameObject:** Básicamente es un contenedor, el cual puede almacenar distintas cosas, ya sean objetos 3D, scripts, sonidos, canvas, etc. Y el elemento al que contiene es el que implementa la verdadera funcionalidad. La forma de crearla es muy simple y se logra dando clic derecho y seleccionando la opción “Crear empty”, lo cual nos creará un GameObject vacío y podrás ir incorporando a él los elementos que desees.
- **Input field:** Es un cuadro en el cual el usuario puede ingresar texto, nos ayuda para solicitar datos y podemos acceder a esa información por medio de script y otras formas, este objeto

se crea de igual manera, con clic derecho solo que esta vez nos dirigiremos a la sección de “UI”, en donde encontramos todos los componentes de ese tipo.

- Canvas: Es el contenedor de todos los elementos UI, es un game object con un componente canvas en él y los elementos ui tienen que ser hijos del mismo, el canvas es mostrado como un rectángulo en la vista de la escena, para crear un elemento canvas tenemos que dirigirnos al apartado mencionado en el párrafo anterior siguiendo la siguiente ruta “**UI>Canvas**”, accediendo a este menú con clic derecho sobre la jerarquía, cabe mencionar que a la hora de crear cualquier elemento ui, se crea el canvas de manera automática, ya que los elementos ui no pueden existir sin él.
- Button: Como en cualquier aplicación, el botón nos ayuda a ejecutar ciertas acciones, lo que cambia en Unity es el que acciones podemos llevar a cabo, ya sea por medio de funciones de script o arrastrando los objetos directo al botón y podríamos modificar todos los componentes de dicho objeto.
- Panel: El panel es como un componente “image”, con la diferencia que el panel tiene diferentes configuraciones, por ejemplo; el panel abarca el 100% del canvas, además de que viene con un color en blanco, pero transparente, no como el componente image que viene en un color en blanco, opaco en su totalidad y las pestañas de responsividad en el centro no como las del panel que se ubican en las esquinas del panel.
- Image: Este componente nos permite colocar una imagen o un fondo en el canvas, siendo que acepta imágenes “Sprite” las cuales el mismo Unity te genera.
- Sprites: son objetos gráficos 2D y básicamente son texturas estándar, pero hay técnicas especiales para combinar y manejar texturas sprites por eficiencia y conveniencia durante el desarrollo [29].
- Asset: Un asset es una representación de cualquier item que puede ser utilizado en su juego o proyecto. Un asset podría venir de un archivo creado afuera de Unity, tal como un modelo 3D, un archivo de audio, una imagen, o cualquiera de los otros tipos de archivos que Unity soporta [30].

Todos los componentes mencionados anteriormente se crean dando clic derecho en la jerarquía. Luego nos dirigimos a la sección de UI, donde encontraremos todos los elementos que hemos mencionado, como se puede observar en la Imagen 37. También puedes generarlos desde

la pestaña GameObject que se encuentra en la parte superior, resaltada en la Imagen 37. La imagen muestra Formas de crear los elementos de la sección 4.2.1 mostrando las 2 opciones, marcando en amarillo la sección del menú superior mencionada y el * marca el espacio que mencionamos como jerarquía y en esa parte es donde damos el clic derecho para crear los componentes.

Como conclusión de esta sección queremos dejar en claro que estos son los elementos que utilizamos en este proyecto. Sin embargo, Unity ofrece muchos más recursos, como elementos de audio, gráficos 3D, gráficos 2D, video, iluminación, entre otros, los cuales requieren tiempo y práctica para ser dominados por el desarrollador.

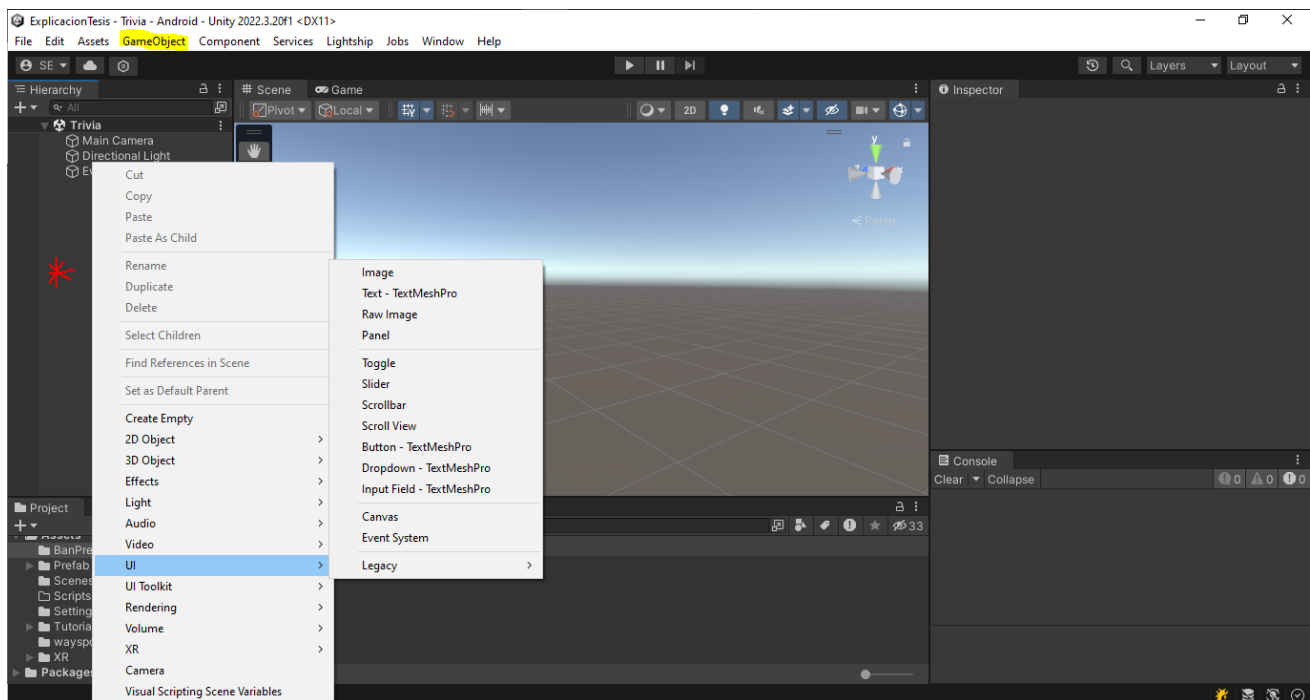


Imagen 37.-Formas de crear los elementos.

Fuente: Imagen capturada de software instalado y editada por el autor.

4.2.2.- Definiciones de instrucciones dentro de los scripts.

Por instrucciones de los scripts, nos referiremos tanto a funciones como a elementos que, mediante código nos permitirán implementar las funcionalidades que necesitemos incorporar a nuestra aplicación.

- TMP_Text: Es una clase que contiene distintos elementos y la cual nos permite controlar distintos componentes UI, ya que debido a que los elementos de texto de esos componentes son de este tipo de texto, si quieres más información de los elementos que contiene y con los

que podrás echar mano por medio de esta clase, la información completa la encuentras en [32].

- Yield: Lo que el operador yield realiza es pausar la ejecución de la iteración y devolver el valor al método que realiza la llamada para que este continúe con su ejecución y cuando termine volverá al siguiente punto del iterador (el que está en el método con yield return) [33].
- Scriptable object: Es un contenedor de datos que puede utilizar para guardar grandes cantidades de datos, independientemente de las instancias de clase. Uno de los principales casos de uso de ScriptableObjects es reducir el uso de memoria de su proyecto evitando copias de valores [34].
- Corrutina: En el caso de un método, este se ejecuta en su totalidad antes de regresar una respuesta, es decir, no puede esperar a que se renderice cada frame y para eso utilizamos las corrutina, con las cuales podremos detener el código por una cierta cantidad de tiempo y después poder continuar.
- Frame: Frame en inglés significa cuadro y lo usamos para hacer referencia a cada una de las imágenes estáticas que forman parte de la sucesión de imágenes de un videojuego [35], aunque se muestre una imagen estática, el juego se actualiza y ejecuta cierta cantidad de frames por segundo.
- Scenes (escenas): Las escenas son donde se trabaja con el contenido en Unity. Son activos que contienen todo o parte de un juego o aplicación. Por ejemplo, usted podría construir un juego simple en una sola escena, mientras que, para un juego más complejo, usted podría utilizar una escena por nivel, cada uno con sus propios entornos, personajes, obstáculos, decoraciones, y UI [36] y esta se crea de una manera sencilla y tenemos que seguir la ruta “Assets > Create > Scene”.
- ScriptableObject: Es un contenedor de datos que puede utilizar para guardar grandes cantidades de datos, independientemente de las instancias de clase [37].
- Serializable: Es el proceso automático de transformar estructuras de datos o estados GameObject en un formato que Unity puede almacenar y reconstruir más tarde [38]. Y lo que logramos con esta instrucción es poder editar desde el inspector a los elementos a los que se los coloquemos por medio de script con la instrucción “[Serializable]” antes del tipo y nombre de la variable o elemento que deseamos volver serializable.

- PlayerPrefs: Es una clase que almacena las preferencias del jugador entre sesiones de juego. Puede almacenar valores string, float e integer en el registro de la plataforma del usuario. Unity almacena PlayerPrefs en un registro local, sin encriptación. No utilice los datos PlayerPrefs para almacenar datos sensibles [31].

4.2.3.-Definiciones ajenas a Unity.

Aquí explicaremos definiciones que no son propias de Unity; sin embargo, es conveniente que las conozcas, ya que se hacen referencia a ellas en las secciones siguientes.

- GitHub: Se trata de una de las principales plataformas para crear proyectos abiertos de herramientas y aplicaciones, y se caracteriza sobre todo por sus funciones colaborativas que ayudan a que todos puedan aportar su granito de arena para mejorar el código [39].
- Repositorio github: Un repositorio es el elemento más básico de GitHub. Es un lugar donde puedes almacenar el código, los archivos y el historial de revisiones de cada archivo [40].
- VPS (Visual Positioning System “Sistema de Posicionamiento Visual”): Determina la ubicación de un dispositivo basándose en imágenes y no en señales GPS. VPS crea primero un mapa tomando una serie de imágenes que tienen una ubicación conocida y analizándolas en busca de características visuales clave, como el contorno de edificios o puentes, para crear un índice a gran escala y de búsqueda rápida de esas características visuales [41].
- PHP: Acrónimo recursivo de PHP: Hypertext Preprocessor, es un lenguaje de código abierto muy popular especialmente adecuado para el desarrollo web y que puede ser incrustado en HTML [42].
- Método POST: Existen 2 métodos con los que el navegador se comunica con el servidor, uno es el método GET y el otro es el método POST y la gran diferencia es que con el método get la información se envía en el header por lo que se ve en la ruta o la dirección a través del navegador y con el método post, esto no sucede.
- PHPMailer: La librería PHPMailer nos permite, entre otras funcionalidades, establecer una conexión SMTP con un servidor de correo. Este servidor SMTP será quien envíe realmente nuestro correo electrónico [43].
- try-catch: Generalmente es utilizado para controlar las excepciones que pueden producirse durante la ejecución del código dentro de un bloque try.

Con esta información concluimos las secciones complementarias de las definiciones y esperamos que te sean de ayuda para comprender mejor el trabajo que realizamos. También nos esforzaremos por ser lo más precisos posible en las secciones en las que explicaremos cómo se llevaron a cabo implementaron todos los elementos de la aplicación.

4.3.- Conociendo Unity.

Antes de explicar ciertos desarrollos y para no hacer las secciones extensas, presentaremos los componentes básicos de Unity. Esto permitirá que, al mencionar ciertos elementos, el lector no se encuentre con conceptos desconocidos y pueda ubicarlos de manera sencilla con solo vistazo a esta sección.

Si se observa la Imagen 38, se podrá cuenta que las secciones están señaladas mediante rectángulos de distintos colores y numeradas del 1 al 7. A continuación, explicaremos de la manera más clara posible el funcionamiento y el objetivo de cada elemento, así como todo lo que se puede hacer y crear en cada uno de ellos.

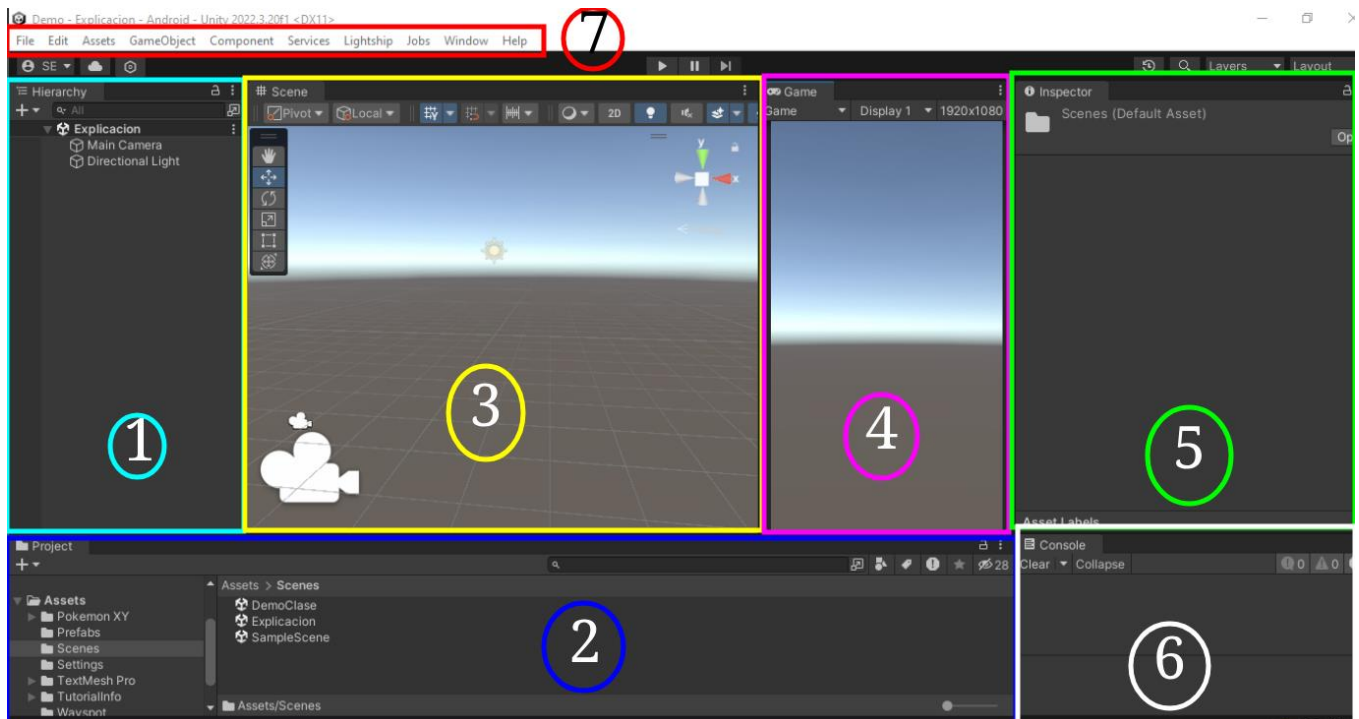


Imagen 38.-Componentes básicos del entorno de Unity.

Fuente: Imagen capturada de software instalado y editada por el autor.

1. Hierarchy: En este apartado vamos a encontrar todos los elementos que se van integrando a la escena, para incorporar los elementos en la escena, tenemos que crearlos dando clic derecho sobre cualquier parte de la que se ve encerrada en el cuadro azul marcado como 1 o bien podemos hacerlo desde el menú superior señalado con el número 7 que más adelante detallaremos. También podemos ocultar, mover y renombrar los elementos que hemos creado y agregado a la escena.
2. Project: En este apartado encontraremos todas las carpetas del proyecto, podremos crear y eliminarlas, todo para tener un mejor orden de tu proyecto, también los scripts se crearán y ubicarán en este apartado, puedes crear una carpeta especial para ellos y eso es lo que te recomendamos, ya que tu trabajo será más fácil y sencillo de realizar, y aquí se encontrarán los elementos que se importen al proyecto.
3. Scene: En este apartado veremos esencialmente el contenido que verá el usuario al utilizar nuestra aplicación. Veremos la cámara, la luz y demás componentes que se agreguen a nuestro juego. Y es ahí donde podremos acomodar los componentes, ya que es aquí donde podemos crear, ver y modificar tanto los componentes 3D, los terrains, las perspectivas de la cámara, los canvas, etc.
4. Game: En el apartado anterior el desarrollador podrá ver y modificar la escena en la que se encuentre trabajando, en el apartado de “Game” podrá ir viendo cómo funciona dicha escena y como interactúan los botones o las acciones que vaya programando. Podrás ejecutar el software que estes desarrollando, así como ver los resultados o los errores que vallan apareciendo antes de compilarlo para la plataforma que desees, también podrás ajustar el tamaño de la pantalla a un tamaño especial, para ir viendo la efectividad de lo responsivo de los botones.
5. Inspector: Aquí podremos ver todos los elementos de los componentes, por ejemplo, un GameObject vacío, siempre tiene como componente inicial un elemento “Transform” que es un elemento en el que se ven las coordenadas en las que se ubicará el GameObject, si tenemos un cuadro, este contendrá elementos como el transform, un componente mesh, un componente image, un componente de materiales, etc. Dichos componentes los podremos ver y modificar desde el inspector.
6. Console: Este componente cualquier desarrollador lo conocerá, esto debido a que es como la consola de cualquier software de desarrollo, donde podrás ver todas las impresiones de

pantalla que se realizarán para ver en funcionamiento de los scripts o para ver lo que desee el programador que no es necesario que el usuario los vea.

7. Menú superior: Básicamente es un menú que encontraremos en la parte superior, el cual será de un acceso más simple si sabes que contiene cada opción, obviamente este editor cuenta con accesos más simples o atajos que podrás encontrar conforme vayas conociendo Unity, pero si vas empezando o te acomodas más con este menú podrás utilizarlo sin problemas, todo dependerá de lo que decidas.

Estos componentes son los que consideramos esenciales para comenzar a desarrollar aplicaciones con Unity. Como mencionamos en el apartado 7, todo dependerá de cómo se adapte el desarrollador a la herramienta. Aunque Unity es una plataforma de desarrollo relativamente sencilla, su complejidad radica en la lógica del programador y la curva de aprendizaje que el programador logre generar al momento de adquirir conocimientos. Para los fines de este trabajo, se eligió Unity por ser una herramienta accesible y de fácil aprendizaje. Ya contábamos con experiencia en el desarrollo de funciones simples en esta plataforma, y hay comentarios que indican que no es necesario tener experiencia previa para trabajar con Unity. Por lo tanto, asumiremos que intentarás familiarizarte con lo básico de Unity antes de replicar los procesos que se mostraron en este trabajo, una vez mencionado esto, aclaramos que debido a esa suposición que hemos realizado, nos tomamos la libertad de excluir ciertas explicaciones.

4.4.- Creación del proyecto.

El primer paso que tenemos que llevar a cabo es crear el proyecto. Para ello nos dirigimos a nuestro Unity Hub y en él seleccionamos la opción New Project, esta opción nos enviara a un apartado en el que podremos seleccionar que tipo de aplicación desarrollaremos, ya sea una aplicación 3D, 2D, etc. Como puedes ver en la Imagen 39. Y para ello deberas tener en claro que tipo de aplicación quieres desarrollar. Como comentario te decimos que en caso de que te equivoques y desees crear un proyecto 3D, pero seleccionaste un proyecto 2D; no te preocupes podrás modificar el proyecto y volverlo 3D sin necesidad de crear otro proyecto.

En este caso seleccionamos el proyecto HRP3D para desarrollar esta aplicación, esto es una recomendación, aunque puedes seleccionar este tipo de proyecto, también puedes seleccionar la opción 3D, sin problemas y la plataforma la cambiaremos después se dará una pequeña instrucción como se hace el cambio. Cabe mencionar que, en dicho panel, debemos seleccionar la

ubicación donde queremos que se guarde el proyecto, así como la organización a la que pertenece. Además, es necesario proporcionar el nombre del proyecto y, a continuación, proceder a seleccionar el botón de aceptar. Ten en cuenta de que dicha opción se habilitará cuando los campos mencionados anteriormente estén completos. La Imagen 39 muestra el Panel principal de Unity Hub, encerrando la opción de crear nuevo proyecto para empezar la creación del proyecto.

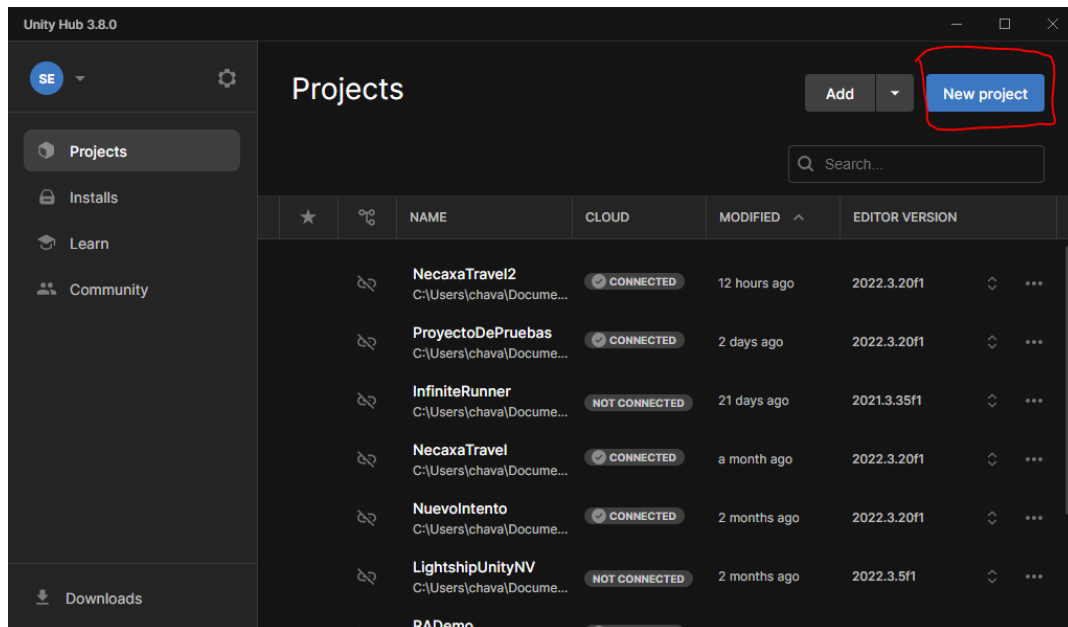


Imagen 39.- Panel principal de Unity Hub.

Fuente: Imagen capturada de software instalado y editada por el autor.

La Imagen 40 muestra el Panel de creación de proyecto, mostrando las opciones que tenemos que llenar de manera obligatoria.

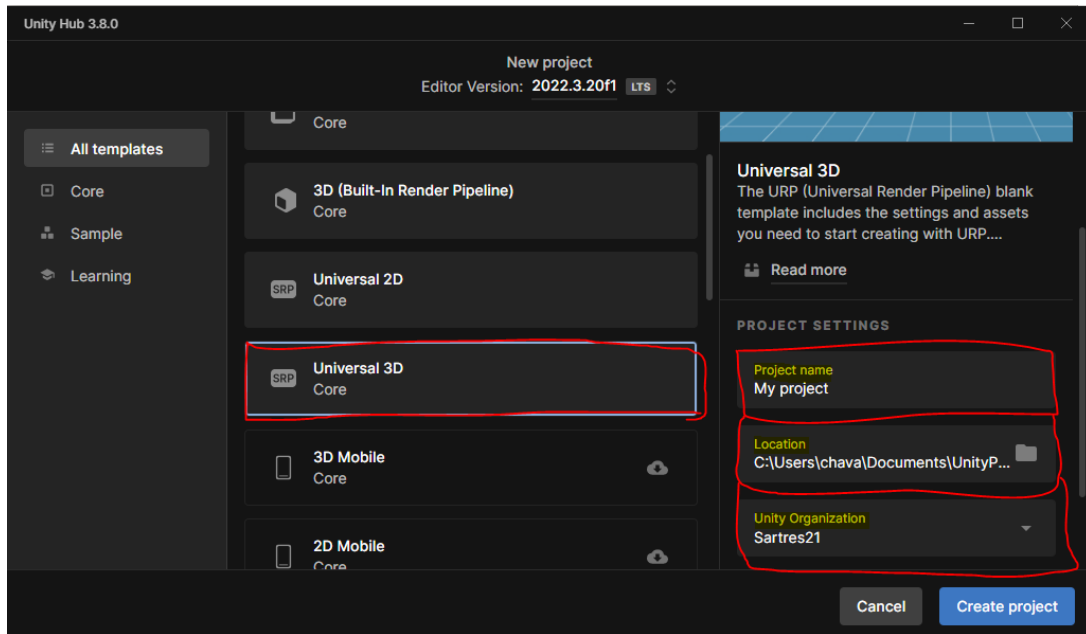


Imagen 40.- Panel de creación de proyecto.

Fuente: Imagen capturada de software instalado y editada por el autor.

4.5.-Volviendo la interfaz responsiva.

En esta sección, haremos que los elementos se redimensionen de acuerdo con la pantalla de los dispositivos que instalen nuestra aplicación. Esto se logra de una manera muy sencilla: si creas un componente UI, en el centro se generan cuatro pestañas y se ubican en el centro de la pantalla. Estas pestañas son las responsables de que los elementos se redimensionen adecuadamente a cada pantalla en la que se instale la aplicación creada, esto debido a que trabaja con porcentajes y no con píxeles.

Se observa la Imagen 41, notará que se ha creado un canvas con un panel sencillo para oscurecer y poder crear un componente image. Además, se han señalado las pestañas, aunque puede ubicarlas en el lugar que desees, tenga en cuenta que la posición que elija determinará el área límite en la que se redimensionará el contenido.

Como conclusión de esta sección, podemos reconocer que la forma en la que se vuelven responsivos los elementos que contiene una escena en Unity es sencilla. A diferencia de otros casos, como en el caso de Bootstrap, donde es necesario establecer diferentes condiciones según diversas medidas, en Unity esto no es así. Basta con tener una noción del espacio que deseamos que nuestro componente use para readaptar estas pestañas.

Se tiene que mencionar que, si deja las pestañas en el centro, es decir, si no la mueve de la posición en donde aparecen al crear el componente, el elemento no se redimensionará. Si instala la aplicación en un dispositivo que tiene las dimensiones con las que estás trabajando, no ocurrirá nada. Sin embargo, si el dispositivo en el que instala la aplicación tiene dimensiones de pantalla menores a las dimensiones con las que está trabajado, los componentes se deformarán y puede suceder que incluso no pueda hacer clic en ellos.

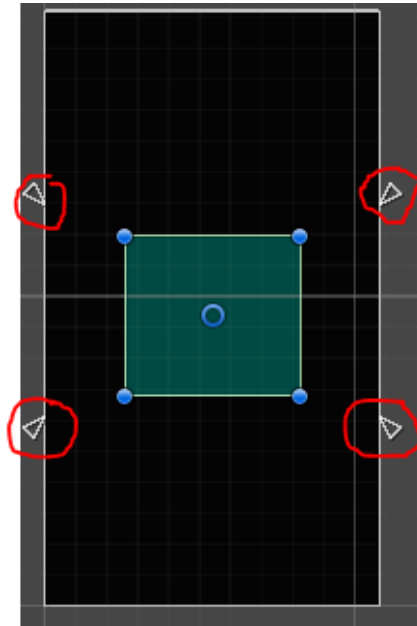


Imagen 41.-Canvas creado con un panel y un componente image.

Fuente: Imagen capturada de software instalado y editada por el autor.

4.6.- Términos y condiciones.

En esta sección, te mostraremos como elaborar los términos y condiciones que se presentan al iniciar la aplicación. Este proceso es realmente sencillo, ya que solo utilizaremos un canvas, un cuadro de texto, dos botones, un toggle y un script que nos permitirá gestionar el texto que mostraremos, así como el player prefab, para que esta escena aparezca únicamente una vez o cada que nosotros lo deseemos.

Primero, creamos un componente “image”. Recuerda que al crear el componente “image”, el “canvas” se generara automáticamente. Después de eso, importaremos la imagen que utilizaremos de fondo para la aplicación. Para importar el recurso que utilizaremos, primero crearemos una carpeta que llamaremos imágenes en la cual iremos colocando los componentes de

este tipo que utilizaremos en el proyecto. Dentro de esa carpeta creamos otra que denominaremos “fondos y logos”.

Para importar elementos a Unity, tenemos dos métodos. El primero consiste en hacer clic en el espacio de la carpeta en donde dejaremos el recurso, luego damos clic derecho y seleccionamos la opción “Import New Asset...”. Esto nos abrirá un cuadro de dialogo en el que debemos buscar el recurso que queremos importar. Una vez seleccionado aceptamos y así se importará a nuestro proyecto.

El segundo método para importar recursos es abrir la carpeta donde se encuentra el componente que deseamos importar y arrastrarlo directamente a Unity, colocándolo en la carpeta creada, así tendremos la imagen, sonido o lo que deseemos importar.

Una vez que haya importado la imagen, el siguiente paso es convertirla en una imagen que sea compatible con el componente, es este caso la convertiremos a “Sprite”. Para ello, seleccionamos la imagen en la carpeta donde la importaste, la seleccionamos y nos dirigimos al inspector. En el inspector encontraras una opción que dice “Texture Shape”, que por defecto está configurada en 2D. Despliega ese menú y selecciona la opción “Sprite (2D and UI)” como se muestra en la Imagen 42. Posteriormente desplázate hasta las últimas opciones del inspector y presionas “Apply”, que es la opción encerrada que se observa en la Imagen 42.

Toma en cuenta que, si no llevas a cabo este procedimiento, algunas imágenes serán aceptadas por el componente, pero en algunos casos no lo serán. Por eso, te recomendamos llevar a cabo este procedimiento para evitar problema, ya que esta acción no te tomará mucho tiempo y Unity no se demorará en procesar la transformación.

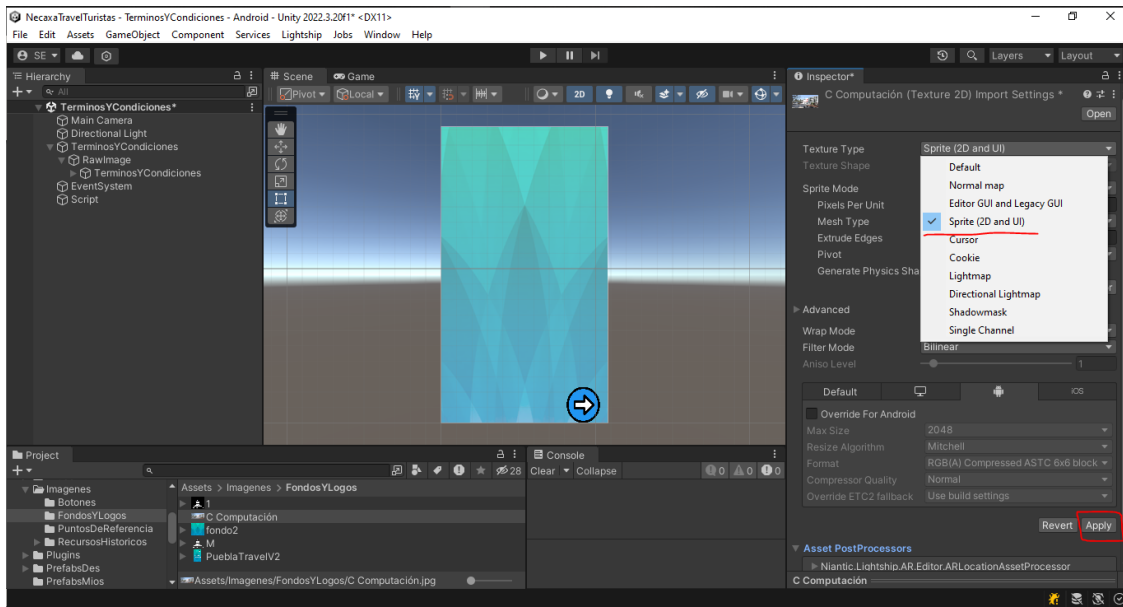


Imagen 42.-Convirtiendo una imagen recién importada a “Sprite 2D”.

Fuente: Imagen capturada de software instalado y editada por el autor.

Cuando cumplimos los pasos anteriores, agregaremos un panel. Dentro de este panel, incluiremos un componente de tipo “Text(TMP)” en el cual colocaremos el texto completo de los términos y condiciones. También agregaremos dos botones: uno para avanzar en el texto y el otro para aceptar los términos y condiciones. El ultimo botón aparecerá al seleccionar el último componente que debemos agregar, que es el componente “Toggle”. Este componente nos permite activar o desactivar la casilla para indicar la selección de algún dato o componente En este contexto, consideraremos que al hacer clic en esta casilla se aceptan los términos y condiciones. Una vez que aceptamos, se mostrará el último botón, que nos llevará a otra escena y creará nuestro player prefab, utilizando el script que se presenta en la imagen 37.

Si observa la Imagen 42, en la jerarquía se encuentra un GameObject llamado “Script”, con este componente podremos manipulas los elementos que se van a ir mostrando y para esto primero tienes que agregar el script al GameObject y lo hacemos seleccionando el GameObject para ver sus componentes en el inspector. Una vez que tenemos aparecen las propiedades del componente en el inspector, entonces arrastramos el script que deseamos. También puedes seleccionar la opción “Add Component” y buscamos el script que deseas agregar al objeto.

Como incorporamos el script al GameObject, tenemos que agregar los elementos necesarios para que se ejecute correctamente, esos elementos los podrás ver en el inspector

(siempre y cuando el componente este seleccionado), ya que estos son los que solicitan el mismo. Como se puede ver en la Imagen 43, estos son los componentes que debemos utilizar en esta sección de la aplicación. Los componentes se referencian simplemente arrastrándolos al inspector (con el componente seleccionado). Es importante asegurarse de que el tipo del componente que estas solicitando en el script sea el correcto; si llegas a referenciar un componente de un tipo diferente, este no será aceptado.

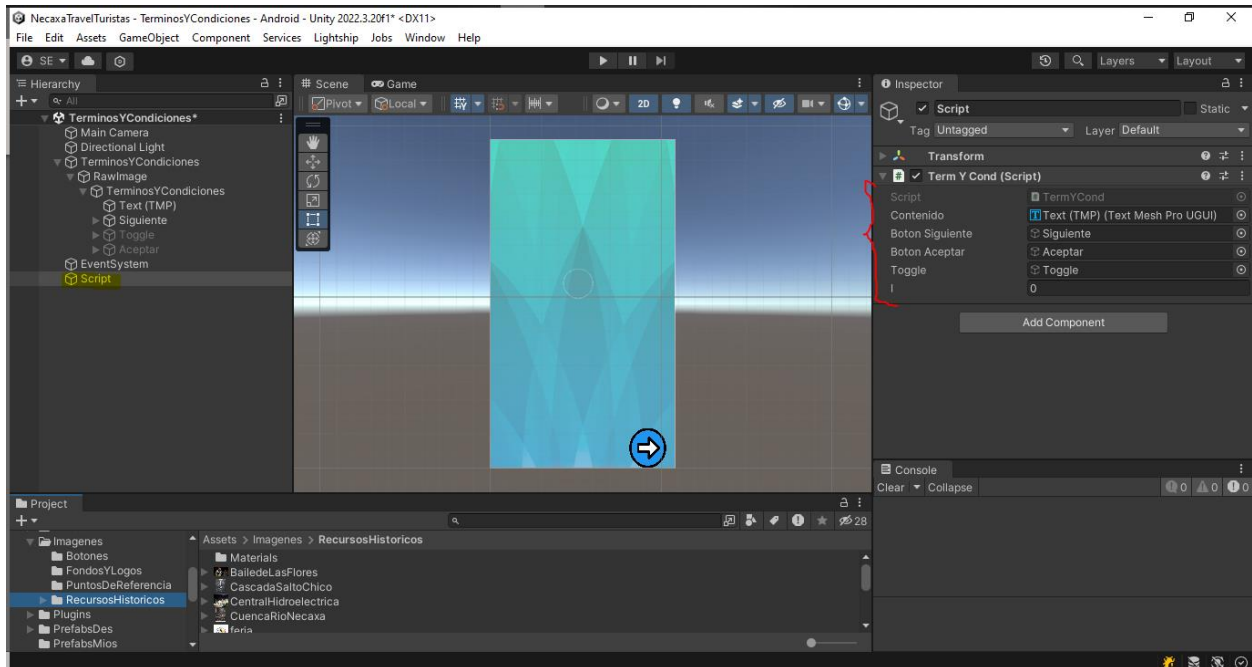


Imagen 43.-Referenciando los elementos a los componentes indicados.

Fuente: Imagen capturada de software instalado y editada por el autor.

Para desarrollar el script que controlará esta escena, debemos solicitar los componentes que vamos a manipular anteponiendo la instrucción “SerializeField”. Esto para que Unity nos permita referenciar los elementos, aunque sean privados. El script queda como se muestra en la imagen 38.

En las primeras líneas, encontrarás los elementos que estamos solicitando para poder implementarlos mediante código simple. Esto se debe a que solo activaremos o desactivaremos elementos dependiendo del caso que necesitemos. También hemos creado un arreglo de strings, en el que se contendrá los términos y condiciones. A través de la función “Actualizar” la cual incrementara la variable siempre y cuando esta sea menor al tamaño del arreglo, si el tamaño llega a ser mayor, entonces se ejecutara el else, el cual nos desactiva el componente de “botonSiguiente”

y activa el “toggle” que utilizaremos para que el usuario acepte los términos y condiciones. Una vez que acepto, creamos nuestro PlayerPrefs que nombramos “TerminosYCond” y le asignaremos el valor “Aceptado” (básicamente con este PlayerPrefs nos servirá de bandera para saber si se salta o no la escena).

Los métodos Start y Update solo se encarga de colocar el contenido de la posición i. Dado que inicializamos a i en 0, esto nos permitirá establecer el contenido al inicio. El método update mostrará el valor correspondiente a la información que contiene el arreglo en posición i, siempre que nuestro método lo modifique. Cuando ya no se cumple la condición es cuando se empiezan a hacer las modificaciones.

```
7 public class TermYCond : MonoBehaviour
8 {
9     [SerializeField] private TMP_Text contenido;
10    [SerializeField] private GameObject botonSiguiente;
11    [SerializeField] private GameObject botonAceptar;
12    [SerializeField] private GameObject toggle;
13    public int i = 0;
14    private string[] terminos =
15    {
16        "Términos y condiciones: \r\nPuebla Travel AR\r\nEsta aplicac
17        "Cuenta con el Doctor Juan Manuel González Calleros, profesor
18        "El objetivo de los datos que se le pedirán en el registro es
19        "La propiedad intelectual de esta aplicación pertenece solame
20        "Queda prohibido su uso para fines políticos o beneficio de p
21    };
22    // Mensaje de Unity 0 referencias
23    private void Awake()
24    {
25        string tyco = PlayerPrefs.GetString("TerminosYCond");
26        if ( tyco != "" )
27        {
28            SceneManager.LoadScene("RegistrarUsuario");
29        }
30    }
31    // Mensaje de Unity 0 referencias
32    public void Start()
33    {
34        contenido.text = terminos[i];
35    }
36    // Mensaje de Unity 0 referencias
37    public void Update()
38    {
39        contenido.text = terminos[i];
40    }
41    // 0 referencias
42    public void Actualizar()
43    {
44        if(i < terminos.Length - 1)
45        {
46            i++;
47        }
48        else if(i == terminos.Length - 1){
49            botonSiguiente.SetActive(false);
50            toggle.SetActive(true);
51            PlayerPrefs.SetString("TerminosYCond", "Aceptado");
52        }
53    }
54 }
```

Imagen 44.-Script con el que trabajamos en la escena de términos y condiciones.

Fuente: Imagen capturada de software instalado y editada por el autor.

Por último, tenemos un método llamado “Awake” que se ejecuta incluso antes que el método start. Este método es el que nos ayudara a que la escena no se inicie si ya aceptamos los términos y condiciones anteriormente. Esto se desarrolló así para que no moleste al usuario. Esta condición se implementa mediante un if, que agregamos previamente el contenido de la etiqueta

a una variable de tipo string con el método “PlayerPrefs.GetString(“Nombre de la llave”, “Valor que queremos guardar”)”. En este caso la llave se relaciona con el dato que queremos verificar. La llave que creamos la nombramos como “TerminosYCond”. Si esa llave se encuentra vacía no hacemos nada, pero si encuentra con algún dato almacenado en ella (que en la sección anterior indicamos que se llenaría con una palabra clave para verificar que el usuario ya acepto los términos y condiciones), nos llevara a la escena de “Registro de usuario” para iniciar dicho proceso.

El cambio de escena lo llevamos a cabo gracias a la instrucción “SceneManager.LoadScene(“Nombre de la escena”)” la cual nos guiara a la escena que deseemos. Como mencionamos en el párrafo anterior, la escena a la que nos dirigiremos es la de registrar usuario. Es importante considerar que puedes hacer el cambio de escena, por nombre o por número, y se logra controlando la posición de la escena al compilar y generar el instalador para el dispositivo que se utilizara. En nuestro caso, generaremos un archivo “.APK” para que la aplicación se instale en un dispositivo Android y esto lo detallaremos mejor en la sección 4.14, donde nos enfocaremos en la generación de dicho archivo.

Por último, contamos con un script básico que nos permite cambiar de escena, al que hemos denominado: “CambiarScene”. Posteriormente lo asignaremos a un botón para facilitar el cambio a la escena que deseemos. En esencia, este script es similar la instrucción que se muestra en la Imagen 44, línea 27, con la diferencia de que en este método especificaremos la escena por medio de una variable de tipo string. En dicha variable, iremos indicando la escena a la que queremos trasladarnos.

4.7.- Registrar Usuario.

En esta sección, abordaremos el registro de usuario, que consiste en un formulario que solicita datos a los usuarios. Esta información se inserta en la base de datos por medio de las UnityWebRequest, las cuales se describen con mayor detalle en la sección 4.8. Por lo tanto, en este momento nos centraremos en explicar cómo creamos la escena de “RegistrarUsuario”, ya que solo varían los datos que solicitamos; sin embargo, el código PHP y el archivo al que nos conectamos si cambian, por lo que detallaremos este archivo en cada sección según corresponda, enfocándonos en los cambios que tendrá el mismo.

Para comenzar a crear la escena, partimos de los elementos que se establecieron en la escena anterior: el canvas y la imagen de fondo. También agregaremos un elemento “image” que contiene

un logo en la parte superior izquierda. Esta imagen la importamos como se mencionó en la sección 4.6, de la misma manera que la imagen de fondo. Cabe destacar que no volveremos a importar el fondo en cada escena, ya que esta imagen queda importada y configurada desde el procedimiento anterior. Además, nuestra escena incluye un componente text que nos indica el título de cada escena, así como un botón que permite cerrar la aplicación. Estos elementos se repetirán en todas las escenas, por lo que ya no lo mencionaremos en secciones posteriores, ya que el panel que creamos es básicamente el mismo, cambiando el título y el contenido.

Comenzaremos con el mismo procedimiento de colocar el componente image para empezar a agregar los elementos necesarios para solicitar los datos. El campo que utilizaremos para capturar la información es el componente “InputFile-TextMeshPro”, al cual accederemos a través de su campo “text”, referenciándolo desde el script principal de la escena.

También crearemos cuatro paneles diferentes que se desarrollarán en distintas escenas y nos permitirán comunicarnos con el usuario. Los paneles son los siguientes:

- Cargando: Este panel es sencillo, solo contendrá un componente text con la palabra escrita “Cargando...” el cual se mostrará mientras el proceso se lleva a cabo esperando la respuesta de la base de datos.
- Correcto: Este panel contiene 2 componentes y es de color verde, el cual tiene como componentes un elemento text y un botón, el cual mostrara el mensaje de registro correcto, y el botón cerrara el panel y nos llevara a la siguiente escena que es la de menú.
- Error: Este panel al igual que el correcto, también cuenta con un componente text en el cual se mostrará el mensaje del servidor, es decir cuál es la causa por la cual no se pudieron insertar los datos en la base de datos y un botón el que cerrara el panel y limpiará los campos de texto de los datos para que el usuario pueda intentar el registro nuevamente.
- Cerrar aplicación: Este panel se repetirá en varias escenas, debido a que este nos sale siempre que queremos salir de la aplicación y el objetivo de este es ver que el usuario este seguro de querer salir de ella, el panel contiene un componente text el cual contiene la pregunta “¿Segur@ que quieres salir de la aplicación?”, y 2 botones, el primero que es el de cancelar lo único que hace es cerrar el panel que se ejecutó y el segundo que es el de aceptar, el cual cierra la aplicación por medio de la instrucción “Application.Quit()” la que encontramos en el método “SalirdeApp()”, función que está contenida en el script que

nombramos “CambiarScene” esto a que este contiene la función con la que podemos cambiar la escena y la función que nos permite cerrar la aplicación, esto se realizó así debido a que son funciones que se utilizarán constantemente, es por eso que se decidió crear este script el cual podemos referenciarlo distintas veces.

Además de estos componentes, la escena incluirá cuatro elementos text, que indican los datos que deben de ir en los cuatro componentes InputFile, ya que cada uno corresponde a un dato requerido para ser insertado en la base de datos. También contendrá un botón que ejecutará la función de inserción de los datos a través del script principal, el cual podrás observar desde la Imagen 45 hasta la Imagen 47.

```
0
1
2
3
4
5
6
7 public class RegistUser : MonoBehaviour
8 {
9     public Servidor servidor;
10    public UsuarioDB user;
11    [SerializeField] private SendEmail sendCorreo;
12    public TMP_InputField inpName;
13    public TMP_InputField inpUserName;
14    public TMP_InputField inpEmail;
15    public TMP_InputField inpPass;
16    public TMP_Text nombre;
17    public TMP_Text mensajeError;
18    public GameObject imgLoading;
19    public GameObject imgBienvenido;
20    public GameObject imgError;
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
```

Imagen 45.-Solicitando los componentes y verificando que no se tenga la sesión iniciada.

Fuente: Imagen capturada de software instalado y editada por el autor.

En la Imagen 45 se ilustra cómo se solicitan los elementos que utilizaremos para gestionar esta escena, desde los inputs donde se solicitaran los datos, hasta los paneles que mostraran las respuestas del servidor.

También contamos con el método Awake, que nos permitirá gestionar las sesiones en nuestra aplicación a través de los player prefabs. Gracias a este método, podremos visualizar y manejar los datos del usuario, que incluye el nombre de usuario y el correo. Para optimizar la

experiencia de usuario y evitar que la aplicación molesta al solicitar estos datos cada ocasión, es fundamental acceder a estos datos para no hacer tediosa la aplicación pidiendo los datos a los usuarios cuando los necesitemos.

En la Imagen 46 podemos observar la función básica de “RegistrarUser()”, la cual nos permite iniciar una corrutina, no sin antes verificar que los datos estén correctamente ingresados en los inputs correspondientes. Si alguno de estos campos se encuentra vacíos, se activa un panel de error que muestra un mensaje indicando que los campos no se completaron adecuadamente. En caso de que los inputs estén llenos, se llama a la función “RegistrarUsuario”, que ver en la Imagen 47. Esta función es la encargada de invocar la clase servidor para insertar los datos.

```
33 0 referencias
34 public void RegistrarUser()
35 {
36     if (inpName.text != "" & inpUserName.text != "" & inpEmail.text != "" & inpPass.text != "")
37     {
38         StartCoroutine(RegistrarUsuario());
39     }
40     else
41     {
42         imgError.SetActive(true);
43         mensajeError.text = "Tienes que llenar todos los datos para realizar el registro";
44     }
45 }
46 1 referencia
```

Imagen 46.-Función que permite registrar usuario y checa que los campos estén llenos.

Fuente: Imagen capturada de software instalado y editada por el autor.

```
47 1 referencia
48 IEnumerator RegistrarUsuario()
49 {
50     imgLoading.SetActive(true);
51     string[] datos = new string[4];
52     datos[0] = inpName.text;
53     datos[1] = inpUserName.text;
54     datos[2] = inpEmail.text;
55     datos[3] = inpPass.text;
56
57     StartCoroutine(servidor.ConsumirServicio("registrar", datos));
58     yield return new WaitForSeconds(1.0f);
59     yield return new WaitUntil(C => !servidor.ocupado);
60     imgLoading.SetActive(false);
61     if (servidor.resp.codigo == 201)
62     {
63         user = JsonUtility.FromJson<UsuarioDB>(servidor.resp.respuesta);
64         sendCorreo.EnviarCorreo(user.nombre, user.email);
65         PlayerPrefs.SetString("Email", user.email);
66         PlayerPrefs.SetString("UserName", user.user_name);
67         nombre.text = "Bienvenid@ " + user.user_name + "\n Ahora ya tienes un usuario creado...";
68         imgBienvenido.SetActive(true);
69     }else
70     {
71         mensajeError.text = "Ups!!! Algo salio mal, intentalo de nuevo";
72         imgError.SetActive(true);
73     }
74 }
```

Imagen 47.-Función que conectara con el script de las UnityWebRequest para insertar los datos.

Fuente: Imagen capturada de software instalado y editada por el autor.

Por último, en la Imagen 47 podemos observar el método que inserta los datos en la base de datos. Comenzamos activando el panel “Cargando”, el cual se mostrará mientras se recibe la respuesta del servidor. Posteriormente, crearemos el arreglo con los datos que serán insertados y llamaremos otra corrutina que será la encargada de realizar la UnityWebRequest. Esperamos un segundo y asignamos el valor true a la variable del servidor que indica que en ese momento se encuentra ocupado, para evitar problemas con esos datos. A continuación, mediante la respuesta que nos enviara el servidor, verificaremos si la inserción de los datos fue correcta o incorrecta. Si el código que se recibe es: “201”, significa que la inserción se llevó a cabo correctamente y procederemos a ejecutar lo siguiente:

- Formatea los datos del servidor en formato JSON para facilitar su manejo.
- Invocamos la función de “EnviarCorreo” la cual detallaremos en la sección 4.6.1
- Creamos dos llaves con los playerprefs que contendrán los datos del usuario, lo que nos permitirá un acceso más fácil a esta información.
- Enviamos el mensaje que se mostrara en el panel Correcto.
- Activamos el panel correcto.

Por otro lado, si se Envío un código diferente al “201”, significa que ha ocurrido un error. En ese caso, se enviará un mensaje de error al panel correspondiente, que se activará de inmediato. Así podremos reintentar el registro, teniendo en cuenta el error cometido. Para finalizar, en la Imagen 48 podrás observar la interfaz de esta escena.



Imagen 48.-Interfaz de Registro de usuario.

Fuente: Imagen capturada de software instalado y editada por el autor.

Como información adicional, en la *Imagen 49* se presentan los códigos que aparecerán en todos nuestros archivos PHP. Esta referencia se mencionará únicamente en esta ocasión, ya que no deseamos repetirla para evitar el aburrimiento del lector. Nuestro objetivo es que la lectura sea amena, por lo que incluimos esta información a modo de comentario único, dado que estos códigos fueron creados por nosotros mismos con el fin de tener una forma más rápida de lograr entender que es lo que sucede al momento en el que está respondiéndonos el servidor.

```

1  Codigos php para nuestra aplicacion:
2
3  Codigos Exitosos:
4  codigo: 201, mensaje: "Usuario creado correctamente", respuesta:"Nombre de usuario"
5  codigo: 202, mensaje: "Inicio de sesion exitoso", respuesta : "Hola user_name bienvenid@"
6  codigo: 203, mensaje: "Correo electronico actualizado correctamente", respuesta:"nuevo Email"
7  codigo: 204, mensaje: "Nombre de usuario actualizado correctamente", respuesta:"Nuevo nombre de usuario"
8  codigo: 205, mensaje: "La operacion se realizo exitosamente", respuesta:"Json con datos extraidos"
9  codigo: 206, mensaje: "Contraseña cambiada correctamente", respuesta:"".$row['user_name']."
10 codigo: 207, mensaje: "Se actualizaron los puntos correctamente", respuesta:"".$varNpuntos.""
11 codigo: 208, mensaje: "Se envio el correo correctamente", respuesta:"El correo se envio correctamente"
12
13 Codigos de error:
14 codigo: 400, mensaje: "Ya existe un usuario registrado con ese correo", respuesta:"Usuario existente"
15 codigo: 401, mensaje: "Error al conectar con el servidor", respuesta:"$e"
16 codigo: 402, mensaje: "Error al intentar crear el usuario", respuesta: "No se puede crear el usuario"
17 codigo: 403, mensaje: "Faltan datos para realizar la operacion", respuesta: "Datos incompletos"
18 codigo: 405, mensaje: "Inicio de sesion incorrecto", respuesta:"El usuario o la contraseña son incorrectos"
19 codigo: 406, mensaje: "Actualizacion incorrecta", respuesta:"El usuario o la contraseña son incorrectos"
20 codigo: 407, mensaje: "No se pudo realizar la operacion", respuesta: "Nombre de usuario o correo incorrectos"
21 codigo: 408, mensaje: "No se pudo enviar el correo, algo salio mal", respuesta: ".{$_mail->ErrorInfo}.";
22 codigo: 409, mensaje: "No se pudo realizar la operacion", respuesta: "El correo del negocio es incorrecto";
23 codigo: 410, mensaje: "No se pudo realizar la operacion", respuesta: "Los puntos que tiene el usuario son
insuficientes";

```

Imagen 49.-Códigos de éxito y error de todos los archivos PHP.

Fuente: Imagen capturada de software instalado y editada por el autor.

4.7.1.- Envío de correo electrónico.

En esta sección, presentaremos de manera detallada el archivo PHP, ya que el correo se envía desde Unity a través de una UnityWebRequest. La herramienta que utilizamos para el envío de correos es “PHPMailer”. Para poder descárgala lo que tenemos que hacer es dirigirnos a la siguiente dirección:

* <https://github.com/PHPMailer/PHPMailer>

Posteriormente, descargaremos la carpeta comprimida en formato zip del proyecto para extraer tres archivos que se encuentran en una subcarpeta llamada “src”. Los archivos contenidos en dicha carpeta son:

- Exception
- PHPMailer
- SMTP

Colocamos estos archivos los en la carpeta que nombraremos “PHPMailer”. Dicha carpeta la subiremos al servidor para que la herramienta que vamos a utilizar funcione correctamente. Posteriormente crearemos el archivo con el que realizaremos el envío de correos. Esto lo hacemos siguiendo el procedimiento que se muestra en la Imagen 50. Primero, debemos referenciar los archivos con la instrucción “use” que se encuentran en las líneas 4, 5 y 6 del archivo, las cuales podrás observar en la imagen mencionada. Además, es necesario indicar los archivos que deben ser requeridos con la instrucción “require” que se encuentran en las líneas 8, 9 y 10 de la imagen previamente referenciada. También hemos decidido incluir un archivo llamado “CuerpoCorreo.PHP” el cual contiene las contraseñas y correo que utilizaremos para el envío de información, que, por seguridad, no deseamos divulgar. Sin embargo, podrás visualizarlo en la imagen 45.

```
1 <?php
2 include 'CuerpoCorreo.php';
3
4 use PHPMailer\PHPMailer\PHPMailer;
5 use PHPMailer\PHPMailer\SMTP;
6 use PHPMailer\PHPMailer\Exception;
7
8 require 'PHPMailer/Exception.php';
9 require 'PHPMailer/PHPMailer.php';
10 require 'PHPMailer/SMTP.php';
11
12 $mail = new PHPMailer(true);
13
14 try {
15
16     $nombreComp = $_POST['nombre'];
17     $email = $_POST['email'];
18
19     //Server settings
20     $mail->SMTPDebug = 0; //Activar la salida de depuración detallada
21     $mail->isSMTP(); //Enviar usando SMTP
22     $mail->Host = $host; //Configura el servidor SMTP a través del cual se enviará el mensaje
23     $mail->SMTPAuth = true; //Activar la autenticación SMTP
24     $mail->Username = $emailfrom; //Nombre de usuario SMTP
25     $mail->Password = $passEmail; //Contraseña SMTP
26     $mail->SMTPSecure = 'ssl'; //Habilitar cifrado TLS implícito
27     $mail->Port = 465; //Puerto TCP al que conectarse; utilice 587 si ha configurado 'SMTPSecure = PHPMailer::ENCRYPTION_STARTTLS'.
28
29     //Recipients
30     $mail->setFrom('sartres96@mexilab.com', 'Puebla Travel AR');
31     $mail->addAddress($email, $nombreComp);
32     //Imágenes anexas
33     // $mail->addAttachment('resources/1.png');
34
35     //Content
36     $mail->isHTML(true); //Set email format to HTML
37     $mail->Subject = 'Salvador Arturo Ramirez Escamilla';
38     $mail->Body = $BodyEmail;
39     $mail->send();
40
41     echo '{"codigo": 208,"mensaje":"Se envió el correo correctamente","respuesta":"El correo se envió correctamente"}';
42
43
44 } catch (Exception $e) {
45     echo '{"codigo": 408,"mensaje":"Algo salió mal, no se pudo enviar el correo","respuesta":"'.$mail->ErrorInfo.'"}';
46 }
```

Imagen 50.- Archivo PHP para enviar correos desde el servidor.

Fuente: Imagen capturada de software instalado y editada por el autor.

```
1 <?php
2 $host = 'mexilab.com';
3 $emailFrom = 'sartres96@mexilab.com';
4 $passEmail = 'XXXXXXXXXX';
5
6 $BodyEmail = '<p>Por medio de la presente queremos agradecerte por apoyar es
participación es muy importante para el crecimiento del proyecto, ya que
si el proyecto ve la luz en un futuro no lejano, serás tomad@ en cuenta
manera te aclaro<br><br>De igual manera te aclaro que tus datos quedara
tampoco serán vendidos o te llegara spam.<br>Sin más por el momento, me
7 ?>
```

Imagen 51.-CuerpoCorreo.PHP, que contiene elementos para el envío de correo.

Fuente: Imagen capturada de software instalado y editada por el autor.

Una vez que referenciamos los archivos, comenzaremos a proporcionar los datos necesarios para el envío del correo. Como se observa en la Imagen 50, en las líneas 16 y 17 solicitamos el nombre del destinatario y la dirección de correo electrónico a la que enviaremos el mensaje. A continuación, debemos completar las variables, tal como se muestra en la Imagen 50. Los datos se encuentran directamente en el servidor que esté utilizando, incluyendo información como el host y el puerto por el cual se enviara el correo. También es necesario agregar la dirección de correo electrónico que del remitente y la contraseña correspondiente. Estos datos, en nuestro caso, están contenidos en el archivo que se muestra en Imagen 51.

Luego, comenzaremos a señalar el remitente y el encabezado del correo electrónico, como se observa en la línea 30. Luego, proporcionamos el nombre y la dirección de correo electrónico de la persona destinataria. Por último, debemos incluir el cuerpo del correo, como se muestra en las líneas 36, 37 y 38 de la Imagen 50, junto con el “Subject” que corresponde al remitente del correo, y el “Body” que es el contenido del correo que admite etiquetas HTML. Finalmente, escribimos “send()” que es la función que realizara el envío del correo. De este modo. podemos enviar los correos electrónicos que deseemos desde nuestro servidor. Si surge alguna duda, el mismo repositorio contiene diferentes ejemplos con instrucciones sencillas y fáciles de seguir.

Por último, mencionamos las funcionalidades de los dos botones que contiene el panel de registro, que se muestra en la Imagen 48. Uno de los botones es el de “Aceptar” que al presionarlo ejecutará el método que registra al usuario. El otro botón es un “toggle”. Dicho componente es con el que nos apoyaremos para verificar si el usuario ya es propietario de una cuenta creada. Si seleccionamos esta opción, se activa un botón que nos traslada a la escena de inicio de sesión, la cual explicaremos en la sección 4.9. Ya que en la sección 4.8, nos enfocaremos en detallar como se integra la base de datos en Unity, utilizando la escena de registro como ejemplo. Posteriormente, solo detallaremos los elementos de la escena y aquellos que requieran datos de la base de datos, y explicaremos a grandes rasgos los archivos PHP que se utilizan para las consultas (dependiendo la funcionalidad que estemos explicando).

4.8.- Integración de bases de datos.

Como ya se mencionó anteriormente, el control de la base de datos en Unity se nos facilita considerablemente a través de las “Unity Web Reques” lo cual es un sistema modular que nos permite realizar peticiones http y gestionar las respuestas que estas generan.

El sistema de UnityWebRequest es compatible con la mayoría de las plataformas de Unity, con la excepción del Unity Web Player. Android y iOS desde soportan las UnityWebRequest desde su versión 5.3.

4.8.1 Arquitectura.

El ecosistema descompone una transacción HTTP en tres operaciones distintas:

- Proporcionando datos al servidor.

- Recibiendo datos del servidor.
- Un flujo de control HTTP.

Creando un flujo similar al que muestra en la Imagen 52, el servidor responde a la aplicación a través de UnityWebRequest, lo que nos permite gestionar la información. Una vez que tengamos una comprensión teórica de cómo funciona la UnityWebRequest, procederemos a explicarte como se lleva a cabo el proceso.

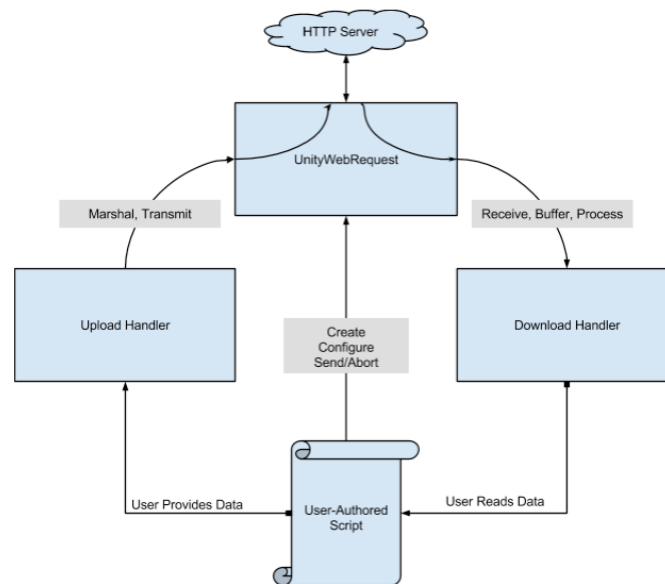


Imagen 52.- Arquitectura de las UnityWebReques.

Fuente: [24].

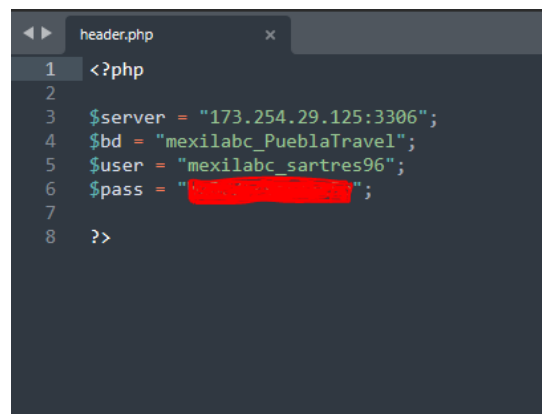
4.8.2 Archivos PHP

Primero, lo que debemos hacer es crear nuestro archivo PHP para generar las consultas y procedimientos que deseamos manejar, ya sea insertar, actualizar o eliminar datos, entre otros. En este ejemplo, lo que haremos es crear el registro de nuestra aplicación. Ten en cuenta que, en la mayoría de las operaciones, tendrás que aplicar tu lógica; básicamente lo que deberás modificar será la consulta. Si deseas realizar tareas más elaboradas o complejas, puedes partir de esta información, pero tendrás que investigar más a fondo, ya que lo que se realizó en este proyecto fue únicamente un CRUD. Si comienzas con este ejemplo, todo será más fácil.

En nuestro caso, crearemos la conexión utilizando “mysqli_connect”, incluyendo las variables en un archivo llamado “header.PHP”, que se muestra en la Imagen 53 (el cual contiene

los datos para realizar la conexión, en el cual ocultamos la contraseña por seguridad de la base de datos), posteriormente, lo incorporaremos al archivo donde realizaremos la consulta. La razón para manejar ese archivo es evitar instrucciones repetitivas; es decir, debemos establecer y cerrar la conexión cada vez que deseamos realizar una consulta. Por lo tanto, ingresar los mismos datos una y otra vez se convierte en una tarea tediosa, es por eso por lo que creamos dicho archivo, para ahorrarnos esas instrucciones.

Una vez que hayamos creados este archivo, lo invocamos en el archivo donde elaboraremos nuestra consulta, el cual podrás observar en la Imagen 54. En dicho archivo, lo primero que hacemos es solicitar los datos mediante el método `$_POST` de PHP, utilizando la instrucción `isset($_POST['dato que solicitaremos'])`. De esta manera, nos aseguraremos de que el dato sea enviado a través de la UnityWebRequest, ya que la función devolverá `false` si una variable ha sido definida o enviada como `null`.

A screenshot of a code editor window titled 'header.php'. The code is as follows:

```
1 <?php
2
3 $server = "173.254.29.125:3306";
4 $bd = "mexilabc_PueblaTravel";
5 $user = "mexilabc_santres96";
6 $pass = "XXXXXXXXXX";
7
8 ?>
```

Imagen 53.- Archivo "header.PHP"

Fuente: Imagen capturada de software instalado y editada por el autor.

```

registratrecaxtravelp.php
1 <?php
2 include 'header.php';
3
4 try {
5     $conn = mysql_connect($server, $user, $pass, $db);
6
7     if (!$conn) {
8         echo '{"codigo": 400, "mensaje": "Error al conectar con el servidor", "respuesta": "No se pudo realizar la conexión"}';
9     } else {
10
11         if (isset($_POST['nombre']) && isset($_POST['user_name']) && isset($_POST['email'])
12             && isset($_POST['pass'])) {
13
14             $varName = $_POST['nombre'];
15             $svarUserName = $_POST['user_name'];
16             $varemail = $_POST['email'];
17             $varPass = hash("sha256", $_POST['pass']); // hash("sha256", $_POST['pass']);
18             $sql = "SELECT * FROM usuarios2 WHERE user_name = '$svarUserName' AND email = '$varemail'";
19             $resultado = $conn->query($sql);
20
21             if (($resultado->num_rows) > 0) {
22
23                 echo '{"codigo": 400, "mensaje": "Ya existe un usuario registrado con ese correo", "respuesta": "Usuario existente"}';
24             } else {
25
26                 $sql = "INSERT INTO usuarios2 ('id_usuario', 'nombre', 'user_name', 'email', 'password', 'puntos_obtenidos') VALUES (NULL, '$varName', '$svarUserName', '$varemail', '$varPass', '0')";
27
28                 if ($conn->query($sql) === TRUE) {
29                     $sql = "SELECT * FROM usuarios2 WHERE nombre = '$varName'";
30                     $resultado = $conn->query($sql);
31                     $jsonRespuesta = "";
32
33                     if ($row = $resultado->fetch_assoc()) {
34                         $jsonRespuesta = ["id_usuario": $row['id_usuario'],
35                             "nombre": $row['nombre'],
36                             "user_name": $row['user_name'],
37                             "email": $row['email'],
38                             "puntos_obtenidos": $row['puntos_obtenidos']];
39                     } else {
40                         echo '{"codigo": 201, "mensaje": "Usuario creado correctamente", "respuesta": "$jsonRespuesta"}';
41                     }
42                 } else {
43                     echo '{"codigo": 401, "mensaje": "Error al intentar crear el usuario", "respuesta": "No se puede crear el usuario"}';
44                 }
45             }
46         }
47     }
48 }

```

Imagen 54.- Archivo PHP en el cual generaremos la consulta para insertar los datos.

Fuente: Imagen capturada de software instalado y editada por el autor.

Una vez que se solicitaron los datos, verificaremos mediante un “if” que los datos no vengan nulos, si la condición se cumple comenzaremos a asignar los datos en variables, utilizando el método “sha256” al asignar la contraseña para que esta se encripte y podamos proteger la información del usuario. Posteriormente asignaremos a una variable la consulta que se muestra en la Imagen 55, en esta consulta, asignaremos los valores que se recibieron a través del método \$_POST. Primero, realizaremos un “SELECT” para comprobar que el correo electrónico y el nombre de usuario recibidos no existan en la base de datos, con el fin de evitar que los datos se dupliquen.

```

$sql = "SELECT * FROM `usuarios2` WHERE `user_name` = '$varUserName' AND `email` = '$varemail.'";

```

Imagen 55.-Consulta “SELECT” que nos ayuda a ver si los datos ingresados ya están en la base de datos.

Fuente: Imagen capturada de software instalado y editada por el autor.

Si el usuario ya existe, se enviará un mensaje indicando que los datos ya están registrados en la base de datos. Si los datos no son duplicados, procederemos a crear y ejecutar un comando

“INSERT” para registrar la información la base de datos. Esta consulta se muestra en la Imagen 56.

```
$sql = " INSERT INTO `usuarios2`(`id_usuario`, `nombre`, `user_name`, `email`, `password`, `
puntos_obtenidos`) VALUES (NULL, '$varName.', '$varUserName.', '$varemail.', '$varPass.',
'');";
```

Imagen 56.-Consulta “INSERT” la cual nos ayuda a registrar los datos en la base de datos.

Fuente: Imagen capturada de software instalado y editada por el autor.

Una vez que se ejecutó la consulta, verificaremos si los datos se insertaron correctamente mediante un “if”; si los datos se agregaron de manera exitosa, realizaremos nuevamente un “SELECT” para recuperar los datos que acabamos de crear y generaremos una respuesta en formato “JSON” para enviar esos datos a Unity por medio de la UnityWebRequest.

En lo que respecta a los archivos PHP, es importante tener en cuenta cada operación requerirá un archivo diferente. Dependiendo de lo que quieras lograr en la base de datos, la lógica que apliques en cada archivo variara. A través de las UnityWebRequest, lo que obtendrás en Unity será el resultado de la información que envíes en cada archivo. Por lo tanto, el resultado que desees reflejar dependerá de las decisiones que tomes a la hora de escribir la consulta.

4.8.3 Scripts para la ejecución desde Unity.

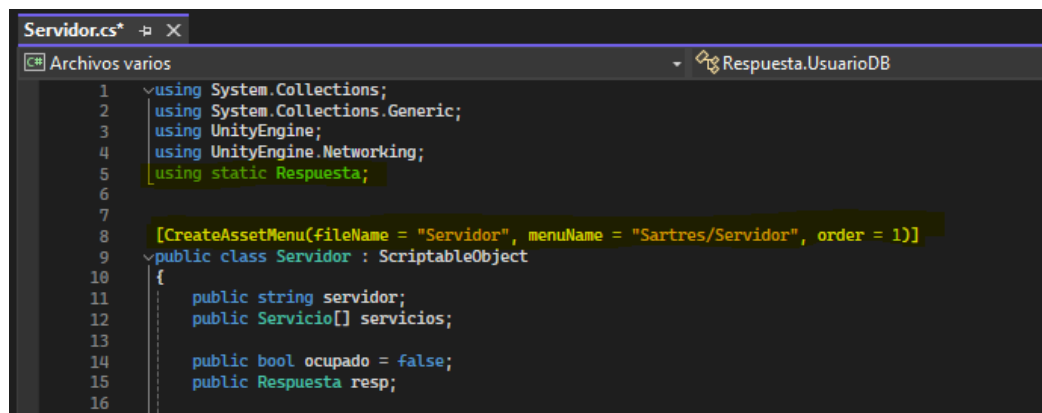
En cuanto a Unity, lo primero que debemos hacer es crear un scrip, al que llamaremos “Servidor.cs”, en el cual controlaremos los servicios que se ejecuten y las solicitudes de UnityWebRequest. Primero, declararemos las variables que utilizaremos, las cuales se mostraran en la Imagen 57. Además, dentro del mismo script, crearemos dos clases: “Servicio” y “Respuesta”, cada una de las cuales contendrá lo siguiente.

Servicio: contiene dos variables y un arreglo de tipo string, las cuales son: nombre, url y el arreglo contendrá los parámetros necesarios para llevar a cabo la UnityWebRequest. Dichos parámetros dependerán de tu archivo PHP, ya que son los que se solicitan en ese archivo, como se muestra en la Imagen 58.

Respuesta: Esta clase dependerá en gran medida del desarrollador, ya que él puede estructurarla y modificarla según las necesidades de la base de datos. En nuestro caso, manejamos varias respuestas, por lo que contiene diferentes variables e incluso una clase adicional dentro de la misma. Inicialmente trabajaremos con tres variables: una de tipo int y dos tipos string. En la

primera, recibiremos un código, al cual podremos asignarle un significado; en la segunda; un mensaje que decidamos asignar; y en la tercera, tendremos la respuesta, que también será asignada por nosotros. En esta clase, creamos un constructor que nos permitirá cargar un código y un mensaje si algo falla, evitando que estos datos queden vacíos. Además, contamos con un método que limpia la respuesta, ya que, para que llegue como JSON a Unity, tenemos una estructura en nuestro PHP, como se muestra en la Imagen 59.

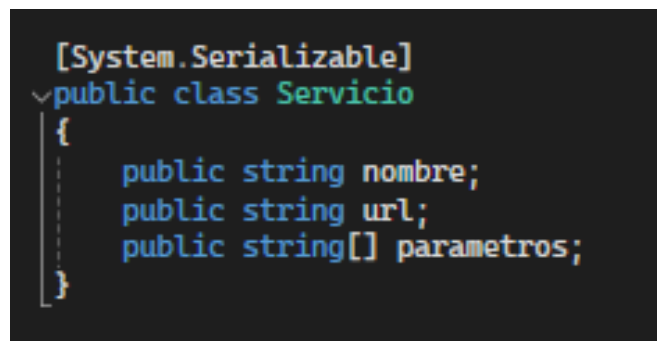
También hemos creado, dentro de la clase Respuesta, una clase UsuarioDB, que nos permitirá organizar los datos de usuarios que utilizamos en algunos servicios para la gestión de nuestra aplicación, los cuales son fundamentales para su funcionamiento.



```
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4 using UnityEngine.Networking;
5 using static Respuesta;
6
7
8 [CreateAssetMenu(fileName = "Servidor", menuName = "Sartres/Servidor", order = 1)]
9 public class Servidor : ScriptableObject
10 {
11     public string servidor;
12     public Servicio[] servicios;
13
14     public bool ocupado = false;
15     public Respuesta resp;
16
```

Imagen 57.- Variables y librerías de la clase y el archivo “Servidor”.

Fuente: Imagen capturada de software instalado y editada por el autor.



```
[System.Serializable]
public class Servicio
{
    public string nombre;
    public string url;
    public string[] parametros;
}
```

Imagen 58.- Variables de la clase “Servicio”.

Fuente: Imagen capturada de software instalado y editada por el autor.

```

[System.Serializable]
public class Respuesta
{
    public int codigo;
    public string mensaje;
    public string respuesta;

    public void LimpiarRespuesta()
    {
        respuesta = respuesta.Replace('#', '');
    }

    public Respuesta()
    {
        codigo = 404;
        mensaje = "Error";
    }

[System.Serializable]
public class UsuarioDB
{
    public int id;
    public string nombre;
    public string user_name;
    public string email;
    public int puntos_obtenidos;
}
}

```

Imagen 59.- Variables de la clase “Respuesta”.

Fuente: Imagen capturada de software instalado y editada por el autor.

Si prestas atención a la Imagen 57, la línea 8 contiene una instrucción que esta resaltada. Esta instrucción marca un tipo derivado de ScriptableObject que nos permite crear instancias que pueden generarse y almacenarse fácilmente en el proyecto como archivos “.asset” y más adelante mostraremos como sacarle provecho a esta instrucción.

Una vez que hayamos configurado lo anterior, comenzaremos a crear una corrutina para ejecutar la UnityWebRequest. Para ello declaramos la función, pero en lugar de especificar un tipo, utilizamos la palabra reservada “IEnumerator” (como se muestra en la Imagen 60) y le asignamos un nombre a la función, que en este caso será “ConsumirServicio”. Esta función recibira dos parámetros: una variable de tipo string y un arreglo de strings, que corresponden al nombre del servicio y el arreglo de los datos que vamos a enviar para realizar la consulta.

Lo primero que haremos en el método es establecer la variable “ocupado” en false, para evitar conflictos en caso de que se llame al método dos veces, ya que de ser así se podría generar un conflicto. Posteriormente inicializamos la variable de la clase “Respuesta” con un mensaje y código que serán los valores por defecto. Luego, creamos el “WWWForm” nombrando la variable como “formulario” y posteriormente inicializamos la variable de la clase “Servicio” nombrando la variable “s” y en un “for” iremos llenando la variable con el arreglo que se pide en el método, tal como se muestra en la Imagen 60 (el cual es básicamente el que hace la UnityWebRequest dependiendo del servicio que se solicite).

Posteriormente, mediante otro “for” añadiremos los parámetros a nuestra variable “formulario” para llevar a cabo la consulta en el archivo “PHP” que creamos anteriormente. Luego ejecutamos la `UnityWebRequest` utilizando la instrucción que se encuentra en la línea 38, la cual puedes ver en la Imagen 60. Posteriormente lanzamos un “yield return [www.SendWebRequest\(\)](#)” que nos permitirá enlazarnos con el archivo “PHP”, como no sabemos cuánto tiempo tardará en respondernos el servidor, es por eso que utilizamos la instrucción “yield” [33].

Una vez que haya recuperado el control, colocaremos un “if else” esto para tomar en cuenta dos casos, si ocurrió algún error, el objeto respuesta se cargara con el código 404 y en el componente mensaje se cargara la palabra “Error”. Si se realizó correctamente la operación, asignaremos la respuesta a la variable “resp” de la siguiente manera:

```
“resp = JSONUtility.DeJSON<Respuesta>(www.doynloadHandker.text)”
```

Que básicamente convierte la respuesta que nos proporciona el archivo PHP a JSON para manipularla en nuestra aplicación. Si observas detenidamente la Imagen 60, notarás que se ejecuta un método llamado “LimpiarRespuesta” que básicamente ejecuta la instrucción “Replace” al texto que llega para eliminar los símbolos que nos sirven como identificadores en el texto, para identificar en donde se encuentra cada dato que llega a la `UnityWebRequest`.

Una vez que terminemos este archivo, procederemos a tratar la solicitud y la respuesta, como podrás observar en la Imagen 59. En dicho archivo, recopilaremos los datos de unos “InputField” que utilizaremos en la interfaz de “Registrar Usuario” y los agregaremos a un arreglo de strings, el cual contendrá los datos necesarios para realizar la inserción en la base de datos.

Posteriormente, ejecutaremos una corrutina para invocar la función del servidor, enviando como parámetros el nombre que asignamos al servicio y los datos que recopilamos de la interfaz. Ejecutaremos dos veces la instrucción “yield”. En la primera instrucción esperamos un segundo y en la segunda nos esperaremos hasta que el servidor se desocupe, si otra clase utiliza el servicio.

```

17
18 public IEnumerator ConsumirServicio(string nombre, string[] datos)
19 {
20     ocupado = true;
21     resp = new Respuesta();
22     WwwForm formulario = new WwwForm();
23     Servicio s = new Servicio();
24     for (int i = 0; i < servicios.Length; i++)
25     {
26         if (servicios[i].nombre.Equals(nombre))
27         {
28             s = servicios[i];
29         }
30     }
31
32     for (int i = 0; i < s.parametros.Length; i++)
33     {
34         formulario.AddField(s.parametros[i], datos[i]);
35     }
36
37     UnityWebRequest www = UnityWebRequest.Post(servidor + s.url, formulario);
38
39     Debug.Log(servidor + s.url);
40     // Debug.Log("Resp 1= " + www.result + " Respuesta 2 = " + UnityWebRequest.Result.Success);
41
42     yield return www.SendWebRequest();
43
44     //Debug.Log("Resp 1= "+www.result+" Respuesta 2 = "+ UnityWebRequest.Result.Success );
45
46     if (www.result != UnityWebRequest.Result.Success)
47     {
48         //Debug.Log("ENTRE AL IF");
49         resp = new Respuesta();
50     }
51     else
52     {
53         Debug.Log("Estoy en el else");
54         Debug.Log(www.downloadHandler.text);
55         resp = JsonUtility.FromJson<Respuesta>(www.downloadHandler.text);
56         //user = JsonUtility.FromJson<DatosUsuario>(www.downloadHandler.text);
57         resp.LimpiarRespuesta();
58         // Debug.Log(resp);
59         // respuestaCompleta.respuesta = resp.codigo;
60     }
61
62     ocupado = false;
63 }
64

```

Imagen 60.- Método "ConsumirServicio"

Fuente: Imagen capturada de software instalado y editada por el autor.

```

36 public void RegistrarUser()
37 {
38     if (inpName.text != "" & inpUserName.text != "" & inpEmail.text != "" & inpPass.text != "")
39     {
40         StartCoroutine(RegistrarUsuario());
41     }
42     else
43     {
44         imgError.SetActive(true);
45         mensajeError.text = "Tienes que llenar todos los datos para realizar el registro";
46     }
47 }
48
49 IEnumerator RegistrarUsuario()
50 {
51     imgLoading.SetActive(true);
52     string[] datos = new string[4];
53     datos[0] = inpName.text;
54     datos[1] = inpUserName.text;
55     datos[2] = inpEmail.text;
56     datos[3] = inpPass.text;
57
58     StartCoroutine(servidor.ConsumirServicio("registrar", datos));
59     yield return new WaitForSeconds(1.0f);
60     yield return new WaitUntil(() => !servidor.ocupado);
61     imgLoading.SetActive(false);
62     if (servidor.resp.codigo == 201)
63     {
64         {
65             user = JsonUtility.FromJson<UsuarioDB>(servidor.resp.respuesta);
66             sendCorreo.EnviaCorreo(user.nombre, user.email);
67             PlayerPrefs.SetString("Email", user.email);
68             PlayerPrefs.SetString("UserName", user.user_name);
69             nombre.text = "Bienvenid@ " + user.user_name + "\n Ahora ya tienes un usuario creado...";
70             imgBienvenido.SetActive(true);
71         }
72     }
73     else
74     {
75         mensajeError.text = "Ups!!! Algo salio mal, intentalo de nuevo";
76         imgError.SetActive(true);
77     }
78 }

```

Imagen 61.-Archivo "RegistUser.cs" el que contiene y utiliza el archivo "Servidor.cs".

Fuente: Imagen capturada de software instalado y editada por el autor.

Cabe mencionar que, al iniciar la función, se activa un panel que indica en la interfaz que la función ha comenzado. Posteriormente, cuando el servicio regresa la respuesta correspondiente, la imagen se desactiva y nos encontramos con un “if”.

Este “if” es el encargado de gestionar las respuestas del usuario, dependiendo del código que nos devuelva el servicio. En el caso del registro de usuario, si el código es 201, consideraremos que el registro fue exitoso, lo que significa que los datos se agregaron correctamente. En este caso mostraremos un mensaje que nos proporciona el servicio en un panel verde, indicando que el procedimiento fue exitoso. Además, mediante un botón se enviará al usuario a la escena de “Menú” para que pueda empezar a utilizar la aplicación sin problemas.

Así mismo utilizaremos la función “PlayerPrefs.SetString(key,string)”, cuya finalidad es guardar los datos de manera persistente para poder acceder a ellos desde cualquier escena en nuestra aplicación. En la Imagen 61 podrás observar cómo se utiliza la instrucción mencionada, ya que la aplicamos en las líneas 67 y 68. Esta acción se realiza para evitar que el usuario tenga que iniciar sesión cada que abra la aplicación; de este modo, la sesión permanecerá activa siempre y cuando él usuario no decida cerrarla. Para ello, se ha añadido un método que eliminara esos datos, el cual explicaremos detalladamente más adelante.

Si el código que el servicio nos devuelve no es 201, se mostrará un panel de color rojo con un mensaje de error que nos solicitará intentar nuevamente. Este panel incluye un botón que desactiva el panel y limpia los campos previamente completados, permitiendo al usuario intentar el registro de nuevo.

Además de eso, se incluye un “if” que comienza en la línea 38 y finaliza en la línea 46, la cual puedes observar en la Imagen 61. Esta condición verifica que los datos no estén vacíos, para evitar datos en blanco en la base de datos al crear el usuario. Si los campos están vacíos, se activa el panel de error mencionado anteriormente, que muestra el mensaje: “Tienes que llenar todos los datos para realizar el registro”, junto con un botón que ejecuta la misma función antes mencionada, es decir que borra los datos que ingresados en los campos requeridos para el registro.

Por último, en la línea 66 se encuentra la siguiente instrucción: “sendCorreo.EnvíarCorreo(user. nombre,user. email)” la cual es una función que nos permite enviar un correo al usuario que se acaba de registrar. Más adelante, dedicaremos una sección de este capítulo a detallar como realizamos el envío de dicho correo. Para ejecutarla desde Unity, lo

hacemos de la misma manera que los servicios descritos en esta sección, ya que este servicio lo agregaremos como uno más al GameObject servidor que creamos en la aplicación.

4.8.4 Agregando el GameObject y trabajando en la interfaz.

Como mencionamos anteriormente y como puedes observar en la Imagen 57, en la cual se ha resaltado una instrucción ubicada en la línea 8, esta instrucción nos permite crear un elemento “.asset” desde un menú, que básicamente corresponde a la ruta que asignamos en dicha instrucción. Para agregarlo, dirígete al apartado “Assets” en el menú que se encuentra en la parte superior. Luego, selecciona la opción “create” y aparecerá un apartado; en nuestro caso, este apartado aparece como “Sartres”, que es el que nombre que se asignado en la instrucción antes mencionada. Una vez que seleccionamos esa opción, se nos mostrara el componente “Servidor” y, al hacer clic en él, podremos crear el componente. La ruta precisa la podrás ver en la Imagen 62. Cabe mencionar que en la instrucción puedes seleccionar las opciones que desees.

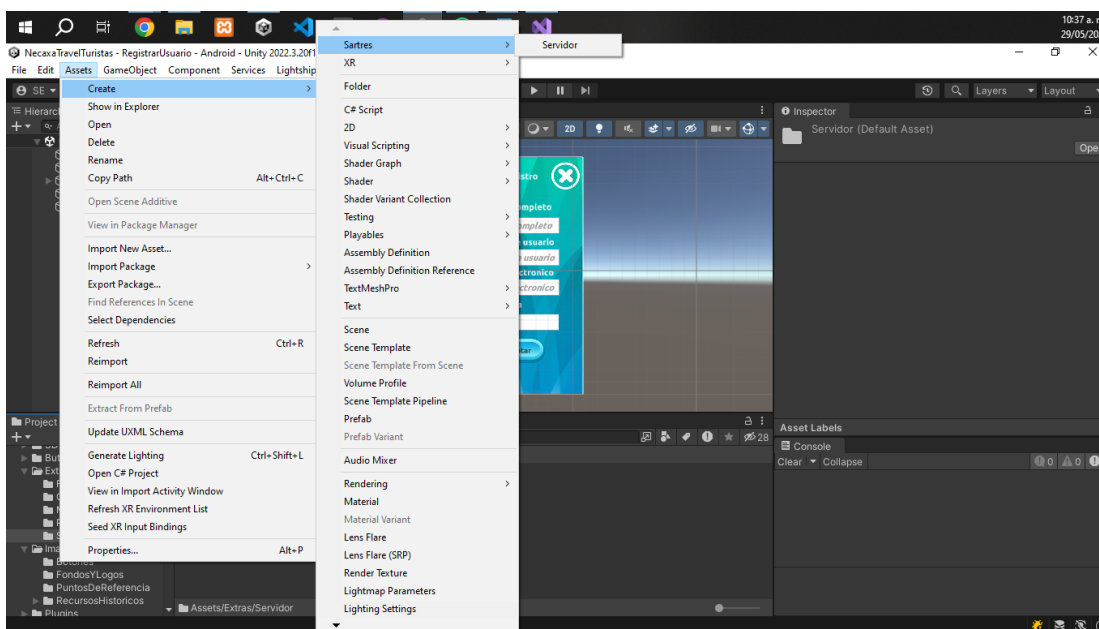


Imagen 62.-Ruta para la creación del componente “. asset” para nuestros servicios.

Fuente: Imagen capturada de software instalado y editada por el autor.

Con este elemento, ya podemos comenzar a asignar los servicios que vamos a crear. Como podrás observar en la Imagen 63, los parámetros que agregamos son la ruta en la que se encuentran los archivos “PHP”. En la imagen, puedes ver que la ruta es “<https://mexilab.com/sartresfolder/>”, que corresponde al servidor en donde tenemos los archivos y también la base de datos, lo que nos permite acceder a ella desde cualquier dispositivo con conexión a internet. Además, podrás notar

que se ha resaltado un servicio y los componentes de dicho servicio. Primero, tenemos el nombre con el que identificaremos el servicio; luego, encontraremos el nombre y la extensión del archivo que vamos a ejecutar y, por último, tenemos los parámetros con los que trabajará el archivo.

Una vez que hayamos creado este componente, podremos ejecutar los servicios de diferente manera. Esto se debe a que solo referenciaremos todos los datos y archivos desde dicho componente, lo que nos facilitará enormemente el trabajo. Con cada archivo y cada clase, nos enfocaremos en cómo obtener los datos y en cómo visualizar la información que enviaremos a la petición.

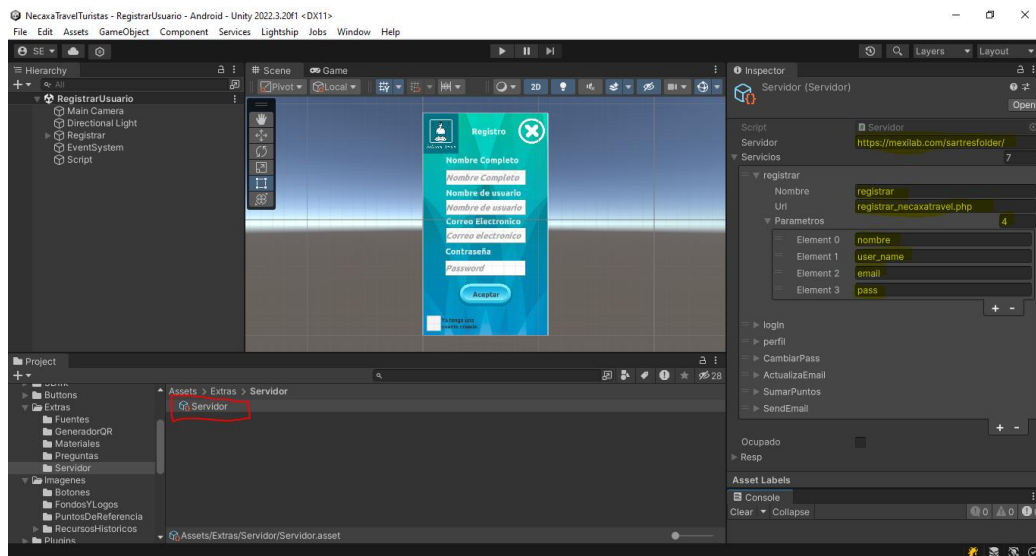


Imagen 63.-Componente “Servidor.asset” con el cual controlaremos los servicios que iremos ejecutando.

Fuente: Imagen capturada de software instalado y editada por el autor.

Si observas la Imagen 64, hemos destacado los elementos con los que trabajará nuestro servicio de “Registrar usuario”, los cuales referenciamos al GameObject que hemos denominado “Script” y esto se logra simplemente arrastrando dichos componentes al rectángulo que aparece después de la etiqueta mostrada.

También se presentan los paneles mencionados, que se muestran según los códigos que devuelve la respuesta del servicio. Asimismo, puedes observar la interfaz que hemos diseñado para el registro, a través de la cual ejecutaremos las funciones y mostraremos la información mediante los scripts. Contamos también con una serie de componentes que podremos utilizar referenciándolos a través del GameObject, lo que nos permite agregar botones y ejecutar las funciones que necesarias. Cuando completemos estas referencias,

podremos utilizar el registro de usuarios. Como se mencionó anteriormente, los demás servicios funcionan de manera similar; lo que varía son los datos que enviamos y las respuestas que recibimos, así como la interfaz.

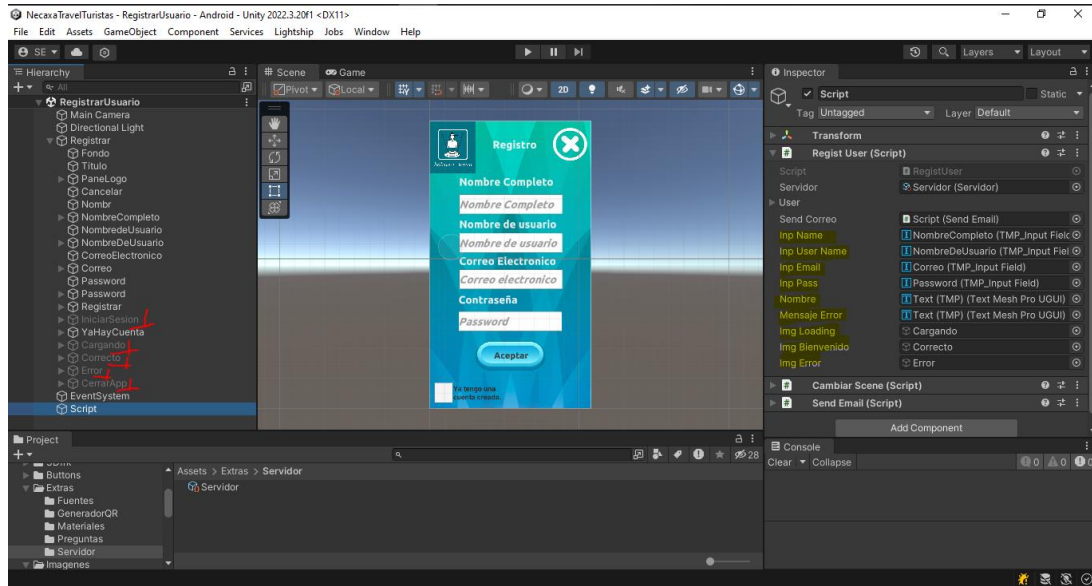


Imagen 64.-Interfaz de registro de usuario, con los componentes señalados y referenciados.

Fuente: Imagen capturada de software instalado y editada por el autor.

4.9.- Iniciar sesión.

En esta sección, te explicaremos cómo se lleva a cabo el inicio de sesión de la aplicación. Esta escena cuenta, al igual que la escena de registro, con un formulario; sin embargo, la diferencia radica en que solo se requieren tres datos: nombre de usuario, correo electrónico y contraseña, tal como se muestra en la Imagen 65.



Imagen 65.- Interfaz de inicio de sesión.

Fuente: Imagen capturada de software instalado.

También contamos con los paneles mencionados en la sección 4.7 de este capítulo, a más de otros dos paneles: el panel de “ReestablecerPassPanel” que contiene un formulario para cambiar la contraseña, como se muestra en la Imagen 66. Los datos que se solicitan son el correo electrónico, el nombre de usuario y la contraseña nueva. Posteriormente, se mostrará el panel adicional mencionado, titulado: “Contraseña correcta”, que indica mediante un mensaje que la contraseña se ha cambiado exitosamente.

Después de realizar el cambio de contraseña, el botón de “aceptar” que aparece en el panel de cambio correcto, nos trasladara a la escena de inicio de sesión, para que podamos acceder a la aplicación.

También tenemos la opción de “Crear cuenta nueva” y si elegimos esta opción, básicamente nos traslada a la escena de “Registro” para poder crear la cuenta y realizar todo el procedimiento que se detalló en la sección 4.7.

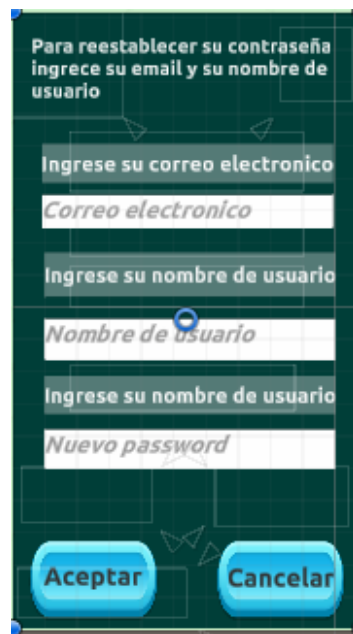


Imagen 66.-Panel para reestablecer la contraseña en la escena de inicio de sesión.

Fuente: Imagen capturada de software instalado.

4.9.1- Actualizando la contraseña en la base de datos.

```
1 <?php
2 include 'header.php';
3 try {
4     $conn = mysqli_connect($server, $user, $pass, $bd);
5     //echo "Conexion exitosa";
6     if (!$conn) {
7         //echo "Lo siento pero algo salio mal :(";
8     }else{
9         if (isset($_POST['email']) && isset($_POST['user_name']) && isset($_POST['nuevoPass'])) {
10            $varemail = $_POST['email'];
11            $varUserName = $_POST['user_name'];
12            $varNPass = hash("sha256", $_POST['nuevoPass']);
13            $sql = "SELECT * FROM `usuarios2` WHERE `email`='".$varemail."' AND `user_name`='".$varUserName.'";";
14
15            $resultado = $conn->query($sql);
16
17            if (($resultado->num_rows) > 0) {
18                $row = mysqli_fetch_assoc($resultado);
19                $sql = "UPDATE `usuarios2` SET `password`='".$varNPass."' WHERE id_usuario='".$row['id_usuario'].";";
20                $res = $conn->query($sql);
21                echo '{"codigo": 206, "mensaje": "contraseña cambiada correctamente", "respuesta":"' . $row['user_name'] . '"}';
22            }else{
23                echo '{"codigo": 406, "mensaje": "Actualizacion incorrecta", "respuesta": "El usuario o la contraseña son incorrectos"}';
24            }
25        }else{
26            echo '{"codigo": 403, "mensaje": "Faltan datos para realizar la operacion", "respuesta": "Datos incompletos"}';
27        }
28    }
29 }
30
31 //Termina el try y sigue el catch
32 }catch (Exception $e) {
33     echo '{"codigo": 401, "mensaje": "Error al conectar cone el servidor", "respuesta": "" . $e . ""}';
34     //echo $e;
35 }
36
37 $conn -> close();
38
39
40 ?>
```

Imagen 67.- Archivo PHP para actualizar el password por medio de una consulta "update".

Fuente: Imagen capturada de software instalado.

Como se muestra en la Imagen 67, la consulta se realiza después de verificar que la conexión ha sido exitosa, tras solicitar los datos y confirmar que no están vacíos, y después de asignar los datos a las variables que utilizaremos. Una vez que hemos superado estos filtros, es cuando ejecutamos la siguiente consulta:

```
"SELECT * DE `usuarios2` WHERE `email`='".$varemail."' AND `user_name`='".$varUserName.'";";
```

El objetivo es verificar si el usuario tiene una cuenta creada. Si la respuesta es afirmativa, pasaremos a realizar la actualización de la contraseña en la base de datos, siguiendo las instrucciones que se encuentran en las líneas 18, 19 y 20. Cabe mencionar que en la línea 18 se utiliza la instrucción "\$row = mysqli_fetch_assoc(\$resultado)", cuyo propósito es obtener el id del usuario, el cual obtenemos gracias a la primera consulta que realizamos. Por último, la consulta que ejecutamos es la siguiente:

```
"UPDATE `usuarios2` SET `password`='".$varNPass."' WHERE id_usuario='".$row['id_usuario'].";";
```

Si no ha ocurrido ningún problema, el servidor nos regresara el código “206”. Si deseas obtener más información o si quieres ver una respuesta más detallada, puedes consultar la Imagen 49.

4.9.2.- Archivo PHP de inicio de sesión.

Por último. Al seleccionar la opción de iniciar sesión, este botón ejecutara el script principal de la escena, el cual se comunica con el archivo PHP nombrado como: “loginNecaxaTravel.PHP”, que se muestra en la Imagen 68. Este proceso está controlado por el script “IniciarSesion.cs”, en el que capturamos los datos, verificamos que no estén vacíos y no exista una sesión previamente iniciada mediante los playerprefs. Si todo esto es correcto, es en ese momento cuando llamamos al archivo PHP para realizar la consulta correspondiente.

```
1 <?php
2
3 include 'header.php';
4
5 try {
6     $conn = mysqli_connect($server, $user, $pass, $bd);
7     //echo "Conexion exitosa";
8
9     if (!$conn) {
10        //echo "Lo siento pero algo salio mal :(";
11    }else{
12        if (isset($_POST['email']) && isset($_POST['pass'])) {
13            $varemail = $_POST['email'];
14            $varPass = hash("sha256", $_POST['pass']); //hash("sha256", $_POST['pass']);
15            $sql = "SELECT * FROM usuarios2 WHERE `email`='".$varemail.'" AND `password`='".$varPass.'"";
16            $resultado = $conn->query($sql);
17
18            //echo $resultado->num_rows;
19            if (($resultado->num_rows) > 0) {
20                $row = $resultado->fetch_assoc();
21                $Json = "{#id#:". $row['id_usuario'].
22                    "#, #nombre#:". $row['nombre'].
23                    "#, #user_name#:". $row['user_name'].
24                    "#, #email#:". $row['email'].
25                    "#, #puntos_obtenidos#:". $row['puntos_obtenidos'].}";
26
27
28                //echo "Hola ".$row['nombre'];
29                //echo "{#codigo#: 202, #mensaje#: Inicio de sesion exitoso, #respuesta#: ".$row['user_name'].}";
30                echo "{#codigo#: 202, #mensaje#: Inicio de sesion exitoso, #respuesta#: ".$Json."}";
31            }else{
32                //echo "Correo o password incorrectos";
33                //echo "</br> La contraseña es: ".$varPass;
34                echo "{#codigo#: 405, #mensaje#: Inicio de sesion incorrecto, #respuesta#: El usuario o la contraseña son incorrectos}";
35            }
36        }
37    }else{
38        //echo "Error: No se puede completar la operacion, ya que no estan los datos que necesitamos";
39        echo "{#codigo#: 403, #mensaje#: Faltan datos para realizar la operacion, #respuesta#: Datos incompletos}";
40    }
41 }
```

Imagen 68.- Archivo “loginNecaxaTravel.PHP” para iniciar sesión en la aplicación.

Fuente: Imagen capturada de software instalado y editada por el autor.

Como no queremos ser repetitivos, en cada archivo PHP verificamos la conexión y aseguramos que los datos no estén vacíos, así como la asignación de las variables que utilizaremos. Lo que más varía es el mensaje y la respuesta que recibimos de este archivo.

La consulta que realizamos es un SELECT, por mediante el cual verificaremos si el usuario exista en la base de datos. La consulta que llevamos a cabo es la siguiente:

```
"SELECT * DE `usuarios2` WHERE `email`='".$varemail.'" AND `password`='".$varPass.'";"
```

Si esta consulta se realiza de manera exitosa y nos devuelve un valor positivo, lo que hacemos es insertar los datos a una variable, intentando darles un formato JSON que procesaremos en Unity mediante C#. El código que se devuelve es “202”, dichos códigos se detallaron a profundidad en la Imagen 49.

4.9.3.- Tratando la información que devolvió el archivo PHP.

Como se mencionó anteriormente, los datos se envían de la misma manera en todos los archivos que utilizan UnityWebRequest. Sin embargo, en esta escena tratamos la información de manera diferente, ya que asignamos los datos del JSON que creamos en el archivo PHP mediante la instrucción que se encuentra encerrada en la Imagen 69.

```
50 IEnumerator LogInUser()
51 {
52
53     imgLoading.SetActive(true);
54     string[] datos = new string[2];
55     datos[0] = inpEmail.text;
56     datos[1] = inpPass.text;
57
58     StartCoroutine(servidor.ConsumirServicio("logIn", datos));
59     yield return new WaitForSeconds(1.0f);
60     yield return new WaitUntil(C => !servidor.ocupado);
61     imgLoading.SetActive(false);
62     if (servidor.resp.codigo == 202)
63     {
64         usuario = JsonUtility.FromJson<UsuarioDB>(servidor.resp.respuesta);
65     }
66     mensajeBienvenido.text = "\t"+ usuario.user_name + "\n Has iniciado sesion, podemos comenzar";
67
68     imgBienvenido.SetActive(true);
69     PlayerPrefs.SetString("Email", usuario.email);
70     PlayerPrefs.SetString("UserName", usuario.user_name);
71     // Debug.Log("Sesion iniciada correctamente");
72     // datosUsuario.AsignarDatos(servidor.resp.respuesta, "jajaja");
73     //ControladorDatos.Instance.LlenarDatos(servidor.resp.respuesta, datos[0]);
74
75 }
76 else
77 {
78     mensajeError.text = servidor.resp.respuesta;
79     imgError.SetActive(true);
80     // Debug.Log("Fallamos");
81 }
```

Imagen 69.- Método de iniciar sesión de la escena login señalando la instrucción donde tratamos los datos del JSON que llega desde el archivo PHP.

Fuente: Imagen capturada de software instalado y editada por el autor.

```
9 referencias
public class UsuarioDB
{
    public int id;
    public string nombre;
    public string user_name;
    public string email;
    public int puntos_obtenidos;
}
```

Imagen 70.- Clase UsuarioDB a la cual le asignaremos los datos del JSON.

Fuente: Imagen capturada de software instalado y editada por el autor.

La instrucción “JSONUtility.DeJSON<UsuarioDB>(servidor.resp.respuesta);” es una función que existe en C# y nos permite asignar un archivo JSON a una variable que contiene las propiedades correspondientes a los datos que se reciben. En este caso, la variable que utilizamos pertenece a una clase que hemos denominado “UsuarioDB”, la cual puedes observar en la Imagen 70. De esta clase, extraeremos los datos que necesitamos; esto se puede ver en la línea 66 de la Imagen 69, donde accedemos al “user_name” para enviar el mensaje de inicio de sesión correcto en el panel correspondiente. Además, insertamos los datos en los playerprefs que utilizaremos para las consultas posteriores.

4.10.- Creación del menú de la aplicación.

Esta sección es sencilla, ya que lo único que haremos es crear el canvas, similar a los que ya hemos creado anteriormente, con la imagen de fondo, el logo, el botón para salir y el título. Sin embargo, en este caso no colocaremos ningún formulario; en su lugar, incluiremos cinco botones y mostraremos un panel inicial que puedes observar en la Imagen 71. El botón de aceptar cierra el panel y crea otro playerpref, el cual utilizaremos para que este panel se ejecute cada vez que se inicie sesión, evitando que se muestre en cada ocasión que ingresemos al menú.



Imagen 71.- Panel que pretende indicarle al usuario que preste mucha atención a las instrucciones de uso.

Fuente: Imagen capturada de software instalado.

Cabe mencionar que el panel para cerrar la aplicación se mantiene, y hemos agregado uno adicional para cerrar sesión, el cual te explicaremos cuando lleguemos a esa sección. Ahora podemos interactuar con el menú, el cual consta de cinco botones, cada uno de los cuales te dirigira a diferentes escenas o paneles, dependiendo de la funcionalidad que debamos implementar. Por esta razón, comenzaremos la explicación desde el último botón hasta el primero, debido a la complejidad de lo que debemos desarrollar. El orden se puede observar en la Imagen 72 que se te mostrará a continuación.



Imagen 72.- Interfaz del menú, mostrando los botones que contiene.

Fuente: Imagen capturada de software instalado.

4.10.1.- Botón “cerrar sesión”.

Este botón es el más simple de la escena, ya que, al presionarlo, se muestra un panel que confirma si deseas cerrar sesión. Este panel contiene dos botones: el de cancelar que prácticamente solo desactiva el panel y el de aceptar, que ejecuta el método “CerrarSesion()”. Este método se encuentra en el script denominado “ControladorGeneral.cs” y utiliza la función “PlayerPrefs.DeleteKey(“Nombre del playerpref que queremos eliminar”)” para eliminar todas las claves que hemos creado a lo largo del desarrollo. En el caso de esta aplicación eliminamos tres claves. Después de eliminarlas, seremos dirigidos a la escena “LogIn”, donde la aplicación permanecerá, puesto que ya no se cumple la condición de que las claves contengan información. Puedes observar el método Imagen 73.

```
27     public void CerrarSesion()
28     {
29         PlayerPrefs.DeleteKey("Email");
30         PlayerPrefs.DeleteKey("UserName");
31         PlayerPrefs.DeleteKey("Menu");
32         SceneManager.LoadScene("LogIn");
33     }
0 referencias
```

Imagen 73.- Método de cerrar sesión, el cual utiliza la instrucción para eliminar las keys.

Fuente: Imagen capturada de software instalado.

Básicamente, esta es toda la funcionalidad de dicho botón. Como se mencionó anteriormente, este es el único que no te dirige a otra escena; simplemente activa y desactiva el panel. Por esta razón, no te mostramos el panel, ya que es similar al de cerrar la aplicación; la única diferencia radica en el mensaje y en la función que ejecuta el botón de aceptar.

4.10.2.- Botón “Créditos”.

Esta es la primera escena simple del menú, esto debido a que solo contiene un panel con el texto de los créditos de la aplicación el cual podrás ver en la Imagen 74. Además, se incluyen las dos imágenes que aparecen en todas las escenas: la imagen de fondo y el logo, así como el botón en la parte superior derecha. Sin embargo, este caso, en lugar de cerrar la aplicación, el botón solo nos regresa a la escena del menú.

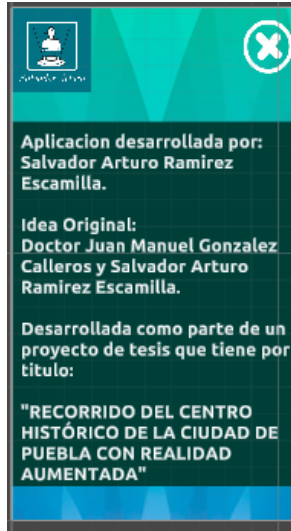


Imagen 74.- Escena de créditos.

Fuente: Imagen capturada de software instalado.

4.10.3.- Botón “Negocios Disponibles”.

Este botón, a pesar de que solo nos muestra información, se puede incluir en una escena por comodidad durante el desarrollo. En el caso del menú que aparece después de iniciar sesión, se mostrará un panel que explicará la dinámica que tendremos con los negocios disponibles para canjear los productos. Además, habrá un botón el cual desactivará el panel, de modo que el usuario ya no lo vea. El panel mencionado en este párrafo se muestra en la Imagen 75.

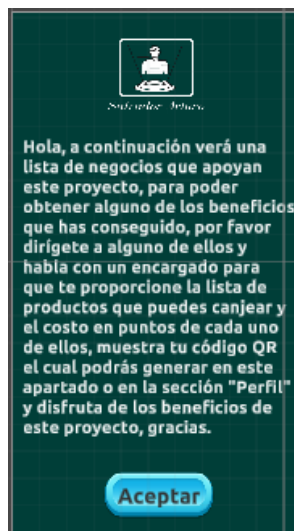


Imagen 75.- Panel de indicaciones para el canjeo de productos en los negocios disponibles.

Fuente: Imagen capturada de software instalado.

Cuando desactivamos el panel, hallaremos un panel similar a los de cada escena, que es una tabla de los negocios asociados a la aplicación realizar los canjes. Este panel cuenta con dos botones: uno en la parte superior izquierda, que nos regresara al menú inicial, y otro que hemos denominado “Generar QR”, que nos llevara a una escena que detallaremos en la sección 4.10.5 debido a que su desarrollo es un poco más complicado.

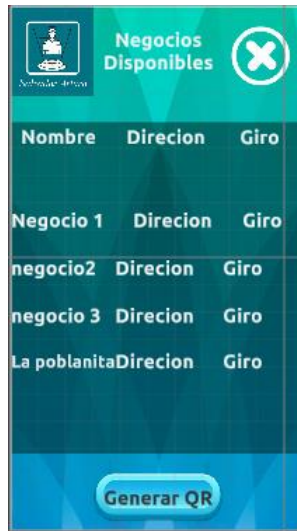


Imagen 76.- Interfaz de negocios afiliados.

Fuente: Imagen capturada de software instalado.

4.10.4.- Botón “Perfil”.

En esta ocasión, el botón nos lleva a una escena en la que se muestran los datos del usuario que inicio sesión en la aplicación. Esto se realiza mediante un controlador general y una consulta simple, similar a la que se lleva a cabo en el inicio de sesión. Primero, nos encontraremos con el panel básico de cada escena. Como podrás observar en la Imagen 77, encontraras que esta escena cuenta con tres botones: el primero te regresara al menú inicial y se encuentra en la parte superior izquierda; el segundo ubicado junto a la leyenda “Correo electrónico”, activara un panel para modificar el correo electrónico; y el último, situado al fondo, contiene el texto “Generar QR” escena que detallaremos en la sección 4.10.5.



Imagen 77.- Interfaz de los datos del perfil.

Fuente: Imagen capturada de software instalado.

4.10.4.1.- Actualizando el correo electrónico.

Cuando presionamos el botón para actualizar nuestro correo electrónico, se activa un panel que solicitará tu nuevo correo y tu contraseña para ejecutar el método “ActualizaCorreo()”, el cual a su vez ejecuta la rutina “CambiaCorreo()”. Esta última envía tres datos: el nuevo correo, el correo registrado y la contraseña. La razón por la que no se solicita nuevamente el correo registrado es que éste se almacena en el playerpref bajo la clave “Email”. Como se muestra en Imagen 78, también notarás que en la línea 43 se ejecuta una instrucción “perfilNuevo.Start()”. El objetivo de esta instrucción es volver a llamar al método que solicita los datos para mostrarlos en la escena en la que nos encontramos al ingresar. Dado que la consulta de datos se realiza una sola vez a través del método start, si dejamos el script sin esa instrucción, al desactivar el panel, el dato que quedaría registrado sería el correo anterior y no el nuevo.

Si la operación nos devuelve el código “203” desactivaremos el panel de actualización y activaremos el panel de cambio correcto, el cual nos mostrara un mensaje que se encuentra en la línea 42. En caso de que la operación no devuelve el código antes mencionado, activaremos el panel de actualización incorrecta, que mostrara el mensaje correspondiente, que puedes observar en la línea 50 de la Imagen 78.

```

20 0 referencias
21 public void ActualizaCorreo()
22 {
23     StartCoroutine(CambiaCorreo());
24 }
25 1 referencia
26 IEnumerator CambiaCorreo()
27 {
28     string[] datos = new string[3];
29     datos[0] = PlayerPrefs.GetString("Email");
30     datos[1] = inpPass.text;
31     datos[2] = inpNuevoEmail.text;
32
33     StartCoroutine(servidor.ConsumirServicio("ActualizaEmail", datos));
34     yield return new WaitForSeconds(1.0f);
35     yield return new WaitUntil(() => !servidor.ocupado);
36     if (servidor.resp.codigo == 203)
37     {
38         PlayerPrefs.DeleteKey("Email");
39         PlayerPrefs.SetString("Email", servidor.resp.respuesta);
40         panelActualizaCorreo.SetActive(false);
41         panelActualizaCorreoGood.SetActive(true);
42         inpPass.text = "";
43         inpNuevoEmail.text = "";
44         mensajeActuTrue.text = "Tu correo se ha actualizado correctamente \nTu nuevo correo es:\n"+servidor.resp.respuesta;
45         perfilnuevo.Start();
46     }
47     else
48     {
49         panelActualizaCorreoFail.SetActive(true);
50         mensajeActuFail.text = "Lo sentimos no se pudo actualizar tu correo\n" + servidor.resp.respuesta;
51     }
52 }

```

Imagen 78.- Script de "ActualizarCorreo.cs" en el que vemos como se realiza la actualización del correo.

Fuente: Imagen capturada de software instalado y editada por el autor.

El archivo PHP encargado de realizar la actualización se muestra en la Imagen 79. Este archivo primero ejecuta la consulta: "SELECT * DE `usuarios2` WHERE `user_name` = ".\$varUserName." AND `email` = ".\$varEmail.";" para verificar si el usuario existe. En este caso también se ejecuta: "SELECT * DE `usuarios2` WHERE `email`=".\$varnuevoEmail."" Para comprobar que el correo proporcionando no esté registrado en la base de datos. Si el correo no se encuentra en la base de datos, validamos la existencia del usuario, lo cual se determina al verificar si la primera consulta devuelve un resultado (se espera solo un resultado, ya que siempre validamos que el correo no exista para poder proceder con el registro). Si el usuario existe, se ejecuta la siguiente consulta: "UPDATE `usuarios2` SET `email`=".\$varnuevoEmail." WHERE id_usuario=".\$row['id_usuario'].";"", y lo que ara es actualizar el correo electrónico del usuario mediante su id de usuario, el cual se obtuvo desde la primera consulta. En caso de que la actualización falle en algún momento, se devolverán egresarán los códigos 406, 408, 403 o 401. dependiendo del tipo de error que se presente.

```

1 |<?php
2 |include 'header.php';
3 |try {
4 |    $conn = mysqli_connect($server, $user, $pass, $bd);
5 |    //echo "Conexion exitosa";
6 |    if (!$conn) {
7 |        echo "Lo siento pero algo salio mal :(";
8 |    }else{
9 |        if (isset($_POST['mail']) && isset($_POST['pass']) && isset($_POST['nemail'])) {
10 |            $varemail = $_POST['mail'];
11 |            $varPass = hash("sha256", $_POST['pass']);
12 |            $varnuevoEmail = $_POST['nemail'];
13 |            $sql = "SELECT * FROM `usuarios2` WHERE `email`='".$varemail.'" AND `password`='".$varPass.'"";
14 |            $resultado = $conn->query($sql);
15 |            $verinEmail = "SELECT * FROM `usuarios2` WHERE `email`='".$varnuevoEmail.'"";
16 |            $resultado2 = $conn->query($verinEmail);
17 |            //echo $resultado->num_rows;
18 |            if (($resultado2->num_rows) == 0) {
19 |
20 |                if (($resultado->num_rows) > 0) {
21 |                    $row = mysqli_fetch_assoc($resultado);
22 |                    $sql = "UPDATE `usuarios2` SET `email`='".$varnuevoEmail.'" WHERE id_usuario='".$row['id_usuario']."'";
23 |                    $res = $conn->query($sql);
24 |                    $Json = $varnuevoEmail;
25 |                }else{
26 |                    echo '{"codigo": 406,"mensaje":"Actualizacion incorrecta","respuesta":"El usuario o la contraseña son incorrectos"}';
27 |                }
28 |                echo '{"codigo": 203,"mensaje": "Correo electronico actualizado correctamente", "respuesta":"'.$Json.'"}';
29 |            }else{
30 |                echo '{"codigo": 408,"mensaje":"Actualizacion incorrecta","respuesta":"El correo con el que desea actualizar los datos, ya existe"}';
31 |            }
32 |
33 |        }else{
34 |            echo '{"codigo": 403,"mensaje":"Faltan datos para realizar la operacion","respuesta":"Datos incompletos"}';
35 |        }
36 |    }
37 |}
38 |
39 |//Termina el try y sigue el catch

```

Imagen 79.- Archivo PHP que realiza la actualización del correo electrónico.

Fuente: Imagen capturada de software instalado.

4.10.4.2.- Visualización de los datos del perfil.

En esta sección, explicaremos únicamente el archivo PHP que nos proporciona la información del usuario, para evitar redundancias, ya que el script es similar al utilizado para actualizar el correo electrónico; solo varían los datos que utilizamos, que son el nombre de usuario y el correo que tenemos almacenados en los playerprefs que se crearon al momento del registro o del inicio de sesión.

En la Imagen 80 podrás observar el archivo “datosNT.PHP”, en el cual realizaremos la consulta para extraer los datos de la base de datos. Dado que los datos se insertan después de verificar que son correctos, no es necesario comprobar si el usuario existe. Por esa razón, enviamos los datos y realizamos directamente la consulta para extraer la información, sin necesidad de verificar si este usuario está registrado.

```

1 <?php
2 include 'header.php';
3
4 try{
5     $conn = mysqli_connect($server, $user, $pass, $bd);
6
7     if(!$conn){
8
9     }else{
10        if(isset($_POST['user_name']) && isset($_POST['email'])){
11
12            $varUserName = $_POST['user_name'];
13            $varEmail = $_POST['email'];
14
15            $sql = "SELECT * FROM usuarios2 WHERE user_name = '$varUserName.' AND `email` = '$varEmail.'";
16            $resultado = $conn->query($sql);
17
18            if(($resultado->num_rows) > 0){
19                $row = mysqli_fetch_assoc($resultado);
20                $Json = "{id#:".$row['id_usuario'],
21                    "#,nombre#:#:".$row['nombre'],
22                    "#,user_name#:#:".$row['user_name'],
23                    "#,email#:#:".$row['email'],
24                    "#,puntos_obtenidos#:".$row['puntos_obtenidos']."}";
25            }else{
26                echo '{"codigo": 407,"mensaje":"No se pudo realizar la operacion","respuesta":"Nombre de usuario o correo
                incorrectos"}';
27            }
28            echo '{"codigo": 205,"mensaje":"La operacion se realizo exitosaeente","respuesta":"'.$Json.'"}';
29            //echo $Json;
30        }else{
31            echo '{"codigo": 403,"mensaje":"Faltan datos para realizar la operacion","respuesta":"Datos incompletos"}';
32        }
33    }
34 }catch(Exception $e){
35     echo '{"codigo": 401,"mensaje":"Error al conectar con el servidor","respuesta":"'.$e.'"}';
36 }
37
38 $conn -> close();
39
40 ?>

```

Imagen 80.- Archivo PHP en el que extraemos los datos de la base de datos.

Fuente: Imagen capturada de software instalado.

Con eso concluimos la sección en la que explicamos el funcionamiento de la escena de perfil. Aunque es simple, puede que te sientas confundido por la gran cantidad de información que debemos tratar y mostrar. Sin embargo, esto es relativamente sencillo, ya que reutilizamos muchas funciones que hemos implementado previamente.

4.10.5.- Generador de códigos QR.

En este apartado te explicaremos el cómo generar un código QR. Primero, abordaremos la razón por la cual hemos incorporado esta función y el objetivo de este código en nuestra aplicación: el cual es facilitar el intercambio de información con los negocios que serán colaboradores. Se tiene previsto desarrollar una aplicación que permitirá canjear productos, aunque esto se considera como trabajo futuro. Si bien contamos con un prototipo que se utilizó para realizar los experimentos, no se mencionará en este documento, ya que no es el objetivo del mismo. La información que se compartirá consistirá en un texto con el nombre de usuario y el correo electrónico, a través del cual aplicaremos el descuento.

Lo primero que haremos es importar un paquete que podrás descargar en https://drive.google.com/file/d/1UB1j8Cs4g8x67caQFg412NGWIwC_-3af/view. Debemos colocarlo en una carpeta; en nuestro caso, hemos creado una carpeta llamada “GeneradorQR”, que

se encuentra dentro de otra carpeta que nombramos “Extras”, la cual contiene todos los recursos que no tienen una clasificación especial.

Una vez importado el elemento, comenzaremos a configurar la cámara. Lo primero que debemos hacer es cambiar la opción de “Skybox” a “solid color”, ajustaremos la proyección a “Orthographic” (para encontrar estas opciones selecciona la “main camera” y búscalas en el inspector). Crearemos un elemento image y lo renombramos como “QR”, ya que este lugar donde se genera el código QR. Lo ubicaremos en la posición deseada y ajustaremos sus dimensiones a las que consideremos adecuadas; en nuestro caso las dimensiones quedaron establecidas en: “Right = 81.41” y “Bottom = 31.33499”.

Seleccionamos el canvas y, en el inspector, buscamos la opción que dice “Render Camera”, que para referenciarlo a “Main Camera”, solo tenemos que arrastrar el elemento que deseamos vincular al campo correspondiente. Ahora procederemos a crear el script que nos permitirá generar el código QR. Este script se te muestra en la Imagen 81 y en la Imagen 82.

Cabe mencionar que la función Awake ya se te explico anteriormente y su funcionalidad es la misma. Esta función se encarga de verificar si la sesión esta iniciada, ya que, si los datos no han sido insertados, no tendremos los recursos necesarios para generar el código QR.

Como se mencionó anteriormente, crearemos un objeto de juego vacío para poder referenciar nuestro script. Lo primero que haremos en el script es eliminar la función “Update”, ya que no la utilizaremos. Posteriormente, procederemos a importar las bibliotecas que se muestran en la Imagen 81, en las líneas 5, 6, 7, 8 y 9. También comenzaremos a crear las variables que utilizaremos; estas variables se encuentran desde la línea 12 hasta la línea 16 en la imagen mencionada anteriormente. Un elemento nuevo que utilizaremos es “Texture2D”, que es la textura encargada de almacenar nuestro texto sin codificar. A través de ella, colocaremos el texto en nuestro componente de image y generaremos el código QR.

```
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4 //Librerias que necesitamos para generar el codigo
5 using ZXing;
6 using ZXing.QrCode;
7 using UnityEngine.UI;
8 using TMPro;
9 using UnityEngine.SceneManagement;
10 public class QRCodeGenerator : MonoBehaviour
11 {
12     [SerializeField]
13     private RawImage imagenqueRecive;
14     [SerializeField]
15     private TMP_Text bienvenido;
16     private Texture2D textoSinCodificar;
17
18     private void Awake()
19     {
20         string email = PlayerPrefs.GetString("Email", "");
21         string user = PlayerPrefs.GetString("UserName", "");
22         if (email == "" || user == "")
23         {
24             SceneManager.LoadScene("Login");
25         }
26     }
27
28     void Start()
29     {
30         textoSinCodificar = new Texture2D(256,256);
31         bienvenido.text = "Bienvenid@ " + PlayerPrefs.GetString("UserName") + " Muestrale este codigo al encargado del negocio" +
32             " para poder canjear tu recompensa";
33         CodificandoTexto();
34     }
35 }
```

Imagen 81.- Primera parte del script que será el encargado de generar el código QR.

Fuente: Imagen capturada de software instalado y editada por el autor.

Lo primero que aremos en el método “start” es inicializar nuestra variable de tipo “Textrure2D” con el valor de “256”, y esta inicialización la encontraras en la línea 30 de la Imagen 81. Posteriormente, mostraremos un mensaje de bienvenida al usuario y ejecutaremos la función “CodificandoTexto” para generar el código QR.

Ahora crearemos la función “Codificar” la cual recibe el texto a codificar y las dimensiones que tendrá el código QR que se generará. Lo primero que debemos hacer es crear un codificador de barras, y esto lo realizamos con las instrucciones que se muestran en las líneas 37 a 45 de la Imagen 82. Dentro de esta instrucción, en la línea 39, lo que hacemos es definir el formato del código que crearemos; por esta razon, le asignamos “BarcodeFormat.QR_CODE”. Luego debemos asignar opciones, como el tamaño del código QR, para lo que asignamos los valores recibidos en la función.

Finalmente, retornamos la matriz de color 32, como se indica en la línea 46. En resumen, la función que acabamos de explicar es la encargada de generar el código QR.

```

35     1 referencia
36     private Color32 [] Codificar(string textoACodificar, int width,int height)
37     {
38         BarcodeWriter writer = new BarcodeWriter
39         {
40             Format = BarcodeFormat.QR_CODE,
41             Options = new QrCodeEncodingOptions
42             {
43                 Height = height,
44                 Width = width,
45             };
46         };
47         return writer.Write(textoACodificar);
48     }
49     0 referencias
50     public void GeneraQR()
51     {
52         CodificandoTexto();
53     }
54     2 referencias
55     private void CodificandoTexto()
56     {
57         //string textWrite = string.IsNullOrEmpty(textoInputFile.text) ? "No has escrito nada compadre" : textoInputFile.text;
58         string textWrite = PlayerPrefs.GetString("Email") + "*" + PlayerPrefs.GetString("UserName");
59         Debug.Log(textWrite);
60         Color32[] _convertirTextoATextura = Codificar(textWrite, textoSinCodificar.width, textoSinCodificar.height);
61         textoSinCodificar.SetPixels32(_convertirTextoATextura);
62         textoSinCodificar.Apply();
63         imagenqueRecive.texture = textoSinCodificar;
64     }
65 }
66

```

Imagen 82.- Segunda parte del script que será el encargado de generar el código QR.

Fuente: Imagen capturada de software instalado y editada por el autor.

Ahora procederemos a crear una función pública que llamaremos “GeneraQR”, la cual se encargara de invocar la función “CodificandoTexto”. Hacemos esto porque función es privada y solo se puede acceder a ella por a través de otro método. En esta función, debemos crear la cadena que codificaremos. Como puedes observar en la Imagen 82, en la línea 57, agregamos a una variable de tipo string el correo que está almacenado en memoria, concatenado con un asterisco (*), y este a su vez, se concatena con el nombre de usuario, que es un dato que también tenemos almacenado en memoria.

Después de esa asignación crearemos una matriz de colores que denominaremos “_convertirTextoATextura” y llamaremos a la función “Codificar”, cuyos parámetros serán:

1. “textWrite”, que es la cadena que creamos al iniciar la función “CodificandoTexto”.
2. “textoSinCodificar.width” que es un valor de nuestro elemento “Texture2D”.
3. “textoSinCodificar.height” que es un valor de nuestro elemento “Texture2D”.

Esta instrucción nos devolverá una matriz de píxeles. Por esta razon, después de que la función “Codificar” nos devuelve la matriz, necesitamos configurar los píxeles. Así, asignamos la textura a nuestro texto sin codificar y ejecutamos la función “Apply” que básicamente aplica la nueva textura a la variable. Estas instrucciones se pueden observar en la Imagen 82,

específicamente en las líneas 60 y 61. Por último utilizaremos la imagen en la que mostraremos el código, asignándolo como se indica en la línea 63.

Con esto hemos creado el código QR, cuyo objetivo es dar información a los negocios asociados a nuestra aplicación. Además, la interfaz cuenta con un botón que nos regresara a la escena de perfil y una imagen que muestra el logo del creador del proyecto (el autor de este trabajo). En la Imagen 83 se te muestra la interfaz de la escena del código QR.



Imagen 83.- interfaz de Genera QR y un ejemplo de un código ya creado con el mensaje de un usuario registrado.

Fuente: Imagen capturada de software instalado.

Como conclusión, podemos afirmar que este módulo puede utilizarse en diversas ocasiones y para diferentes medios, ya que nuestra implementación te convierte cualquier texto a código QR. Anué la dinámica que hemos desarrollado no ha recibido la dedicación necesaria, esta funcionalidad será de gran utilidad para crear actividades en futuros proyectos.

4.11.- Colocación de contenido virtual con Lightship.

En esta sección, mostraremos cómo integrar contenido virtual en la realidad mediante el Sistema de Posicionamiento Visual (VPS, por sus siglas en inglés). Para ello, utilizaremos la herramienta proporcionada por Niantic, la empresa responsable del desarrollo y mantenimiento de Lightship, que nos permitirá posicionar el contenido de manera efectiva.

4.11.1.- ¿Qué es lightship?

“Lightship te brinda las herramientas para crear experiencias de RA en el mundo real que inspiran a las personas a explorar el mundo juntos” [45].

Es una tecnología que pertenece a la empresa Niantic, una compañía estadounidense de desarrollo de software conocida por la creación y publicación de juegos móviles de realidad aumentada, como Ingress, Harry Potter “Wizards Unite” y Pokémon GO.

Esta empresa se está enfocando en el desarrollo de diversas herramientas para la creación de realidad aumentada, ya que Lightship es un conjunto de herramientas que nos ayudaran a crear experiencias de realidad aumentada, que inspiran al movimiento y fomentan la exploración.

4.11.3.- Descarga de Lightship ARDK 3.0

Como se mencionó anteriormente, la herramienta Lightship le pertenece a la empresa Niantic. Por lo tanto, nos dirigiremos a su página para acceder a la documentación y de descargar los recursos que necesitamos. Como podrás ver en la Imagen 84, se muestra el apartado principal de la página <https://lightship.dev/account/dashboard>, a la que accedemos ingresando a <https://lightship.dev> e iniciando sesión. Si no tienes una cuenta, podrás crear una sencilla, proporcionando una dirección de correo electrónico y una contraseña.

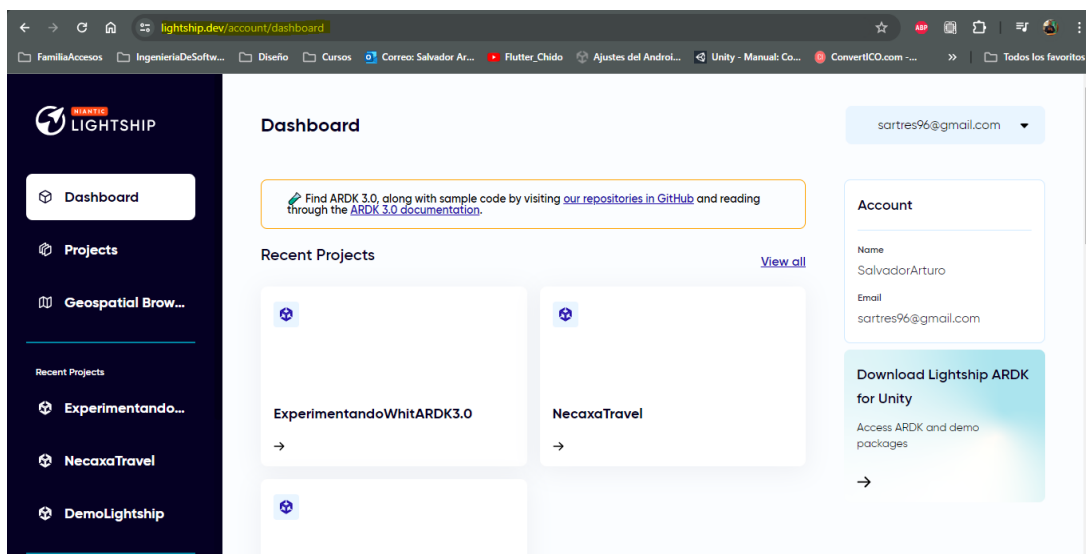


Imagen 84.-Página principal de la cuenta de lightship, de donde descargaremos todos los recursos.

Fuente: Imagen capturada de <https://lightship.dev/account/dashboard>

Una vez allí, nos dirigiremos a la sección de “Documentation” por medio del cual tendremos acceso a la documentación de ARDK 3.0 y podremos descargar los recursos necesarios para integrar contenido 3D en la realidad. La documentación que necesitas se encuentra en el siguiente enlace: <https://lightship.dev/docs/ardk/setup/> .

En dicha documentación se explica cómo instalar los recursos en Unity de tres maneras:

1. Por URL de github.
2. Seleccionando desde Tarball.
3. Agregarlo como un package.

En este documento, te explicaremos como instalar los paquetes mediante tarball, que es una forma sencilla y cómoda para nosotros. Para ello, debes dirigirte a la página que se te proporciono anteriormente, en la sección “Installing the ARDK Plugin de Tarball”. Allí encontrarás dos enlaces, como podrás ver en la Imagen 85, que te guiara a dos repositorios de GitHub desde los cuales descargaremos los recursos que necesitamos.

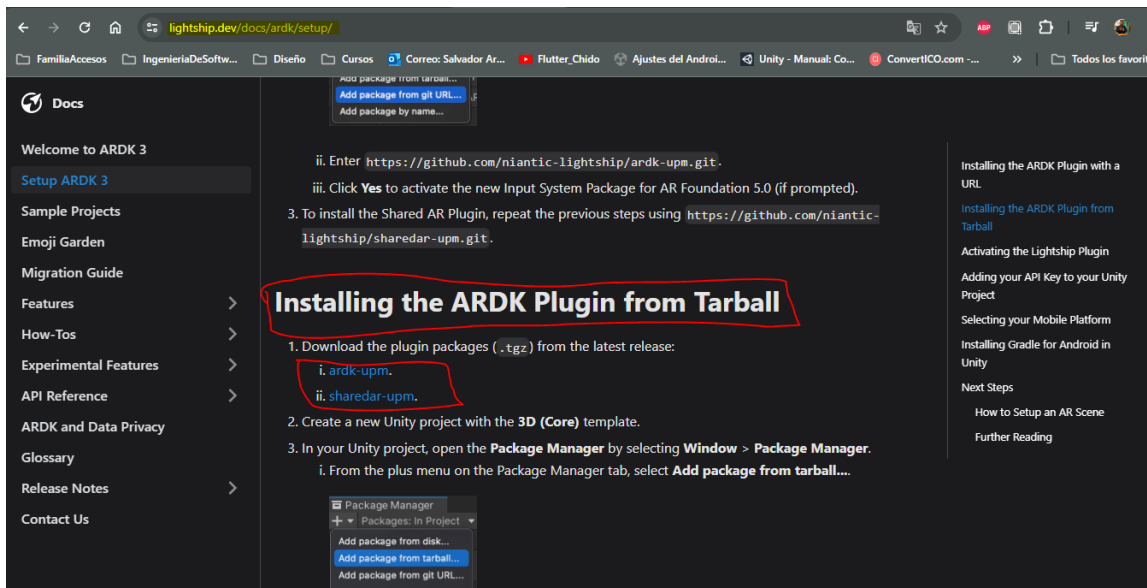


Imagen 85.-Documentación para la instalación de Lightship en Unity de diferentes modos

Fuente: Imagen capturada de <https://lightship.dev/docs/ardk/setup/> y editada por el autor.

Como puedes observar en la Imagen 85, se muestran dos hipervínculos por medio de los cuales accederemos a los siguientes enlaces:

- I. <https://github.com/niantic-lightship/ardk-upm/releases/tag/3.4.0>
- II. <https://github.com/niantic-lightship/sharedar-upm/releases/tag/3.4.0>

Los enlaces te llevarán a los repositorios correspondientes, desde donde podrás desplazarte hasta la parte inferior y descargar los archivos con extensión “.tgz” simplemente haciendo clic en las opciones que se muestran en la Imagen 86 y en la Imagen 87. Estas imágenes ilustran los repositorios desde los cuales descargaremos los archivos “ardk-upm” y “sharedar-upm”, que, a la fecha de redacción de este documento, se encuentran en la versión “3.4.0”, que es la versión más reciente para trabajar con Unity.

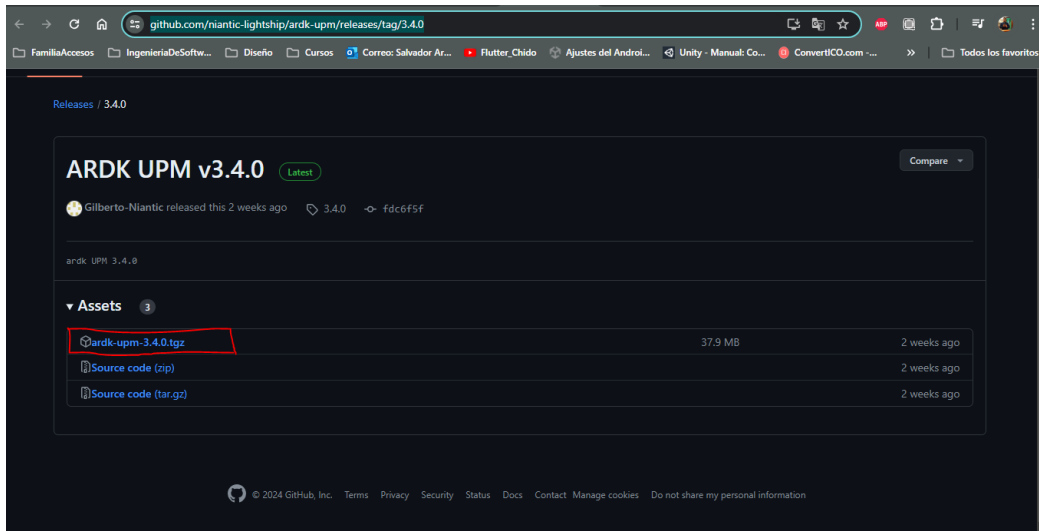


Imagen 86.-Repositorio del vínculo I.

Fuente: Imagen capturada de <https://github.com/niantic-lightship/ardk-upm/releases/tag/3.4.0> y editada por el autor.

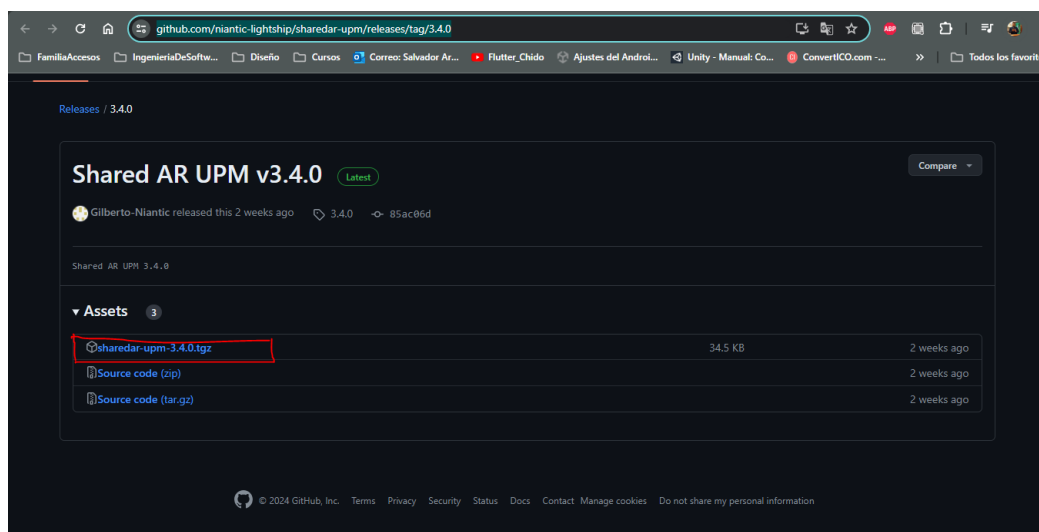


Imagen 87.-Repositorio del vínculo II.

Fuente: Imagen capturada de <https://github.com/niantic-lightship/sharedar-upm/releases/tag/3.4.0/> y editada por el autor.

4.11.4.- Importando los recursos en Unity.

Una vez que hayamos descargados los recursos, procederemos a importarlos en Unity. Para ello, nos trasladaremos al proyecto que creamos anteriormente. Seleccionaremos la opción “Windows” en el menú ubicado en la parte superior, lo que desplegará una lista de opciones. A continuación, elegiremos la opción “Package Manager”, esta nos llevará a la vista que se muestra en la Imagen 88.

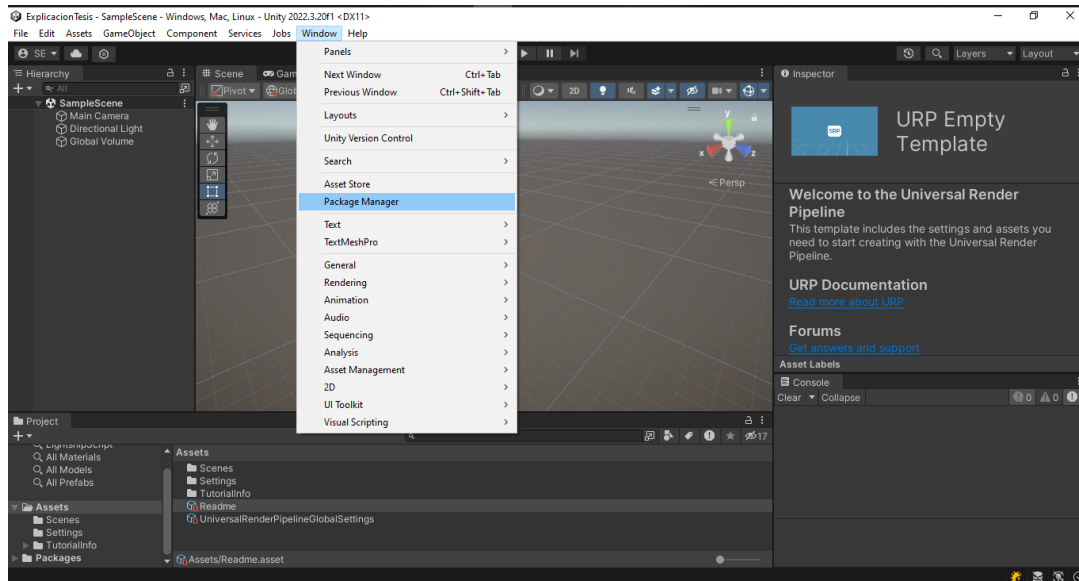


Imagen 88.-Opciones seleccionadas para desplegar el “Package Manager”.

Fuente: Imagen capturada de software instalado.

Posteriormente, seleccionaremos la opción que se te muestra, la cual es un símbolo de “+” con una flecha apuntando hacia abajo. Esto desplegará las opciones disponibles para importar “packages”. Elegiremos la opción que dice “Add package de tarball...”, lo que abrirá una ventana en la que se te pedirá que selecciones el archivo que deseas importar.

Para ello, deberás buscar la carpeta donde se encuentran los archivos “.tgz” que descargamos anteriormente. Importaremos primero el archivo “ardk-upm-3.4.0”. Después de seleccionar esta opción, Unity te pedirá reiniciarse; aceptamos la solicitud y esperamos a que el software se reinicie. Posteriormente, seguiremos los mismos pasos para importar el archivo “sharedar-upm-3.4.0”, de la misma manera en que se importamos el archivo anterior. Cabe mencionar que debes buscar este archivo en la carpeta en la que lo colocaste al descargarlo desde el repositorio de GitHub mencionado anteriormente.

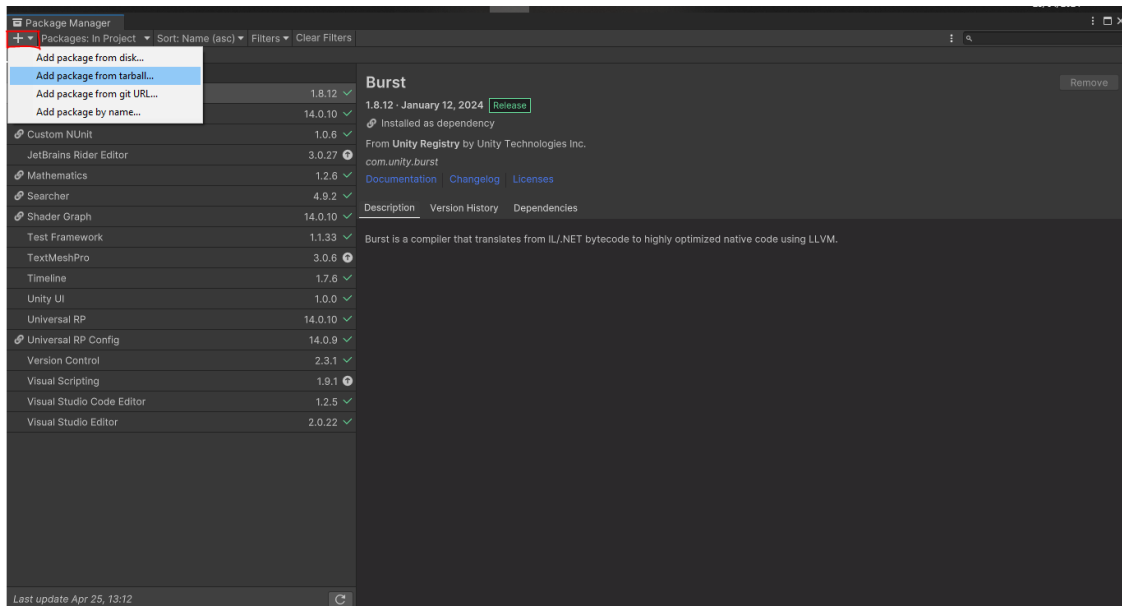


Imagen 89.-Package manager, seleccionando la opción para importar los recursos desde tarball.

Fuente: Imagen capturada de software instalado.

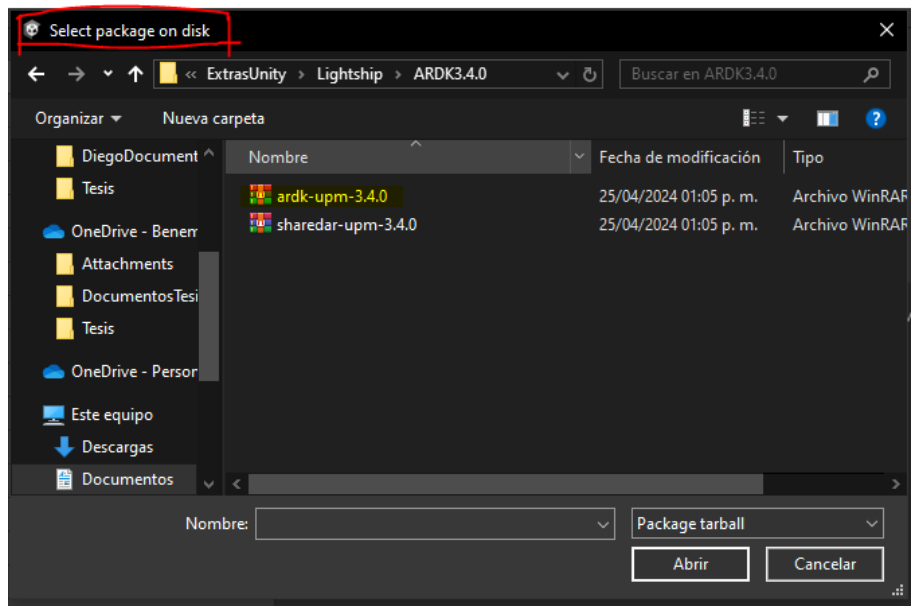


Imagen 90.-Importando "ardk-upm-3.4.0" a Unity desde tarball.

Fuente: Imagen capturada de software instalado.

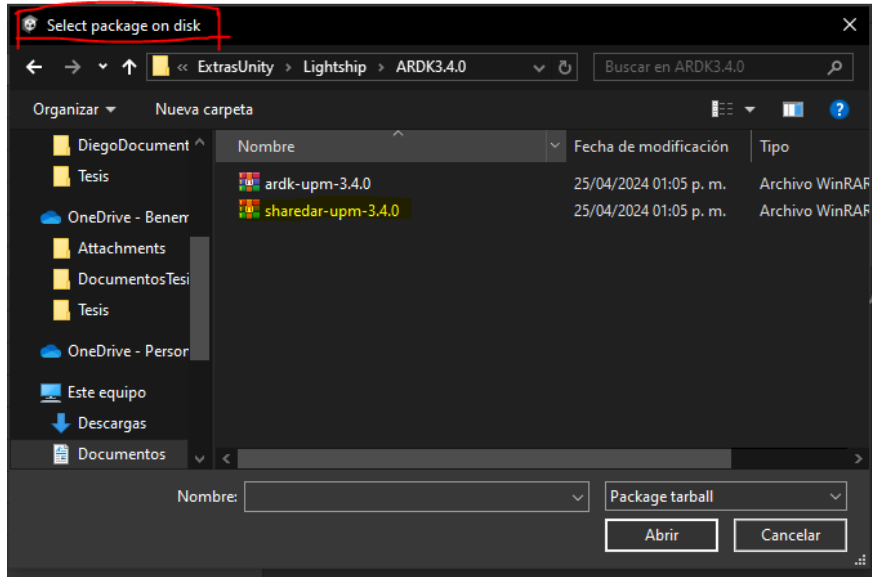


Imagen 91.-Importando “sharedar-upm-3.4.0”.

Fuente: Imagen capturada de software instalado.

Una vez que se importaron los paquetes, notarás que Unity te muestra un error. Esto se debe a que no tienes una clave del producto, la cual podemos obtener fácilmente. Debemos regresar a la página de la cuenta de Lightship; esta vez seleccionaremos la opción de “Projects” y crearemos un proyecto nuevo, el cual solo es colocar nombre y seleccionar la clave que se crea automáticamente como lo puedes ver en Imagen 92, siguiendo los pasos que se muestran en la Imagen 93.

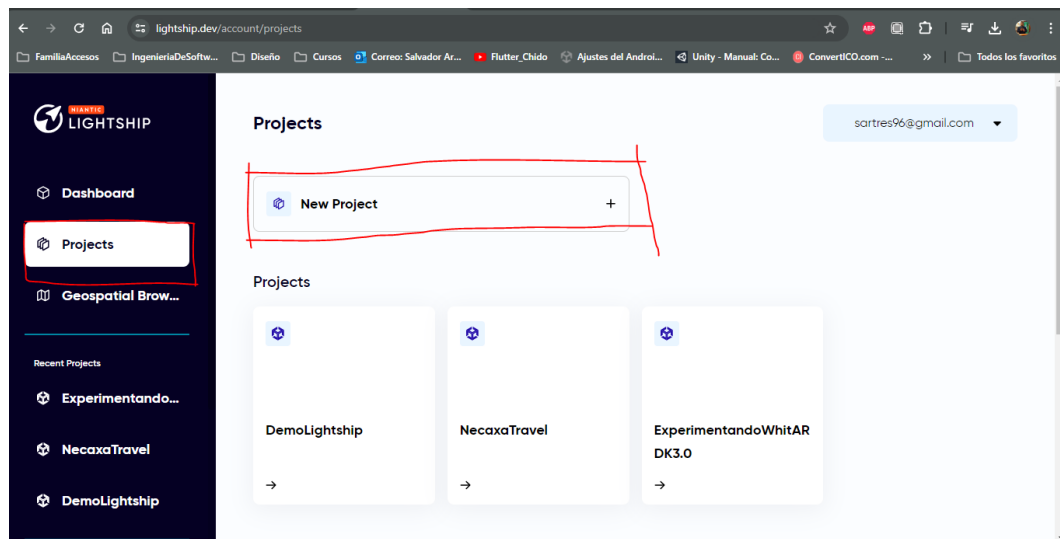


Imagen 92.-Creando la “Key” para el proyecto.

Fuente: Imagen capturada de <https://lightship.dev/account/projects>.

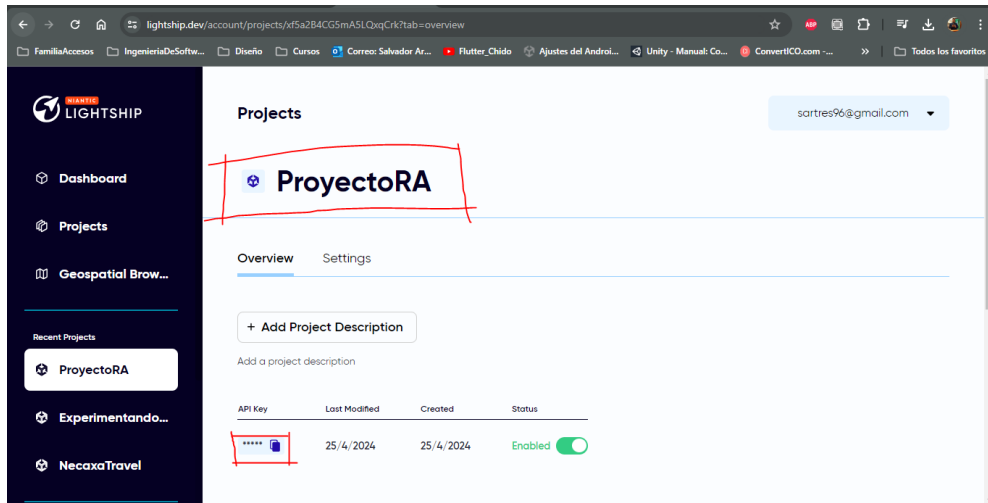


Imagen 93.-Copiando la “key” seleccionando la opción que se muestra en la imagen.

Fuente: Imagen capturada de <https://lightship.dev/account/projects> .

Una vez que hemos creado la clave del proyecto, la copiamos, regresamos a Unity. Nos ubicamos en el menú superior, desplegamos un menú con la opción que dice Lightship, y debemos seleccionar “Settings”. En esta parte proporcionaremos la clave del producto, lo cual nos eliminara los errores que se muestran en la Imagen 94. En la Imagen 95, podrás ver las opciones en las que agregamos dicha clave.

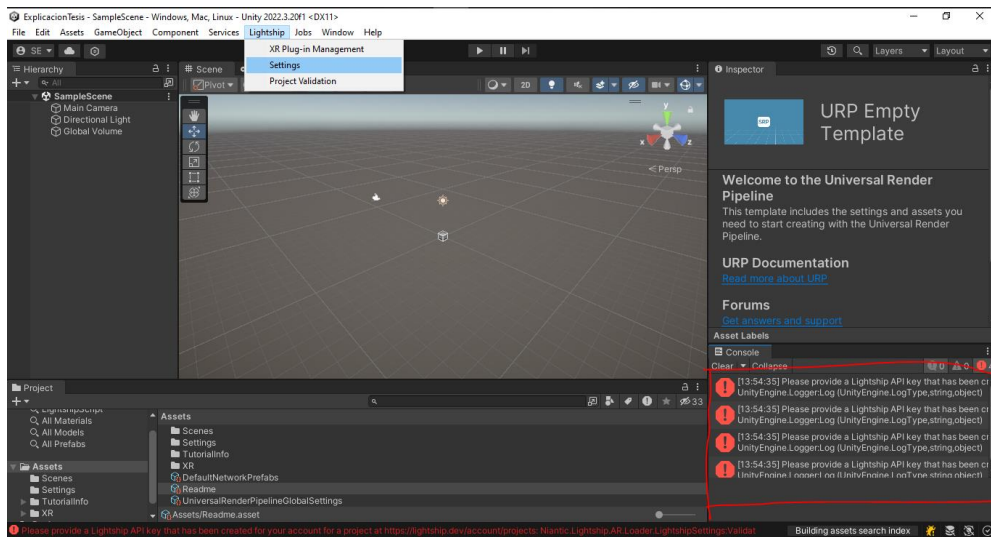


Imagen 94.-Menú para proporcionar la clave del proyecto, así como los errores que se nos muestran por no haber agregado la clave antes.

Fuente: Imagen capturada de software instalado y editada por el autor.

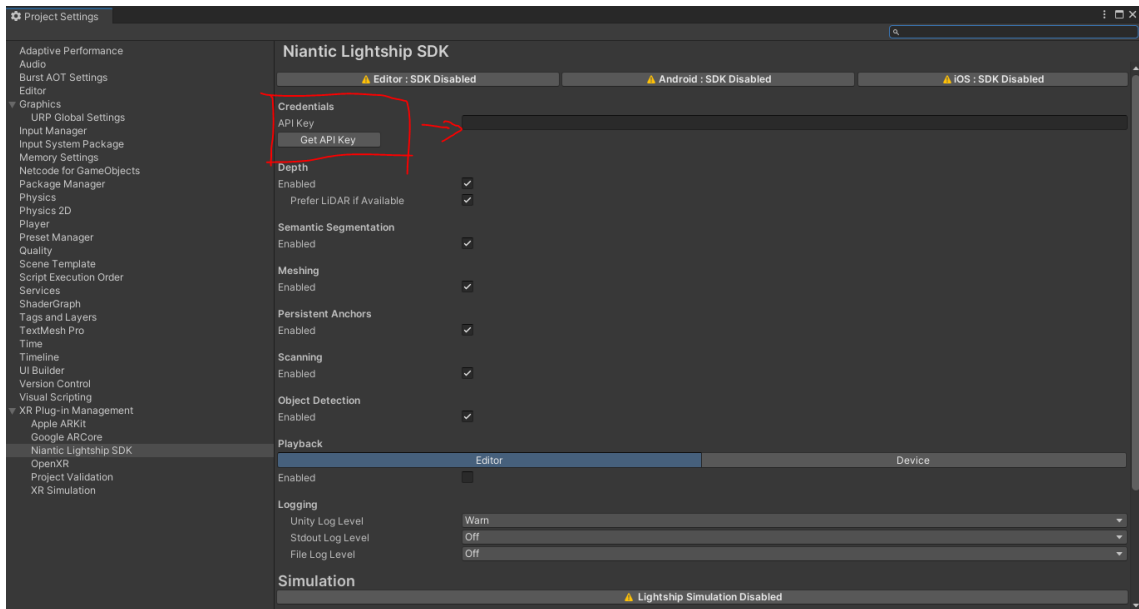


Imagen 95.-Apartado en el que agregaremos la clave del proyecto.

Fuente: Imagen capturada de software instalado y editada por el autor.

Una vez que hemos agregado la clave al proyecto, debemos proceder a configurar el mismo, ya que Lightship tiene requisitos mínimos específicos que deben cumplirse para que la herramienta se compile correctamente.

4.11.5.- Configurando el proyecto.

Para comenzar a configurar el proyecto, debemos dirigirnos al menú de superior y seleccionar la opción “File”, esto nos desplegará un menú . Seleccionamos la opción “Build and settings”, nos llevará a una ventana en la cual cambiaremos la plataforma en la que se corra el proyecto. Tenemos que seleccionar “Android”, Presionamos la opción “Switch platform” como se muestra en la Imagen 97.

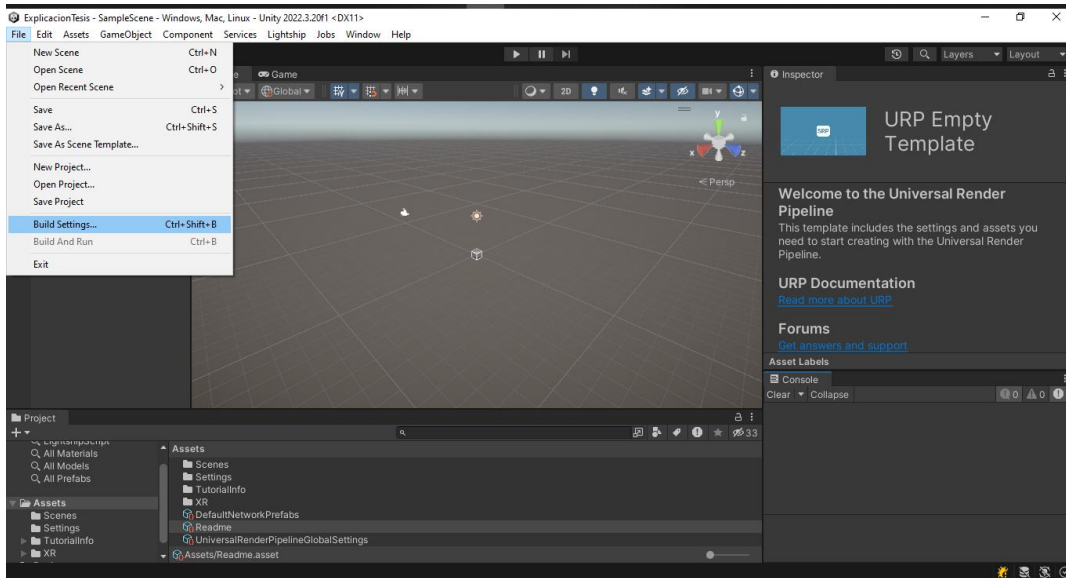


Imagen 96.-Seleccionando la opción de “Build Settings”.

Fuente: Imagen capturada de software instalado y editada por el autor.

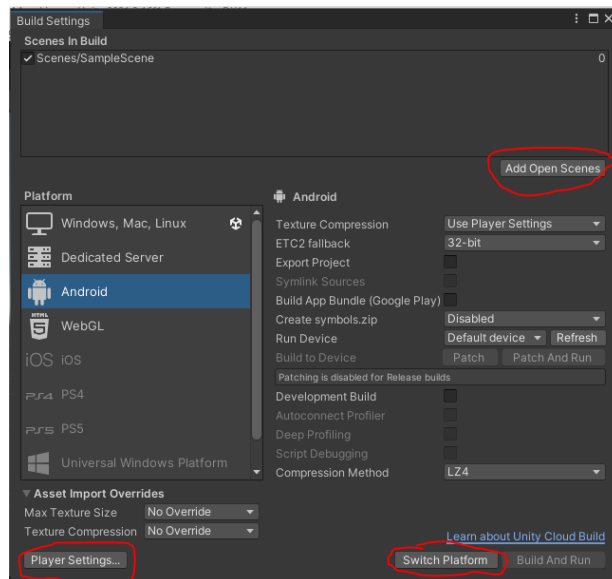


Imagen 97.-Ventana de “Build settings” en donde cambiaremos la plataforma a “Android”.

Fuente: Imagen capturada de software instalado y editada por el autor.

Una vez que cambiemos de plataforma, debemos seleccionar la opción “Player Settings” y en ese momento comenzaremos la configuración del proyecto.

Lo primero que debemos configurar es el Rendering. Para ello, nos trasladaremos a la opción de “Other Settings” y tenemos que asegurarnos que esté seleccionada la opción “Linear”. Eliminaremos la opción “Vulkan” con el botón “-” que se encuentra debajo de los elementos, como se muestra en la Imagen 98, dejando únicamente “OpenGL ES3”.

Posteriormente, debemos configurar el nivel mínimo de Android que utilizara la aplicación. Esto es necesario debido a los requerimientos, ya que las funcionalidades de la herramienta requieren como mínimo, Android 7.0 “Nougat”. Para este proyecto el nivel mínimo que seleccionaremos será Android 9, ya que ese es la versión más baja a la que tenemos acceso. No hay inconveniente en la versión de android que elijas, siempre y cuando sea igual o superior a la 7.0. Además, debemos cambiar el “Scripting Backend” de “Mono” a “IL2CPP” y el “Api compatibility level” de “.NET estándar 2.1” a “.Net Framework”, activaremos la opción “ARM64” en el apartado de “Target Architectures”, la configuración debe quedar como se muestran en la Imagen 99.

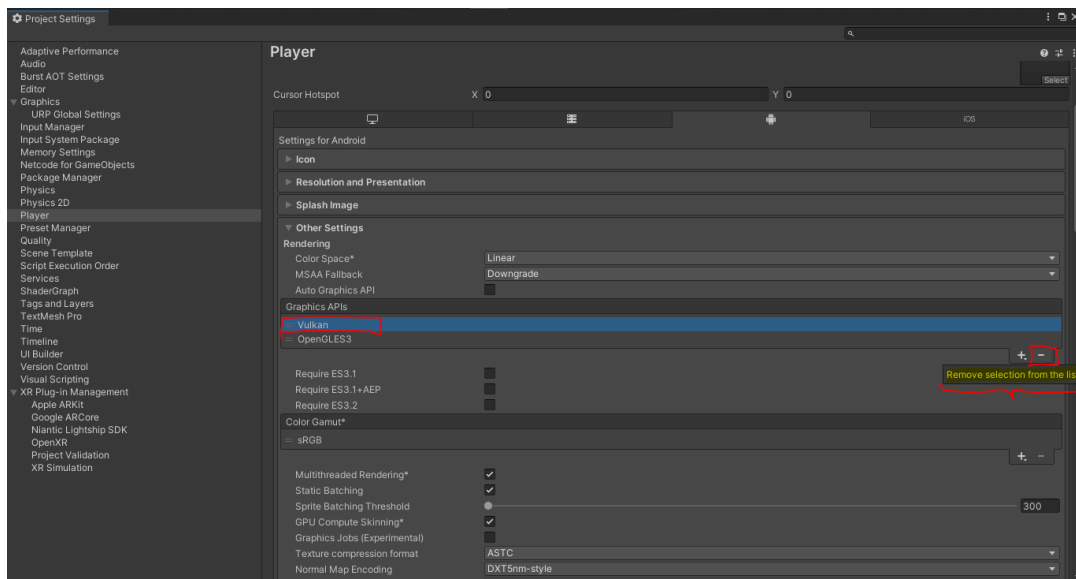


Imagen 98.-Primeras configuraciones del proyecto.

Fuente: Imagen capturada de software instalado y editada por el autor.

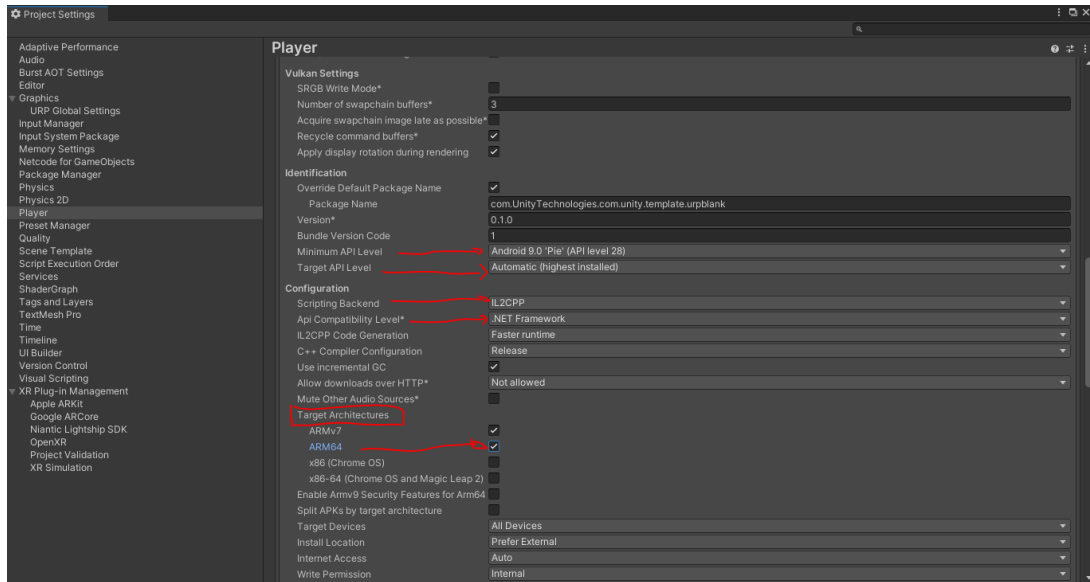


Imagen 99.-Cambiando las configuraciones del nivel mínimo de Android, así como el scripting, api compatibility y target architectures.

Fuente: Imagen capturada de software instalado y editada por el autor.

Ahora hemos terminado con las configuraciones para Android. En caso de que desees compilar para iOS, las configuraciones se encuentran en la documentación que se te proporcionó en la sección 4.2.3, en el apartado “Player Settings for iOS:”.

Para finalizar, debemos acceder a “XR Plug-in Management”. Se abrirá la ventana que se muestra en la Imagen 100. En esta ventana, seleccionaremos la opción “Niantic Lightship SDK + Apple ARKit”, la activaremos. Luego nos dirigiremos a la opción que tiene el icono de android y activamos la opción de “Niantic Lightship SDK + Google ARCore” también marcándola, tal como se observa en la Imagen 101. De esta manera, concluiremos con las configuraciones de este proyecto.

Nota: Se realizaron estas configuraciones para iOS, ya que, de no llevarse a cabo, podrían generar alertas constantes que resultan ser incómodas para el desarrollador.

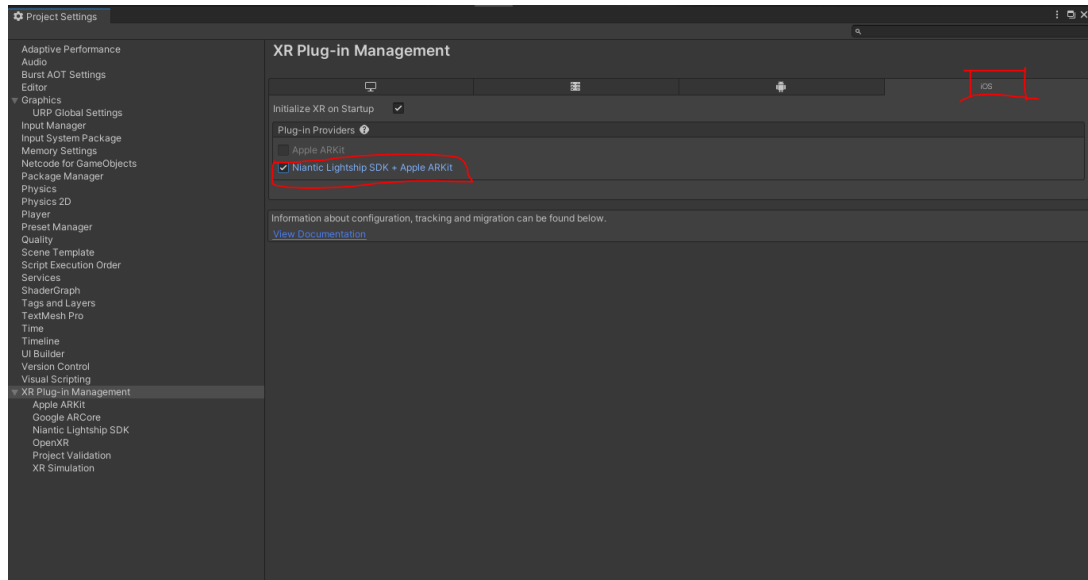


Imagen 100.-Seleccionando las configuraciones del manejador de lightship para iOS.

Fuente: Imagen capturada de software instalado y editada por el autor.

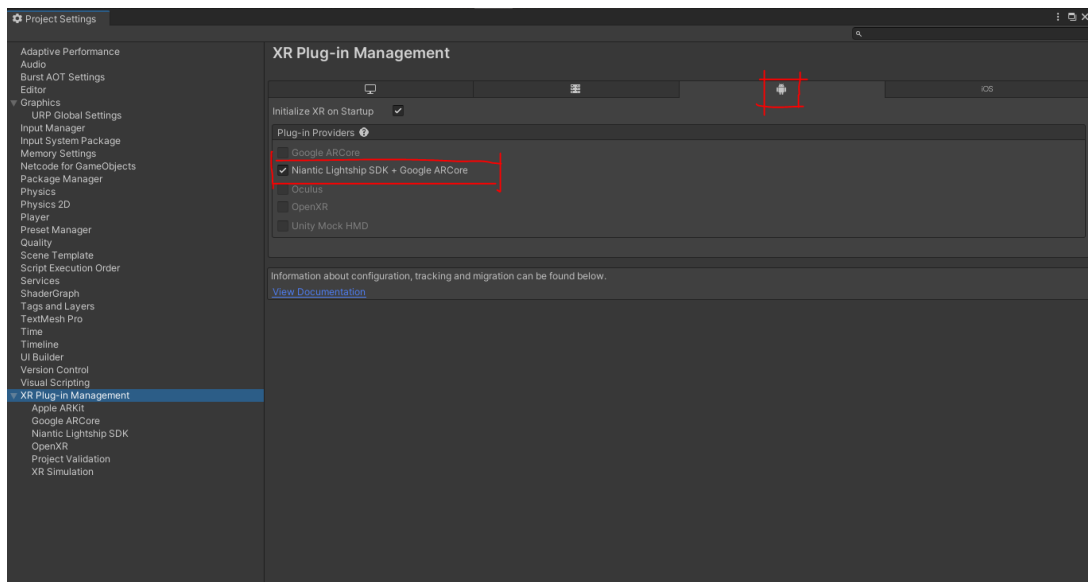


Imagen 101.-Configuraciones del manejador de lightship para android.

Fuente: Imagen capturada de software instalado y editada por el autor.

Con estas configuraciones ya estamos listos para posicionar contenido virtual en la realidad a través de Lightship. Ahora procederemos a mostrar cómo se lleva a cabo este procedimiento.

4.11.6.- Empezando a posicionar contenido.

Ahora que hemos configuramos todo el proyecto, procederemos a mostrar el contenido en la realidad mediante la herramienta que hemos configurado. Comenzaremos por eliminar la “Main Camera” del proyecto y luego agregaremos un componente “XR Origin” y un componente “AR Session”, dando clic derecho en la jerarquía, seleccionaremos la opción “XR”, lo cual nos desplegara un menú de componentes, y allí elegiremos los mencionados anteriormente, como se puede observar en la Imagen 102.

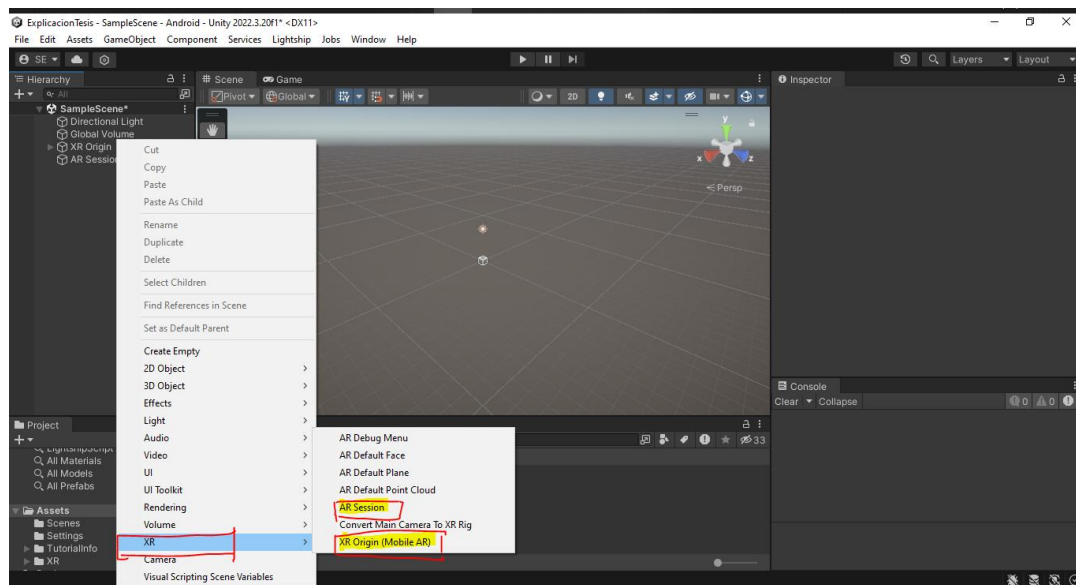


Imagen 102.-Menú para agregar los componentes para colocar el contenido.

Fuente: Imagen capturada de software instalado y editada por el autor.

Ahora seleccionamos el componente “XR Origin” que creamos anteriormente. En la ventana “Inspector”, hacemos clic en la opción “Add Component”, donde buscaremos “AR Location Manager” y lo seleccionamos para incluirlo. Activamos la opción de “Auto Track” y seleccionamos “Add AR Location”. Esto nos llevara a un apartado donde debemos referenciar el wayspot que utilizaremos para colocar el contenido, tal como se muestra en la Imagen 103.

Primero, te explicaremos como descargar un wayspot. Para eso tenemos que regresar a la página de Niantic y seleccionar la opción “Geospatial Browser”. Esto nos mostrara un mapa en donde podrás buscar la zona en la que quieres colocar el contenido (en este ejemplo buscaremos

nuevo Necaxa). Se te mostrarán los waypoints que están activados o propuestos en dicha área, dependiendo de las opciones que selecciones, tal como se muestra en la Imagen 104.

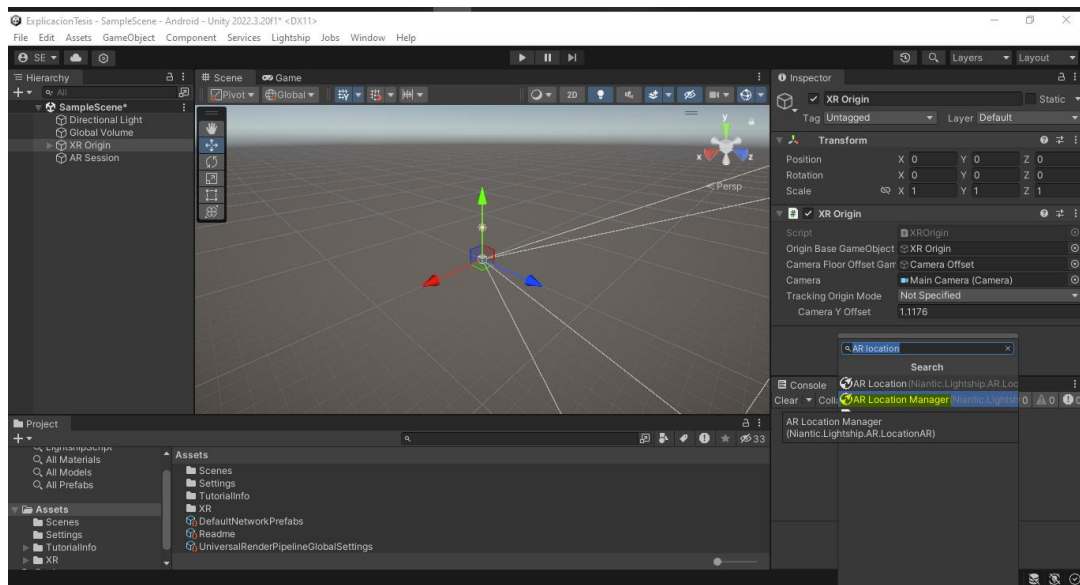


Imagen 103.-Agregando “AR Location Manager” al componente XR Origin.

Fuente: Imagen capturada de software instalado y editada por el autor.

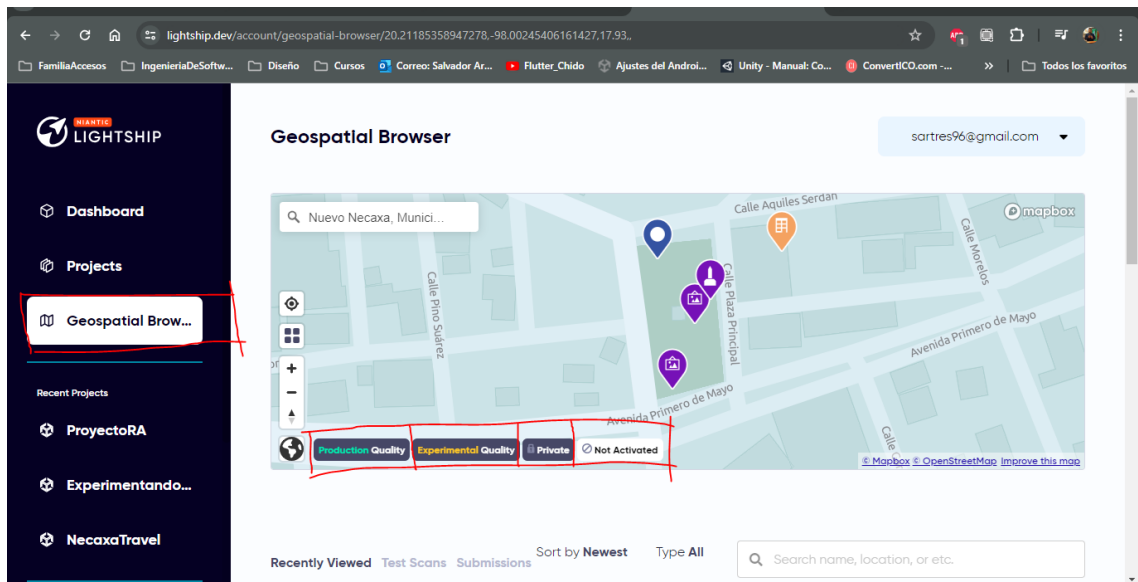


Imagen 104.-Ventana de Geospatial Browser, con la opción “Not Activated” desmarcada, para que solo se muestren los waypoints que se tienen activados en dicha zona.

Fuente: Imagen capturada de <https://lightship.dev/account/geospatial-browser/20.211238,-98.007258,15,>, y editada por el autor.

Por el momento, para fines de la explicación, descargaremos un wayspot que ya está activado. Procederemos a seleccionar uno de estos wayspots; para ello, debemos hacer clic sobre el wayspot que queremos descargar, tal como se muestra en la Imagen 105. Esto nos mostrará un elemento generado por la empresa Niantic y una opción que nos permitirá descargar dicho elemento en un archivo “.zip”. Una vez que se haya descargado el archivo, debemos ir a la carpeta donde lo guardamos y lo arrastramos a Unity. De esta manera, Unity convertirá el archivo descargado en dos archivos, uno de los cuales referenciaremos al componente “AR Location” como se puede observar en la Imagen 106.

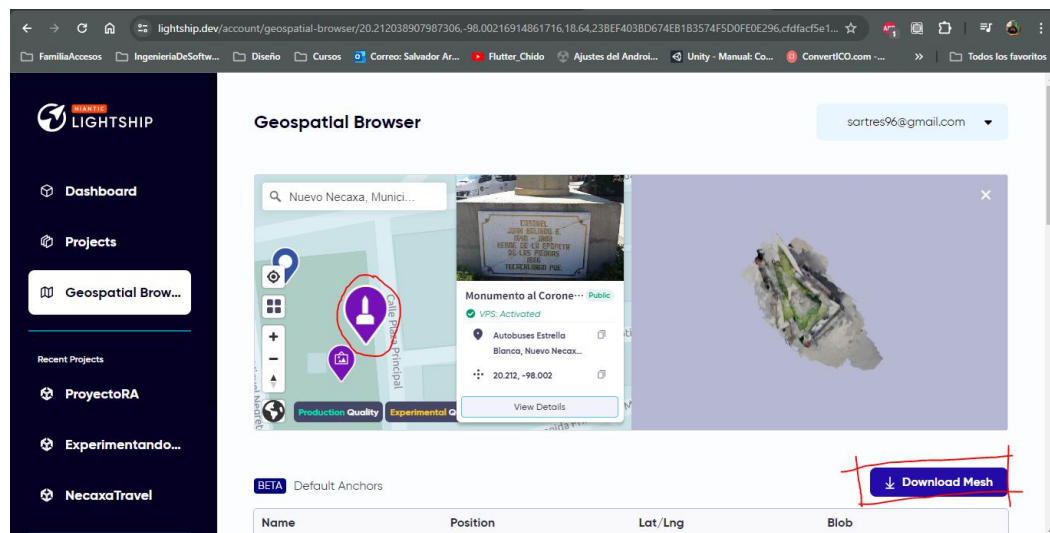


Imagen 105.-Descargando el wayspot para colocar el contenido.

Fuente: Imagen capturada de <https://lightship.dev/account/geospatial-browser/20.211238,-98.007258,15,>, y editada por el autor.

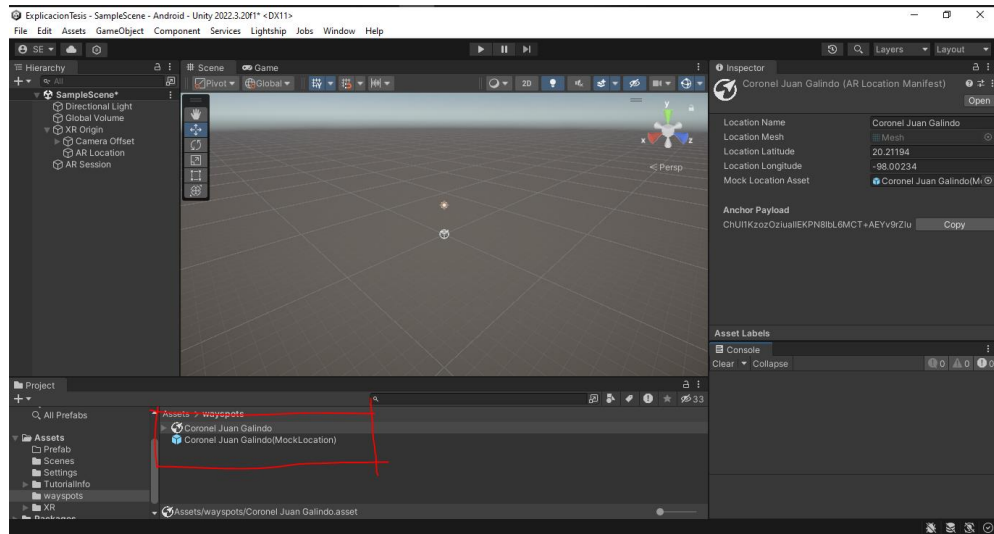


Imagen 106.- Wayspot importado en Unity, convertido en los 2 archivos automáticamente.

Fuente: Imagen capturada de software instalado y editada por el autor.

Ahora procederemos a referenciar el archivo correspondiente al componente “AR Location”, arrastraremos el archivo que se muestra en la Imagen 106 hacia el apartado que se muestra en la Imagen 107. Al realizar esta acción, la herramienta nos mostrara una réplica en 3D de la ubicación donde colocaremos el contenido. Esta representación se utiliza como referencia, ya que no aparece en el resultado final; sin embargo, es de gran ayuda para situar el contenido en la posición deseada.

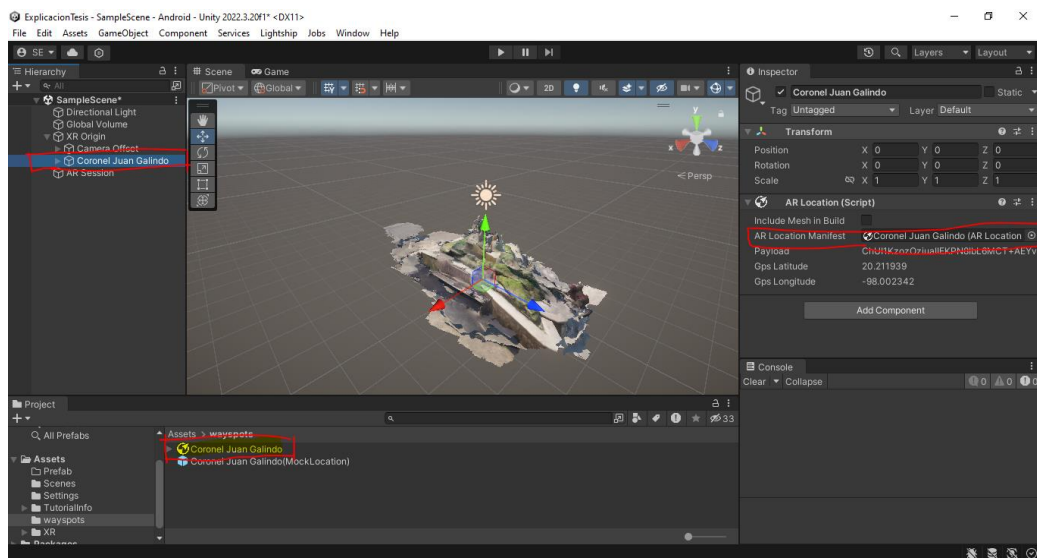


Imagen 107.-Referenciando el waspot al componente que nos permitirá colocar el contenido.

Fuente: Imagen capturada de software instalado y editada por el autor.

Una vez que llegamos a este punto, podremos colocar el contenido que deseamos en la ubicación deseada de manera sencilla, simplemente arrastrando el contenido que queremos ubicar al componente “AR Location”, el cual adopta el nombre del wayspot que hemos descargamos y le asignamos.

Colocamos el contenido que deseamos mostrar como hijos del componente y los situamos en la posición que queremos que aparezcan, como se puede observar en la Imagen 108. Posteriormente, compilamos la aplicación y generamos el instalador para nuestro dispositivo, nos dirigimos al lugar donde se encuentra el wayspot que seleccionamos. Al abrir la aplicación, se mostrará el contenido que hemos incorporado, tal como se puede ver en la Imagen 108. En la aplicación, este contenido puede manipularse de diversas maneras, ya sea a través de código o mediante acciones directamente sobre el mismo contenido.

Como conclusión de este apartado, podemos afirmar que mostrar contenido a través de Lightship es un proceso sencillo y existen diversas alternativas para controlarlo, ya sea mediante código u otros modos. A continuación, se explicará cómo se propone y activa un wayspot.

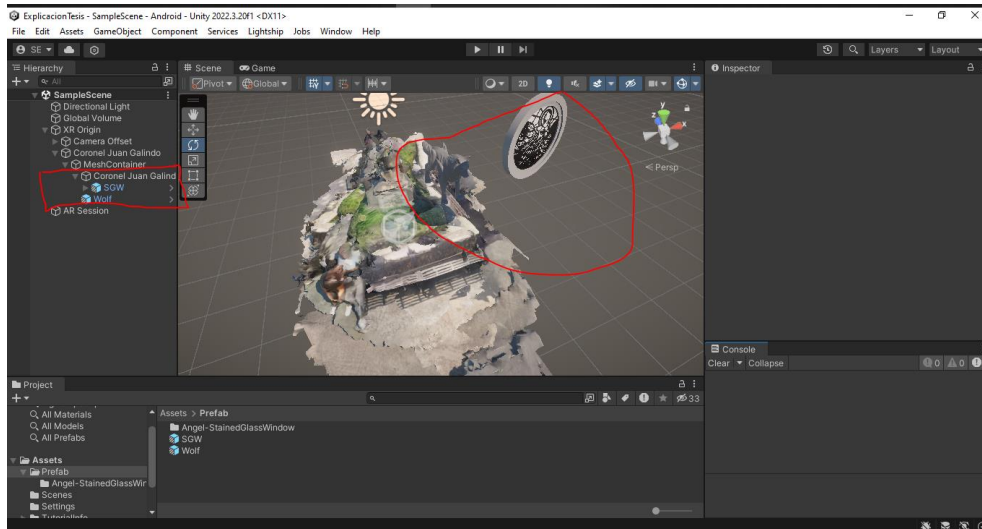


Imagen 108.-Colocando el contenido en la aplicación que creamos.

Fuente: Imagen capturada de software instalado y editada por el autor.



Imagen 109.-Resultados de la aplicación que creamos, mostrando como se ve el contenido que colocamos.

Fuente: Imagen capturada de software instalado.

4.12.- Proponiendo y activando wayspots.

Como se mencionó anteriormente, podemos proponer o activar wayspot, lo cual lograremos mediante la aplicación “wayfarer” que fue desarrollada por la empresa Niantic. Su interfaz es sencilla y nos permite observar los wayspots cercanos a nosotros, ya que funciona con acceso a nuestro GPS. Tendrás acceso a ella a través de la Play Store o, de manera más directa, descargándola desde la página de Lightship, accediendo a ella mediante el código QR que se muestra en la Imagen 110.

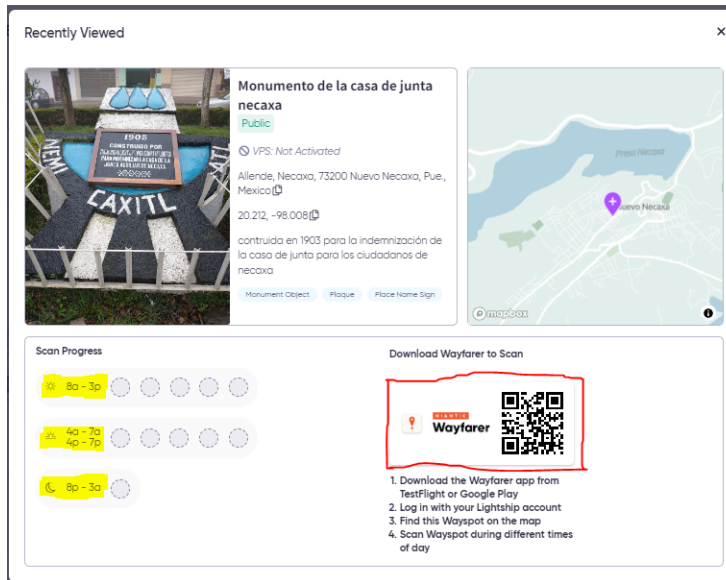


Imagen 110.-Apartado desde donde podemos descargar la aplicación para realizar los escaneos, así como los horarios en los que nos solicitan que realicemos los mismos.

Fuente: Imagen capturada de <https://lightship.dev/account/geospatial-browser/20.21099263316961,-98.007426445872,16.78,,40cee2cbf08d49b1a0afaee53318ae5.16/scan-wayspot>

4.12.1.- Proponiendo un Wayspot.

Para proponer un wayspot, debemos acceder a la aplicación Wayfarer y presionar el botón que se muestra en la Imagen 111, lo cual nos permitirá presentar nuestra propuesta de wayspot.

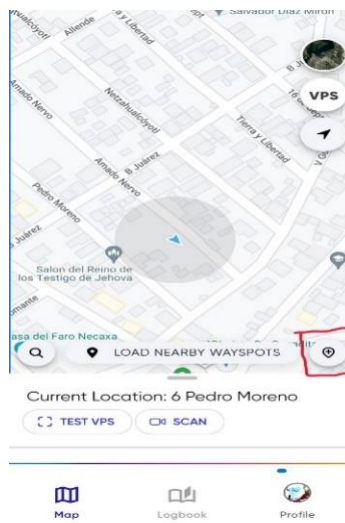


Imagen 111.- Opción que seleccionaremos para iniciar la propuesta del wayspot.

Fuente: Imagen capturada de software instalado y editada por el autor.

Cuando seleccionamos la opción, esta nos llevará a una pantalla que nos permitirá elegir la ubicación del wayspot, como se muestra en la Imagen 112. Aquí podemos seleccionar donde queremos que aparezca el wayspot. Posteriormente, al seleccionar la opción “Next”, nos dirigirá a la sección que se muestra en la Imagen 113, en la cual se explica las condiciones necesarias para que el wayspot sea activado. Estas condiciones o indicaciones vienen en inglés y, en esencia, explican que la ubicación debe ser de fácil acceso para los peatones y que debe tratarse de una plaza cívica o una institución educativa para que sea considerada por la empresa.

Después de leer las indicaciones, presionamos la opción “Next” que te indicara los datos que se te solicitarán, como una foto, nombre, descripción, etc. También se te recordara que no debes proponer residencias privadas, lugares estacionales o no permanentes, así como tampoco lugares que interfieran con los servicios de emergencia. Como podrás ver en la Imagen 114, presionamos clic al botón “Next” para posteriormente seguir leyendo indicaciones, volvemos a presionar clic a “Next” donde empezaremos la nominación del wayspot.

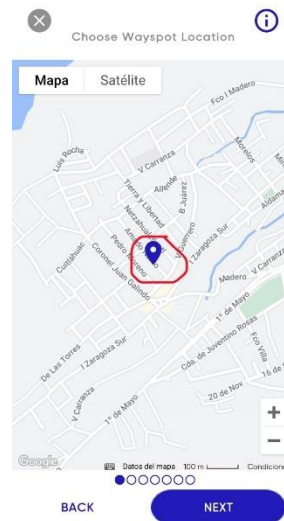


Imagen 112.- Seleccionando la ubicación del wayspot que estamos sugiriendo.

Fuente: Imagen capturada de software instalado y editada por el autor.

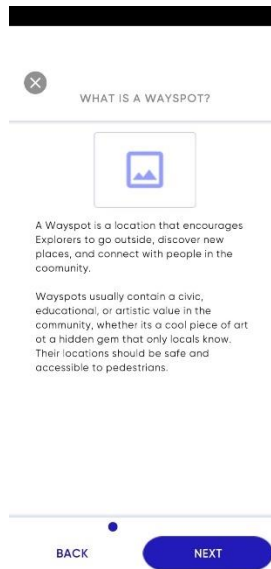


Imagen 113.- Condiciones para que se lleve a cabo la activación de los wayspots.

Fuente: Imagen capturada de software instalado y editada por el autor.

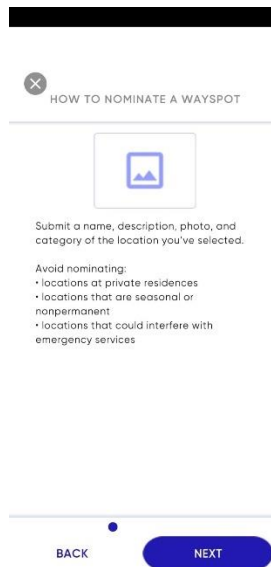


Imagen 114.- Indicaciones y advertencias para la activación.

Fuente: Imagen capturada de software instalado y editada por el autor.

Una vez que leímos las indicaciones, debemos rectificar la ubicación que seleccionamos para colocar el wayspot. Esto lo haremos de manera similar a lo que se muestra en la Imagen 112, pero en esta ocasión seleccionamos la vista de satélite para estar seguros de apuntar a la dirección

correcta, como podrás observar en la Imagen 115. Después de seleccionar el punto, presionamos “Next”, lo que nos llevara a las indicaciones para tomar la foto, como se muestra en la Imagen 116. Seleccionamos “TAKE PHOTO”, tomamos la foto y luego damos “Next”, tal como se ve en la Imagen 117.



Imagen 115.- Reafirmando la ubicación de nuestra propuesta.

Fuente: Imagen capturada de software instalado y editada por el autor.

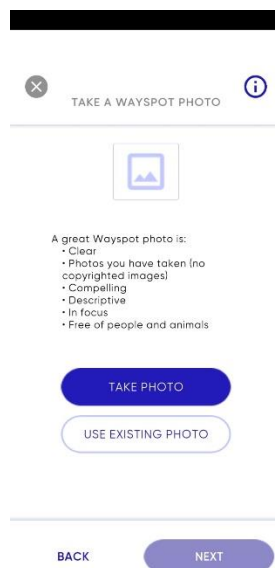


Imagen 116.- Indicaciones para tomar la foto de la propuesta.

Fuente: Imagen capturada de software instalado y editada por el autor.



Imagen 117.- Subiendo la foto de nuestra propuesta.

Fuente: Imagen capturada de software instalado y editada por el autor.

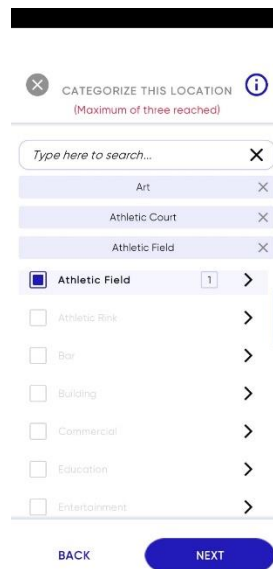
Después de subir la foto, se nos pedirá que tomemos otra de los alrededores de la ubicación. Una vez que la hayamos tomado, debemos presionar “Next” y, a continuación, se nos solicitará la descripción de la ubicación, tal como se muestra en la Imagen 118. Después de proporcionar la descripción y hacer clic en “Next”, tendremos que elegir las categorías en las que entra nuestra nominación, como se puede observar en la Imagen 119. Luego, presionamos “Next”, lo que nos llevara a confirmar los datos que hemos proporcionamos estén correctos. Si algún dato es incorrecto, en esta sección en podrás editar la información errónea. Si la información es correcta como se ve en la Imagen 120, presionamos “Next” y se nos pedirá una breve explicación sobre lo que encontraremos en dicho punto y porque es importante su activación. Con estos datos, los colaboradores de la empresa decidirán si la activación es viable o no. Proporcionamos un texto como puedes ver en la Imagen 121.

Una vez que llegamos al paso anterior y presionamos “Next” se nos mostrara una vista en la que se nos indica que nuestra solicitud ha sido enviada, tal como se muestra en la Imagen 122. Después de eso, podremos dar seguimiento a nuestra solicitud en el apartado “Logbook”, que se encuentra en la parte inferior de la aplicación. allí podremos vere si nuestra solicitud fue aceptada

o rechazada, como se observa en la Imagen 123. Así es como se solicita un wayspot en la aplicación.



*Imagen 118.- Empezamos ingresando la descripción de la ubicación.
Fuente: Imagen capturada de software instalado y editada por el autor.*



*Imagen 119.- Seleccionando la categoría de las propuestas.
Fuente: Imagen capturada de software instalado y editada por el autor.*



Imagen 120.- Confirmando la información que proporcionamos.

Fuente: Imagen capturada de software instalado y editada por el autor.

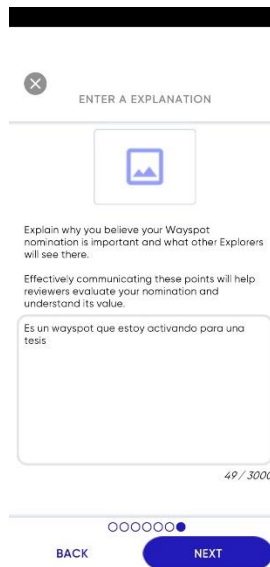


Imagen 121.- Explicando la importancia de la activación.

Fuente: Imagen capturada de software instalado y editada por el autor.

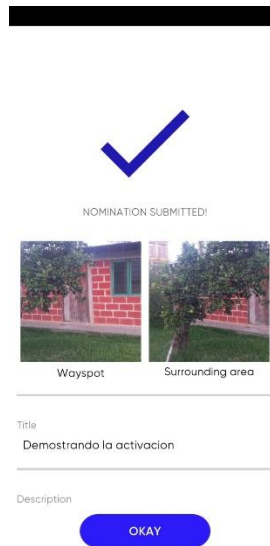


Imagen 122.- Nuestra solicitud enviada, para la activación del wayspoy.

Fuente: Imagen capturada de software instalado y editada por el autor.

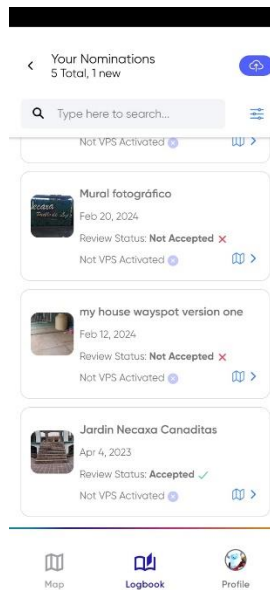


Imagen 123.- Selección en donde podremos ver el estatus de nuestra solicitud.

Fuente: Imagen capturada de software instalado y editada por el autor.

4.12.2.- Activando un Wayspot.

Para realizar activar un wayspot, es necesario tener en cuenta que la aplicación requiere realizar al menos diez escaneos en diferentes horarios. Estos escaneos son fundamentales para crear el modelo 3D que nos servirá de referencia para colocar el contenido en el sitio que representa el wayspot. Es importante visitar el lugar y escanearlo en distintos momentos del día: de 8 a.m. – 3 p.m., 4 p.m. – 7 p.m., 8 p.m. – 3 a.m. y de 4 a.m. – 7 a.m.

Para realizar estos escaneos, debemos dirigirnos al punto que deseamos activar y seleccionamos el punto en la aplicación de wayfarer. Esto nos mostrará la opción de “SCAN” o “VPS Test”. Primero tenemos que mencionar que el vps test se puede utilizar siempre y cuando el waypot este activado, de otra forma la aplicación no te dejará seleccionar esa opción, solo te permitirá seleccionar la opción de scan, como se muestra en la Imagen 124.

Posteriormente a eso, la aplicación abrirá la cámara y deberás apuntarla al punto que deseas escanear. Al realizar el escaneo, se mostrará una especie de malla que rodea los elementos del entorno que estamos mapeando, como podrás observar en la Imagen 125. Es importante tener en cuenta que el escaneo debe durar más de 20 segundos y menos de 5 minutos. Sin embargo, por experiencia personal, si la duración excede los 3 minutos, el tamaño del archivo generado es demasiado grande, lo que dificulta considerablemente el procesamiento del contenido por parte de nuestro equipo.

Una vez que hemos escaneado el sitio deseado, hacemos clic en el botón con un cuadro rojo que puedes ver en la Imagen 125 . Esto nos llevará a una pantalla en la que se muestra el progreso de la generación del archivo 3D que la aplicación crea, tal como se ilustra en la Imagen 126 . Una vez que se ha generado el contenido, la aplicación nos llevará a una pantalla que muestra el escaneo realizado y nos informa sobre la calidad con la que se generó el recurso, como puedes observar en la Imagen 127.

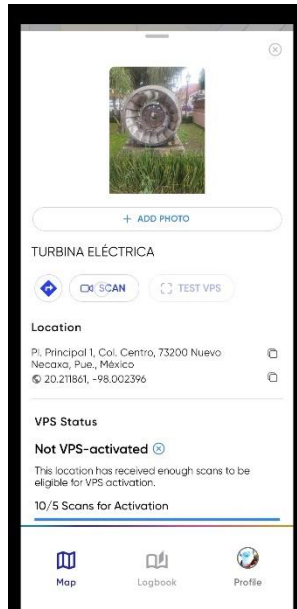


Imagen 124.- Menú que nos permitirá realizar el escaneo del wayspot.

Fuente: Imagen capturada de software instalado.

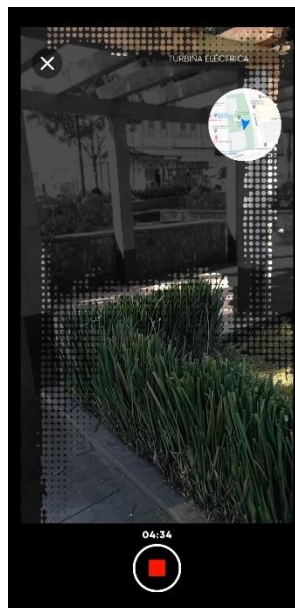


Imagen 125.- Malla que rodea el entorno que estamos mapeando.

Fuente: Imagen capturada de software instalado.

Cuando aceptamos la calidad del material generado, es importante que sea, al menos, de calidad media o buena. Si el escaneo es de mala calidad, la aplicación no nos permitirá cargarlo.

Los factores que influyen a la calidad son la luz, el movimiento lento a la hora de escanear el elemento, entre otros.

Una vez que aceptamos la calidad, se presentaron dos opciones, las cuales son “Upload later” o “Upload now”, que básicamente nos pide si queremos cargar el archivo en ese momento o subirlo más tarde, como se muestra en la Imagen 128. Si se genera el archivo con datos, se recomienda subirlo a través de una conexión Wifi, ya que estos archivos suelen ser pesados y, por lo tanto, el uso de datos móviles podría agotarse rápidamente. Una vez que seleccionamos la opción para subir el archivo, se nos muestra un mensaje que indica que la operación se ha realizado con éxito y que hemos terminado el escaneo. Después de realizar diez escaneos en los diferentes horarios, la aplicación habilita una opción que dice “Activate wayspot” seleccionamos dicha opción y tenemos que esperar a que el wayspot sea activado. Habrá veces que no se te permitirá activar el wayspot debido a que la página tiene un límite de activaciones, una especie de pila (se da como ejemplo esta estructura de datos, aunque realmente no se sabe el cómo funcionan las activaciones, solo se tiene conocimiento del límite que existe) de activaciones, lo que se tiene que hacer es esperar e insistir con la activación en diferentes horarios hasta que la pila se libere.

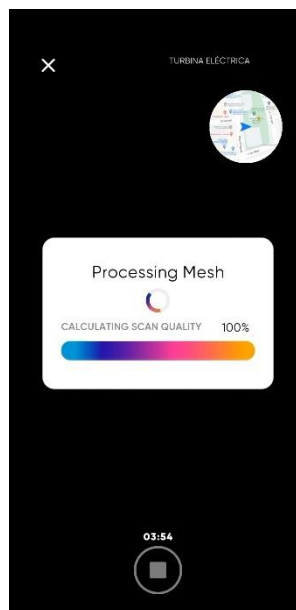


Imagen 126.- Imagen que muestra el proceso de cómo se genera el contenido 3D para generar el archivo 3D que se descarga en la misma página.

Fuente: Imagen capturada de software instalado.



Imagen 127.- Se nos muestra la calidad del contenido que se generó.

Fuente: Imagen capturada de software instalado.

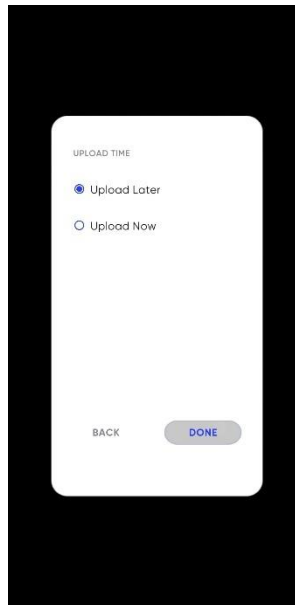


Imagen 128.-Opciones para subir el archivo que se generó.

Fuente: Imagen capturada de software instalado.

De esta manera es como se lleva a cabo la sugerencia, escaneo y activación de wayspot para incorporar el contenido en nuestra aplicación. Aunque el proceso puede resultar un poco tedioso, Lightship es una herramienta extremadamente útil y conveniente para colocar contenido 3D en la

realidad. Se ha comprobado que esta herramienta es fácil de usar y nos permite adaptarnos a ella sin dificultad, ya que no requiere configuraciones complejas y el contenido se presenta de manera accesible.

4.13.- Desarrollando la trivía.

Como se mencionó a lo largo de este trabajo, nuestro objetivo es crear actividades que permitan al usuario acumular puntos, los cuales podrá canjear por producto en los negocios que acepten asociarse a nuestra aplicación. Actualmente, en el desarrollo de esta aplicación, la dinámica que hemos diseñado es un juego de una trivía. En esta sección explicaremos y mostraremos cómo se programó dicha dinámica.

Lo primero que aremos será crear nuestro banco de preguntas para la trivía. Para ello comenzaremos por desarrollar dos scripts: “Option.cs” y “Question.cs”. El primero contendrá el texto de la respuesta y una variable booleana que indicara si es la opción es correcta o no. El segundo, por su parte, incluirá el texto de la pregunta y una lista de opciones, que seran las respuestas que se mostrarán al jugador.

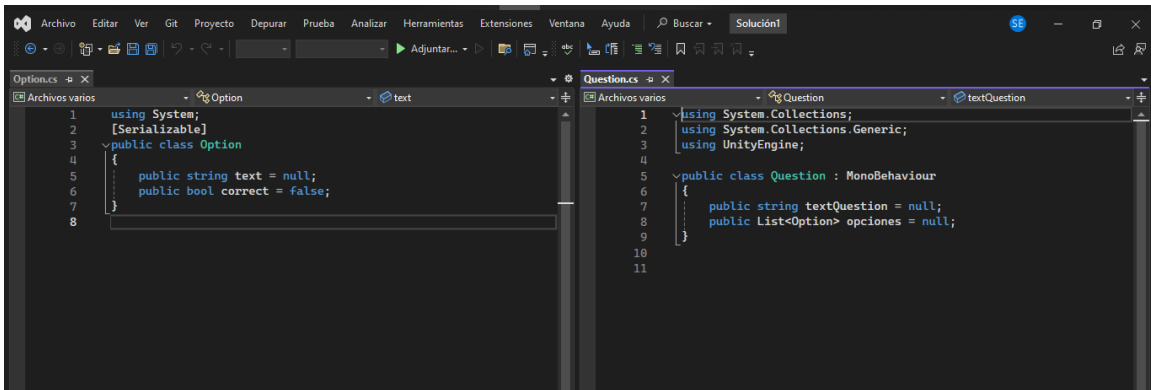
Como puedes observar en la imagen 88, la clase option es “Serializabe” y esto se debe a que, a través de ella, podremos modificar los objetos desde el inspector. Por otro lado, nuestra clase question no tiene ninguna de estas condiciones.

Una vez que hayamos creado estos dos scripts creados, regresaremos a Unity para crear un GameObject. Le añadiremos nuestro script question y podremos agregar las opciones que deseemos, ya sea una o muchas más, tantas como nosotros queramos. Además, podremos seleccionar cuál de las opciones es la correcta.

Una vez que hayamos creado el GameObject, procederemos a generar un prefab a partir de él. Con este prefab, comenzaremos a construir nuestro banco de preguntas, creando tantos prefabs como deseemos. Cada prefab se asemejará al de la Imagen 130. Cada pregunta contendrá un campo de texto, y cada opción incluirá un campo de texto y una variable booleana que nos permitirá indicar si es la respuesta correcta.

Una vez que hayamos creado el prefab, procederemos a desarrollar otro script que almacenará nuestras preguntas. Nombramos este script como “QuizDB.cs”. Este archivo contendrá, básicamente, una lista de las preguntas que deseamos almacenar, para lo cual crearemos

una lista de tipo Question y serán de esta lista las preguntas que el usuario ira respondiendo en la dinámica.



```
Option.cs
1 using System;
2 [Serializable]
3 public class Option
4 {
5     public string text = null;
6     public bool correct = false;
7 }

Question.cs
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4
5 public class Question : MonoBehaviour
6 {
7     public string textQuestion = null;
8     public List<Option> opciones = null;
9 }
10
11
```

Imagen 129.-Scripts “Option.cs” y “Question.cs” los cuales nos permitirán crear y manejar nuestras preguntas.

Fuente: Imagen capturada de software instalado.

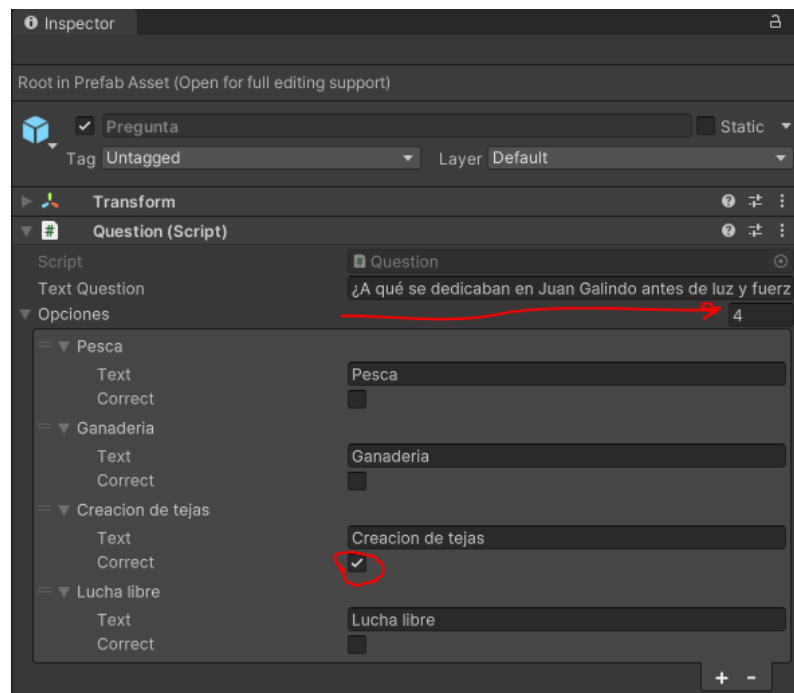
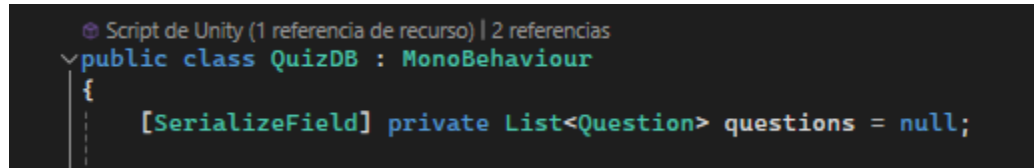


Imagen 130.- Prefab de la pregunta, el cual clonaremos y modificaremos para el banco de preguntas.

Fuente: Imagen capturada de software instalado.

Posteriormente a creación del archivo que se muestra en la Imagen 131, regresamos a Unity y creamos otro GameObject al que le asignaremos el script QuizDB. Este GameObject sera el encargado de contener nuestro banco de preguntas, las cuales serán los prefabs que creamos

anteriormente. Cada prefab representará una pregunta que podrá mostrarse en el juego y el objeto que contendrá el banco de preguntas deberá tener una apariencia similar a la del objeto de la Imagen 131.

A screenshot of the Unity code editor showing a C# script named QuizDB.cs. The script is a public class that inherits from MonoBehaviour. It contains a single line of code: a private List<Question> named 'questions' is declared and initialized to null. The 'SerializeField' attribute is applied to this field. The script has 2 references and is associated with 1 resource.

```
Script de Unity (1 referencia de recurso) | 2 referencias
public class QuizDB : MonoBehaviour
{
    [SerializeField] private List<Question> questions = null;
}
```

Imagen 131.- “QuizDB.cs” donde crearemos la lista de preguntas para el banco.

Fuente: Imagen capturada de software instalado.

También crearemos tres funciones que nos ayudaran más adelante. La primera nos permitirá obtener una pregunta de manera aleatoria; la segunda nos permitirá volver a llenar el arreglo principal (en caso de que este se quede vacío) con un respaldo que creamos previamente; y una función Awake que es la que respaldara el arreglo de preguntas para cuando el arreglo de respaldo quede vacío.

En el método de “GetRandoom” le agregaremos una variable booleana que nos ayudara a evitar que las preguntas se repitan en la trivía. La primera condición que veremos en la función es un if que nos permitirá realizar un respaldo cuando el arreglo de preguntas se quede vacío. En nuestro caso esto no debería ocurrir, ya que contamos con quince preguntas y al usuario solo lo cuestionamos diez veces. Sin embargo, incluiremos estas condiciones por razones de seguridad y para que, si alguien desee repetir nuestro trabajo con menos preguntas, no tenga problemas.

Posteriormente a ese if seleccionaremos la pregunta que enviaremos generando un número aleatorio entre el cero y la cantidad de preguntas contenidas en el arreglo (esta cantidad la conseguimos con la instrucción “questions.Count”, ya que “Count” es una función que nos regresa el tamaño del arreglo). Luego preguntamos si deseamos remover la pregunta del banco mediante un if. Después guardamos la pregunta, la eliminamos del banco y retornamos la pregunta que salió gracias al número aleatorio generado. El código resultante se muestra en la Imagen 132.

Recuerda que cada elemento del arreglo es un prefab que referenciamos a este GameObject. Esto se realiza arrastrando los objetos al arreglo para que sean asignados. En la Imagen 133, puedes observar que hay quince preguntas. Puedes agregar las que desees para que tu banco este bien alimentado y para evitar que se repitan en la trivía.

```
9
10 private List<Question> respaldo = null;
11
12 private void Awake()
13 {
14     respaldo = questions.ToList();
15 }
16 public Question GetRandom(bool remove = true)
17 {
18     if (questions.Count == 0)
19     {
20         YaNoHayPreguntas();
21     }
22
23     int index = Random.Range(0, questions.Count);
24
25     if (!remove)
26     {
27         return questions[index];
28     }
29     Question q = questions[index];
30     questions.RemoveAt(index);
31
32     return q;
33 }
34
35
36 private void YaNoHayPreguntas()
37 {
38     questions = respaldo.ToList();
39 }
40 }
```

Imagen 132.- Funciones para que se envíen las preguntas de manera aleatoria.

Fuente: Imagen capturada de software instalado.

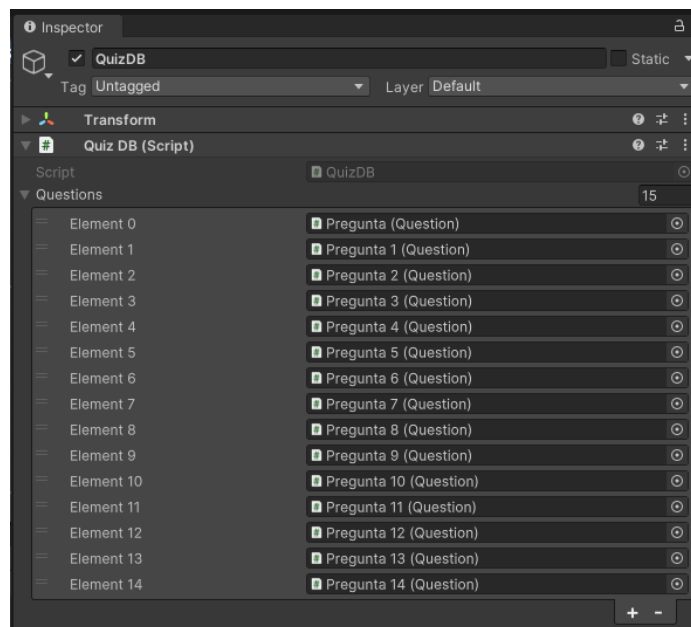


Imagen 133.- GameObject que funcionara como nuestro banco de preguntas.

Fuente: Imagen capturada de software instalado.

Cuando tengamos estos elementos, desarrollaremos un script que nos ayudará a controlar el comportamiento de las respuestas a las preguntas. Este script lo denominaremos

“OptionButtons.cs”. Vamos a incluir “[RequireComponent(typeof(Button))]

 y “[RequireComponent(typeof(Image))] con el fin de que, al implementar este script, se exija la presencia de los componentes necesarios para su funcionamiento. Solicitaremos tanto el botón como la imagen. Además, crearemos un método público que utilizaremos como constructor, el cual nos permitirá inicializar todos los elementos que empleemos. Este mismo constructor recibirá la opción que vamos a mostrar, es decir, recibirá un dato de tipo “Option” y un callback que determinará la acción a seguir cuando seleccionemos una opción.

A continuación, en el mismo constructor, vamos a crear un action listener en el que invocaremos el callback, tal como se muestra en la Imagen 134. Posteriormente implementaremos un método get y un método set (como se observa en la línea 17) para almacenar la opción que vamos a crear. Debemos asegurarnos de que el botón este deshabilitado y guardar el color original de la imagen. Cabe mencionar que también necesitamos el texto de la opción.

Creamos un método Awake para obtener los componentes necesarios. El texto lo buscaremos mediante la función que se encuentra en la línea 23. Por último, crearemos un método para asignar el color a la opción; si la respuesta es correcta, la opción se coloreará de verde, y si es incorrecta se coloreará de rojo, tal como se muestra en el método “SetColor”, que puedes ver en la Imagen 134.

Cuando creamos estos archivos, desarrollaremos la interfaz gráfica. Primero, crearemos el canvas y, dentro de él, generaremos un objeto vacío al que denominaremos "QuizUi" y lo modificamos para que se adapte a todo el canvas. Luego, crearemos otro objeto vacío, al que llamaremos pregunta, que contendrá el texto de la pregunta. Adaptaremos su tamaño para que abarque el área deseada en la pantalla y ajustaremos las pestañas que lo hacen responsivo, convirtiéndolo en hijo del primer objeto. A este objeto le añadiremos el componente "TextMeshPro" y configuraremos el texto con el color y el tamaño que deseemos, tal como se muestra en la Imagen 135. Recuerda que estas configuraciones se realizan y visualizan en el inspector.

Volvemos a crear otro GameObject que contendrá todas las opciones de la pregunta, es decir, las respuestas correspondientes. A este componente lo llamaremos "Respuestas" y debe ser hijo del primer objeto de juego que denominamos "QuizUi".

```

8 [RequireComponent(typeof(Button))]
9 [RequireComponent(typeof(Image))]
10 public class OptionButtons : MonoBehaviour
11 {
12     private TMP_Text preguntaTexto = null;
13     private Button miButton = null;
14     private Image miImage = null;
15     private Color miColor = Color.black;
16
17     4 referencias
18     public Option Option { get; set; }
19
20     @ Mensaje de Unity | 0 referencias
21     private void Awake()
22     {
23         miButton = GetComponent<Button>();
24         miImage = GetComponent<Image>();
25         preguntaTexto = transform.GetChild(0).GetComponent<TMP_Text>();
26
27         miColor = miImage.color;
28     }
29
30     1 referencia
31     public void Constructor(Option op, Action<OptionButtons> call)
32     {
33         preguntaTexto.text = op.text;
34
35         miButton.onClick.RemoveAllListeners();
36         miButton.enabled = true;
37         miImage.color = miColor;
38
39         Option = op;
40         miButton.onClick.AddListener(delegate
41         {
42             call(this);
43         });
44     }
45
46     3 referencias
47     public void SetColor(Color c)
48     {
49         miButton.enabled = false;
50         miImage.color = c;
51     }
52 }

```

Imagen 134.- Archivo "OptionButtons.cs".

Fuente: Imagen capturada de software instalado.

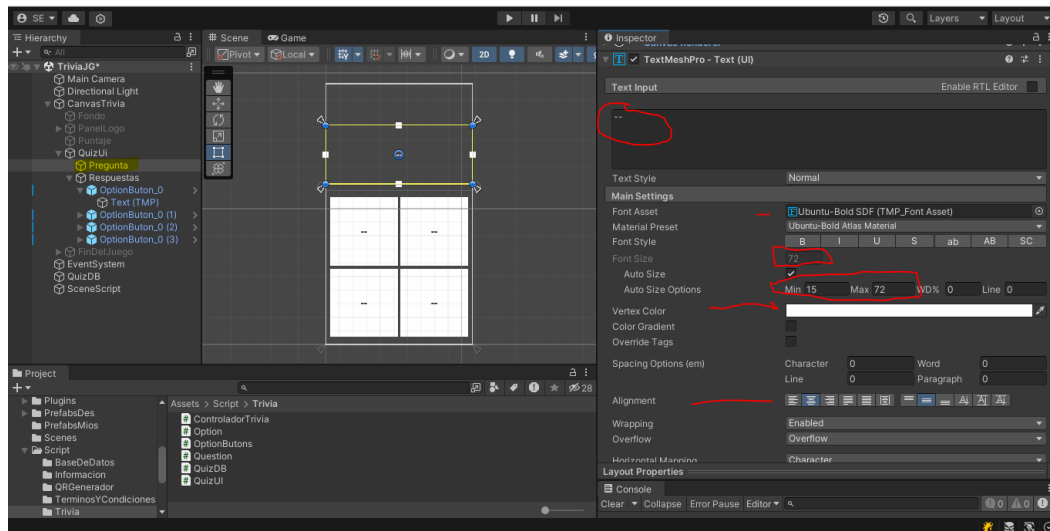


Imagen 135.- Primeras configuraciones de la interfaz de la trivia.

Fuente: Imagen capturada de software instalado.

Una vez que se haya creado el GameObject denominado "Respuestas" y también cuando ya le hayamos añadido el componente llamado "grid layout", este componente nos permitira organizar los hijos de este objeto. Sin embargo, incluiremos estas condiciones por razones de

seguridad y para que, si alguien desee repetir nuestro trabajo con menos preguntas, no tenga problemas. Además, podremos elegir el espacio que deseamos que exista entre ellos, el tamaño de los hijos, el número de columnas que queremos que tenga el objeto padre, entre otros aspectos.

Dentro de ese elemento, creamos otro GameObject al que nombraremos “OptionButton_0” y al que le asignaremos el script “OptionButtons” que creamos anteriormente. Como recordaras, en ese script buscamos que el hijo de este componente tenga un hijo de tipo texto. Por ello creamos un componente “TextMeshPro” como hijo de este objeto y lo ajustamos para que ocupe todo el espacio del botón. Una vez realizado esto, arrastramos el objeto a una carpeta para convertirlo en prefab, en caso de que más adelante deseemos modificarlo. Después de lograr esto, determinamos el tamaño que queremos para los botones. Clonamos el “OptionButton_0” tres veces más y configuramos el elemento “grid layout” para que los botones queden dispuestos según nuestras preferencias. Estos elementos en nuestro trabajo quedaron configurados como se observa en la Imagen 136. Después de terminar la interfaz gráfica, debemos crear un guion que nos ayudará a controlar la interfaz gráfica de las preguntas. Lo denominaremos “QuizUI.cs” y comenzaremos su codificación.

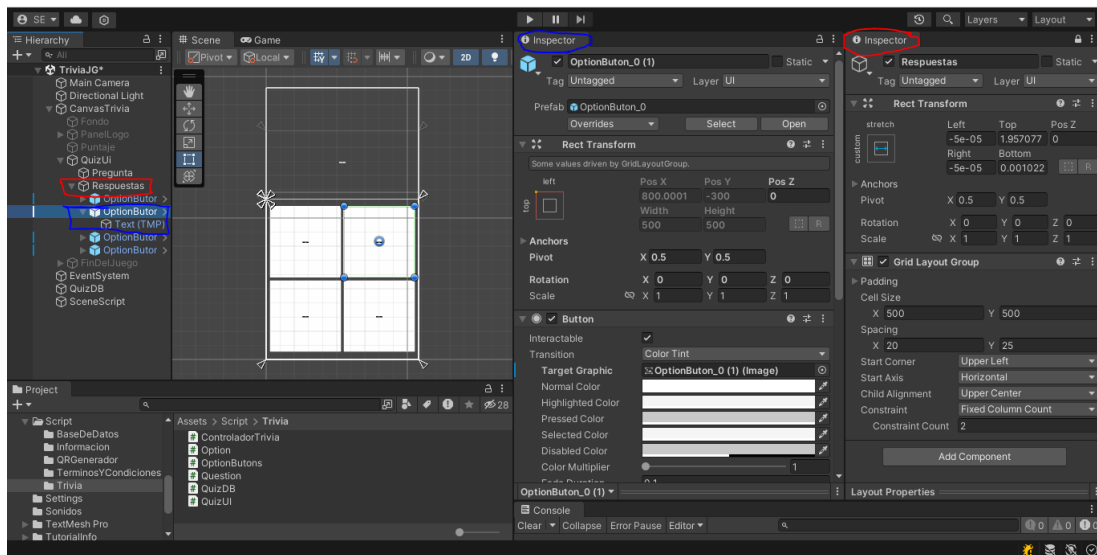


Imagen 136.- Configuraciones de los elementos de preguntas y respuestas, encerrando el nombre del elemento del mismo color que si respectiva ventana inspector.

Fuente: Imagen capturada de software instalado.

Los componentes que solicitaremos en este script son un texto para formular la pregunta y una lista de opciones, que representaran las respuestas. Todo ello llevará el prefijo

“[SerializeField]” para poder editar los elementos desde el inspector. Crearemos una función que denominaremos “Constructor”, la cual recibirá una variable de tipo “Question” y un callback. Básicamente, un callback es un método que permite pasar una función que devuelve el valor que se coloque entre los símbolos mayor que y menor que.

```
1  using System;
2  using System.Collections;
3  using System.Collections.Generic;
4  using TMPro;
5  using UnityEngine;
6
7  public class QuizUI : MonoBehaviour
8  {
9      [SerializeField] private TMP_Text pregunta = null;
10     [SerializeField] private List<OptionButtons> m_buttonList = null;
11
12
13     1 referencia
14     public void Construc(Question q, Action<OptionButtons> call)
15     {
16         pregunta.text = q.textQuestion;
17
18         for (int i = 0; i < m_buttonList.Count; i++)
19         {
20             m_buttonList[i].Constructor(q.opciones[i], call);
21         }
22     }
23
24 }
```

Imagen 137.- Script “QuizUI.cs” que se encargara del manejo de la interfaz de la trivia.

Fuente: Imagen capturada de software instalado.

Lo primero que aremos es colocar el texto de la pregunta que queremos formular. Posteriormente, con la ayuda de un ciclo for, inicializaremos los botones que son las respuestas a las preguntas. Al volver a invocar el mismo método, simplemente construimos la pregunta y cada opción con las respuestas correspondientes. Básicamente, esto es lo que se desarrolla en el código que puedes ver en Imagen 137.

Ahora crearemos el controlador principal de la trivia. Para ello, crearemos un script que denominaremos “ControladorTrivia”, en el cual también solicitamos un componente de audio mediante la instrucción “[RequireComponent(typeof(AudioSource))]”. A continuación comenzaremos a solicitar todos los elementos que necesitemos; estas referencias se obtienen a través de las instrucciones que se muestran en la Imagen 138.

Después crearemos un método start, en este caso utilizaremos el método start y no el awake, debido a que en las clases anteriores queremos tener todo referenciado antes de que el método comience a ejecutarse.

```
1 using System.Collections;
2 using TMPro;
3 using UnityEngine;
4
5 [RequireComponent(typeof(AudioSource))]
6 public class ControladorTrivia : MonoBehaviour
7 {
8     [SerializeField] GameObject panelFinal;
9     [SerializeField] private AudioClip m_correctSound = null;
10    [SerializeField] private AudioClip m_failSound = null;
11    [SerializeField] private Color m_colorCorrect = Color.white;
12    [SerializeField] private Color m_colorFail = Color.white;
13    [SerializeField] private float espera = 0.0f;
14    [SerializeField] private int preguntas;
15    [SerializeField] private TMP_Text puntaje;
16    [SerializeField] private ActualizarPuntos nuePuntos;
17
18
19    public int puntajeValor = 0;
20
21    private QuizDB m_quizDB = null;
22    private QuizUI m_quizUI = null;
23    private AudioSource m_audioSource = null;
```

Imagen 138.- Elementos con los que trabajara el controlador de la trivia.

Fuente: Imagen capturada de software instalado.

En el método start, obtendremos la referencia al banco de preguntas y a nuestra interfaz gráfica mediante de las instrucciones de las líneas 27 y 28, que podrás observar en la Imagen 139. A continuación, llamaremos a la función “NextQuestion” para iniciar el juego con una pregunta. Posteriormente, crearemos la función “Update”, la cual nos ayudara a visualizar el puntaje que va obteniendo el jugador.

Posteriormente, crearemos la función “NextQuestion”, que básicamente invoca el constructor de la clase "QuizUI" y envía el callback que nos devuelve la función "GiveAnswer", la cual inicia una corrutina. Esta corrutina determina el comportamiento de las dinámicas, dependiendo de la respuesta seleccionada por el usuario, ya sea correcta o incorrecta. Además, también establece cuándo detener el juego, ya que en este caso solo queremos realizar diez preguntas al usuario. Recuerda inicializar el contador de preguntas en cero, ya que, si no lo haces, no podrás llevar el control de las preguntas y se realizarán más de diez.

```

25  Mensaje de Unity | 0 referencias
26  private void Start()
27  {
28      m_quizDB = GameObject.FindObjectOfType<QuizDB>();
29      m_quizUI = GameObject.FindObjectOfType<QuizUI>();
30      m_audioSource = GetComponent<AudioSource>();
31      preguntas = 0;
32      NextQuestion();
33  }
34  Mensaje de Unity | 0 referencias
35  private void Update()
36  {
37      puntaje.text = puntajeValor.ToString();
38  }
39  2 referencias
40  private void NextQuestion()
41  {
42      m_quizUI.Construc(m_quizDB.GetRandom(), GiveAnswer);
43  }
44  1 referencia
45  private void GiveAnswer(OptionButons opSelec)
46  {
47      StartCoroutine(GiveAnswerRoutine(opSelec));
48  }

```

Imagen 139.- Funciones: Start, Update, NextQuestion y GiveAnswer del controlador de trivia.

Fuente: Imagen capturada de software instalado.

```

47  1 referencia
48  private IEnumerator GiveAnswerRoutine(OptionButons op)
49  {
50      if (m_audioSource.isPlaying)
51      {
52          m_audioSource.Stop();
53      }
54      preguntas++;
55
56      if (op.Option.correct)
57      {
58          m_audioSource.clip = m_correctSound;
59          op.SetColor(m_colorCorrect);
60          puntajeValor += 50;
61      }
62      else
63      {
64          m_audioSource.clip = m_failSound;
65          op.SetColor(m_colorFail);
66          puntajeValor -= 5;
67      }
68
69      m_audioSource.Play();
70
71      if (preguntas == 10)
72      {
73          panelFinal.SetActive(true);
74          nuePuntos.SumaPuntos(puntajeValor);
75      }
76
77      yield return new WaitForSeconds(espera);
78
79      NextQuestion();
80  }
81  }

```

Imagen 140.- Corrutina que determinara el comportamiento de la trivia.

Fuente: Imagen capturada de software instalado.

Esta corrutina recibirá el option button como parámetro. Lo primero que haremos es verificar si se está reproduciendo un sonido; si es así, lo detenemos para posteriormente reproducir otro. Incrementamos el contador de preguntas y luego realizaremos un if para determinar si la opción seleccionada fue la correcta. Esta condición se encuentra desde de la línea 56 hasta la línea 67, las cuales puedes ver en la Imagen 140. Si la respuesta es correcta, reproduciremos un sonido similar a un destello y el botón se colorea de verde. Además, sumaremos 50 al puntaje del jugador. Si la respuesta seleccionada no es correcta, reproduciremos un sonido similar al de corno y el botón se coloreará de rojo, además de restar 5 al puntaje del jugador.

Después de que reproducimos el audio, procederemos a verificar si hemos realizados las diez preguntas al jugador. Si ya las hemos hecho, se actualizarán los puntos que el jugador tenía previamente en la base de datos, sumando los puntos obtenidos en la trivía que acaba de responder. Se mostrará un panel con la suma de los puntos obtenidos y los puntos previamente acumulados. Este panel contara con un botón que te regresara al menú principal.

Para finalizar, agregaremos una imagen de fondo, nuestro logotipo y un campo de texto en el que mostraremos el puntaje, de manera que la interfaz quede como se observa en la Imagen 141.



Imagen 141.- Interfaz de la trivía junto con el panel de fin de la trivía.

Fuente: Imagen capturada de software instalado.

Por último, mencionamos que los puntos se actualizan mediante un archivo PHP que ejecuta una consulta de actualización, así como un script que realiza la UnityWebRequest de manera sencilla. Como se mencionó anteriormente, para este proceso el usuario debe tener una sesión iniciada correctamente, lo que garantiza que su información ya existe en nuestra base de datos. Por lo tanto, no es necesario verificar los datos para insertar los nuevos puntos, ya que, gracias a los PlayerPrefs, estos se han guardado desde que el usuario inició sesión.

4.14.- Generación el APK para la instalación en Android.

En esta última sección de este capítulo, le mostraremos al lector cómo generar el archivo de instalación para dispositivos Android, el cual resulta en un archivo de extensión “.APK”, este archivo será el que compartiremos para que se puedan realizar las pruebas del recorrido.

Primero, debemos dirigirnos a la ventana “Build Settings” seleccionando la opción “File” en el menú superior o presionando las teclas “Ctrl + Shift + B”, esto nos permitirá trasladarnos a la ventana que se muestra en la Imagen 142.

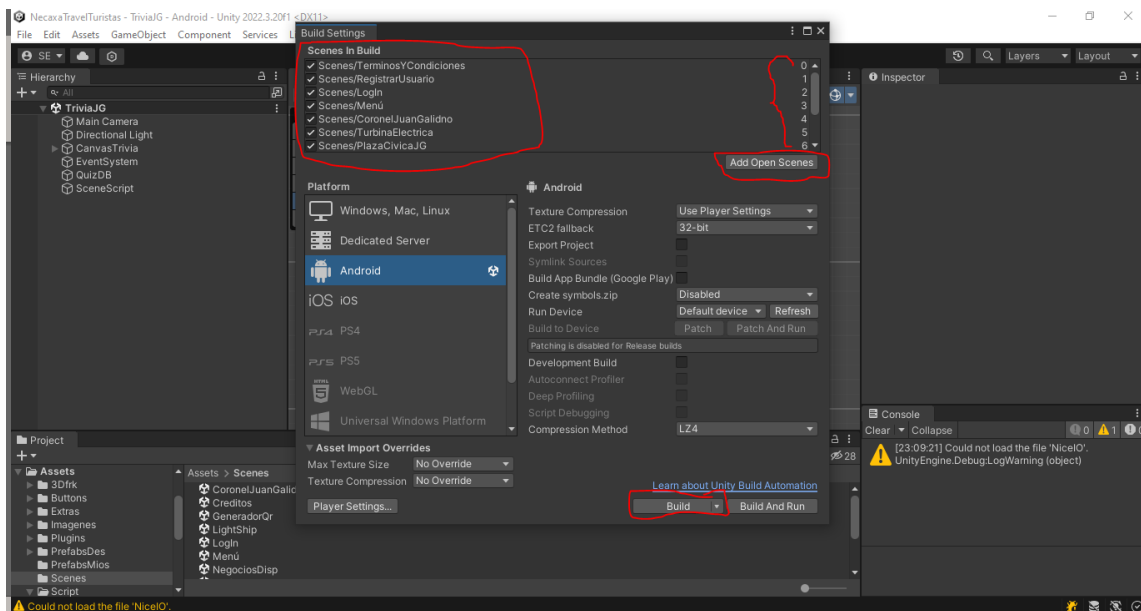


Imagen 142.-Ventana de “Build Settings” desde donde generaremos el APK.

Fuente: Imagen capturada de software instalado y editada por el autor.

Como puedes observar en el apartado de “Scenes In Build”, aquí se encuentran todas las escenas que componen la aplicación. Si generas el archivo APK y alguna escena no está incluida en esa sección, aparecerá en tu aplicación. Para que una escena se agregue a esa sección, debes

abrirlo en el editor, acceder a las build settings y seleccionar la opción “Add Open Scenes”. O bien puedes seleccionar todas las escenas en la carpeta en donde estén guardadas en la estructura del proyecto y las arrastras hasta este apartado; de este modo, se agregarán en un orden automático generado por unity.

Una vez que están contenidas en el apartado mencionado anteriormente, podremos organizarlas una encima de la otra o debajo, dependiendo lo que deseemos. Esto permitirá que el número que aparece en el extremo opuesto a los nombres de las escenas se modifique. El propósito de este número es facilitar el cambio de escena; aunque puedes ingresar el nombre de la escena a la deseas trasladarte, eso no es fundamental. Si conoces el orden de compilación, puedes enviar directamente el número o la posición en la que se encuentra, evitando así la necesidad de poner el nombre de la escena. Aunque esta opción es útil, su uso dependerá mucho del programador. En el caso de este proyecto, como no sabíamos el orden en el que quedarían las escenas, dado que se desarrollaron por partes, optamos por utilizar el nombre de la escena.

Una vez que tengas las escenas asignadas, selecciona la opción “Build”, esto abrirá una ventana en la que tendrás que dar una ruta en la que se guardará tu archivo APK. Una vez generado el archivo, podrás compartirlo al dispositivo en el que instalaras la aplicación. Si observas detenidamente la Imagen 142, notarás que también existe otra opción que dice “Build And Run”. Para utilizar esta opción es necesario que tu dispositivo móvil esté conectado por medio de cable USB, lo que permitirá realizar la instalación directamente desde Unity. Después de presionar build y diste la dirección donde se guardará el archivo, se iniciará una barra de estado en la que podrás ver el proceso de compilación de tu aplicación.

Si todo salió bien, el archivo APK aparecerá en la carpeta que seleccionaste para su almacenamiento. En caso contrario, Unity te notificara que ha ocurrido un problema y en la consola se mostraran los errores que impidieron la generación de tu archivo. Si no corriges dichos errores, no podrás generar tu APK. Por último, lo que debes hacer es transferir el archivo al dispositivo en el que deseas instalar tu aplicación; este realizará la instalación de manera automática, solicitando los permisos que requiera la aplicación para funcionar correctamente.

Con esto, podremos comenzar a realizar las pruebas y recorridos para los experimentos de nuestra aplicación. Ten en cuenta que el dispositivo en el que instales la aplicación te solicitará que otorgues ciertos permisos por motivos de seguridad. En todos los casos, el mismo dispositivo

te guiara o indicara cómo realizar las configuraciones necesarias para instalarse correctamente. Además, que las configuraciones que realizamos en el apartado 4.11.5 de este proyecto serán las responsables de impedir la instalación si no se cumplen dichas configuraciones.

Existen condiciones que no se verifican durante la compilación, ya que no afectan o no son consideradas por la herramienta. Sin embargo, pueden influir en el funcionamiento de tu aplicación. En nuestro caso, no se presentó ningún problema. Si al intentar replicar este trabajo te encuentras con alguna dificultad, te recomendamos que te comuniques directamente con Niantic, ya que suelen atender rápidamente y brindan una buena asistencia.

Capítulo 5. Resultados y conclusiones.

Como se mencionó en el capítulo 3, se diseñó un experimento que se llevó a cabo con 10 personas, a las cuales se les aplicó la encuesta SUS, detallada en la sección 3.3 del capítulo mencionado. Además, se les aplicó una encuesta elaborada por nosotros para evaluar qué tan innovadora les resultaba la aplicación. También se solicitó un recuadro de recomendaciones generales, así como las se pidieron al aplicar la encuesta SUS y una muestra del recorrido que realizamos para las pruebas.

5.1.- Recorrido realizado.

Como se dijo antes, realizamos un experimento para evaluar la usabilidad de nuestro proyecto, realizado en el municipio de Juan Galindo. A continuación, presentaremos los resultados de la aplicación utilizada en el experimento, aclarando que en el capítulo 4 mostramos diversas interfaces de cada sección, en las cuales se detalló el desarrollo. Por esta razón, en esta sección solo mostraremos el contenido que visualizaron los usuarios que interactuaron con la aplicación.

El recorrido fue en el jardín central del municipio de Juan Galindo y consta de cuatro escenas, que representan cuatro puntos relevantes que usamos para mostrar el contenido. El primer punto de inicio del recorrido es el monumento al coronel Juan Galindo, donde comenzamos con un modelo 3D y un marco con imágenes que se irán presentando conforme se proporciona la información, además de las flechas en 3D.

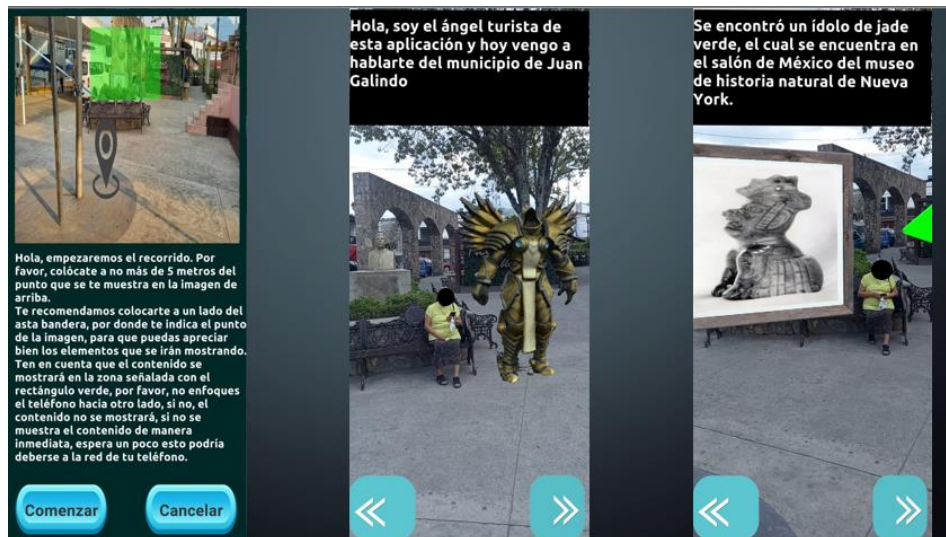


Imagen 143.- Primeras vistas del contenido que se muestra en el monumento al coronel Juan Galindo.

Fuente: Imagen capturada de software instalado.



Imagen 144.- Segunda parte del contenido que se mostró en el monumento al coronel Juan Galindo.

Fuente: Imagen capturada de software instalado.

Como podrás observar en la Imagen 143 y la Imagen 144, el contenido que se presenta en el monumento al coronel Juan Galindo incluye las imágenes en el cuadro y las flechas que indican tanto el monumento como el siguiente punto de interés. Además, se incluye un modelo 3D de un ángel que se descargaba de internet bajo licencia de uso libre.



Imagen 145.- Segunda escena, monumento de la turbina eléctrica.

Fuente: Imagen capturada de software instalado.

Si observas la Imagen 145 y la Imagen 146, la segunda escena del recorrido se centra en el monumento de la turbina ubicado en el jardín mencionado anteriormente. Continuamos

presentando imágenes y flechas que respaldan la información que se muestra a través del texto en el panel superior. Es importante señalar que el texto y el contenido es controlado por los botones situados en la parte inferior del canvas.



Imagen 146.- Segunda muestra del contenido de la escena de la turbina eléctrica y la placa por los 100 años de lyfc.

Fuente: Imagen capturada de software instalado.

La tercera escena se centra en la Plaza Cívica Armada de México, la cual se muestra en la Imagen 147 y en la Imagen 148.

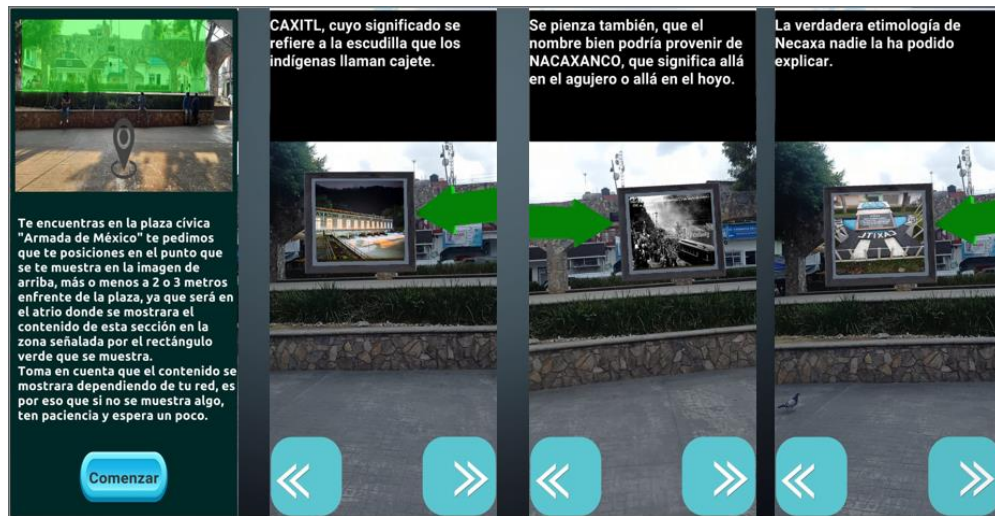


Imagen 147.- Contenido de la escena de la plaza cívica, tercer punto del recorrido.

Fuente: Imagen capturada de software instalado.



Imagen 148.- Segunda parte del contenido del tercer punto.

Fuente: Imagen capturada de software instalado.

Para concluir, el último punto del recorrido es la presidencia municipal de Juan Galindo. Parte del contenido que presentamos se puede observar en la Imagen 149. El material mostrado a lo largo del recorrido consiste principalmente en imágenes y texto.



Imagen 149.- Contenido del último punto del recorrido.

Fuente: Imagen capturada de software instalado.

Con esto concluimos la presentación del contenido utilizado en los experimentos realizados para la evaluación de nuestra aplicación. A lo largo de este proceso, recibimos comentarios positivos, así como numerosas ideas y sugerencias de mejora para futuros trabajos. Esto se debe a la necesidad de desarrollar personajes y dinámicas adicionales que hagan la aplicación aún más

atractiva para los usuarios. Sin embargo, para dar por finalizado este proyecto, se decidió no continuar con su expansión, a fin de evitar que se convierta en un trabajo más prolongado.

5.2.- Encuestas SUS.

Esta encuesta es un instrumento diseñado para medir la usabilidad de una aplicación. En esta sección, presentaremos las preguntas que formulamos y los resultados obtenidos tras el recorrido realizado por los usuarios. A continuación, listaremos las preguntas que se realizaron a través de un formulario de Google. Estas preguntas son las preguntas que se aplican al momento de evaluar una aplicación con el método SUS, pero han sido traducidas al español.

A continuación, te listaremos las preguntas y en la Tabla 2 podrás ver la calificación que nos otorgaron en cada pregunta. Si tienes una duda sobre qué pregunta se está respondiendo recuerda que la abreviatura P es para pregunta y el número indica el número de la pregunta de la lista mostraremos a continuación. Recuerda que el rango va de 1 a 5 y que en la tabla solo mostraremos las respuestas de la pregunta 1 a 10 debido a que la última opción se responde con texto y no con un valor numérico como las anteriores a ella.

1. ¿Creo que me gustaría ocupar Puebla Travel AR frecuentemente?
2. ¿Encontré Puebla Travel AR innecesariamente complicado?
3. ¿Pienso que Puebla Travel AR es fácil de usar?
4. ¿Creo que requeriré el apoyo de una persona experta para ser capaz de usar Puebla Travel AR?
5. ¿Encuentro las diversas tareas en Puebla Travel AR bien integradas?
6. ¿Pienso que Puebla Travel AR tenía muchas inconsistencias?
7. ¿Imagino que la mayoría de personas aprenderán a ocupar Puebla Travel AR bastante rápido?
8. ¿Encuentro Puebla Travel AR bastante problemático de usar?
9. ¿Siento bastante confianza al usar Puebla Travel AR?
10. ¿Necesito aprender muchas cosas antes de poder trabajar con Puebla Travel AR?
11. Describe qué harías diferente.

	P. 1	P. 2	P. 3	P. 4	P. 5	P. 6	P. 7	P. 8	P. 9	P.10
Marisol	5	5	5	3	5	1	5	1	5	3
Yadira	4	1	5	1	4	2	5	1	5	1
Ficel	4	1	5	2	4	2	5	1	5	3
Elizabeth	5	2	1	1	5	1	5	1	5	1
Jesús	5	2	3	4	4	1	3	1	5	1
Yoselin	5	1	5	2	4	2	5	1	5	1
Betzy	3	2	4	2	4	4	4	2	3	2
Ana Luisa	5	1	5	5	4	1	5	1	5	1
Gilberto	1	1	5	1	5	2	4	2	5	1
Diego Gael	2	1	4	3	4	3	5	1	5	1

Tabla 2.- Tabla que muestra como respondieron los usuarios la encuesta sus.

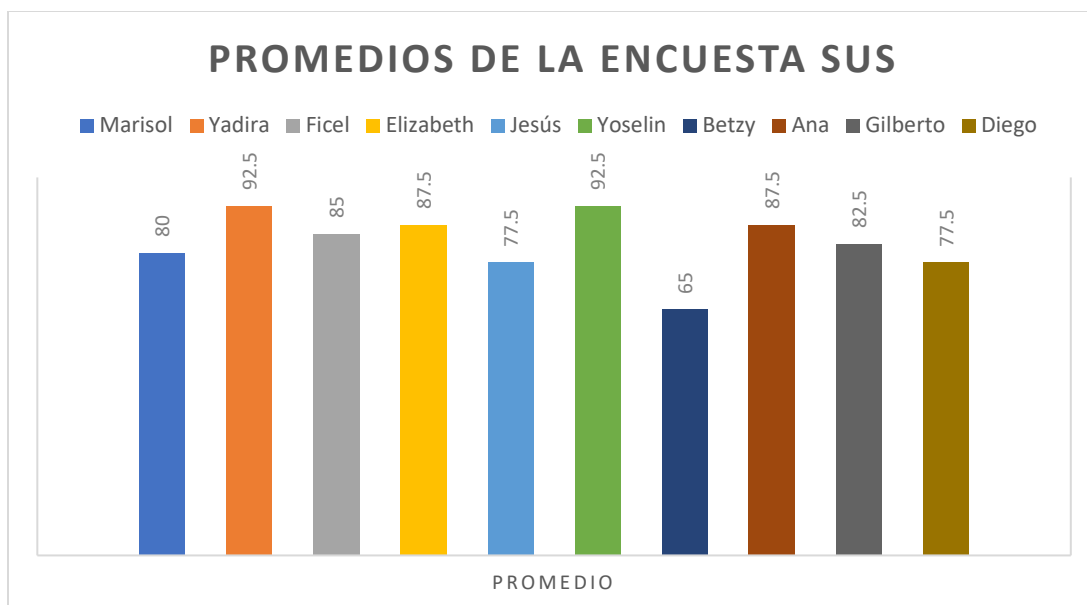
Posteriormente, comenzaremos a aplicar las operaciones sugeridas en el sistema SUS. Restamos 1 al valor que las personas que hicieron el experimento asignaron a las preguntas 1, 3, 5, 7 y 9. Al valor que asignaron a las preguntas 2, 4, 6, 8, y 10 le restamos 5.

Una vez realizadas las operaciones correspondientes, sumaremos los valores resultantes de cada pregunta, sumaremos los valores de las preguntas 1, 3, 5, 7 y 9 tras restarle 1 a cada valor y sumaremos los valores de las preguntas 2, 4, 6, 8 y 10, tras restarle 5 al valor de cada pregunta.

Ya casi tenemos el promedio que resultó después de las evaluaciones de los usuarios. Una vez que tengamos los cálculos de la Tabla 3, procederemos a sumar los dos resultados. Luego, multiplicaremos el resultado de la suma por 2.5 para obtener un promedio. Si este promedio es mayor a 70, se considerará que la aplicación es fácil de utilizar; en cambio, si el promedio es inferior a 70, será necesario tener en cuenta ciertas recomendaciones para mejorar.

Pregunta	Marisol	Yadira	Ficel	Elizabeth	Jesús	Yoselin	Betzy	Ana	Gilberto	Diego
Restando 1 al valor de la pregunta	20	18	18	16	15	19	13	19	15	15
Restando 5 al valor resultante	12	19	16	19	16	18	13	16	18	16

Tabla 3.- Tabla de los resultados de la evaluación después de las operaciones sugeridas por sus.



Gráfica 1.- Gráfica de promedios de las personas que evaluaron nuestra aplicación con el método sus.

Como se observa en la Gráfica 1, la mayoría de los promedios que obtuvimos se situaron por encima de 70; únicamente el promedio con el que una persona calificó nuestra aplicación resultó con valor de 65. Creemos que la calificación resultante fue gracias a la mala interpretación de las indicaciones iniciales, ya que las recomendaciones que nos proporcionó están relacionadas con la herramienta que utilizamos. En la Tabla 4, podrás ver las sugerencias que nos hicieron a través de la encuesta SUS.

Usuario	Recomendaciones
Elizabeth	Quizá las instrucciones para el corrido
Betzy	Poner descripciones cortas y concisas de cómo por el ejemplo “sigue la flecha” o “gira a tu izquierda/derecha” para saber que tienes que ir hacia allá ya que cuando la utilice no sabía qué el significado de las flechas o ponerlas de distinto color también funcionaría.
Gilberto	Reducir el tamaño de las flechas

Tabla 4.- Aquí mostramos las respuestas que nos dieron a la pregunta 11, solo 3 personas nos hicieron recomendaciones.

Como mencionamos, solo una persona nos evaluó de manera reprobatoria y nos proporcionó recomendaciones claras. Sin embargo, podemos mejorar, al igual que en cada

aplicación; a medida que más personas utilicen la aplicación, surgirán más especificaciones o solicitudes. Este aspecto se abordará en trabajos futuros.

5.3.- Resultados de la encuesta propia.

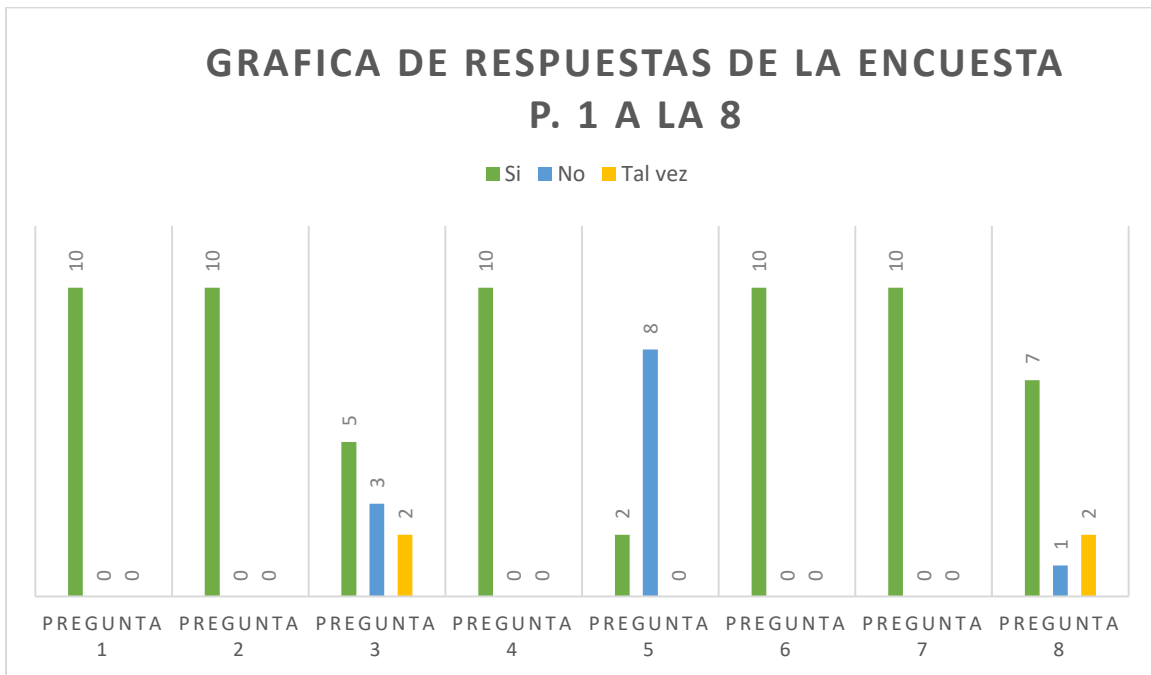
Nuestra encuesta no abarcaba aspectos relacionados con la usabilidad, nuestro objetivo era obtener indicios sobre el grado de interés que la idea propuesta generó en los usuarios. Se trata de un estudio de mercado informal que busca determinar si la aplicación tiene competencia y si resultó atractiva para las personas que la utilizaron.

La encuesta consta de 10 preguntas con respuestas de sí, no y una opción variable. Por ello, te presentaremos las preguntas que se realizaron a los encuestados e indicaremos cuántos de los 10 participantes seleccionaron cada opción.

Las preguntas son:

1. ¿Le llama la atención la implementación de tecnologías nuevas para los recorridos turísticos?
2. ¿Cree que la realidad aumentada sea una herramienta útil para el turismo?
3. ¿Ha escuchado sobre los museos interactivos?
4. ¿Cree que es una buena idea implementar realidad aumentada para conocer más detalles de la historia del lugar que se está visitando?
5. ¿Conoce alguna aplicación de realidad aumentada parecida a la que se le acaba de mostrar?
6. ¿Cree que el regalar descuentos en comercios cercanos servirá para reactivar la economía?
7. ¿Estaría dispuesto(a) a visitar otros lugares con tal de hacer un recorrido como el que hizo con nuestra aplicación?
8. ¿Cree que un cupón sería suficiente para hacer que consuma en el local que proporciona el descuento?
9. ¿Qué dispositivo móvil tiene usted?
10. ¿En caso de haber seleccionado la opción “c” escriba que sistema operativo utiliza su dispositivo?

En esta encuesta solicitamos respuestas de sí, no y tal vez, dándonos que el 100% de los encuestados, fueron constantes en cuanto a sus respuestas, como se muestra en la Gráfica 2.



Gráfica 2.- Gráfica de las respuestas de la encuesta que creamos nosotros, que muestra que tan innovadora les pareció la aplicación a los usuarios encuestados.

Como se observa en la Gráfica 2, la mayoría de las personas consideraron innovadora nuestra aplicación. Esto es porque todas las respuestas que verifican esta afirmación se han respondido afirmativamente. Estos resultados confirman los datos que, obtenidos en la investigación, ya que no existe aplicaciones similares o idénticas. Por lo tanto, este proyecto es muy viable y altamente aceptable para su implementación, no solo en Puebla, sino también en diversos destinos turísticos de la República Mexicana.

Las preguntas 1, 2, 4, 6 y 7 abordan si los usuarios consideran interesante la implementación de la realidad aumentada en el sector turístico y en qué medida este tipo de aplicaciones incentivaría a los usuarios para realizar los recorridos propuestos. La pregunta 3 busca investigar el conocimiento que tienen los usuarios sobre las tecnologías en el sector turístico. Aunque la mayoría de las personas que respondieron la encuesta mencionan haber escuchado hablar de los museos interactivos, pero al conversar con ellas nos descubrimos que su conocimiento es, en gran medida, vago y se basa en lo que han escuchado sobre el metaverso. Así

mismo las personas que respondieron “Alguna vez escuche algo” se refirieron a esos mismos datos. Sin embargo, ninguna de las personas mencionó proyectos de este tipo desarrollados en México.

La pregunta 8 evalúa que los usuarios que visitan alguna zona turística pueden consumir en algún local cercano, motivados solo por un descuento o un producto gratuito que obtienen a través de los puntos acumulados por las actividades realizadas con la aplicación.

Por último, las preguntas 9 y 10 se refieren a los dispositivos utilizados durante los experimentos. En la pregunta 9 se analiza si los usuarios utilizan dispositivo Android, iOS u otro. Si usan otro dispositivo, se indaga sobre el sistema operativo de dicho dispositivo.



Gráfica 3.- Gráfica de los dispositivos utilizados para los recorridos del experimento.

Como se observa en la Gráfica 3 la mayoría de las personas cuentan con un dispositivo Android y solo una contaba con iOS, cabe mencionar que la persona que utiliza iOS no quiso instalar la aplicación debido a la garantía de su dispositivo, es por eso que realizó el recorrido con un dispositivo Android; sin embargo, con esto nos damos cuenta de que el mercado está dominado por estos 2 sistemas y que si es factible crear la aplicación para ambos dispositivos.

5.4.- Recomendaciones.

Después de los experimentos realizados la gente nos dio distintas recomendaciones, las cuales se centraban en el sitio en donde se mostraba el contenido, en la distancia en la que aparece

el contenido, se nos mencionaba el tamaño de los elementos que se muestran como las flechas o el color de estas, otro comentario recurrente fue la velocidad con la que se veía el contenido.

Si bien no todo depende de nosotros, si mejoramos varios detalles que, si dependían de nosotros como desarrolladores siendo estos los colores de los elementos, la orientación de las imágenes, el tamaño del cuadro con el que mostramos el contenido y mejoramos las indicaciones mediante paneles en donde detallamos el cómo y dónde se mostrara el contenido, sin embargo, nos dimos cuenta de que la mayoría de las personas no son curiosas o pacientes para leer tantas indicaciones.

A continuación, transcribiremos las recomendaciones y errores que vieron los usuarios en nuestra aplicación, ya que las recomendaciones del puño y letra de los usuarios con los que experimentamos se encontraran en el apartado de anexos y las recomendaciones se muestran en la Tabla 5, como comentario toma en cuenta que solo 1 usuario no quiso responder este recuadro, esto debido a que no encontró más comentarios que realizar.

Nombre	Recomendaciones	Errores encontrados
Marisol	<p>En cuanto al texto más centrado con color negro ya que dificulta un poco por la imagen.</p> <p>Revisar las fotografías que no estén en efecto espejo.</p> <p>Checar las flechas y fecha de encuesta, así como ortografía.</p>	<p>Ortografía</p> <p>Imágenes</p>
Yadira	<p>Revisar gramática y hacer más llamativos los textos para que sea llamativo para el usuario.</p>	<p>Cambio de contraseña</p> <p>Enfoque</p>
Ficel	<p>Que los colores de la aplicación sean más llamativos o alusivos al lugar turístico.</p> <p>Que tenga más imágenes llamativas.</p> <p>Que haya interacción con los monumentos.</p>	<p>Faltas de ortografía.</p> <p>En un caso me regreso al inicio.</p>

Elizabeth	La flecha de otro color más llamativo Las letras de otro color	Especificar puntos de parada para encontrar las imágenes.
Jesús	Quizá mejorar las indicaciones en el recorrido	Dificultad al escanear
Yoselin		Al iniciar la visita, me sacaba de la aplicación. No aparecieron todas las imágenes.
Betzy	Recomendación como tal no tengo ya que considero qué, al ser una aplicación en desarrollo, no se debe “exigir algo más producido” ya que considero que al desarrollar una se tiene que ir puliendo a lo que el creador tiene como expectativa hasta llegar a como lo había previsto o planeado. Solo algo que si me agradaría sería que objetos o algún edificio tomara forma animada.	La distancia en la que debes tener para observar la realidad siento que deberían adaptarse a cualquier punto del lugar que te indica.
Ana Luisa	Agregar algunos otros puntos de interacción.	En el contenido, no se podía apreciar muy bien.
Ana Luisa segunda visita	En esta segunda ocasión pude apreciar con más claridad las imágenes que se presentan de apoyo para la explicación breve del municipio de Juan Galindo.	Un pequeño error que note fue que las flechas que indican la ubicación de ciertas imágenes no se alcanzaron a apreciar.
Gilberto	Las flechas que indican los siguientes puntos, son un poco grandes e intrusivas, quizá un poco más pequeñas estaría muy bien.	

Tabla 5.- Recomendaciones que nos dieron en el formato que se llenó después del recorrido.

En la Tabla 5 tenemos las recomendaciones y errores que encontraron los usuarios en nuestra aplicación, varias recomendaciones fueron tomadas en cuenta e implementadas para una mejor experiencia de usuario y si observas bien hay un caso en el que un usuario realizó el recorrido

2 veces y encontró mejoras, aunque también encontró un error y esto se debe a que las 2 personas que comentaron que el contenido no se mostraba bien, fue que hicieron el recorrido cuando ya estaba oscureciendo y esto debido a los escaneos que se realizaron en lightship, puesto que no se han de haber realizado los escaneos en horarios nocturnos, pero se pueden volver a realizar escaneos en esos horarios para mejorar el funcionamiento de la herramienta.

Por último, en este formato había un apartado de percepción de la aplicación, en el cual todos coincidieron en ser una aplicación innovadora, intuitiva y prometedora, comentarios como los de “Gilberto” quien es desarrollador de software, nos indican que vamos por buen camino y que esta aplicación cumple con su objetivo.

5.5.- Conclusiones.

Como conclusión, este proyecto resultó satisfactorio, porque los comentarios recibidos sobre la aplicación han sido buenos y la aplicación es demasiado innovadora para los usuarios que utilizaron el proyecto.

Además, gracias a la investigación realizada verificamos que no existe una aplicación similar a la propuesta en este trabajo y por eso nos resulta muy motivador llevar a la realidad nuestro trabajo porque creo que la realidad tiene muchas posibilidades que aún no se han explorado y que en mucho tiempo no se explorarán y no solo en el sector turístico, también en el sector empresarial, ya que en el sector educativo podemos apoyar la enseñanza de distintas materias mediante la realidad aumentada, recreando batallas, teoremas matemáticos, etc.

En el sector empresarial, se pueden implementar tutoriales para máquinas demasiado costosas o peligrosas y así no tener pérdidas y no poner en peligro a personas inexpertas, también podríamos utilizar la realidad aumentada como medio de orientación para los visitantes que lleguen a tener en las oficinas de la empresa o crear recorridos dentro de ellas dependiendo del giro de la empresa.

Por último, en el sector turístico le encontramos muchas más aplicaciones además de las propuestas de este proyecto, algunas aplicaciones que vemos son los museos interactivos, la replicación de batallas de las cuales solo se tiene información verbal, el rescate de arquitecturas antiguas por medio de la digitalización, también encontramos muchas propuestas de aplicaciones que se encarguen de publicidad por medio de la realidad aumentada y un sinnúmero de aplicaciones que sé que irán apareciendo conforme este campo vaya incrementando con el apoyo de las nuevas

tecnologías se han ido desarrollando, esto gracias a empresas como niantic, que se están encargando de desarrollar herramientas y dinámicas para incrementar la demanda de desarrolladores de realidad aumentada, así como se han encargado de ir ampliando las tecnologías en las que podemos utilizar esta poderosa herramienta.

5.6.- Trabajo futuro.

El prototipo que conseguimos obtener es muy bueno, pero aún se puede mejorar demasiado y esta propuesta tiene demasiadas posibilidades, por eso comentaremos varios puntos que pretenden implementarse después y por eso no se implementaron por 2 razones económicas, ya que para tener una mejor calidad tendríamos que invertir un poco de capital y por no alargas más el tiempo de desarrollo, ya que su implementación sería posible después de aprender nuevas tecnologías. El trabajo futuro es el siguiente:

- Recorrido narrado: en este apartado pretendemos eliminar el texto que se le muestra a los usuarios, esto debido a que varios usuarios no leen detalladamente y con esta implementación pretendemos que el usuario preste atención a un audio y no tenga que estar leyendo además se pretende realizar este punto para que personas con debilidad visual puedan utilizar nuestra aplicación.
- Tutoriales rápidos: también pretendemos implementar pequeñas animaciones para que el usuario tenga una mejor idea de cómo utilizar la aplicación, ya que gracias a los experimentos nos dimos cuenta de que a pesar de que se pretendió realizar una aplicación demasiado intuitiva, los usuarios necesitan de tutoriales o una pequeña explicación para facilitar el uso de la aplicación.
- Contenido alusivo al sitio que se visita: pretendemos tener “mascotas” alusivas al lugar turístico del recorrido, por ejemplo, el municipio de Juan Galindo es conocido por ser la cuna de la industria eléctrica en México, entonces sería muy interesante que el modelo encargado de guiar el recorrido o de dar las explicaciones sea algo alusivo a ello.
- Interacción con el entorno: pretendemos realizar el desarrollo de pequeños juegos que provoquen que el usuario interactúe con el entorno que se encuentra visitando, esto gracias a que niantic está generando muchas herramientas que nos ayudarían a crear dichas dinámicas.

- Competencias entre usuarios: otra modalidad que estan implementando recientemente en lightship son funciones que nos permiten llevar a cabo videojuegos multijugador y gracias a ello podríamos crear dinámicas con las que los usuarios comparen su comprensión del contenido que estan obteniendo.
- Orientación: se tiene planeado implementar un sistema de orientación, que nos indique si nos estamos dirigiendo al siguiente punto de manera correcta, tener una especie de navegación como la que tiene Google Maps.
- Más recorridos: buscar más lugares para implementar estos recorridos, no solo centrarnos en el estado de puebla, sino en zonas interesantes de la república mexicana.

De momento estos son los puntos que más nos interesa desarrollar para un futuro no muy lejano, sin embargo, las ideas siempre llegan de manera constante y siempre estamos dispuestos a mejorar

Anexos.

Repositorio git en donde tenemos todo el proyecto, así como el apk para Android.

<https://github.com/SArturoRE/NecaxaTravelTuristas.git>

Correo en el que te atiende niantic para solucionar duda o problemas que tengas durante el desarrollo de tus aplicaciones con la herramienta.

support@lightship.dev

Consentimientos informados de las personas que realizaron el experimento, así como la encuesta que respondieron en papel y los cuadros de comentarios que llenaron.

Glosario.

- API: Por sus siglas en inglés “Application Programming Interface”, se trata de un software intermediario que permite que las aplicaciones se comuniquen entre sí, intercambiando datos, características y funcionalidades [46].
- Motor de videojuegos: Los motores de videojuegos son una serie de librerías de programación que permite trabajar con el diseño, la creación y la representación de un videojuego [47].
- Realidad Virtual: La realidad virtual es un entorno de escenas y objetos simulados, que emulan un entorno real en el que las personas tienen la sensación de estar mediante un dispositivo visual y pueden desenvolverse dentro de ellos [48].
- Realidad Aumentada: La realidad aumentada son tecnologías que permiten que el usuario visualice parte del mundo real mediante un dispositivo con contenidos virtuales [49].
- Anchor (anclas en español): Son elementos que se utilizan cuando utilizamos el API de geospatial creator de ARCore, con estos elementos podemos colocar el contenido virtual por medio de este api.
- Geolocalización: La geolocalización es el proceso de identificación de datos geográficos captados mediante dispositivos conectados a internet, radiofrecuencia o satélite [50].
- Motion graphics: En esencia es un tipo de animación, cuyo propósito es distinto en varios sentidos. El término animación puede referirse a distintas técnicas para hacer que las imágenes tengan movimiento. Por otro lado, los motion graphics son propiamente un estilo

de diseño, en el que el movimiento se aplica a ciertos elementos de una composición para atraer la atención del espectador a ellos [51].

- Render: Es una imagen digital que se crea a partir de un modelo o escenario 3D realizado en algún programa de computadora especializado, cuyo objetivo es dar una apariencia realista desde cualquier perspectiva del modelo [52].
- C#: Es un lenguaje de programación moderno, basado en objetos y con seguridad de tipos. C# permite a los desarrolladores crear muchos tipos de aplicaciones seguras y sólidas que se ejecutan en .NET [53].
- C++: Es un lenguaje de programación que proviene de la extensión del lenguaje C para que pudiese manipular objetos [54].
- Streaming: Es enviar y recibir datos en un flujo continuo a través de una red informática, desde un servidor remoto hasta un dispositivo [55].
- Juegos AAA: Son aquellos videojuegos que están desarrollados por grandes compañías de la industria con altos costes tanto de desarrollo como de marketing.
- SDK (Kit de desarrollo de software): Es un conjunto de herramientas de creación específicas de plataformas para desarrolladores. Se necesitan componentes como depuradores, compiladores y bibliotecas para crear código que se ejecute en una plataforma, sistema operativo o lenguaje de programación específicos.
- APK: Son las siglas de “Android Application Package” y se trata de un archivo ejecutable que contiene todos los datos que se necesitan para instalar y hacer funcionar una aplicación Android.



Mariela Patricia Sánchez

Nombre y firma.



Encuesta

1. ¿Le llama la atención la implementación de tecnologías nuevas para los recorridos turísticos?
a) ~~Si~~ b) No c) Creo haber escuchado algo
2. ¿Cree que la realidad aumentada sea una herramienta útil para el turismo?
a) ~~Si~~ b) No c) Prefiero las visitas tradicionales
3. ¿Ha escuchado sobre los museos interactivos?
a) Si b) ~~No~~ c) ~~Alguna vez escuche algo~~
4. ¿Cree que es una buena idea implementar realidad aumentada para conocer más detalles de la historia del lugar que se está visitando?
a) ~~Si~~ b) No c) Seria cuestión de estudiarlo
5. ¿Conoce alguna aplicación de realidad aumentada parecida a la que se le acaba de mostrar?
a) Si b) ~~No~~
6. ¿Cree que el regalar descuentos en comercios cercanos servirá para reactivar la economía?
a) ~~Si~~ b) No c) Es mala estrategia
7. ¿Estaría dispuesto(a) a visitar otros lugares con tal de hacer un recorrido como el que hizo con nuestra aplicación?
a) ~~Si~~ b) No c) Depende de la cercanía
8. ¿Cree que un cupón sería suficiente para hacer que consuma en el local que proporciona el descuento?
a) ~~Si~~ b) No c) tendrían q
9. ¿Qué dispositivo móvil tiene usted?
a) ~~Android~~ b) IOS c) Otro
10. ¿En caso de haber seleccionado la opción "c" escriba que sistema operativo utiliza su dispositivo?

Consentimiento informado

Nuevo Necaxa, Puebla a 14 de mayo de 2024.

Yo, Yadira Franco Santos,
confirmando de manera voluntaria que mis datos sean utilizados solamente con fines académicos en la muestra de resultados en el trabajo de tesis: "RECORRIDO DEL CENTRO HISTÓRICO DE LA CIUDAD DE PUEBLA CON REALIDAD AUMENTADA".

Dichos datos son; el nombre completo, los comentarios referentes a la aplicación, los cuales serán plasmados en la tesis como recomendaciones para ser tomadas en cuenta para el desarrollo final de la aplicación de realidad aumentada y las respuestas que doy en la encuesta final que me fue aplicada.

Por medio de la presente, el PASANTE DE INGENIERIA EN CIENCIAS DE LA COMPUTACIÓN; Salvador Arturo Ramirez Escamilla, con matrícula 201513123, en calidad de estudiante, autorizo de manera informada y voluntaria, el uso de mis datos personales para fines académicos, conforme a las finalidades establecidas que me fueron informadas previamente.

Yadira Franco Santos

Nombre y firma.



Encuesta

1. ¿Le llama la atención la implementación de tecnologías nuevas para los recorridos turísticos?
a) Si b) No c) Creo haber escuchado algo
2. ¿Cree que la realidad aumentada sea una herramienta útil para el turismo?
a) Si b) No c) Prefiero las visitas tradicionales
3. ¿Ha escuchado sobre los museos interactivos?
a) Si b) No c) Alguna vez escuche algo
4. ¿Cree que es una buena idea implementar realidad aumentada para conocer más detalles de la historia del lugar que se está visitando?
a) Si b) No c) Seria cuestión de estudiarlo
5. ¿Conoce alguna aplicación de realidad aumentada parecida a la que se le acaba de mostrar?
a) Si b) No
6. ¿Cree que el regalar descuentos en comercios cercanos servirá para reactivar la economía?
a) Si b) No c) Es mala estrategia
7. ¿Estaría dispuesto(a) a visitar otros lugares con tal de hacer un recorrido como el que hizo con nuestra aplicación?
a) Si b) No c) Depende de la cercanía
8. ¿Cree que un cupón sería suficiente para hacer que consuma en el local que proporciona el descuento?
a) Si b) No c) tendrían q
9. ¿Qué dispositivo móvil tiene usted?
a) Android b) IOs c) Otro
10. ¿En caso de haber seleccionado la opción "c" escriba que sistema operativo utiliza su dispositivo?

Nombre	Marisol Patricia Sánchez			Fecha	14-mayo-2024
Edad	27	Ocupación	Psicóloga		
Recomendaciones	<p>En cuanto al texto más centrado con color en negro ya que dificulta un poco por la imagen.</p> <p>Revisar las fotografías que no estén en efecto espejo.</p> <p>Checar fechas y fecha de encuesta, así como ortografía.</p>				
Errores encontrados	<ul style="list-style-type: none"> - Ortografía - Imágenes. 				
Percepción de la aplicación.	<ul style="list-style-type: none"> * Dinámica * Fácil uso * Interactiva * 				

Nombre	Yadira Franco Santos			Fecha	
Edad	28	Ocupación	Lic. RR. II.		
Recomendaciones	<p>Revisar gramática y hacer más llamativos los textos para que sea llamativo para el usuario.</p>				
Errores encontrados	<p>Cambio de contraseña</p> <p>Enfoque</p>				
Percepción de la aplicación.	<p>Innovadora y llamativo al público</p>				

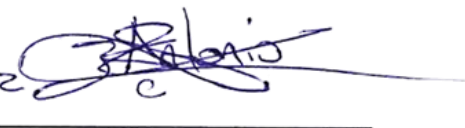
Consentimiento informado

Nuevo Necaxa, Puebla a 14/Mayo de 2024.

Yo, GERMAN ANTONIO CRUZ,
en papel de tutor legal confirmo de manera voluntaria que los datos de mi tutorado
Ficel Antonio Valdez
sean utilizados solamente con fines académicos en la muestra de resultados en el
trabajo de tesis: "RECORRIDO DEL CENTRO HISTÓRICO DE LA CIUDAD DE
PUEBLA CON REALIDAD AUMENTADA".

Dichos datos son; el nombre completo, los comentarios referentes a la aplicación,
los cuales serán plasmados en la tesis como recomendaciones para ser tomadas
en cuenta para el desarrollo final de la aplicación de realidad aumentada y las
respuestas que doy en la encuesta final que me fue aplicada.

Por medio de la presente, el PASANTE DE INGENIERIA EN CIENCIAS DE LA
COMPUTACIÓN; Salvador Arturo Ramirez Escamilla, con matrícula 201513123,
en calidad de estudiante, autorizo de manera informada y voluntaria, el uso de mis
datos personales para fines académicos, conforme a las finalidades establecidas
que me fueron informadas previamente.

GERMAN ANTONIO CRUZ 

Nombre y firma.

Encuesta

1. ¿Le llama la atención la implementación de tecnologías nuevas para los recorridos turísticos?
a) Si b) No c) Creo haber escuchado algo
2. ¿Cree que la realidad aumentada sea una herramienta útil para el turismo?
a) Si b) No c) Prefiero las visitas tradicionales
3. ¿Ha escuchado sobre los museos interactivos?
a) Si b) No c) Alguna vez escuche algo
4. ¿Cree que es una buena idea implementar realidad aumentada para conocer más detalles de la historia del lugar que se está visitando?
a) Si b) No c) Seria cuestión de estudiarlo
5. ¿Conoce alguna aplicación de realidad aumentada parecida a la que se le acaba de mostrar?
a) Si b) No
6. ¿Cree que el regalar descuentos en comercios cercanos servirá para reactivar la economía?
a) Si b) No c) Es mala estrategia
7. ¿Estaría dispuesto(a) a visitar otros lugares con tal de hacer un recorrido como el que hizo con nuestra aplicación?
a) Si b) No c) Depende de la cercanía
8. ¿Cree que un cupón sería suficiente para hacer que consuma en el local que proporciona el descuento?
a) Si b) No c) tendrían q
9. ¿Qué dispositivo móvil tiene usted?
a) Android b) IOS c) Otro
10. ¿En caso de haber seleccionado la opción "c" escriba que sistema operativo utiliza su dispositivo?

Nombre	Ficel Antonio Valdez		Fecha 14/05/24
Edad	16	Ocupación	Estudiante
Recomendaciones	<ul style="list-style-type: none"> - Que los colores de la aplicación sean más llamativos o alusivos al lugar turístico. - Que tenga más imágenes llamativas. - Que haya interacción con los monumentos. 		
Errores encontrados	<ul style="list-style-type: none"> - Faltas de ortografía. - En un caso me regresó al inicio. 		
Percepción de la aplicación.	Es llamativa, porque no hay aplicaciones sobre ese tipo en mi municipio, y menos sobre la historia de él.		

Consentimiento informado

Nuevo Necaxa, Puebla a 15 de Mayo de 2024.

Yo, Elizabeth Yarissa Rios Ramirez,
confirmando de manera voluntaria que mis datos sean utilizados solamente con fines académicos en la muestra de resultados en el trabajo de tesis: "RECORRIDO DEL CENTRO HISTÓRICO DE LA CIUDAD DE PUEBLA CON REALIDAD AUMENTADA".

Dichos datos son; el nombre completo, los comentarios referentes a la aplicación, los cuales serán plasmados en la tesis como recomendaciones para ser tomadas en cuenta para el desarrollo final de la aplicación de realidad aumentada y las respuestas que doy en la encuesta final que me fue aplicada.

Por medio de la presente, el PASANTE DE INGENIERIA EN CIENCIAS DE LA COMPUTACIÓN; Salvador Arturo Ramirez Escamilla, con matrícula 201513123, en calidad de estudiante, autorizo de manera informada y voluntaria, el uso de mis datos personales para fines académicos, conforme a las finalidades establecidas que me fueron informadas previamente.

Elizabeth Y. Rios Rmz. EYariss

Nombre y firma.

Encuesta

1. ¿Le llama la atención la implementación de tecnologías nuevas para los recorridos turísticos?
a) Si b) No c) Creo haber escuchado algo
2. ¿Cree que la realidad aumentada sea una herramienta útil para el turismo?
a) Si b) No c) Prefiero las visitas tradicionales
3. ¿Ha escuchado sobre los museos interactivos?
a) Si b) No c) Alguna vez escuche algo
4. ¿Cree que es una buena idea implementar realidad aumentada para conocer más detalles de la historia del lugar que se está visitando?
a) Si b) No c) Seria cuestión de estudiarlo
5. ¿Conoce alguna aplicación de realidad aumentada parecida a la que se le acaba de mostrar?
a) Si b) No
6. ¿Cree que el regalar descuentos en comercios cercanos servirá para reactivar la economía?
a) Si b) No c) Es mala estrategia
7. ¿Estaría dispuesto(a) a visitar otros lugares con tal de hacer un recorrido como el que hizo con nuestra aplicación?
a) Si b) No c) Depende de la cercanía
8. ¿Cree que un cupón sería suficiente para hacer que consuma en el local que proporciona el descuento?
a) Si b) No c) tendrían q
9. ¿Qué dispositivo móvil tiene usted?
a) Android b) IOs c) Otro
10. ¿En caso de haber seleccionado la opción "c" escriba que sistema operativo utiliza su dispositivo?

Encuesta

1. ¿Le llama la atención la implementación de tecnologías nuevas para los recorridos turísticos?
a) Si b) No c) Creo haber escuchado algo
2. ¿Cree que la realidad aumentada sea una herramienta útil para el turismo?
a) Si b) No c) Prefiero las visitas tradicionales
3. ¿Ha escuchado sobre los museos interactivos?
a) Si b) No c) Alguna vez escuche algo
4. ¿Cree que es una buena idea implementar realidad aumentada para conocer más detalles de la historia del lugar que se está visitando?
a) Si b) No c) Seria cuestión de estudiarlo
5. ¿Conoce alguna aplicación de realidad aumentada parecida a la que se le acaba de mostrar?
a) Si b) No
6. ¿Cree que el regalar descuentos en comercios cercanos servirá para reactivar la economía?
a) Si b) No c) Es mala estrategia
7. ¿Estaría dispuesto(a) a visitar otros lugares con tal de hacer un recorrido como el que hizo con nuestra aplicación?
a) Si b) No c) Depende de la cercanía
8. ¿Cree que un cupón sería suficiente para hacer que consuma en el local que proporciona el descuento?
a) Si b) No c) tendrían q
9. ¿Qué dispositivo móvil tiene usted?
a) Android b) IOs c) Otro
10. ¿En caso de haber seleccionado la opción "c" escriba que sistema operativo utiliza su dispositivo?

Nombre	Elizabeth Yarissa Rios Ramirez			Fecha	15-05-24
Edad	42 años	Ocupación	Negocio Propio		
Recomendaciones	La flecha de otro color más llamativo Las letras de otro color ✓ -				
Errores encontrados	Especificar puntos de parada para encontrar imágenes.				
Percepción de la aplicación.	Muy buena				

Nombre	Jesús Márquez Granado			Fecha	15/Mayo/24
Edad	43 años	Ocupación	Médico Cirujano		
Recomendaciones	Quizá mejorar las indicaciones en el recordo.				
Errores encontrados	Dificultad al escanear.				
Percepción de la aplicación.	Excelente.				


Consentimiento informado

Nuevo Necaxa, Puebla a 17 Mayo de 2024.

Yo, Ana Luisa Ibarra Escamilla,
confirmando de manera voluntaria que mis datos sean utilizados solamente con fines académicos en la muestra de resultados en el trabajo de tesis: "RECORRIDO DEL CENTRO HISTÓRICO DE LA CIUDAD DE PUEBLA CON REALIDAD AUMENTADA".

Dichos datos son; el nombre completo, los comentarios referentes a la aplicación, los cuales serán plasmados en la tesis como recomendaciones para ser tomadas en cuenta para el desarrollo final de la aplicación de realidad aumentada y las respuestas que doy en la encuesta final que me fue aplicada.

Por medio de la presente, el PASANTE DE INGENIERIA EN CIENCIAS DE LA COMPUTACIÓN; Salvador Arturo Ramirez Escamilla, con matrícula 201513123, en calidad de estudiante, autorizo de manera informada y voluntaria, el uso de mis datos personales para fines académicos, conforme a las finalidades establecidas que me fueron informadas previamente.

Ana Luisa Ibarra Escamilla 

Nombre y firma.

PR. Ana.
17 (Mayo) 24

Encuesta

1. ¿Le llama la atención la implementación de tecnologías nuevas para los recorridos turísticos?
a) Si ✓ b) No c) Creo haber escuchado algo
2. ¿Cree que la realidad aumentada sea una herramienta útil para el turismo?
a) Si ✓ b) No c) Prefiero las visitas tradicionales
3. ¿Ha escuchado sobre los museos interactivos?
a) Si ✓ b) No c) Alguna vez escuche algo
4. ¿Cree que es una buena idea implementar realidad aumentada para conocer más detalles de la historia del lugar que se está visitando?
a) Si ✓ b) No c) Seria cuestión de estudiarlo
5. ¿Conoce alguna aplicación de realidad aumentada parecida a la que se le acaba de mostrar?
a) Si b) No ✓
6. ¿Cree que el regalar descuentos en comercios cercanos servirá para reactivar la economía?
a) Si ✓ b) No c) Es mala estrategia
7. ¿Estaría dispuesto(a) a visitar otros lugares con tal de hacer un recorrido como el que hizo con nuestra aplicación?
a) Si ✓ b) No c) Depende de la cercanía
8. ¿Cree que un cupón sería suficiente para hacer que consuma en el local que proporciona el descuento?
a) Si ✓ b) No c) tendrían q
9. ¿Qué dispositivo móvil tiene usted?
a) Android ✓ b) IOs c) Otro
10. ¿En caso de haber seleccionado la opción "c" escriba que sistema operativo utiliza su dispositivo?

Nombre	Ana Luisa Ibarra Escamilla			Fecha	17/05/2024
Edad	32 años	Ocupación	Amo de casa		
Recomendaciones	Agregar Algunos otros puntos de Interacción				
Errores encontrados	En el contenido, no se podía apreciar muy bien.				
Percepción de la aplicación.	En mi punto de vista es una aplicación muy buena, aunque hay que mejorar la apreciación del contenido.				

Nombre	Ana Luisa Ibarra Escamilla			Fecha	23/05/24
Edad	32 años	Ocupación	amade casa		
Recomendaciones	En esta segunda ocasión pude apreciar con más claridad las imágenes que se presentaban de apoyo para la explicación breve del Municipio de Juan Guinelo.				
Errores encontrados	Un pequeño error que note fue que las flechas que indican la ubicación de ciertas imágenes no se alcanzaban a apreciar.				
Percepción de la aplicación.	Es una muy buena aplicación para Brindar información concisa y precisa a los turistas.				


Consentimiento informado

Nuevo Necaxa, Puebla a 20 DE MAYO D de 2024.

Yo, BETSY ESCAMILLA VARGAS,
confirmando de manera voluntaria que mis datos sean utilizados solamente con fines académicos en la muestra de resultados en el trabajo de tesis: "RECORRIDO DEL CENTRO HISTÓRICO DE LA CIUDAD DE PUEBLA CON REALIDAD AUMENTADA".

Dichos datos son; el nombre completo, los comentarios referentes a la aplicación, los cuales serán plasmados en la tesis como recomendaciones para ser tomadas en cuenta para el desarrollo final de la aplicación de realidad aumentada y las respuestas que doy en la encuesta final que me fue aplicada.

Por medio de la presente, el PASANTE DE INGENIERIA EN CIENCIAS DE LA COMPUTACIÓN; Salvador Arturo Ramirez Escamilla, con matrícula 201513123, en calidad de estudiante, autorizo de manera informada y voluntaria, el uso de mis datos personales para fines académicos, conforme a las finalidades establecidas que me fueron informadas previamente.



BETSY ESCAMILLA VARGAS

Nombre y firma.

B-122

Encuesta

1. ¿Le llama la atención la implementación de tecnologías nuevas para los recorridos turísticos?
a) Si b) No c) Creo haber escuchado algo
2. ¿Cree que la realidad aumentada sea una herramienta útil para el turismo?
a) Si b) No c) Prefiero las visitas tradicionales
3. ¿Ha escuchado sobre los museos interactivos?
a) Si b) No c) Alguna vez escuche algo
4. ¿Cree que es una buena idea implementar realidad aumentada para conocer más detalles de la historia del lugar que se está visitando?
a) Si b) No c) Seria cuestión de estudiarlo
5. ¿Conoce alguna aplicación de realidad aumentada parecida a la que se le acaba de mostrar?
a) Si b) No
6. ¿Cree que el regalar descuentos en comercios cercanos servirá para reactivar la economía?
a) Si b) No c) Es mala estrategia
7. ¿Estaría dispuesto(a) a visitar otros lugares con tal de hacer un recorrido como el que hizo con nuestra aplicación?
a) Si b) No c) Depende de la cercanía
8. ¿Cree que un cupón sería suficiente para hacer que consuma en el local que proporciona el descuento?
a) Si b) No c) tendrían q
9. ¿Qué dispositivo móvil tiene usted?
a) Android b) IOs c) Otro
10. ¿En caso de haber seleccionado la opción "c" escriba que sistema operativo utiliza su dispositivo?

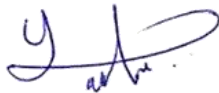
Consentimiento informado

Nuevo Necaxa, Puebla a 20 de Mayo de 2024.

Yo, Yoselin Flores Venegas,
confirmando de manera voluntaria que mis datos sean utilizados solamente con fines académicos en la muestra de resultados en el trabajo de tesis: "RECORRIDO DEL CENTRO HISTÓRICO DE LA CIUDAD DE PUEBLA CON REALIDAD AUMENTADA".

Dichos datos son; el nombre completo, los comentarios referentes a la aplicación, los cuales serán plasmados en la tesis como recomendaciones para ser tomadas en cuenta para el desarrollo final de la aplicación de realidad aumentada y las respuestas que doy en la encuesta final que me fue aplicada.

Por medio de la presente, el PASANTE DE INGENIERIA EN CIENCIAS DE LA COMPUTACIÓN; Salvador Arturo Ramirez Escamilla, con matrícula 201513123, en calidad de estudiante, autorizo de manera informada y voluntaria, el uso de mis datos personales para fines académicos, conforme a las finalidades establecidas que me fueron informadas previamente.



YOSELIN FLORES VENEGAS

Nombre y firma.

Encuesta

1. ¿Le llama la atención la implementación de tecnologías nuevas para los recorridos turísticos?
a) Si b) No c) Creo haber escuchado algo
2. ¿Cree que la realidad aumentada sea una herramienta útil para el turismo?
a) Si b) No c) Prefiero las visitas tradicionales
3. ¿Ha escuchado sobre los museos interactivos?
a) Si b) No c) Alguna vez escuche algo
4. ¿Cree que es una buena idea implementar realidad aumentada para conocer más detalles de la historia del lugar que se está visitando?
a) Si b) No c) Seria cuestión de estudiarlo
5. ¿Conoce alguna aplicación de realidad aumentada parecida a la que se le acaba de mostrar?
a) Si b) No
6. ¿Cree que el regalar descuentos en comercios cercanos servirá para reactivar la economía?
a) Si b) No c) Es mala estrategia
7. ¿Estaría dispuesto(a) a visitar otros lugares con tal de hacer un recorrido como el que hizo con nuestra aplicación?
a) Si b) No c) Depende de la cercanía
8. ¿Cree que un cupón sería suficiente para hacer que consuma en el local que proporciona el descuento?
a) Si b) No c) tendrían q
9. ¿Qué dispositivo móvil tiene usted?
a) Android b) IOs c) Otro
10. ¿En caso de haber seleccionado la opción "c" escriba que sistema operativo utiliza su dispositivo?

Nombre	BETSY ESCAMILLA JARGAS			Fecha	20-MAYO-2024
Edad	28	Ocupación	EMPLEADA		
Recomendaciones	<p>RECOMENDACIÓN COMO TAL NO TENGO YA QUE CONSIDERO QUE AL SER UNA APLICACIÓN EN DESARROLLO, NO SE DEBE "EXIGIR ALGO MAS PRODUCIDO" YA QUE CONSIDERO QUE AL DESARROLLAR UNA SE TIENE QUE IR PULIENDO A LO QUE EL CREADOR TIENE COMO EXPECTATIVA HASTA LLEGAR A COMO LO HABIA PREVISTO ORANEARLO. SOLO ALGO QUE SI ME AGRADARIA SERIA QUE OBJETOS O ALGUN EDIFICIO TOMARA FORMA ANIMADA.</p>				
Errores encontrados	<p>LA DISTANCIA EN LA QUE DEBES TENER PARA PODER OBSERVARLA REALIDAD SIENTO QUE DEBERIA ADAPTARSE A CUALQUIER PUNTO DEL LUGAR QUE TE INDICA.</p>				
Percepción de la aplicación.	<p>ES BUENA, PERO COMO DICE SI SE SIGUE TRABAJANDO o PULIENDO PUEDE REALMENTE LLEGAR A SER MUY BUENA.</p>				

Nombre	YOSIELIN FLORES VENEGAS			Fecha	20-Mayo-2024
Edad	26 años	Ocupación	SN		
Recomendaciones					
Errores encontrados	<p>Al iniciar la visita, me sacaba de la aplicación, NO aparecían todos los imagenes.</p>				
Percepción de la aplicación.	<p>Super Interesante</p>				192


Consentimiento informado

Nuevo Necaxa, Puebla a 18 de mayo de 2024.

Yo, Hilda Morales Nolasco,
en papel de tutor legal confirmo de manera voluntaria que los datos de mi tutorado
Diego Gael Escamilla Morales
sean utilizados solamente con fines académicos en la muestra de resultados en el
trabajo de tesis: "RECORRIDO DEL CENTRO HISTÓRICO DE LA CIUDAD DE
PUEBLA CON REALIDAD AUMENTADA".

Dichos datos son; el nombre completo, los comentarios referentes a la aplicación,
los cuales serán plasmados en la tesis como recomendaciones para ser tomadas
en cuenta para el desarrollo final de la aplicación de realidad aumentada y las
respuestas que doy en la encuesta final que me fue aplicada.

Por medio de la presente, el PASANTE DE INGENIERIA EN CIENCIAS DE LA
COMPUTACIÓN; Salvador Arturo Ramirez Escamilla, con matrícula 201513123,
en calidad de estudiante, autorizo de manera informada y voluntaria, el uso de mis
datos personales para fines académicos, conforme a las finalidades establecidas
que me fueron informadas previamente.

Hilda Morales Nolasco 

Nombre y firma.

Encuesta

1. ¿Le llama la atención la implementación de tecnologías nuevas para los recorridos turísticos?
a) Si b) No c) Creo haber escuchado algo
2. ¿Cree que la realidad aumentada sea una herramienta útil para el turismo?
a) Si b) No c) Prefiero las visitas tradicionales
3. ¿Ha escuchado sobre los museos interactivos?
a) Si b) No c) Alguna vez escuche algo
4. ¿Cree que es una buena idea implementar realidad aumentada para conocer más detalles de la historia del lugar que se está visitando?
a) Si b) No c) Seria cuestión de estudiarlo
5. ¿Conoce alguna aplicación de realidad aumentada parecida a la que se le acaba de mostrar?
a) Si b) No
6. ¿Cree que el regalar descuentos en comercios cercanos servirá para reactivar la economía?
a) Si b) No c) Es mala estrategia
7. ¿Estaría dispuesto(a) a visitar otros lugares con tal de hacer un recorrido como el que hizo con nuestra aplicación?
a) Si b) No c) Depende de la cercanía
8. ¿Cree que un cupón sería suficiente para hacer que consuma en el local que proporciona el descuento?
a) Si b) No c) tendrían q
9. ¿Qué dispositivo móvil tiene usted?
a) Android b) IOs c) Otro
10. ¿En caso de haber seleccionado la opción "c" escriba que sistema operativo utiliza su dispositivo?


Consentimiento informado

Nuevo Necaxa, Puebla a 28 mayo de 2024.

Yo, Gilberto Rodríguez Ramírez,
confirmando de manera voluntaria que mis datos sean utilizados solamente con fines académicos en la muestra de resultados en el trabajo de tesis: "RECORRIDO DEL CENTRO HISTÓRICO DE LA CIUDAD DE PUEBLA CON REALIDAD AUMENTADA".

Dichos datos son; el nombre completo, los comentarios referentes a la aplicación, los cuales serán plasmados en la tesis como recomendaciones para ser tomadas en cuenta para el desarrollo final de la aplicación de realidad aumentada y las respuestas que doy en la encuesta final que me fue aplicada.

Por medio de la presente, el PASANTE DE INGENIERIA EN CIENCIAS DE LA COMPUTACIÓN; Salvador Arturo Ramirez Escamilla, con matrícula 201513123, en calidad de estudiante, autorizo de manera informada y voluntaria, el uso de mis datos personales para fines académicos, conforme a las finalidades establecidas que me fueron informadas previamente.

Gilberto Rodríguez Ramírez 

Nombre y firma.

Encuesta

1. ¿Le llama la atención la implementación de tecnologías nuevas para los recorridos turísticos?

a) Sí

b) No

c) Creo haber escuchado algo

2. ¿Cree que la realidad aumentada sea una herramienta útil para el turismo?

a) Sí

b) No

c) Prefiero las visitas tradicionales

3. ¿Ha escuchado sobre los museos interactivos?

a) Sí

b) No

c) Alguna vez escuche algo

4. ¿Cree que es una buena idea implementar realidad aumentada para conocer más detalles de la historia del lugar que se está visitando?

a) Sí

b) No

c) Seria cuestión de estudiarlo

5. ¿Conoce alguna aplicación de realidad aumentada parecida a la que se le acaba de mostrar?

a) Sí

b) No

6. ¿Cree que el regalar descuentos en comercios cercanos servirá para reactivar la economía?

a) Sí

b) No

c) Es mala estrategia

7. ¿Estaría dispuesto(a) a visitar otros lugares con tal de hacer un recorrido como el que hizo con nuestra aplicación?

a) Sí

b) No

c) Depende de la cercanía

8. ¿Cree que un cupón sería suficiente para hacer que consuma en el local que proporciona el descuento?

a) Sí

b) No

c) tendrían q

9. ¿Qué dispositivo móvil tiene usted?

a) Android

b) IOs

c) Otro

10. ¿En caso de haber seleccionado la opción "c" escriba que sistema operativo utiliza su dispositivo?

Nombre	Gilberto Rodríguez Ramírez			Fecha	28/05/24
Edad	33	Ocupación	Desarrollador software		
Recomendaciones	Las flechas que indican los siguientes puntos, son un poco grandes o intrusivas, quiz un poco más pequeños estarían muy bien.				
Errores encontrados					
Percepción de la aplicación.	Buena, bastante intuitiva y fácil de usar.				

Bibliografía

- [1] Muñoz, R. V. (2012). La senda del saber en el centro histórico de Puebla, México: una propuesta de turismo cultural a través de micronódulos turísticos en ciudades patrimoniales. *CULTUR: Revista de Cultura e Turismo*, 6(2), 5-19.
- [2] Yung, R., & Khoo-Lattimore, C. (2019). New realities: a systematic literature review on virtual reality and augmented reality in tourism research. *Current Issues in Tourism*, 22(17), 2056-2081.
- [3] Bello, C. R. (2017). La realidad aumentada: lo que debemos conocer. *Tecnología Investigación y Academia*, 5(2), 257-261.
- [4] Prendes Espinosa, C. (2015). Realidad aumentada y educación: análisis de experiencias prácticas. *Píxel-Bit. Revista de Medios y Educación*, 46, 187-203.
- [5] Torres, D. R. (2011). Realidad Aumentada, educación y museos. *ICONO 14, Revista de comunicación y tecnologías emergentes*, 9(2), 212-226.
- [6] Corporativa, I. (n.d.). ¿Qué es la Realidad Aumentada? Iberdrola. Recuperado el 23 de noviembre del 2023, de <https://www.iberdrola.com/innovacion/que-es-realidad-aumentada>
- [7] Silva, P. D. (2005). Vinculación de la ciudad históricoturística de Puebla a través de itinerarios culturales. *Economía Sociedad y Territorio*.
- [8] Boboc, R. G., Băutu, E., Gîrbacia, F., Popovici, N., & Popovici, D. M. (2022). Augmented reality in cultural heritage: an overview of the last decade of applications. *Applied Sciences*, 12(19), 9859.
- [9] Basogain, X., Olabe, M., Espinosa, K., Rouèche, C., & Olabe, J. C. (2007). Realidad Aumentada en la Educación: una tecnología emergente. *Escuela Superior de Ingeniería de Bilbao, EHU*. Recuperado de <http://bit.ly/2hpZokY>.
- [10] Telefónica, F. (2011). *Realidad Aumentada: una nueva lente para ver el mundo*. Fundación Telefónica.
- [11] Bohórquez, I. M. M. (2018). Realidad aumentada y aplicaciones. *Tecnología Investigación y Academia*, 6(1), 28-35.
- [12] Christopoulos, D., Mavridis, P., Andreadis, A., & Karigiannis, J. N. (2011, May). Using Virtual Environments to Tell the Story: "The Battle of Thermopylae". In *2011 Third International Conference on Games and Virtual Worlds for Serious Applications* (pp. 84-91). IEEE.
- [13] Gooding, M. (2016, November 2). Augmented reality on your iPhone with ViewRanger. *Cambridgeshire Live*. <https://www.cambridge-news.co.uk/business/technology/augmented-reality-your-iphone-viewranger-12112780> Gooding, M. (2016). Augmented reality on your iPhone with ViewRanger Skyline.
- [14] OkMap Software. (n.d.). Mountains AR, to discover mountains with iPhone and iPad. Okmap.org. Recuperado el 20 de noviembre del 2023, de https://www.okmap.org/en/en_mountainsARFeatures.aspx

- [15] ArtLens App. (2012, November 16). Cleveland Museum of Art. <https://www.clevelandart.org/artlens-gallery/artlens-app>
- [16] iCandi Apps ®. (n.d.). Icandiapps.com. Recuperado el 20 de noviembre del 2023, de <https://icandiapps.com/night-sky-4/>
- [17] Hardware de HoloLens 2. (n.d.). Microsoft.com. Recuperado el 20 de noviembre del 2023, de <https://learn.microsoft.com/es-es/hololens/hololens2-hardware>
- [18] (N.d.). Meta.com. Recuperado el 20 de noviembre del 2023, de <https://www.meta.com/quest/quest-3/>
- [19] Apps de productividad de Apple Vision Pro. Recuperado el 10 de octubre del 2023, de <https://www.apple.com/apple-vision-pro/>
- [20] Scripting: qué es y lenguajes. (n.d.). Santanderopenacademy.com. Recuperado el 1 de abril del 2024, de <https://www.santanderopenacademy.com/es/blog/scripting.html>
- [21] Unity Technologies. (n.d.). Prefabs. Unity Learn. Recuperado el 1 de abril 1, 2024, de <https://learn.unity.com/tutorial/prefabs-e>
- [22] Unity Technologies. (n.d.-b). Scenes. Unity3D.com. Recuperado el 1 de abril del 2024, de <https://docs.unity3d.com/Manual/CreatingScenes.html>
- [23] Splash screen. (n.d.). Cambridge.org. Recuperado el 2 de abril del 2024, de <https://dictionary.cambridge.org/us/dictionary/english/splash-screen>
- [24] Unity Technologies. (n.d.). UnityWebRequest. Unity3d.com. Recuperado el 2 de abril del 2024, de <https://docs.unity3d.com/es/530/Manual/UnityWebRequest.html>
- [25] Peres, S. & Phillips, Ronald. (2013). Validation of the System Usability Scale (SUS). Proceedings of the Human Factors and Ergonomics Society Annual Meeting. 57. 192-196. 10.1177/1541931213571043.
- [26] Saldaño, V., Martín, A., Gaetan, G., & Vilte, D. (2014). Focusing on older web users: An experience in Patagonia, Argentina. *International Journal on Advances in Internet Technology*, 7, 172–182.
- [27] Amaro, C. G. (2 de julio del 2004). Rescate del testimonio de los moradores del cajete de agua “Nemi cacitl atl.”
- [28] Necaxa · Nuevo necaxa, Puebla, Mexico. (n.d.). Necaxa · Nuevo Necaxa, Puebla, Mexico. Recuperado el 24 de marzo del 2024, de https://www.google.com.mx/maps/place/Necaxa,+Nuevo+Necaxa,+Pue./@20.211971,-98.0023669,71m/data=!3m1!1e3!4m6!3m5!1s0x85d0891bb08c808f:0x8588e8146632dfda!8m2!3D20.2113026!4d-98.0117493!16s%2Fg%2F11c265fh_z?entry=ttu

- [29] Unity Technologies. (n.d.). Sprites. Unity3D.com. Recuperado el 4 de junio del 2024, de <https://docs.Unity3D.com/es/530/Manual/Sprites.html>
- [30] Unity Technologies. (n.d.-a). Flujo de trabajo de los Assets (Asset Workflow). Unity3D.com. Recuperado el 4 de junio del 2024, de <https://docs.Unity3D.com/es/530/Manual/AssetWorkflow.html>
- [31] Unity Technologies. (n.d.-b). PlayerPrefs. Unity3D.com. Recuperado el 5 de junio del 2024, de <https://docs.Unity3D.com/ScriptReference/PlayerPrefs.html>
- [32] Class TMP_Text. (n.d.). Unity3D.com. Recuperado el 4 de junio del 2024, de https://docs.Unity3D.com/Packages/com.Unity.textmeshpro@1.0/api/TMPro.TMP_Text.html
- [33] NetMentor. (n.d.). Cómo utilizar Yield en programación. <https://www.netmentor.es/entrada/yield-programacion>. Recuperado el 28 de mayo del 2024, de <https://www.netmentor.es/entrada/yield-programacion>
- [34] Unity Technologies. (n.d.-b). ScriptableObject. Unity3D.com. Recuperado el 4 de junio del 2024, de <https://docs.Unity3D.com/Manual/class-ScriptableObject.html>
- [35] Puchades, D. (2019, April 15). Frame. Geekno. <https://www.geekno.com/glosario/frame>
- [36] Unity Technologies. (n.d.-b). Scenes. Unity3D.com. Recuperado el 5 de junio del 2024, de <https://docs.Unity3D.com/Manual/CreatingScenes.html>
- [37] Unity Technologies. (n.d.-c). ScriptableObject. Unity3D.com. Recuperado el 4 de junio del 2024, de <https://docs.Unity3D.com/Manual/class-ScriptableObject.html>
- [38] Unity Technologies. (n.d.-c). Script serialization. Unity3D.com. Recuperado el 5 de junio del 2024, de <https://docs.Unity3D.com/Manual/script-Serialization.html>
- [39] Fernández, Y. (2019, October 30). Qué es Github y qué es lo que le ofrece a los desarrolladores. Xataka.com; Xataka Basics. <https://www.xataka.com/basics/que-github-que-que-le-ofrece-a-desarrolladores>
- [40] Acerca de los repositorios. (n.d.). <https://docs.github.com/es/repositories/creating-and-managing-repositories/about-repositories>
- [41] Dhande, M. (2023, January 12). Getting lost with GPS? Don't worry, Google Maps VPS is here. Geospatial World; Geospatial Media & Communications. <https://www.geospatialworld.net/prime/business-and-industry-trends/what-is-visual-positioning-system-vps/>

- [42] ¿Qué es PHP? (n.d.). PHP.net. Recuperado el 5 de junio del 2024, de <https://www.PHP.net/manual/es/intro-what-is.PHP>
- [43] (N.d.). Swhosting.com. Recuperado el 11 de junio del 2024, de <https://www.swhosting.com/es/comunidad/manual/PHPmailer>
- [44] BillWagner. (n.d.). Instrucciones de control de excepciones: throw y try, catch, finally - C# reference. Microsoft.com. Recuperado el 12 de junio del 2024, de <https://learn.microsoft.com/es-es/dotnet/csharp/language-reference/statements/exception-handling-statements>
- [45] Niantic lightship. (n.d.). Lightship.dev. Recuperado el 25 de Abril del 2024, de <https://lightship.dev/>
- [46] ¿Qué es una API? (n.d.). OVHcloud. Recuperado el 6 de noviembre del 2024, de <https://www.ovhcloud.com/es/learn/what-is-api/>
- [47] Carrasco, A. C. (s/f). ¿Qué es un motor de videojuegos? Upm.es. Recuperado el 6 de noviembre de 2024, de <https://blogs.upm.es/observatoriogate/2018/07/04/que-es-un-motor-de-videojuegos/>
- [48] Realidad Virtual: otro mundo al alcance de tus ojos. Iberdrola.com. Recuperado el 6 de noviembre de 2024, de <https://www.iberdrola.com/innovacion/realidad-virtual>
- [49] Realidad Aumentada: el mundo real con otros ojos. Iberdrola.com. Recuperado el 6 de noviembre del 2023, de <https://www.iberdrola.com/innovacion/que-es-realidad-aumentada>
- [50] ¿Para qué sirve la geolocalización en instituciones financieras? (2022, octubre 6). Docusign.com. <https://www.docusign.com/es-mx/blog/geolocalizacion>
- [51] Constant, I. (2020, julio 10). Motion graphics: ¿qué es y para qué sirve? Domestika. <https://www.domestika.org/es/blog/4241-motion-graphics-que-es-y-para-que-sirve>
- [52] ¿qué es un render? (s/f). Renders en México. Recuperado el 6 de noviembre del 2023, de <https://www.arqing-mexico.com/renderers/qu%C3%A9-es-un-render/>
- [53] Universidad de Guanajuato. (2022, marzo 30). Clase digital 3. Lenguajes de desarrollo para dispositivos móviles. Recursos Educativos Abiertos; Sistema Universitario de Multimodalidad Educativo (SUME) - Universidad de Guanajuato. <https://blogs.ugto.mx/rea/clase-digital-3-lenguajes-de-desarrollo-para-dispositivos-moviles/>

[54] Zermeño, R. M. (2023, enero 25). ¿Qué significa C++? Este es el origen del nombre del famoso lenguaje de programación. Azul School. <https://www.azulschool.net/que-significa-c-este-es-el-origen-del-nombre-del-famoso-lenguaje-de-programacion/>

[55] ¿Qué es un SDK?. Amazon.com. Recuperado el 6 de noviembre de 2024, de <https://aws.amazon.com/es/what-is/sdk/#:~:text=Un%20kit%20de%20desarrollo%20de,o%20lenguaje%20de%20programaci%C3%B3n%20espec%C3%ADficos.>