



ESPE
UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA

Diseño e implementación de un Sistema de Internet de las cosas (IoT) con realidad aumentada para el monitoreo de un banco de calibración de medidores de caudal de la empresa ESRW S.A.

Quintero Tanguila, Carlos Andrés

Departamento de Ciencias de la Energía y Mecánica

Carrera de Ingeniería Mecatrónica

Trabajo de titulación, previo a la obtención del Título de Ingeniero Mecatrónico

Ing. Singaña Amaguaña, Marco Adolfo

18 de agosto de 2023

Latacunga

Reporte de verificación de contenido

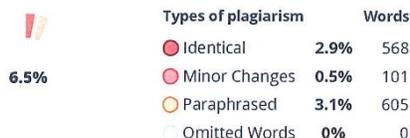


TESIS - QUINTERO CARLOS_rev.pdf

Scan details

Scan time: August 18th, 2023 at 20:42 UTC Total Pages: 79 Total Words: 19710

Plagiarism Detection



AI Content Detection

N/A

Text coverage
 AI text
 Human text

Plagiarism Results: (105)

- Open Access proceedings Journal of Physics: Confer...** 0.7%

<https://iopscience.iop.org/article/10.1088/1757-899x/743/1/0...>

George Evans

IOP Conference Series: Materials Science and Engineering PAPER • OPEN ACCESS Development of Internet of Things (IOT) Based Electronic R...
- Internet of Things (IoT): A Literature Review** 0.6%

https://file.scirp.org/pdf/jcc_2015052516013923.pdf

Somayya Madakam, R. Ramaswamy, Siddharth Tripathi

Journal of Computer and Communications, 2015, 3, 164-173 Published Online May 2015 in SciRes. <http://www.scirp.org/journal/jcc> <http://dx...>
- PLANIFIQUEMOS UN EXPERIMENTO PARA DEMOSTR...** 0.4%

<https://brainly.lat/tarea/37881308>

valeriavictoria78 valeriavictoria78 27.03.2021 Biología Se...

Ing. Singaña Amaguaña, Marco Adolfo
 CC.: 0501627889
 Director



Departamento de Ciencias de la Energía y Mecánica

Carrera de Ingeniería Mecatrónica

Certificación

Certifico que el trabajo de titulación: “**Diseño e implementación de un Sistema de Internet de las cosas (IoT) con realidad aumentada para el monitoreo de un banco de calibración de medidores de caudal de la empresa ESRAW S.A**”, fue realizado por el señor **Quintero Tanguila, Carlos Andrés**, el mismo que cumple con los requisitos legales, teóricos, científicos, técnicos y metodológicos establecidos por la Universidad de las Fuerzas Armadas ESPE, además fue revisado y analizado en su totalidad por la herramienta de prevención y/o verificación de similitud de contenidos; razón por la cual me permito acreditar y autorizar para que lo sustente públicamente.

Latacunga, 18 de agosto del 2023

A handwritten signature in blue ink, appearing to read 'Marco Adolfo Singaña Amaguaña', is positioned above a horizontal dotted line.

Ing. Singaña Amaguaña, Marco Adolfo

C.C.: 0501627889



Departamento de Ciencias de la Energía y Mecánica

Carrera de Ingeniería Mecatrónica

Responsabilidad de Autoría

Yo, **Quintero Tanguila, Carlos Andrés**, con cédula de ciudadanía n° **2200114417**, declaro que el contenido, ideas y criterio del trabajo de titulación: **“Diseño e implementación de un Sistema de Internet de las cosas (IoT) con realidad aumentada para el monitoreo de un banco de calibración de medidores de caudal de la empresa ESRW S.A.”** es de mi autoría y responsabilidad, cumpliendo con los requisitos legales, teóricos, científicos, técnicos y metodológicos establecidos por la Universidad de las Fuerzas Armadas ESPE, respetando los derechos intelectuales de terceros y referenciando las citas bibliográficas.

Latacunga, 18 de agosto de 2023



.....
Quintero Tanguila, Carlos Andrés

C.C.: 2200114417



Departamento de Ciencias de la Energía y Mecánica

Carrera de Ingeniería Mecatrónica

Autorización de Publicación

Yo, **Quintero Tanguila, Carlos Andrés**, con cédula de ciudadanía n° **2200114417**, autorizo a la Universidad de las Fuerzas Armadas ESPE publicar el trabajo de titulación: “**Diseño e implementación de un Sistema de Internet de las cosas (iot) con realidad aumentada para el monitoreo de un banco de calibración de medidores de caudal de la empresa ESRAW S.A.**” en el Repositorio Institucional, cuyo contenido, ideas y criterios son de mi responsabilidad.

Latacunga, 18 de agosto de 2023



.....
Quintero Tanguila, Carlos Andrés

C.C.: 2200114417

Dedicatoria

El presente trabajo se lo dedico a Dios, a mi familia, de manera especial a mi padre y madre a quienes admiro por formar mi persona con valores, por sus sabios consejos, confianza y apoyo condicional necesarios para tomar fortaleza y culminar este proyecto.

A mis hermanos y amigos que de una u otra forma han estado presentes en los buenos y malos momentos durante mi período de formación profesional, desarrollo y culminación del proyecto.

QUINTERO TANGUILA CARLOS ANDRÉS

Agradecimientos

Agradezco a Dios que está siempre conmigo Dios, por ser parte y guía en mi vida.

A mis padres por siempre motivarme a mejor y apoyarme durante toda mi vida.

A la Universidad de las Fuerzas Armadas ESPE, por su formación académica y consolidar mis conocimientos profesionales, en especial al Ingeniero Marco Adolfo Singaña Amaguaña por su valioso aporte y colaboración en la terminación de este proyecto.

A la compañía Esraw SA., por las facilidades brindadas en la elaboración de este proyecto, de manera especial a todos y a cada uno de los trabajadores de la empresa.

QUINTERO TANGUILA CARLOS ANDRÉS

ÍNDICE DE CONTENIDOS

Carátula	1
Reporte de verificación de contenido.....	2
Certificación	3
Responsabilidad de Autoría.....	4
Autorización de Publicación	5
Dedicatoria	6
Agradecimientos.....	7
Índice de contenidos	8
Índice de tablas.....	13
Índice de figuras	14
Resumen.....	18
Abstract	19
Capítulo I: Introducción.....	20
Antecedentes.....	20
Planteamiento del problema.....	21
Descripción resumida del proyecto	23
Justificación e importancia	24
Objetivos.....	25
<i>Objetivo general</i>	25
<i>Objetivos específicos</i>	25
Hipótesis	26

Variables de la investigación.....	26
<i>Variable dependiente</i>	26
<i>Variable independiente</i>	26
Capítulo II: Marco Teórico	27
Internet de las cosas (IoT)	27
<i>Arquitectura IoT</i>	28
Realidad aumentada.....	29
<i>Componentes de Realidad Aumentada (AR)</i>	30
<i>Herramientas de desarrollo de realidad aumentada</i>	31
<i>Seguimiento sin marcadores</i>	32
<i>Seguimiento basado en marcadores</i>	33
Monitorización Industrial	33
Android	34
<i>Android Studio</i>	34
<i>Kotlin</i>	35
Google Firebase o Plataforma Firebase	35
<i>Firebase Authentication</i>	36
<i>Firebase Realtime Database</i>	37
Google Apps Script.....	37
Google Looker Studio	38
Tarjeta UNO WiFi R3 ATmega328P ESP8266.....	39

Metrología	41
<i>Calibración</i>	<i>41</i>
<i>Datos Primarios</i>	<i>42</i>
<i>Medidor de Temperatura</i>	<i>42</i>
<i>Medidor de Presión.....</i>	<i>43</i>
<i>Medidor de caudal</i>	<i>43</i>
Capítulo III: Diseño e implementación.....	45
Parámetros generales de diseño.....	45
Diagrama de Conexión.....	46
<i>Arquitectura Lógica</i>	<i>47</i>
<i>Arquitectura Física</i>	<i>47</i>
Software y lenguaje de programación	48
Base de Datos con Firebase.....	49
Obtención de datos de los sensores del Banco de Pruebas.....	52
<i>Programación en Arduino</i>	<i>53</i>
Diseño del complemento AR	55
<i>Marcadores.....</i>	<i>55</i>
<i>Modelos 3D de los estados de los sensores</i>	<i>57</i>
<i>Base de Datos para Realidad Aumentada.....</i>	<i>59</i>
<i>Implementación en Unity.....</i>	<i>67</i>
<i>Exportación de Herramienta AR como librería Unity</i>	<i>76</i>

<i>Importación de librería Unity correspondiente a la herramienta AR.....</i>	79
Diseño de la aplicación Android	85
<i>Splash Screen</i>	85
<i>Conexión de Firebase con Android Studio</i>	90
<i>Autenticación</i>	95
<i>Barra navegación.....</i>	98
<i>Pantalla Menú.....</i>	101
<i>Pantalla Monitoreo en Tiempo Real.....</i>	102
<i>Interfaz de registro de datos</i>	104
<i>Notificaciones y eventos</i>	109
<i>Interfaz de AR.....</i>	111
Capítulo IV: Pruebas y resultados	113
Pruebas de Autenticación de Usuario	113
Pruebas de Monitoreo en Tiempo Real.....	114
Pruebas de Informes Generados.....	120
Prueba de generación de Notificaciones	128
Prueba de reconocimiento del estado del sensor.....	131
Validación de la hipótesis.....	133
Alcances	138
Limitaciones	138
Capítulo V: Conclusiones y Recomendaciones.....	139

Conclusiones.....	139
Recomendaciones.....	141
Bibliografía.....	142
Anexos.....	149

ÍNDICE DE TABLAS

Tabla 1 <i>Especificaciones de la tarjeta UNO WIFI R3 ATmega328P ESP8266</i>	40
Tabla 2 <i>Modos de trabajo de Tarjeta UNO WiFi R3 ATmega328P ESP8266</i>	41
Tabla 3 <i>Características de Software y lenguaje</i>	48
Tabla 4 <i>Comparación de mediciones de Presión</i>	115
Tabla 5 <i>Comparación de mediciones de Caudal</i>	117
Tabla 6 <i>Comparación de mediciones de Temperatura</i>	118
Tabla 7 <i>Comparación de mediciones de Frecuencia</i>	119
Tabla 8 <i>Comparación de datos de Registro en Frecuencia – Prueba 1</i>	120
Tabla 9 <i>Comparación de datos de Registro en Presión – Prueba 1</i>	121
Tabla 10 <i>Comparación de datos de Registro en Temperatura – Prueba 1</i>	122
Tabla 11 <i>Comparación de datos de Registro en Caudal – Prueba 1</i>	122
Tabla 12 <i>Comparación de datos de Registro en Frecuencia – Prueba 2</i>	125
Tabla 13 <i>Comparación de datos de Registro en Presión – Prueba 2</i>	125
Tabla 14 <i>Comparación de datos de Registro en Temperatura - Prueba 2</i>	126
Tabla 15 <i>Comparación de datos de Registro en Caudal - Prueba 2</i>	126
Tabla 16 <i>Prueba de notificaciones</i>	131
Tabla 17 <i>Mediciones del sensor de caudal</i>	134
Tabla 18 <i>Condiciones presentadas para realizar la validación por el método de T-Student</i>	135
Tabla 19 <i>Tabla de distribución t de Student</i>	136

ÍNDICE DE FIGURAS

Figura 1 <i>Arquitectura IoT</i>	28
Figura 2 <i>Head Mounted - Display</i>	31
Figura 3 <i>Ejemplo de marcadores</i>	33
Figura 4 <i>UNO WiFi R3 ATmega328P ESP8266</i>	39
Figura 5 <i>Indicador de temperatura</i>	42
Figura 6 <i>Transmisor Rosemount 3051T en línea</i>	43
Figura 7 <i>Caudalímetro electromagnético</i>	44
Figura 8 <i>Diagrama de Conexión</i>	47
Figura 9 <i>Creación de la base de datos en Firebase</i>	50
Figura 10 <i>Reglas de Base de datos</i>	51
Figura 11 <i>Secreto de base de datos</i>	51
Figura 12 <i>Diagrama de conexión de las señales de sensores</i>	52
Figura 13 <i>Conexión de señales de sensores del banco de pruebas de caudal</i>	53
Figura 14 <i>Programación lectura de señales del banco de pruebas</i>	54
Figura 15 <i>Programación Arduino-Firebase</i>	55
Figura 16 <i>Marcador de señal de Temperatura</i>	56
Figura 17 <i>Marcador de señal Presión</i>	56
Figura 18 <i>Marcador de la señal de Caudal</i>	57
Figura 19 <i>Marcador de la señal de Frecuencia</i>	57
Figura 20 <i>Aviso en 3D de señal de estado de la señal</i>	58
Figura 21 <i>Aviso de estado de Apagado del sensor</i>	58
Figura 22 <i>Aviso de estado de Encendido del sensor</i>	59
Figura 23 <i>Aviso de estado de ALERTA del sensor</i>	59
Figura 24 <i>Pasos en Vuforia para creación de Base de Datos</i>	60
Figura 25 <i>Creación de Base de Datos Vuforia</i>	60

Figura 26 <i>Add Target</i>	62
Figura 27 <i>Descarga de Base de Datos Vuforia</i>	63
Figura 28 <i>Importar Base de datos Vuforia</i>	64
Figura 29 <i>Creación de AR Camera</i>	65
Figura 30 <i>Clave de Licencia Vuforia</i>	66
Figura 31 <i>Creación de Image Target</i>	67
Figura 32 <i>Cambio de nombre a los Image Target</i>	68
Figura 33 <i>Selección de marcador en Image Target</i>	69
Figura 34 <i>Objetos 3D de avisos de estados de sensores</i>	70
Figura 35 <i>Objetos 3D dentro de las Image Target</i>	71
Figura 36 <i>Declaración de GameObjects para cada sensor</i>	72
Figura 37 <i>Programación para activar o desactivar objetos 3D de presión</i>	73
Figura 38 <i>Asignación de funciones de activación y desactivación con Image Target</i>	74
Figura 39 <i>Asignación de objetos 3D al Script Database</i>	75
Figura 40 <i>Acceso a la configuración compilación para exportar</i>	76
Figura 41 <i>Build Settings de proyecto Unity</i>	77
Figura 42 <i>Exportación de proyecto Unity</i>	78
Figura 43 <i>Proyecto de Unity para exportar como librería</i>	79
Figura 44 <i>Pasos para abrir el proyecto en Android Studio</i>	80
Figura 45 <i>Pasos para la importación de librería Unity</i>	81
Figura 46 <i>Configuración del archivo settings.gradle</i>	82
Figura 47 <i>Corrección en la estructura de AndroidManifest</i>	83
Figura 48 <i>Corrección de código de build.gradle</i>	84
Figura 49 <i>Sincronizar estructura de herramienta AR</i>	85
Figura 50 <i>Pantalla de Splash Screen</i>	86
Figura 51 <i>Exportar imágenes de Adobe XD con diferentes densidades</i>	87

Figura 52 <i>Dirección de las carpetas a cargar las imágenes de splash screen</i>	87
Figura 53 <i>Líneas de código para Splash Screen</i>	88
Figura 54 <i>Identificación de la pantalla inicial</i>	89
Figura 55 <i>Inicialización de Splash Screen</i>	89
Figura 56 <i>Primera parte para agregar el servicio Firebase a Android</i>	90
Figura 57 <i>Identificación del nombre de proyecto</i>	91
Figura 58 <i>Pasos para hallar la huella digital SHA1 de depuración</i>	92
Figura 59 <i>Segunda parte para agregar el servicio Firebase</i>	93
Figura 60 <i>Dependencia en “build.gradle Project” para servicio Firebase</i>	94
Figura 61 <i>Plugin y dependencia en “build.gradle Module” para servicio Firebase</i>	95
Figura 62 <i>Autenticación con Firebase para ESRAW S.A.</i>	96
Figura 63 <i>Codificación para autenticación de Usuarios de ESRAW S.A.</i>	97
Figura 64 <i>Pantalla de Autenticación</i>	98
Figura 65 <i>Barra de navegación</i>	99
Figura 66 <i>Codificación de barra de navegación</i>	99
Figura 67 <i>Codificación de la función navegar entre pantallas desde la barra de navegación</i> .	100
Figura 68 <i>Implementación de función de navegación entre pantallas opción Monitor</i>	101
Figura 69 <i>Pantalla Menú</i>	102
Figura 70 <i>Codificación para obtener valores de la base de datos</i>	103
Figura 71 <i>Pantalla de Monitoreo en tiempo real</i>	104
Figura 72 <i>Registro de Datos en Looker Studio</i>	105
Figura 73 <i>Codificación para WebView</i>	105
Figura 74 <i>Interfaz de Registro de Datos</i>	106
Figura 75 <i>Codificación para obtener información de la base de Datos de Firebase</i>	107
Figura 76 <i>Envío de datos de Firebase a la hoja de cálculo de Google</i>	108
Figura 77 <i>Implementación de datos en Hoja de Cálculo de Google</i>	108

Figura 78 <i>Pasos para crear un servicio en Android Studio</i>	110
Figura 79 <i>Codificación de ForegroundService</i>	111
Figura 80 <i>Comunicación entre Unity y Android</i>	112
Figura 81 <i>Asignación de estados de los sensores</i>	112
Figura 82 <i>Ingreso de usuario erróneo</i>	113
Figura 83 <i>Autenticación de Usuario exitosa</i>	114
Figura 84 <i>Comparación de valor de magnitud de presión</i>	115
Figura 85 <i>Comparación de valor de magnitud Caudal</i>	116
Figura 86 <i>Comparación de valor de magnitud Temperatura</i>	117
Figura 87 <i>Comparación de valor de magnitud Frecuencia</i>	119
Figura 88 <i>Registro de datos primarios de ESRAW S.A. para prueba 1</i>	123
Figura 89 <i>Registro de Datos del Banco de Calibración del 03/02/2022</i>	124
Figura 90 <i>Registro de datos primarios de ESRAW S.A. para prueba 2</i>	127
Figura 91 <i>Registro de Datos del Banco de Calibración del 04/02/2022</i>	128
Figura 92 <i>Notificación Inicial de encendido</i>	129
Figura 93. <i>Notificación de Apagado del Banco de Calibración</i>	129
Figura 94 <i>Notificación de estado de alerta</i>	130
Figura 95 <i>Herramienta AR detectando el sensor en estado de apagado</i>	131
Figura 96 <i>Herramienta AR detectando el sensor en estado de alerta</i>	132
Figura 97 <i>Herramienta AR detectando el sensor en estado de encendido</i>	133

Resumen

En el presente proyecto se realiza la implementación de un sistema internet de las cosas (IOT) complementándolo con el uso de realidad aumentada para monitorear, digitalizar y generar eventos relevantes del procedimiento de calibración de medidores de caudal en la empresa ESRAW S.A.; a través de la creación de una aplicación móvil para dispositivos Android, en la cual se puede visualizar en tiempo real las mediciones de los sensores del banco de calibración, además de revisar los registros de datos obtenidos durante todo el proceso y obtener ayuda en la identificación de equipos con realidad aumentada. La adquisición de los datos de las magnitudes a monitorear del banco de calibración es obtenida por una tarjeta UNO WiFi R3 ATmega328P ESP8266 que a su vez es la encargada de enviar los datos a la base de datos por medio de conexión WiFi. Mediante la utilización de servicios de Google Database, App Scripts y Firebase, se crea una base de datos y generan registros. La App notifica de algún evento relevante como son encendido o apagado, alarmas, desconexión de equipos. Por medio de la integración de una librería de Unity a la App, se da la función de realidad aumentada, que permite al operador reconocer que elementos del banco de calibración se encuentran fallando. El sistema no necesita de un computador o tarjeta procesadora dado que se utilizan los recursos del dispositivo Android y servicios de Google que se encuentran en línea, el sistema solo necesita estar conectado a una red con conexión a internet.

Palabras clave: IoT, realidad aumentada, monitoreo en tiempo real, base de datos
Firebase.

Abstract

In the present project, the implementation of an Internet of Things (IOT) system is carried out, complementing it with the use of augmented reality to monitor, digitize and generate relevant events of the flow meter calibration procedure in the company ESRAW S.A.; through the creation of a mobile application for Android devices, in which the measurements of the calibration bench sensors can be viewed in real time, in addition to reviewing the data records obtained throughout the process and obtaining help in the identification equipment with augmented reality. The acquisition of the data of the magnitudes to be monitored from the calibration bench is obtained by a UNO WiFi R3 ATmega328P ESP8266 card, which in turn is in charge of sending the data to the database through a WiFi connection. By using Google Database services, App Scripts and Firebase, a database is created and records are generated. The App notifies of any relevant event such as power on or off, alarms, equipment disconnection. Through the integration of a Unity library to the App, the augmented reality function is given, which allows the operator to recognize which elements of the calibration bench are failing. The system does not need a computer or processor card since the resources of the Android device and Google services that are online are used, the system only needs to be connected to a network with an internet connection.

Key words: IoT, augmented reality, real time monitoring, Firebase database.

Capítulo I

Introducción

Antecedentes

Hoy en día el uso de las Tecnologías de la Información y la Comunicación (TIC), y especialmente el Internet de las Cosas (IoT), en la industria se han vuelto ineludibles, principalmente por ser vitales para incrementar la eficiencia organizacional y su nivel de competitividad. Esto ha impulsado la adopción de las TIC y la IoT en la mayoría de las actividades de la industria, lo que ha llevado al embrión de lo que es conocida como Industria 4.0 (también llamada Internet industrial de las cosas - IIoT). Es decir, la industria 4.0 busca reinventar la interacción entre trabajadores y máquinas mediante la integración de fabricación, TIC y sistemas ciber-físicos. (Alonso et al., 2020)

Una aplicación de esta nueva tecnología, es un “Sistema IoT inalámbrico de baja potencia para monitoreo y alertas en tiempo real en ambientes interiores”, en la que (Fonović et al. 2021) se desarrolló un sistema de monitoreo de ambiente interior en tiempo real basado en código abierto para recopilar, almacenar y visualizar datos con un sistema de notificación de alerta simple, demuestran que sistemas IoT basados en código abierto, de bajo costo y bajo consumo de energía pueden proporcionar una alta confiabilidad al tiempo que garantizan un bajo consumo de energía.

La realidad aumentada se ha desarrollado como una de las tecnologías más importantes con referencia a obtener un apoyo didáctico, buscando de esta manera que un estudiante pueda desarrollar y utilizar un conocimiento sobre un fenómeno que pueda ser observado. (Zarate Nava, Mendoza González, & Aguilar Galicia, 2013)

En la última década la realidad aumentada y su programación ha ido en crecimiento en distintas zonas: a nivel industrial, educativo, servicios y recreativos. Los indicios de la aparición de realidad aumentada fueron dentro del mundo cinematográfico y fotografía. (Abud Figueroa, 2012)

Una de las amplias aplicaciones y un claro ejemplo es el “MagicBook” o “El Libro de los Hechizos” (Fernández, 2013) , que implementan realidad aumentada incursionado en la educación obteniendo excelentes resultados. En diversas aplicaciones industriales y domésticas, existe una amplia gama de datos disponibles que están vinculados a elementos presentes en el entorno físico. En este contexto, la realidad aumentada emerge como la herramienta que fusiona y amalgama eficazmente esta información con los objetos tangibles del mundo real. (Basagain et al., 2014)

Otro ejemplo de aplicación con realidad aumentada es el trabajo de “Programación integrada de producción y mantenimiento a través del monitoreo de máquinas y la realidad aumentada: un enfoque de la Industria 4.0” (Mourtzis et al., 2017), desarrolla un sistema integrado, bajo el concepto de Industria 4.0, consistente en una herramienta de monitoreo de máquina herramienta y una aplicación móvil de realidad aumentada, que están interconectados con una herramienta de programación de taller. La aplicación móvil permite al operador monitorear el estado de la máquina en base a los datos de la herramienta de monitoreo y decida llamar inmediatamente al mantenimiento remoto de AR o programar tareas de mantenimiento para más tarde.

ESRAW S.A. es una empresa Servicios de Instrumentación que brinda servicios de Calibración, Inspección e Instrumentación para equipos de trabajo del sector industrial y petrolero, sus laboratorios se encuentran en la ciudad de El Coca (ESRAW, 2018), actualmente los procesos de calibración de sensores de caudal los realiza un técnico manualmente.

Planteamiento del problema

En 2018 se realizó una encuesta entre los 15 Institutos Metrológicos Nacionales (INM) de América Latina y El Caribe para analizar la situación en la que se encuentran en tecnología en relación a la industria 4.0 (Valqui, Csaburi, & Suaznabar, 2019), en los cuales se llegó a la conclusión que dos tercios de los INM entrevistados manifestó tener los procesos de calibración digitalizados y automatizados; y casi dos tercios de los INM entrevistados declaró

tener la mayoría o todos los laboratorios con monitoreo y control ambiental digitalizado y automatizado. Este tema representa el estado más vanguardista en relación con la digitalización de los servicios metrológicos. Un quinto de los Institutos Nacionales de Metrología (INM) encuestados indicó que no ha implementado ninguna iniciativa para la atención digital al cliente, mientras que otro quinto manifestó carecer de un servicio de calibración para instrumentos de medición con componente digital.

Por ende, los centros de metrología deben de invertir en actualizar sus métodos de calibración para estar a la vanguardia de la tecnología actual.

En nuestro país, la utilización de Sistemas IoT con Realidad Aumentada se limita a un reducido grupo de estudiantes y profesionales que vienen desarrollando proyectos de este tipo de sistemas y sus respectivos complementos de manera personal.

La empresa ESRAW S.A., actualmente realiza procesos de calibración de instrumentos y equipos para el sector petrolero de la provincia de Orellana, los cuales son realizados por técnicos que realizan mediciones en puntos designados y posteriormente suben los datos obtenidos a una planilla. Debido a que realizan diferentes procesos de calibración, un técnico debe estar concentrado en qué magnitudes operan y cómo las evalúan, pero no están exentos a cometer errores, tanto al tomar mediciones como al ingresar los datos en la planilla, dado que si esto sucede puede ocasionar problemas tanto en la empresa cliente como en la empresa que calibra.

La implementación de nuevas tecnologías tales como realidad aumentada en los procesos industriales o en el sector de metrología es una potente herramienta para aumentar la competitividad en las empresas, así como también el uso de la industria 4.0 que consiste en obtener un monitoreo de diferentes variables dentro de una planta de producción/procesos de manera remota y sencilla, por tal motivo la empresa ESRAW S.A. se ha fijado en mejorar sus servicios implementando nuevas tecnologías.

Descripción resumida del proyecto

El presente proyecto busca realizar la implementación de un sistema de internet de las cosas (IOT) complementándolo con el uso de realidad aumentada para monitorear, digitalizar y generar eventos relevantes del procedimiento de calibración de medidores de caudal en la empresa ESRAW S.A.

El proyecto inicia con una fase de investigación bibliográfica, la cual consistirá en recolectar información referente a la aplicabilidad de un sistema IoT dentro de plantas de aprendizaje o a nivel industrial, así como también desarrollar realidad aumentada como guías en el proceso de calibración.

Este proyecto está enfocado en proporcionar una aplicación móvil para dispositivos Android, la aplicación proporciona al supervisor o técnico encargado una herramienta para monitorear en tiempo real y revisar los registros de datos obtenidos durante todo el proceso de calibración de los medidores de caudal, así como también será notificado de algún evento relevante.

Los eventos más relevantes como alarmas, arranques, desconexión de equipos, entre otros, serán notificados en el celular en que se encuentre instalada la aplicación y será presentada en primer plano para tener mayor importancia y no dejar pasar por alto la información de gran trascendencia del banco de calibración.

La aplicación de monitoreo y su procesamiento no necesitan de un computador o tarjeta procesadora dado que se utilizarán herramientas y aplicaciones de Google que se encuentran disponibles para brindar servicio en línea; con la única necesidad de estar conectados a una red proveedora de internet.

Mientras que, en la parte de realidad aumentada, va a permitir al operador reconocer los elementos del banco de calibración e indicar si los sensores se encuentran operando o no, para que sea más fácil identificar si uno de ellos está fallando.

Finalmente, se realizarán las respectivas pruebas de funcionamiento del banco de calibración con un técnico y con el sistema IoT operando, se evaluará los datos obtenidos por un técnico, con las obtenidas por el sistema IoT implementado, se procederá a verificar tiempos de operación y porcentaje de error para determinar el grado de eficiencia logrado, esta aplicación busca obtener una realimentación del funcionamiento del sistema para realizar una puesta a punto del mismo y desarrollar la validación de la hipótesis planteada.

Justificación e importancia

La empresa ESRAW S.A. posee mucha competencia local, por lo que implementar nuevos servicios para mejorar los procesos de calibración e instrumentación es de gran importancia.

Se hacen indispensables mediciones confiables para obtener una mejora sustancial en la eficacia de los procedimientos de producción y la calidad de los productos en mercados que experimentan creciente competencia. (Valqui, Csaburi, & Suaznabar, 2019).

El proyecto es importante en virtud que las empresas dentro de diversas industrias están adoptando esta forma avanzada de interconexión de dispositivos, el propósito de enfrentar eficazmente el flujo de información y el procesamiento instantáneo de datos, con el objetivo de transformarse en entidades empresariales interconectadas. (Proaño, 2017)

Los sistemas IoT permiten que equipos industriales estén conectados entre sí, además presta servicios remotos de almacenamiento y gestión de datos.

La implementación de tecnología IoT es de gran importancia porque permite notificar a supervisores y operarios sobre eventos relevantes o incidentes que sucedan en los equipos, además de poder realizar una revisión en tiempo real de los datos que se están generando en el proceso de calibración. La realidad aumentada proporciona mayor comprensión de la información del procedimiento de calibración de los medidores de caudal, así como también disminuir el tiempo de capacitación al personal y guiar durante proceso para evitar accidentes y daños al equipo.

Implementando un sistema IoT con realidad aumentada se logrará mejorar la comunicación hombre-máquina, se obtendrá respaldos del proceso de calibración mediante la generación de registros, permitirá la digitalización de las variables del proceso datos y su revisión mediante internet en tiempo real.

Objetivos

Objetivo general

Diseñar e implementar un IoT con realidad aumentada para la digitalización, monitoreo y generación de eventos de un banco de calibración de medidores de caudal de la empresa ESRAW S.A.

Objetivos específicos

- Recopilar información necesaria para la correcta maniobrabilidad del equipo por parte del estudiante.
- Investigar acerca de IoT, técnicas de simulación de objetos en realidad aumentada, su importancia y características.
- Seleccionar componentes para la implementación del sistema IoT con respecto a su mecánica y electrónica.
- Diseñar e implementar esquemas gráficos en 3D de los componentes, marcas de reconocimiento y aplicación de realidad aumentada.
- Digitalizar y registrar variables del proceso: presión, temperatura, caudal y frecuencia del banco de calibración.
- Generar eventos a través de IoT: alarmas, arranque, paro del proceso, fallo de comunicación en la aplicación.
- Implementar el software de monitoreo de datos del banco de calibración.
- Evaluar el funcionamiento de la aplicación en tiempo real del banco de calibración de medidores de caudal.

- Validar que la implementación del Sistema IoT con realidad aumentada permitirá la digitalización, monitoreo y generación de eventos de un banco de calibración de medidores de caudal de la empresa ESRAW S.A.

Hipótesis

¿La implementación del Sistema Internet de las Cosas con realidad aumentada permitirá la digitalización, monitoreo y generación de eventos de un banco de calibración de medidores de caudal de la empresa ESRAW S.A.?

Variables de la investigación

Variable dependiente

Digitalización, monitoreo y generación de eventos de un banco de calibración de medidores de caudal.

Variable independiente

Sistema de Internet de las Cosas con realidad aumentada.

Capítulo II

Marco Teórico

Internet de las cosas (IoT)

Nunberg (2012) señala que el internet de las cosas es un cambio de paradigma novedoso en el ámbito de las tecnologías de la información. La frase *Internet de las cosas*, que también se conoce como IoT, se acuñó a partir de dos palabras, es decir, la primera palabra es *Internet* y la segunda palabra es *Cosas*. Internet es un conjunto mundial de sistemas de redes de computadoras interconectadas que emplean el protocolo estándar de Internet (TCP/IP) para atender a una amplia población global de usuarios. Se trata de una interconexión de múltiples redes, que incluye redes privadas, públicas, educativas, empresariales y gubernamentales, que varían en tamaño desde locales hasta globales. Estas redes se unen mediante diversas tecnologías electrónicas, inalámbricas y ópticas. Hoy en día, más de 100 países están vinculados a intercambios de datos, noticias y opiniones a través de Internet.

La mejor definición de IoT según (Madakam, Ramaswamy, & Tripathi, 2015) es: “Una red abierta e integral de objetos inteligentes que tienen la capacidad de autoorganizarse, compartir información, datos y recursos, reaccionando y actuando ante situaciones y cambios en el entorno” (p. 165).

Se considera que Internet de las cosas (IoT) es uno de los facilitadores de la próxima revolución industrial. Está impulsado por el avance de las tecnologías digitales, además de cambiar drásticamente la forma en que las empresas se involucran en las actividades comerciales y las personas interactúan con su entorno. (Vermesan & Friess, 2016)

Internet de las Cosas (IoT) trae consigo ventajas en la mejora de la administración y el monitoreo de activos y productos, incrementa la disponibilidad de datos informativos, facilita la optimización de equipos y la utilización eficiente de recursos, lo cual puede llevar a la reducción de gastos. Además, brinda la posibilidad de concebir dispositivos interconectados inteligentes novedosos y explorar modelos de negocios innovadores. (Salazar & Silvestre, 2016)

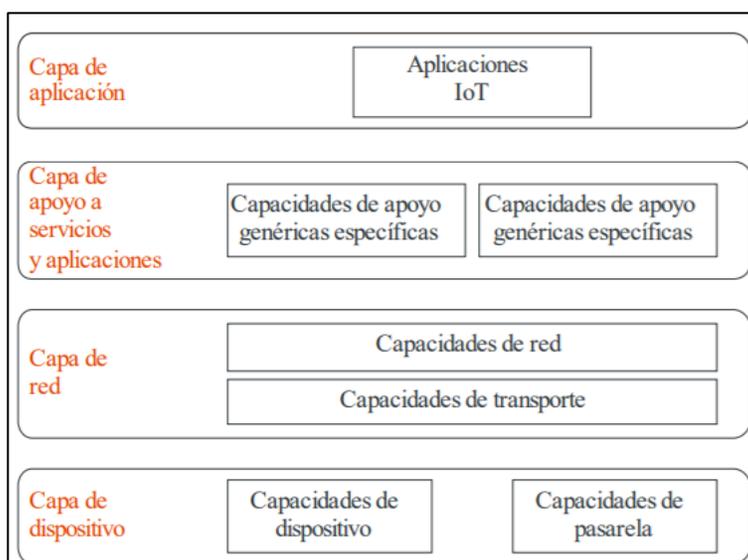
Arquitectura IoT

El Internet de las cosas posee varias arquitecturas en base a sus capas, en (UIT-T, 2012) definen un modelo como se muestra en la Figura 1, que consta de cuatro:

- Capa de aplicación
- Capa de apoyo a servicios y aplicaciones
- Capa de red
- Capa de dispositivo

Figura 1

Arquitectura IoT



Nota. Tomado de Overview of the Internet of things por (UIT-T, 2012)

Capa de aplicación

En esta capa se desarrollan aplicaciones y servicios que utilizan la información procesada para dar solución a problemas específicos. Las aplicaciones de monitoreo, control y automatización se encuentran entre los posibles usos para ellos. (Guinard & Trifa, 2009)

Capa de soporte de servicios y aplicaciones

La capa de soporte de servicios y aplicaciones consiste procesar, analizar y extraer datos para identificar patrones significativos. Esto puede abarcar técnicas de minería de datos, aprendizaje automático y análisis. (Gubbi, Buyya, Marusic, & Palaniswami, 2013)

Capa de red

La capa de red se encarga de establecer y gestionar la conectividad entre los dispositivos y sistemas, permitiendo la transferencia eficiente de datos en entornos heterogéneos y escalables. Incluye tecnologías como 5G, LPWAN, y protocolos de enrutamiento. (Borgia, 2014)

Capa de dispositivo

En esta capa se encuentran los sensores, actuadores, dispositivos embebidos. Esta capa tiene la función de interactuar directamente con la red de comunicación o los que interactúan indirectamente con la red de comunicación que necesitan de un dispositivo con función de pasarela. (UIT-T, 2012)

Realidad aumentada

La realidad aumentada (AR) emerge como una modalidad experiencial en la cual el entorno físico (RW) se enriquece mediante elementos generados por computadora que están asociados a ubicaciones o actividades específicas. En términos simples, AR posibilita que el contenido digital se integre de manera perfecta con nuestras percepciones del mundo real. Más allá de los objetos en 2D o 3D que se esperan comúnmente, los recursos digitales como archivos de audio y video, datos de texto e incluso información de carácter olfativo o táctil pueden fusionarse con la percepción del mundo real por parte de los usuarios. En conjunto, estas adiciones pueden contribuir a facilitar y enriquecer el entendimiento y conocimiento de las personas acerca de lo que está aconteciendo a su alrededor. (Yuen, Yaoyuneyong, & Johnson, 2011)

Componentes de Realidad Aumentada (AR)

Generador de escenas

El generador de escenas en realidad aumentada es un componente esencial que permite crear y representar objetos virtuales en un entorno del mundo real. Proporciona herramientas para diseñar, configurar y ubicar elementos tridimensionales y bidimensionales en el espacio físico, permitiendo la interacción de los usuarios con contenido virtual inmersivo.

(Dey et al., 2018)

Sistema de rastreo

En el contexto de la realidad aumentada (AR), un sistema de seguimiento hace referencia a una integración sinérgica de tecnologías, sensores y algoritmos especialmente concebidos para supervisar y registrar de forma exacta y en tiempo real la ubicación y orientación de elementos virtuales en relación con el entorno físico circundante. Esta estructura posibilita una superposición fluida y dinámica de información digital sobre la realidad con coherencia, lo cual enriquece la experiencia del usuario al crear un sentido inmersivo y persuasivo. (Klein & Murray, 2007)

Display

La tecnología para AR continúa en desarrollo y las soluciones se basan en decisiones de diseño. HMD (Head Mounted Display) es el dispositivo de visualización más utilizado para AR. (Desai et. al, 2020)

En la Figura 2, se muestra un ejemplo HMD.

Figura 2*Head Mounted - Display*

Nota. Tomado de (Desai et al., 2020)

Herramientas de desarrollo de realidad aumentada**ARKit**

ARKit es un framework de desarrollo de Apple que brinda a los desarrolladores las herramientas necesarias para crear experiencias de realidad aumentada (AR) envolventes en dispositivos iOS. Proporciona capacidades avanzadas de seguimiento de posición y orientación, detección de planos y objetos, y la posibilidad de interactuar con contenido digital dentro del entorno físico, lo que permite a los usuarios experimentar la realidad aumentada de manera realista y cautivadora. (Apple Inc., 2020)

ARCore

En el estudio de Chen et al. (2019) se define ARCore como la plataforma de software de Google para crear aplicaciones AR, que es similar al ARKit de Apple. Puede aprovechar los avances en el software de la nube y el hardware del dispositivo para llevar los objetos digitales al mundo real. Las funciones principales son la captura de movimiento, la percepción del entorno y la percepción de la fuente de luz.

Vuforia

Es una plataforma líder en el desarrollo de aplicaciones de realidad aumentada (AR) que ofrece a los desarrolladores herramientas y recursos para crear experiencias interactivas de AR en una variedad de dispositivos, incluidos teléfonos inteligentes, tabletas y lentes inteligentes. Utiliza tecnologías de seguimiento visual y detección de objetos para superponer contenido digital en el mundo real, lo que permite a los usuarios interactuar y explorar de manera envolvente. (Vuforia, 2023)

Wikitude SDK

Wikitude SDK es un kit de desarrollo de software (SDK) permite a los desarrolladores crear experiencias de realidad aumentada (AR) altamente interactivas. Esta plataforma ofrece capacidades avanzadas de detección de objetos, seguimiento de posición y orientación, y superposición de contenido digital en el entorno real. Wikitude SDK es utilizado para crear aplicaciones de AR que pueden funcionar en diversos dispositivos móviles y plataformas. (Wikitude GmbH, 2023)

Seguimiento sin marcadores

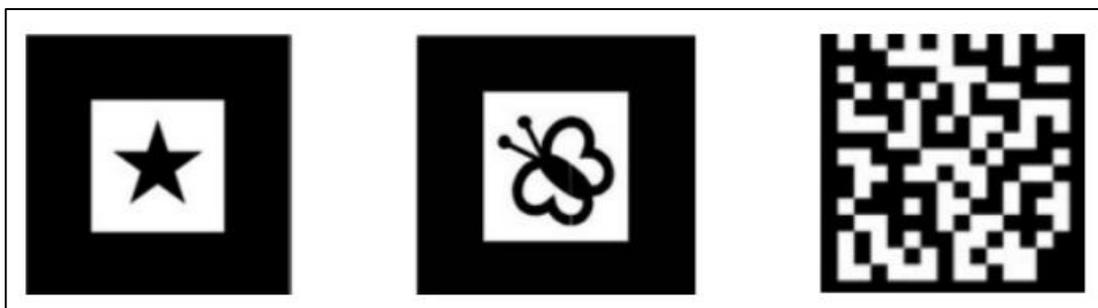
El rastreo sin utilizar marcadores se fundamenta en la identificación de puntos claves. Estos puntos claves deben ser identificables con rapidez, presentar una estabilidad considerable en diversas condiciones lumínicas o en casos de imágenes borrosas, ser resistentes ante variaciones en los ángulos de observación y deben exhibir características distintivas en una amplia gama de distancias entre el usuario y los objetos. En el contexto de los sistemas de realidad aumentada que carecen de marcadores, los métodos de detección se pueden categorizar en dos grupos: detectores de esquinas y detectores de manchas. (Daponte, et al., 2014)

Seguimiento basado en marcadores

Un marcador es un signo-objeto predefinido como fácilmente detectable en el entorno real, un ejemplo se puede observar en la Figura 3. Un enfoque basado en marcadores resuelve fácilmente el problema de alinear objetos virtuales en el entorno real con marcadores visuales, que son detectables con métodos comunes de visión por computadora y no requieren un trabajo explícito en la localización y el mapeo detectado, define el estado correcto de la cámara. La facilidad de uso y una extensa lista de kits de herramientas basados en marcadores hicieron que este método fuera muy popular entre los desarrolladores de aplicaciones de AR. (Durnev, 2017)

Figura 3

Ejemplo de marcadores



Nota. Tomado de (Siltanen, 2012)

Monitorización Industrial

Monitorización industrial también llamado seguimiento de procesos en entornos industriales, aprovecha las innovaciones tecnológicas actuales para supervisar el estado operativo de cualquier máquina o sistema. Mediante la incorporación de estos avances tecnológicos, es viable determinar si las máquinas están funcionando de manera adecuada y si se requiere llevar a cabo algún tipo de mantenimiento. Estas herramientas de la revolución digital en la industria, como el Internet de las Cosas, posibilitan la realización del seguimiento

de los procesos. Surge entonces un concepto nuevo, el IIoT (Industrial Internet of Things, por sus siglas en inglés), que se refiere a la aplicación de esta tecnología en los sectores industriales. (Zitelia, 2019)

Android

Android es un sistema operativo basado en Linux para teléfonos móviles como teléfonos inteligentes y tabletas, proporciona una plataforma abierta para que los desarrolladores creen sus propias aplicaciones a fin que las usen una variedad de dispositivos móviles. Android fue desarrollado originalmente por Android, Inc., con el apoyo financiero de Google, que luego lo adquirió en 2005. Este sistema operativo se lanzó oficialmente en 2007. Luego, para desarrollar Android, se formó Open Handset Alliance, un consorcio de 34 empresas de hardware, software y telecomunicaciones, incluidas Google, HTC, Intel, Motorola, Qualcomm, T-Mobile y Nvidia. (Oktaviyana, Ronaldo, & Arifin, 2019)

Es un sistema operativo utilizado por la mayoría de los teléfonos inteligentes en la actualidad. Además, la aplicación del sistema operativo Android en dispositivos móviles conduce a que este sistema operativo pueda realizar adecuadamente la difusión y toma de información en tiempo real, y sea capaz de brindar un sistema de información que sea práctico, conciso, interactivo y de fácil acceso para todos en un dispositivo móvil. (Putri, 2022)

Android Studio

Afnan Uda *et al.* (2020) define a Android Studio como el entorno de desarrollo IDE oficial para crear aplicaciones Android, que está diseñado específicamente para agilizar el proceso de desarrollo de Android y ayudar a los usuarios a crear aplicaciones de alta calidad para todos los dispositivos Android. Esta plataforma ofrece diversas herramientas específicas para desarrolladores, como funciones de edición de código, depuración, pruebas y creación de perfiles.

En 2021, Wu et al., especifica Android Studio es compatible con herramientas populares de gestión de código, como SVN y GITHUB. Los desarrolladores pueden importarlo fácilmente y utilizarlo para su trabajo. El IDE viene con una función inteligente de autocompletado que mejora enormemente la eficiencia de la escritura de código. Es la herramienta oficial de desarrollo de aplicaciones de Google diseñada específicamente para Android, y actualmente es el único IDE recomendado por Google. Ya no es compatible con otros IDE.

Kotlin

Kotlin es un lenguaje de programación moderno que representa una alternativa relevante a Java en varios dominios de desarrollo. En particular, se ha adoptado como lenguaje de desarrollo oficial para el sistema operativo Android. (Ardito et al., 2020)

Naik & Oza (2021), lo describen como un lenguaje de programación multiplataforma, de código abierto, sólido y estático, que se ejecuta en una máquina virtual Java (JVM) y combina varios paradigmas de programación, como orientado a procedimientos, incluyendo el enfoque orientado a procedimientos, orientado a objetos y funcional, todo en una única plataforma. Lo que destaca a Kotlin es su robustez, ya que la mayoría de los errores se identifican durante la fase de compilación. Los desarrolladores eligen Kotlin como su lenguaje de preferencia debido a su código conciso con un mínimo de errores. Kotlin equilibra adecuadamente la legibilidad y la concisión del código.

Google Firebase o Plataforma Firebase

Firebase es una empresa proveedora de BaaS (Backend como servicio) operada por Google. Esencialmente, se trata de una base de datos en tiempo real, lo que significa que funciona como una base de datos NoSQL. Firebase almacena y organiza datos en una estructura jerárquica a través de un extenso árbol JSON. (Mishra et al., 2019)

Según el artículo de investigación de Nizam et al. (2020), Google Firebase emerge como una plataforma que pone su enfoque en el desarrollo de aplicaciones móviles y web,

brindando a los creadores una amplia gama de recursos y servicios. Esta plataforma representa una solución integral de backend en la nube, lo que simplifica significativamente la labor de los desarrolladores al permitirles forjar aplicaciones eficientes y fácilmente escalables con rapidez y sin complicaciones. Firebase engloba un abanico de prestaciones y utilidades valiosas para los programadores, que incluyen alternativas para la autenticación de usuarios, el almacenamiento en nube, la prestación de servicios web, la ejecución de funciones en la nube, la evaluación y supervisión de aplicaciones, así como la emisión de notificaciones push. Adicionalmente, Firebase se acopla de manera fluida con otras herramientas de desarrollo renombradas, como Unity y Android Studio, lo que optimiza aún más el ciclo de desarrollo de aplicaciones y aporta una experiencia cohesiva y sin fisuras para los profesionales de la programación.

Firebase Authentication

Firebase Authentication es un servicio esencial que proporciona autenticación y acceso a aplicaciones. En el ámbito de aplicaciones móviles y de ordenadores, es común la necesidad de identificar a los usuarios de manera confiable. Al obtener información sobre la identidad del usuario, la aplicación puede almacenar sus datos en la nube de forma segura y ofrecer una experiencia personalizada y consistente en todos los dispositivos utilizados por el usuario. Firebase Authentication ofrece una variedad de herramientas, incluyendo servicios de backend, SDKs de fácil implementación y bibliotecas de interfaz de usuario predefinidas, diseñadas específicamente para la autenticación de usuarios en una aplicación. Además, es compatible con múltiples métodos de autenticación, tales como contraseñas, números de teléfono y proveedores de identidad federada populares como Google, Facebook, Twitter, entre otros. (Google Developers, 2020)

Firestore Realtime Database

Firestore Realtime Database es una base de datos NoSQL que se encuentra alojada en la nube, lo que permite la sincronización en tiempo real de datos entre todos los clientes y ofrece funcionalidad incluso cuando los dispositivos están sin conexión. Los datos almacenados en esta base de datos son representados en formato JSON y todos los clientes conectados comparten una misma instancia, lo que les permite recibir actualizaciones automáticas con la información más reciente. (Moroney, 2017)

La plataforma de Firestore permite la creación de aplicaciones potentes y colaborativas al brindar un acceso seguro a la base de datos directamente desde el código del cliente. Los datos se mantienen localmente, incluso en situaciones sin conectividad a internet, lo que permite el funcionamiento en tiempo real de los eventos y garantiza una aplicación adaptable para el usuario. Una vez que el dispositivo recupera la conexión, los cambios realizados en los datos locales se sincronizan con las actualizaciones remotas que ocurrieron mientras el cliente estuvo sin conexión. Además, Firestore Realtime Database proporciona un lenguaje flexible de reglas basadas en expresiones conocido como 'Reglas de Seguridad', que permiten definir la estructura de los datos y los momentos en los que se puede acceder a su lectura o escritura. (Firestore Documentation, 2020)

Google Apps Script

Google Apps Script (GAS) es una plataforma de desarrollo de secuencias de comandos creada por Google para la creación de aplicaciones ágiles en el entorno de Google. Proporciona API integradas para casi todos los servicios ofrecidos por Google. Mediante el uso de Google Apps Script, los desarrolladores tienen la capacidad de configurar conexiones externas para sistemas de administración de bases de datos relacionales o hojas de cálculo, permitiendo que estas últimas funcionen como una base de datos para sus aplicaciones. (Ekanayake, Ihalage, & Abytesundara, 2021)

La plataforma en cuestión ofrece una amplia variedad de oportunidades, como la creación de funciones personalizadas para Google Sheets, el desarrollo de aplicaciones basadas en hojas de cálculo y la escritura de aplicaciones "macro" sencillas. Además, permite la integración con diversos productos y servicios de Google, facilita el desarrollo de una interfaz gráfica de usuario (GUI) que actúa como una aplicación web, y ofrece la posibilidad de interactuar con bases de datos relacionales basadas en la nube mediante los servicios de Google JDBC (Java Database Connectivity). Google Apps Script desempeña un papel fundamental al extender la funcionalidad de las hojas de cálculo de Google, permitiendo un mayor control y gestión de las mismas, así como su integración con otras aplicaciones y servicios. (Radenković et al., 2021)

Google Looker Studio

Google Looker Studio, previamente conocida como Google Data Studio, se configura como una aplicación de visualización de datos que proporciona una herramienta amigable para presentar conjuntos de datos complejos de manera atractiva y fácilmente comprensible. Este servicio se encuentra en la nube, permitiendo acceso desde cualquier ubicación, de manera gratuita, y posibilita la compartición de informes con seleccionados destinatarios. Google Data Studio facilita la fusión de informes desde diversas fuentes de datos, garantizando la posibilidad de compartir informes utilizando esta plataforma sin comprometer la confidencialidad de los datos suministrados. (Apriani, Aan, & Saputra, 2022)

Google Data Studio se erige como una destacada plataforma para la difusión en tiempo real de información pública relacionada con los datos de una entidad. Representa el enfoque de Google en el ámbito de la generación de informes, análisis y representación visual de datos. (Mucchetti, 2020)

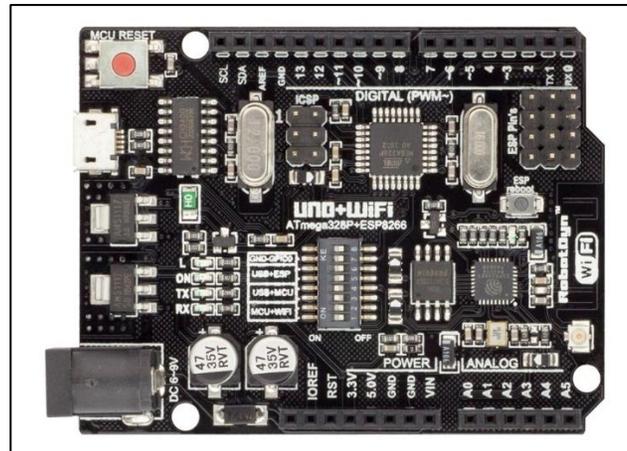
Tarjeta UNO WiFi R3 ATmega328P ESP8266

Es una variante personalizada de la placa ARDUINO UNO R3 tradicional, la cual incorpora de manera integral el microcontrolador Atmel ATmega328 y el circuito integrado WiFi ESP8266, como se muestra en la figura 4. Además, cuenta con una capacidad de memoria flash de 32 MB y un conversor USB-TTL CH340G, todo ello compactado en un único tablero, (RobotDyn, 2017).

Todos los módulos pueden trabajar juntos o de forma independiente.

Figura 4

UNO WiFi R3 ATmega328P ESP8266



Nota. Tomado de (RobotDyn, 2017)

En la Tabla 1, se describen las especificaciones más importantes de la tarjeta UNO WIFI R3 ATmega328P ESP8266.

Tabla 1

Especificaciones de la tarjeta UNO WIFI R3 ATmega328P ESP8266

Propiedad	Características/Valor
Microcontrolador	ATmega328
WiFi IC	ESP8266
Convertidor USB-TTL	CH340G
Sin electricidad	5V-800mA
ENTRADA DE ENERGÍA. USB	5 V (500 mA máx.)
ENTRADA DE ENERGÍA. Conector VIN / DC	9-24 V
El consumo de energía	5V 800mA
Nivel lógico	5V
WiFi	Wi-Fi 802.11 b / g / n 2.4 GHz
USB	Micro USB
Frecuencia de reloj	16 MHz
Voltaje de suministro operativo	5V
E / S digital	14
E / S analógicas	6
Tamaño de la memoria	32 MB
Tipo de interfaz	serial \ OTA
Temperatura de funcionamiento	-40C ° / + 125C °
Largo x Ancho	53,34 x 68,58 mm
Peso	8
antena	Antena incorporada \ externa

Nota. Tomado de (RobotDyn, 2017)

La tarjeta puede trabajar de 5 modos, variando las posiciones de los interruptores. La configuración de los modos de trabajo se presenta en la Tabla 2.

Tabla 2

Modos de trabajo de Tarjeta UNO WiFi R3 ATmega328P ESP8266

Modo de trabajo	Interruptores							
	1	2	3	4	5	6	7	8
CH340 conectar a ESP8266 (cargar boceto)	OFF	OFF	OFF	OFF	ON	ON	ON	NoUSE
CH340 conectar a ESP8266 (conectar)	OFF	OFF	OFF	OFF	ON	ON	OFF	NoUSE
CH340 conectar a ATmega328 (cargar boceto)	OFF	OFF	ON	ON	OFF	OFF	OFF	NoUSE
Mega328 + ESP8266	ON	ON	OFF	OFF	OFF	OFF	OFF	NoUSE
Todos los módulos funcionan de forma independiente	OFF	OFF	OFF	OFF	OFF	OFF	OFF	NoUSE

Nota. Tomado de (RobotDyn, 2017)

Metrología

La metrología abarca el dominio del saber referente a las mediciones, englobando todos los aspectos, tanto de índole teórica como práctica, que se vinculan con las mediciones, sin importar su nivel de precisión y en cualquier esfera de la ciencia y la tecnología. La meta de la metrología consiste en asegurar la uniformidad y confiabilidad de las mediciones, abarcando tanto las transacciones comerciales y de servicios, como los procesos industriales, así como los respectivos empeños de investigación científica y desarrollo tecnológico. (Iglesias, 2022)

Calibración

La calibración es la comparación de un equipo o instrumento bajo prueba (IBP) con un equipo patrón, el equipo patrón debe tener una exactitud 3 veces mayor al del instrumento bajo prueba.

Otro aspecto de la calibración incluye la documentación de los desvíos registrados entre el instrumento bajo investigación y el patrón de referencia; el cálculo de la incertidumbre resultante y la creación del certificado de calibración conteniendo los datos obtenidos y la trazabilidad. (OILM, 1986)

Datos Primarios

Los datos primarios en el contexto de la metrología se refieren a las mediciones experimentales o lecturas directas obtenidas mediante instrumentos de medición o sensores. Estos datos representan información cruda e inalterada que se recopila durante el proceso de medición y son esenciales para evaluar la precisión, la exactitud y la trazabilidad de los resultados. Los datos primarios sirven como base para el análisis, la validación y la calibración de los instrumentos de medición, lo que garantiza la fiabilidad y la calidad de los resultados en la metrología.

Medidor de Temperatura

Es un instrumento electrónico cuya función consiste en medir la temperatura del fluido que circula a través de una tubería; se encuentra compuesto por una sonda de temperatura y un indicador. El indicador posee un transmisor y un display que muestra los valores obtenidos por la sonda, como se presenta en la Figura 5.

Figura 5

Indicador de temperatura



Nota. Tomado de (Endress Hauser, 2020)

Medidor de Presión

Es un instrumento electrónico cuya función consiste en medir la presión del fluido que circula a través de una tubería, se encuentra compuesto por transmisor y un indicador.

Los transmisores de presión de la marca Rosemount 3051T en línea, como se muestra en la Figura 6, son el estándar de la industria para mediciones de presión manométrica y absoluta. El diseño en línea y compacto permite conectar el transmisor directamente a un proceso; para una instalación rápida, sencilla y económica. Entre sus capacidades principales destaca el diagnóstico de integridad del lazo; monitoriza continuamente el lazo eléctrico a fin de detectar cambios que afecten la integridad de la señal de salida de 4-20 mA del transmisor.

(Rosemount, 2020)

Figura 6

Transmisor Rosemount 3051T en línea



Nota. Tomado de (Rosemount, 2020)

Medidor de caudal

También llamado caudalímetro, es un instrumento electrónico cuya función consiste en medir la cantidad del flujo que circula por una tubería.

En la Figura 7 se muestra un caudalímetro electromagnético de la marca Endress Hauser, modelo Proline Promag 10W, sirve para una amplia variedad de aplicaciones, combinado con el transmisor Promag 10 para aplicaciones básicas e integración directa, Promag 10W ofrece mediciones precisas de caudal de líquidos para una amplia variedad de aplicaciones. (Endress Hauser, 2020)

Figura 7

Caudalímetro electromagnético



Nota. Tomado de (Endress Hauser, 2020)

Capítulo III

Diseño e implementación

El sistema de monitoreo en tiempo real se realizó mediante implementación de una tarjeta de adquisición de datos, una base de datos Firebase, una aplicación Android, Google Apps Script, Google Docs, Looker Studio y un complemento AR de Unity.

La tarjeta de adquisición de datos obtiene los valores de los sensores del banco de calibración de medidores de caudal y los envía a la base de datos de Firebase, posteriormente una aplicación Android obtiene los datos de la Base de Datos y los muestra en una interfaz de monitoreo. Mediante un script de Google Apps Script se obtiene los datos de la base de datos y son enviados a una hoja de cálculo de Google y Looker Studio toma los datos de la hoja de cálculo y los muestra como un registro de datos.

La aplicación Android se desarrolló para emitir notificaciones relevantes del banco de pruebas de medidores de caudal, mostrar los datos de los sensores en tiempo real, ver los registros de Looker Studio y observar los estados de los sensores mediante la integración de un complemento AR realizado en Unity.

Parámetros generales de diseño

El enfoque del trabajo se basa en crear una aplicación android para el monitoreo en tiempo real y que posee una función de asistencia en caso de falla con realidad aumentada.

Los parámetros de diseño que deberán ser considerados se tienen:

- Sistema operativo Android
- Conexión a Internet
- Bajo costo
- Notificaciones en primer plano
- Monitoreo de 4 señales

Los parámetros de diseño mencionados son esenciales y de gran importancia al seleccionar el software que dirigirá la configuración de la estructura del sistema a ser diseñado, siendo esta la piedra angular del sistema en su totalidad. De ahí surge la necesidad de iniciar el proceso a través de la selección precisa de los parámetros de diseño. A medida que el proyecto se desenvuelve, surgen oportunidades para ajustes y adaptaciones que permiten al sistema evolucionar de manera coherente, asegurando así el cumplimiento de su funcionalidad requerida.

Diagrama de Conexión

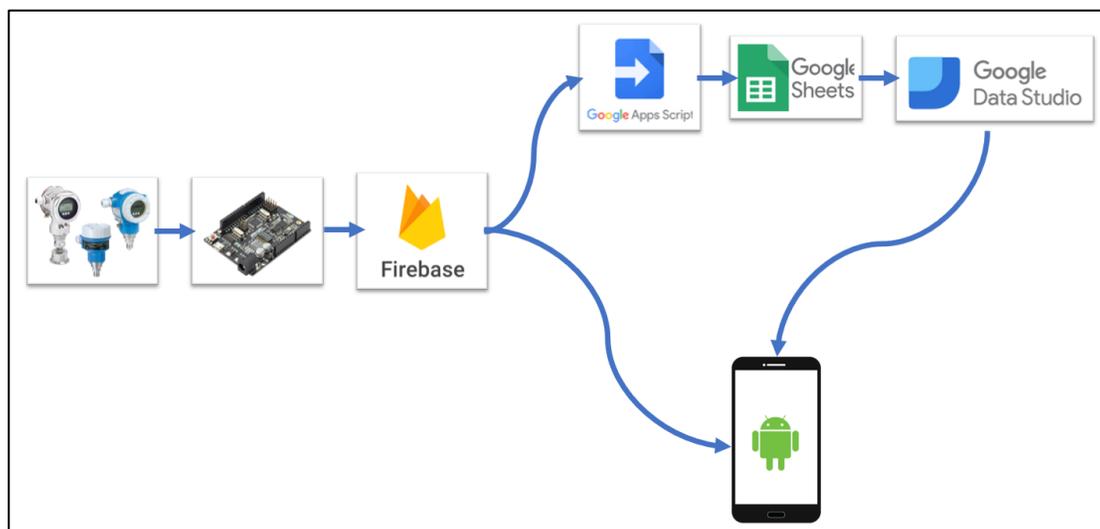
En la arquitectura del sistema de IoT deben constar de dos apartados según Donayre Monteza (2017):

- Arquitectura lógica: expresa cómo se descompone el software en módulos y como se relacionan entre sí.
- Arquitectura física: expresa cómo se distribuye el software entre los componentes físicos y cómo se relación entre ellos.

En la Figura 8, se observa el recorrido de los datos desde los sensores hasta el dispositivo móvil android.

Figura 8

Diagrama de Conexión



Nota. La tarjeta capta las medidas de los sensores del banco de pruebas y los envía a la base de datos, la aplicación android muestra la información de la base de datos y de los reportes de Google Data Studio.

Arquitectura Lógica

Comienza en Arduino enviando los datos a la base de datos Firebase, posteriormente Google Script capta los valores de la base de datos y los registra en una hoja de Cálculo de Google Sheets y Google Data Studio muestra los registros con gráficos. Android Studio muestra los datos de Firebase mediante interfaz gráfica y los registros creados en Google Data Studio.

Arquitectura Física

Arduino lee los datos de la salida de señal de los sensores del banco de pruebas de caudal.

Software y lenguaje de programación

Para el desarrollo y puesta en funcionamiento del sistema de monitoreo, se recomienda usar los siguientes software y lenguaje de programación con las características mencionadas, que se describen en la Tabla 3.

Tabla 3

Características de Software y lenguaje

Software/Lenguaje	Características
Lenguaje de programación android	Java y Kotlin
Editor Android	Android Studio 4.1.1
Base de datos	Firestore
Google Scripts	JavaScript
Hoja de cálculo	Google Sheets
Registros y gráficos	DataStudio
Realidad Aumentada	Unity 2021.2.7f1
Programación Unity	C#
Arduino	Arduino 1.8+

Nota. La Tabla 3, muestra software y lenguajes de programación basados en los puntos de vista tanto económicos como técnicos, los cuales en base a experticia y desarrollo se recomiendan.

En el aspecto económico, optar por el uso de los programas y lenguajes de programación sugeridos resulta adecuado, ya que las ediciones fundamentales de estos recursos son de libre acceso. Esto es más que suficiente para satisfacer las necesidades del sistema de monitoreo IoT, generando así un ahorro considerable al evitar la inversión en licencias.

En el aspecto técnico, se realizó la investigación bibliográfica y se encontró mayor relevancia en el autor Donayre Monteza (2017), en la que se fundamenta en la perspicacia de expertos técnicos, y los lenguajes de programación recomendados, tanto en términos de software como de funcionalidades, satisfacen los criterios esenciales de seguridad, escalabilidad, adaptabilidad y confiabilidad que son indispensables para el sistema. Esto ocurre de manera análoga a cómo lo harían soluciones de software y lenguajes de programación que demandan costosas licencias de uso.

Base de Datos con Firebase

Ingresando al sitio web *firebase.google.com*, se da clic en “Ir a consola”, se accede con un usuario o correo de *Gmail*, se genera un proyecto y se escoge el servicio de *Realtime Database (Base de Datos en Tiempo Real)*.

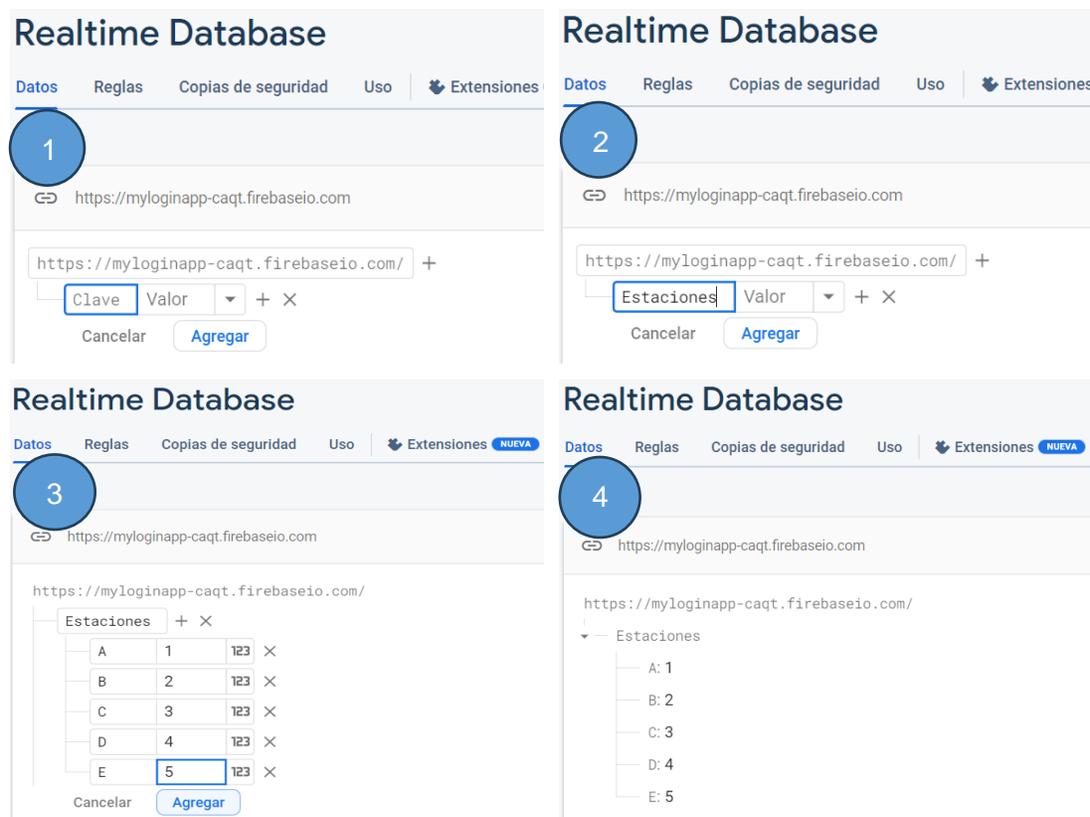
En el servicio de Realtime Database, en la parte de *Datos*, se agregan los campos que almacenarán los datos enviados por la tarjeta UNO WiFi R3 ATmega328P ESP8266, para este proyecto se denominó ESTACIONES, A-E, como se muestra en la Figura 9.

Se deben definir las reglas de la base de datos, en el apartado Reglas, se da clic para visualizar las reglas y se procede a establecer las reglas necesarias para poder leer y modificar los datos que existen en la base de datos. Para acceder a las reglas, se debe ubicar en el servicio de Realtime Database, en la parte de *Reglas*.

En la Figura 10, se indica las reglas establecidas para el proyecto.

Figura 9

Creación de la base de datos en Firebase



Nota. Al crear los bloques de datos se introducen números para habilitar el modo numérico, que son el tipo de datos que se almacenarán.

. Para acceder a la base de datos desde la aplicación a realizar se debe programar e insertar el código secreto de la base de datos del proyecto el cual se encuentra en la configuración del proyecto – cuentas de servicio – Secretos de la base de datos, como se muestra en la Figura 11.

El secreto de la cuenta es un código que permite visualizar la base de datos y dependiendo las reglas establecidas puede modificarse, por ende se debe tener en cuenta de no compartir dicho código.

Figura 10

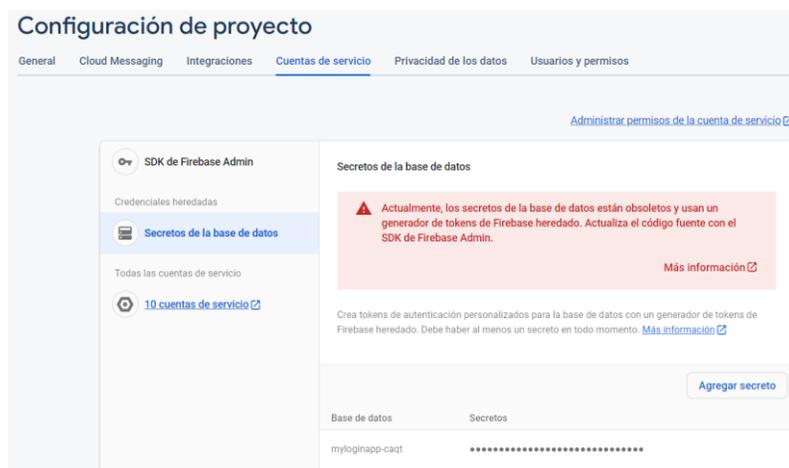
Reglas de Base de datos



Nota. Las reglas mostradas en la figura enuncian que se encuentra habilitado el modo de lectura y escritura.

Figura 11

Secreto de base de datos



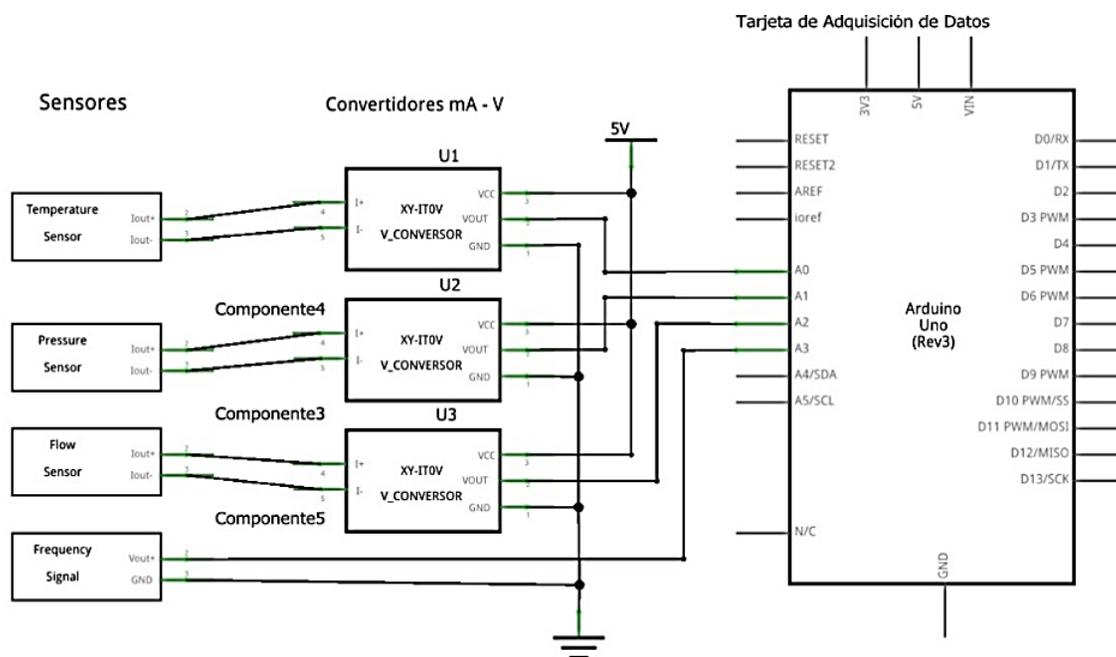
Nota. El secreto de la base de datos es una medida de seguridad crítica para prevenir el acceso no autorizado a la base de datos.

Obtención de datos de los sensores del Banco de Pruebas

En el banco de pruebas de caudal de la empresa ESRW S.A. se monitorearán 4 magnitudes, las cuales provienen de 3 sensores tipo transmisores que proporcionan salida de señal de 4 a 20 mA y variador de frecuencia que proporciona una salida analógica de 5V, por lo cual estas señales serán medidas y tratadas en la tarjeta UNO WiFi R3 ATmega328P ESP8266. Se utilizó conversores de señal de miliamperios a voltaje para la lectura de la señal de los sensores del banco de pruebas, como se indica en la Figura 12 y Figura 13.

Figura 12

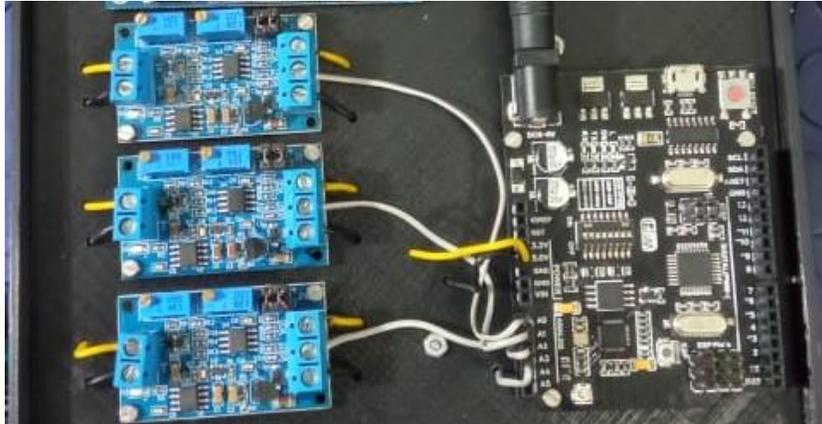
Diagrama de conexión de las señales de sensores



Nota. Las hojas de datos técnicos o datasheets de los sensores y convertidores mA-V se encuentran en el Anexo 1.

Figura 13

Conexión de señales de sensores del banco de pruebas de caudal



Nota. Implementación del diagrama de conexión eléctrica de la Figura 12.

Programación en Arduino

Mediante la tarjeta UNO WiFi R3 ATmega328P ESP8266 se reciben las señales a monitorear y se envían a base de datos Firebase, para ello se generan dos programas.

En el primero se recibe la señal en voltaje de los sensores, se escala y envía al módulo esp8266 de la tarjeta a través de comunicación serial, como se detalla en la Figura 14.

En el segundo, toma los valores enviados por serial del programa uno, realiza la conexión con la red WiFi y envía los datos a la base de datos Firebase, así como se muestra en la Figura 15.

Figura 14

Programación lectura de señales del banco de pruebas

```

ReceptordeDatos Arduino 1.8.13
Archivo Editar Programa Herramientas Ayuda

ReceptordeDatos $
27 //Se realiza la lectura del sensor 1
28 valorSensor1 = analogRead(sensor1);
29 //cambia la escala a 0.0 - 100.0
30 vSensor1 = map(valorSensor1, 0, 204.6, 0.0, 250);
31 vSensor1=vSensor1/100;
32 //Se realiza la lectura del sensor 2
33 valorSensor2 = analogRead(sensor2);
34 //cambia la escala a 0.0 - 100.0
35 vSensor2 = map(valorSensor2, 0, 368, 0, 2390);
36 vSensor2=vSensor2/100;
37 //Se realiza la lectura del sensor 3
38 valorSensor3 = analogRead(sensor3);
39 //cambia la escala a 0.0 - 100.0
40 vSensor3 = map(valorSensor3, 0, 409, 0.0, 10000);
41 vSensor3=vSensor3/100;
42 //Se realiza la lectura del sensor 4
43 valorSensor4 = analogRead(sensor4);
44 //cambia la escala a 0.0 - 100.0
45 vSensor4 = map(valorSensor4, 0, 1084, 0.0, 6000);
46 vSensor4=vSensor4/100;
47 // Serial.println(vSensor4);
48
49 //Enviamos los valores de los sensores por comunicación serial al ESP8266 para su posterior env
50 //Preparación para enviar valores de los sensores por serial al esp8266
51 DynamicJsonBuffer jbuffer;
52 JsonObject& root = jbuffer.createObject();
53 root["Sensor1"] = String(vSensor1); // configurando la variable Sensor1 para contener el valor
54 root["Sensor2"] = String(vSensor2); // configurando la variable Sensor2 para contener el valor
55 root["Sensor3"] = String(vSensor3); // configurando la variable Sensor3 para contener el valor
56 root["Sensor4"] = String(vSensor4); // configurando la variable Sensor4 para contener el valor
57 root.printTo (Serial); // enviando los valores de los sensores por serial al esp8266
58 root.remove ("Sensor1"); // elimina el campo temporal del objeto raiz JSON para poder reutilize
59 root.remove ("Sensor2"); // elimina el campo temporal del objeto raiz JSON para poder reutilize
60 root.remove ("Sensor3"); // elimina el campo temporal del objeto raiz JSON para poder reutilize
61 root.remove ("Sensor4"); // elimina el campo temporal del objeto raiz JSON para poder reutilize
62 delay(2000);
63 }

```

Nota. Se carga la programación cuando la tarjeta UNO WiFi R3 ATmega328P ESP8266 se encuentra con modo *CH340* conectar a *ATmega328*, como se indica en la Tabla 2.

Figura 15

Programación Arduino-Firebase

```

40 // En el ciclo loop se desarrolla el código que se va a estar repitiendo constantemente
41 void loop() {
42   // Definición de variables para leer los valores del Arduino
43
44   while(1){
45     // Aceptar datos del arduino
46     while (Serial.available()) { // Leer datos en serie del Arduino
47       json = Serial.readString();
48       StringReady = true;
49     }
50     if (StringReady) {
51       StaticJsonBuffer <200> jsonBuffer; // memoria preasignada para almacenar el JsonObject max 200 bytes
52       JsonObject & root = jsonBuffer.parseObject (json); // convertir la cadena en datos JSON
53
54       if (!root.success()) {
55         Serial.println ("parseObject () failed");
56         return;
57       }
58       //Declaración de variables para almacenar los valores leído del arduino
59       String rawSensor1 = root["Sensor1"]; //
60       String rawSensor2 = root["Sensor2"]; //
61       String rawSensor3 = root["Sensor3"]; //
62       String rawSensor4 = root["Sensor4"]; //
63       // Conversión en número flotantes a los valores obtenidos
64       vSensor1=rawSensor1.toFloat();
65       vSensor2=rawSensor2.toFloat();
66       vSensor3=rawSensor3.toFloat();
67       vSensor4=rawSensor4.toFloat();
68       root.remove ("Sensor1"); // elimina el campo temporal del objeto raíz JSON para poder reutilizar el ob
69       root.remove ("Sensor2"); // elimina el campo temporal del objeto raíz JSON para poder reutilizar el ob
70       root.remove ("Sensor3"); // elimina el campo temporal del objeto raíz JSON para poder reutilizar el ob
71       root.remove ("Sensor4"); // elimina el campo temporal del objeto raíz JSON para poder reutilizar el ob
72     }
73
74     /// Programación para enviar a la base de datos
75
76     //Si el valor del sensor ha cambiado respecto a la lectura previa actualiza el valor en la base de dat

```

Nota. Se carga la programación cuando la tarjeta UNO WiFi R3 ATmega328P ESP8266 se encuentra con modo *CH340* conectar a *ESP8266*, como se indica en la Tabla 2.

Diseño del complemento AR

El complemento AR se realizará en el motor gráfico Unity, con la ayuda del complemento Vuforia.

El reconocimiento se basará en marcadores

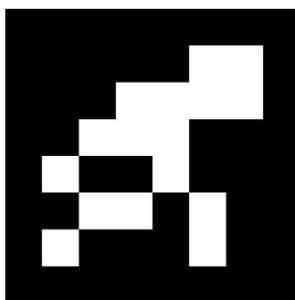
Marcadores

Para el reconocimiento de los sensores se necesitará de marcadores que pueden ser creados o generados por diferentes herramientas, lo importante de un marcador es que se distinga su forma.

Los marcadores ocupados en el presente trabajo son de la librería de ARUCO, para la señal de temperatura se ocupó el marcado de ARUCO #33 de 6*6 (Figura 16), para la señal de presión se ocupó el marcado de ARUCO #91 de 7*7 (Figura 17), para la señal de caudal se empleó el marcado de ARUCO #107 de 7*7 (Figura 18), para la señal de frecuencia se utilizó el marcado de ARUCO #146 de 7*7 (Figura 19).

Figura 16

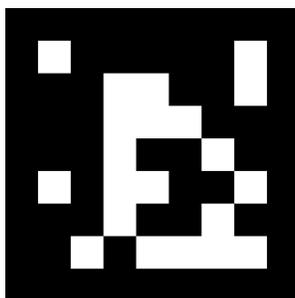
Marcador de señal de Temperatura



Nota. Tomado de (Kalachev, 2021)

Figura 17

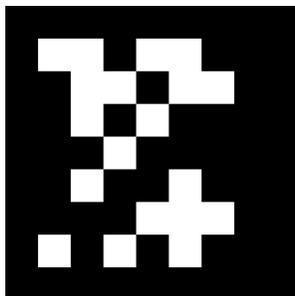
Marcador de señal Presión



Nota. Tomado de (Kalachev, 2021)

Figura 18

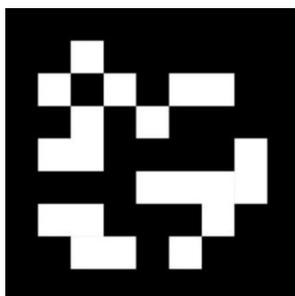
Marcador de la señal de Caudal



Nota. Tomado de (Kalachev, 2021)

Figura 19

Marcador de la señal de Frecuencia



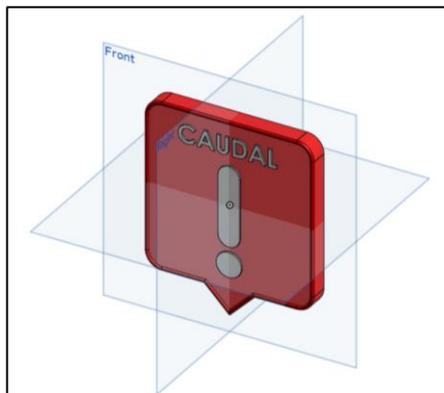
Nota. Tomado de (Kalachev, 2021)

Modelos 3D de los estados de los sensores

Cuando la aplicación reconozca el marcador se mostrará un objeto en 3D, el cual proporciona el estado del sensor, para lo cual se creó 3 tipos de objetos, que representan estado de APAGADO, estado de ENCENDIDO y estado de ALERTA, en las Figuras 20, 21, 22 y 23, se indica los señalado anteriormente.

Figura 20

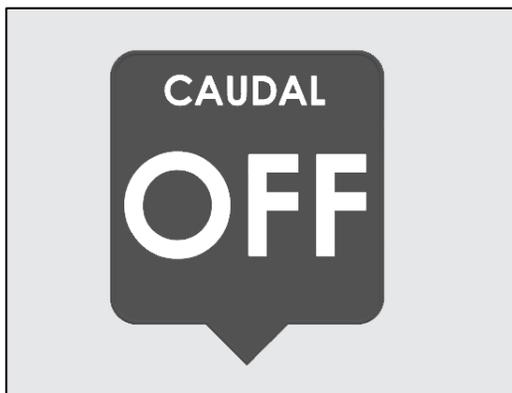
Aviso en 3D de señal de estado de la señal



Nota. Boceto en 3D de la señal de alerta de caudal

Figura 21

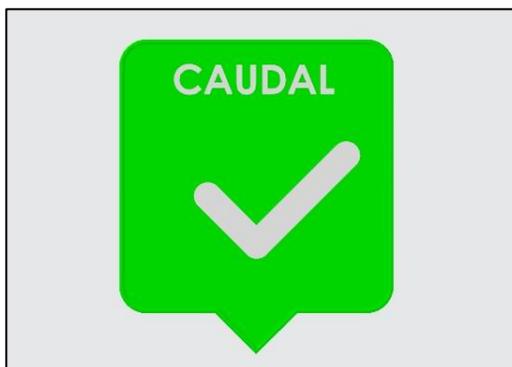
Aviso de estado de Apagado del sensor



Nota. En la señal de apagado se optó por mostrar el aviso de color gris y la palabra OFF.

Figura 22

Aviso de estado de Encendido del sensor



Nota. En la señal de encendido se optó por mostrar el aviso de color verde y el símbolo de un visto.

Figura 23

Aviso de estado de ALERTA del sensor



Nota. En la señal de error se optó por mostrar el aviso de color rojo y el símbolo de exclamación.

Base de Datos para Realidad Aumentada

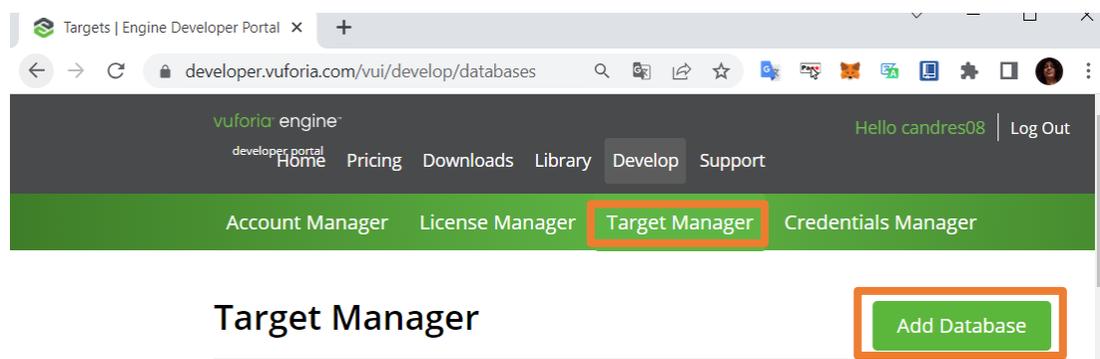
Para el desarrollo de la herramienta de realidad aumentada, se utilizará en el motor gráfico Unity, con el complemento de Vuforia.

Vuforia es la encargada de reconocer el marcador, para lo cual se deberá crear y subir las imágenes de los marcadores a una base de datos en Vuforia Developer.

En el sitio web de desarrollo de Vuforia Developer Portal developer.vuforia.com debe crear un usuario, como siguiente se da clic en la pestaña *Develop*, luego se selecciona *Target Manager* y se da clic en el botón *Add Database*, como se muestra en la Figura 24.

Figura 24

Pasos en Vuforia para creación de Base de Datos

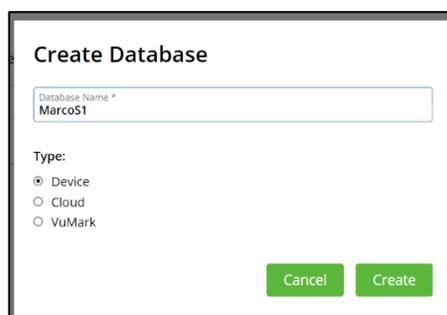


Nota. Se puede acceder al sitio web de Vuforia desde cualquier navegador web.

En el cuadro que se despliega, se ingresa un nombre y se selecciona el tipo para la nueva Base de Datos, como se indica en la Figura 25.

Figura 25

Creación de Base de Datos Vuforia

A screenshot of a 'Create Database' dialog box. The title is 'Create Database'. There is a text input field labeled 'Database Name *' with the value 'MarcoS1' entered. Below the input field, there is a 'Type:' section with three radio button options: 'Device' (which is selected), 'Cloud', and 'VuMark'. At the bottom right of the dialog, there are two buttons: 'Cancel' and 'Create'.

Nota. El tipo de *Database* que se selecciona es *Device* debido a que se cargaran a un conjunto de imágenes o características visuales que se almacenan y utilizan directamente en el dispositivo para el reconocimiento de objetos.

Posteriormente, se selecciona la base de datos creada y se añade las imágenes dando clic en “Add Target” (se debe añadir individualmente cada imagen correspondiente a cada marcador), en el cuadro que aparece se selecciona el tipo de target, en este caso tipo imagen, se carga la dirección en la cual se tiene la imagen del marcador dando clic en “Browse...” y buscando en la carpeta donde se guardaron las imágenes, se le asigna un nombre y se da clic en “Add”, así como presenta en la Figura 26.

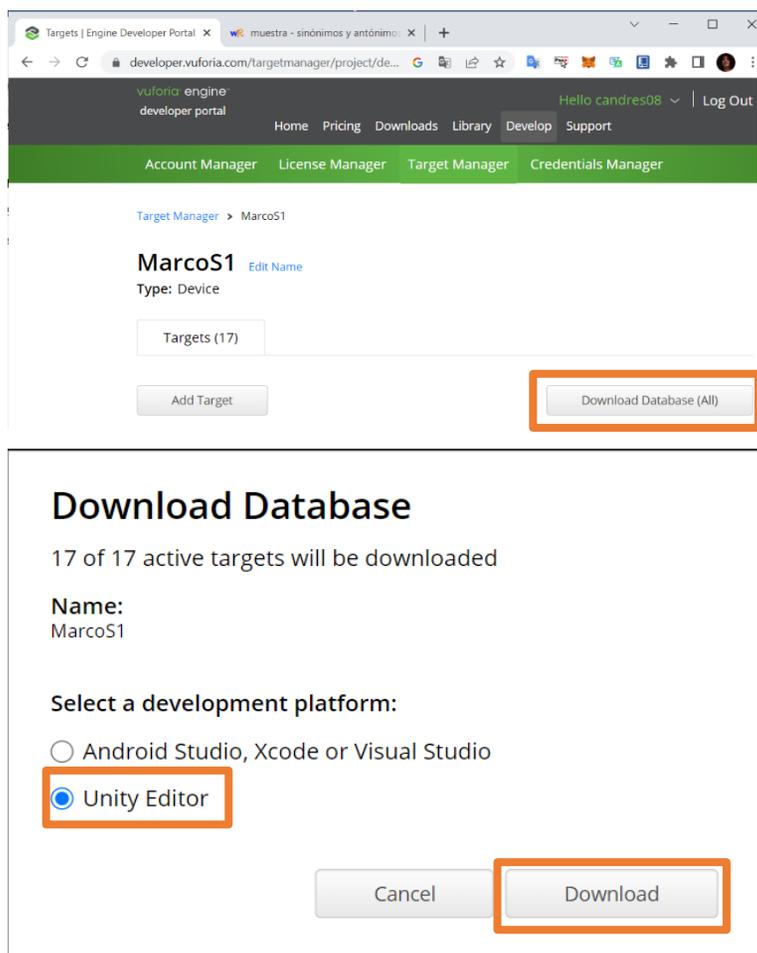
Al tener todas las imágenes cargadas en la base de datos se debe descargar la Base de datos dando clic en “Download Database”, seleccionar la opción de “Unity Editor” y dar clic en “Download”, como se indica en la Figura 27.

Figura 26

Add Target

The screenshot shows the Vuforia Target Manager interface. The main header includes the Vuforia logo, 'engine developer portal', and user information 'Hello candres08'. The navigation menu includes 'Home', 'Pricing', 'Downloads', 'Library', 'Develop', and 'Support'. Below the navigation, there are links for 'Account Manager', 'License Manager', 'Target Manager', and 'Credentials Manager'. The main content area shows the 'Target Manager' section for 'MarcoS1'. A red box highlights the 'MarcoS1' name and the 'Edit Name' link. Below this, there is a 'Type: Device' label and a 'Targets (17)' button. A red box highlights the 'Add Target' button. Below the 'Add Target' button, there is a section titled 'Add Target' with a 'Type:' label. A red box highlights the 'Image' type selection. Below the 'Image' type, there are four options: 'Image', 'Multi', 'Cylinder', and 'Object'. Below the 'Image' type, there is a 'File:' label and a text input field containing '7x7_1000-00.jpg'. A red box highlights the 'File:' label and the input field. Below the input field, there is a 'Browse...' button. Below the 'File:' label, there is a 'Width:' label and a text input field. Below the 'Width:' label, there is a paragraph of text: 'Enter the width of your target in scene units. The size of the target should be on the same scale as your augmented virtual content. Vuforia uses meters as the default unit scale. The target's height will be calculated when you upload your image.' Below the 'Width:' label, there is a 'Name:' label and a text input field containing '7x7_1000-00'. A red box highlights the 'Name:' label and the input field. Below the input field, there is a paragraph of text: 'Name must be unique to a database. When a target is detected in your application, this will be reported in the API.' Below the 'Name:' label, there are two buttons: 'Cancel' and 'Add'. A red box highlights the 'Add' button.

Nota. Se puede agregar un nombre adicional en el espacio de *Name*, pero de preferencia se lo deja por defecto con el mismo nombre de la imagen.

Figura 27*Descarga de Base de Datos Vuforia*

Nota. Se selecciona la opción *Unity Editor* debido a que se va a utilizar la base de datos de Vuforia en la plataforma Unity.

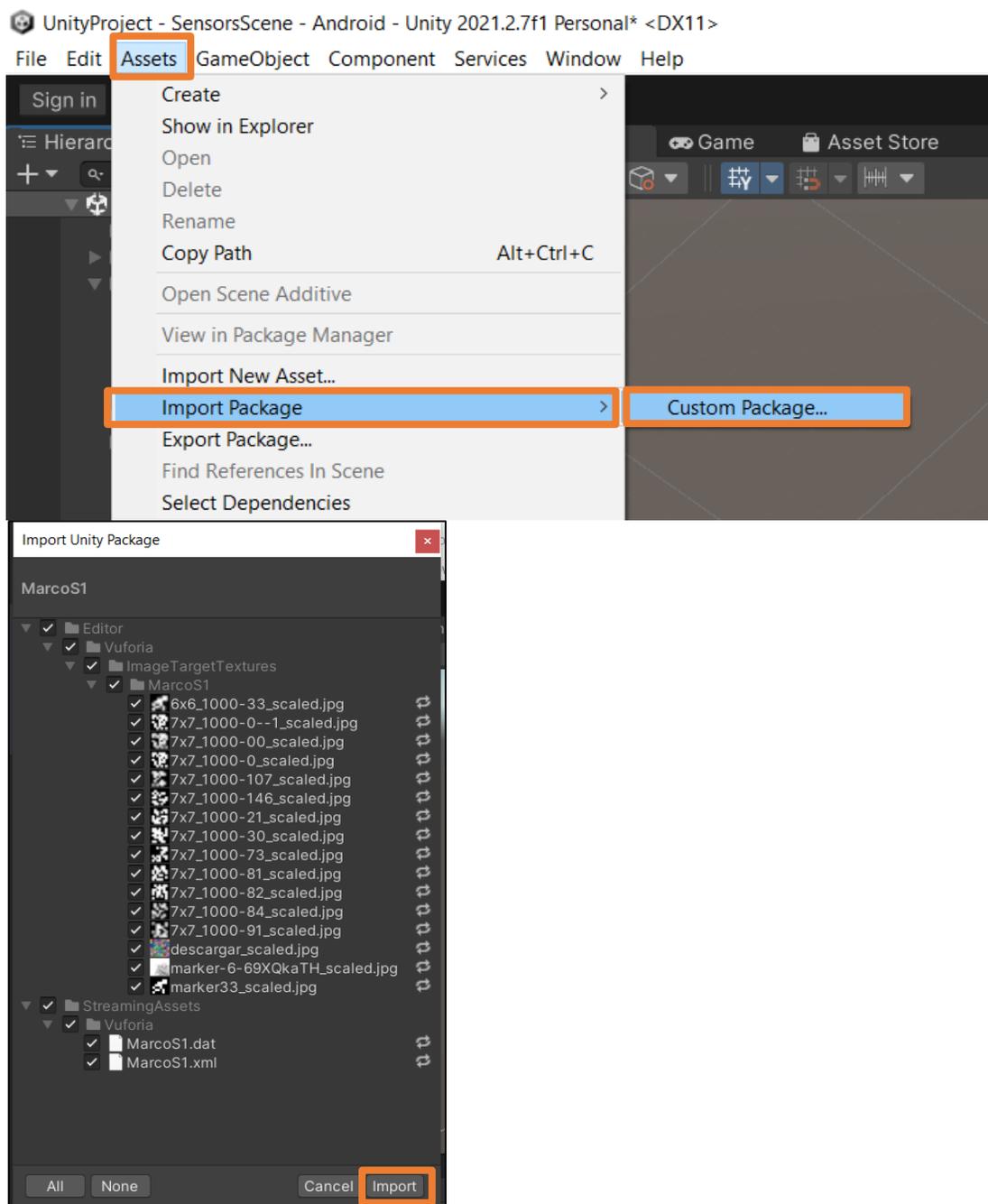
Una vez descargada la base de datos de Vuforia, se debe crear un proyecto de Unity en el que se realizará el módulo de realidad aumentada para la aplicación de monitoreo.

Con el proyecto de Unity abierto, se debe importar la base de datos de Vuforia dando doble clic, o a su vez dando clic en la pestaña *Assets* y eligiendo la opción *Import Package* –

Cumston Package..., se busca la dirección donde se descargó la base de datos Vuforia y se añade. Se abrirá una ventana como en la Figura 28 y se le da clic en *Import*.

Figura 28

Importar Base de datos Vuforia

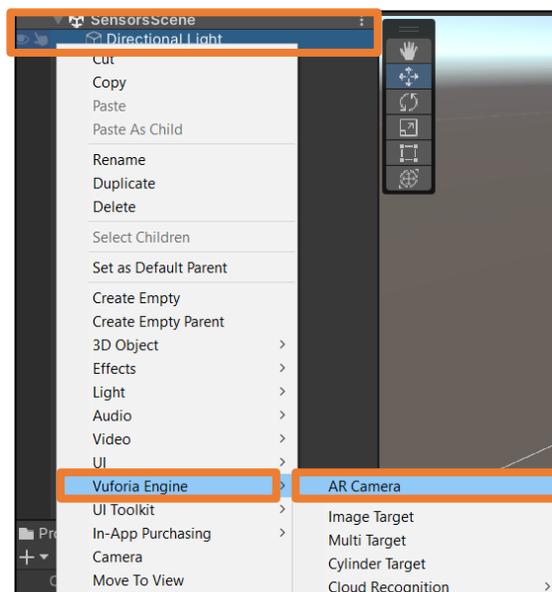


Nota. Verificar que se encuentre seleccionados todos los casilleros con un visto.

Se debe añadir un AR Camera, dando clic derecho en el proyecto, luego se debe seleccionar *Vuforia Engine* y por último *AR Camera*, así como se muestra en la Figura 29.

Figura 29

Creación de AR Camera



Nota. AR Camera es un componente de Unity que se ocupa como cámara virtual en una escena de realidad aumentada para mostrar objetos virtuales superpuestos en el mundo real a través de la vista de la cámara del dispositivo.

Se debe copiar la Licencia de Vuforia que se encuentra en la página de Vuforia Developer, en la pestaña *Develop*, opción *License Manager*, se escoge la licencia IoT y se copia la licencia que se muestra en el cuadro gris, luego se debe ubicar en el proyecto de Unity creado, dar clic en AR Camera, se debe abrir la configuración de Vuforia, luego se da clic en el menú *Window*, a continuación seleccionar la opción *Vuforia Configuration*, para pegar la clave de licencia de Vuforia y dar clic en *Add Licence*, como en la Figura 30.

Figura 30

Clave de Licencia Vuforia

The figure consists of two screenshots. The top screenshot shows the Vuforia Developer Portal website. The URL in the browser is `developer.vuforia.com/develop/licenses/3e862b7c83bf4a7ab0b045401e7ef881/info`. The navigation menu includes 'Home', 'Pricing', 'Downloads', 'Library', 'Develop', and 'Support'. The 'License Manager' tab is active, showing a license for 'IoT'. The license key is displayed in a text box, and the 'Plan Type' is 'Basic', 'Status' is 'Active', and 'Created' is 'Mar 19, 2021 00:18'. The 'License UUID' is '3e862b7c83bf4a7ab0b045401e7ef881'. The bottom screenshot shows the Unity 2021.2.7f1 Personal* interface. The 'Window' menu is open, and 'Vuforia Configuration' is selected. The 'App License Key' field is populated with the license key from the Vuforia portal, and the 'Add License' button is highlighted.

Vuforia Developer Portal License Information:

- License Name: IoT
- License Key: `Ac1134j/////AAABmYvKgE/u+EpPnWzL3s01NBhPWtmsHFn/hkna9k+tRCVq3AK5q5mwTg9cqy8e3nmmQzUJH2NIIJL4M/wclAAppbukHWm0JRPvEZMo3dWcVUGdakMBZkNMZFxYwT6GAUn0QrX+iwfXWWKsYTwyS5eN6RHoFFUpsfx+34t8W13lqU/131AwFOkFisXOxHtIJIEiTVQ0EdS7Mzd6VjJTQTN7koWfWGY4DCLi2u5Gjbc+TNvT17mqJN5BeR08R6KydMFINOEB8PjS1GAV+Ufo/1XIQBtX0XzCbPfiy7lt5vIExZqrQhaLyoY1s0Fa0f1XHYijmK+YG/zgj8nlywXvgD5Rh2rQqmqyh`
- Plan Type: Basic
- Status: Active
- Created: Mar 19, 2021 00:18
- License UUID: 3e862b7c83bf4a7ab0b045401e7ef881
- History: License Created - Mar 19, 2021 00:18

Unity Configuration:

- Window Menu: Vuforia Configuration (Ctrl+Shift+V)
- App License Key: `Ac1134j/////AAABmYvKgE/u+EpPnWzL3s01NBhPWtmsHFn/hkna9k+tRCVq3AK5q5mwTg9cqy8e3nmmQzUJH2NIIJL4M/wclAAppbukHWm0JRPvEZMo3dWcVUGdakMBZkNMZFxYwT6GAUn0QrX+iwfXWWKsYTwyS5eN6RHoFFUpsfx+34t8W13lqU/131AwFOkFisXOxHtIJIEiTVQ0EdS7Mzd6VjJTQTN7koWfWGY4DCLi2u5Gjbc+TNvT17mqJN5BeR08R6KydMFINOEB8PjS1GAV+Ufo/1XIQBtX0XzCbPfiy7lt5vIExZqrQhaLyoY1s0Fa0f1XHYijmK+YG/zgj8nlywXvgD5Rh2rQqmqyh`
- Add License Button

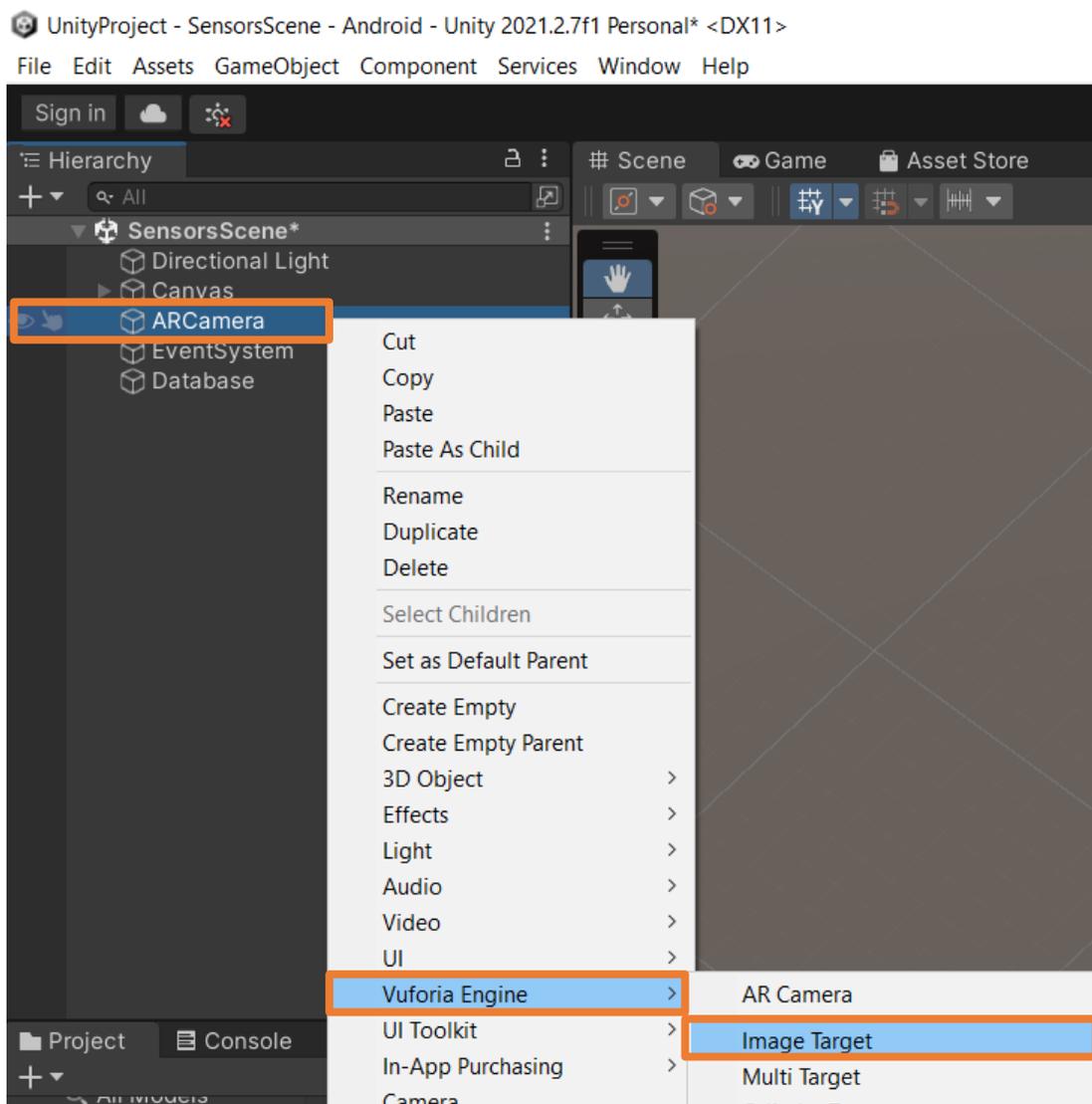
Nota. La clave de la licencia de Vuforia permite acceder a las funciones de Vuforia desde la plataforma de Unity.

Implementación en Unity

Dando derecho clic en *ARCamera* se añade cuatro *Image Target*, como se muestra en la Figura 31.

Figura 31

Creación de Image Target

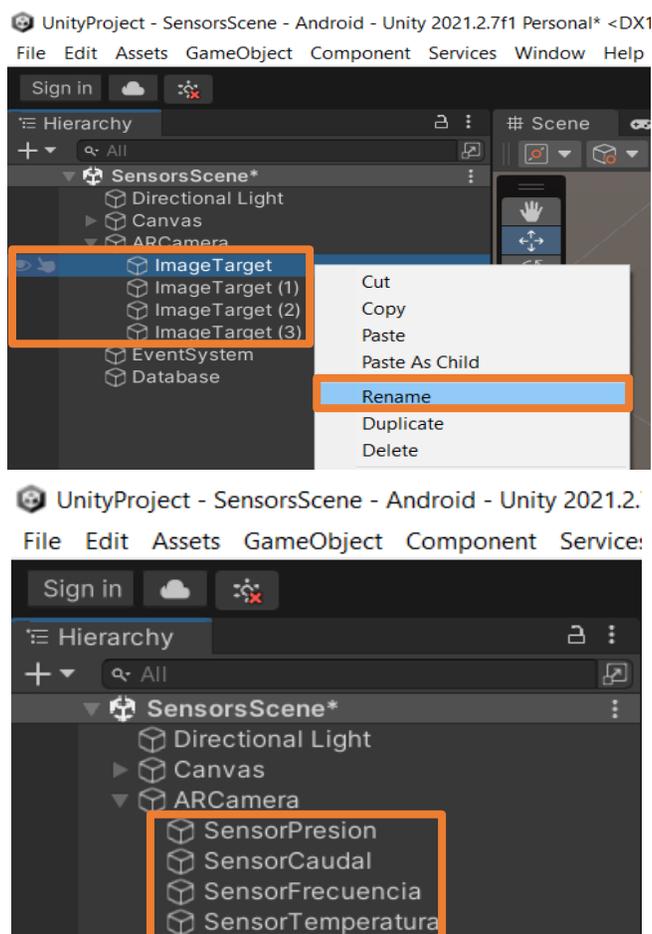


Nota. Image Target es un componente de Unity, el cual se refiere a una imagen o patrón visual en el mundo real que se utiliza como marcador o referencia para que una aplicación pueda reconocer y ubicar objetos virtuales sobre él.

A los cuatro *Image Target* se les cambia el nombre para poder identificarlos, dando clic derecho en el cada *Image Target* y seleccionando la opción *Rename*, como se muestra en la Figura 32.

Figura 32

Cambio de nombre a los Image Target

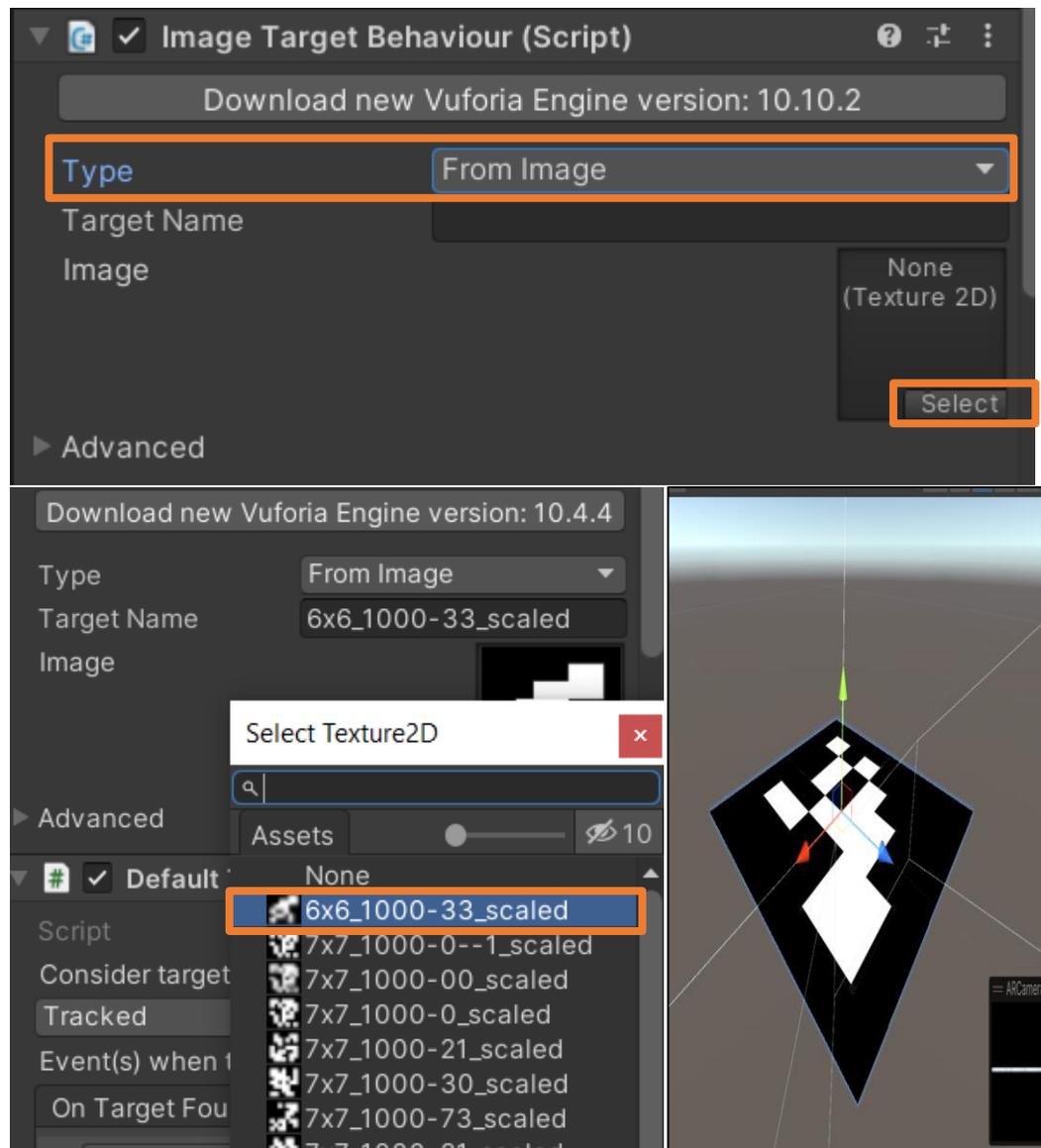


Nota. A cada *Image Target* se le asignó el nombre de cada uno de los sensores del banco de pruebas de medidores de caudal.

Se selecciona el tipo de marcador, en este caso *From Image* y la imagen que se le asignará al marcador va a reconocer la *Image Target*, como se muestra en la Figura 33.

Figura 33

Selección de marcador en Image Target



Nota. Se selecciona el marcador asignado a cada sensor.

Posteriormente se debe añadir los modelos 3D de los sensores realizados anteriormente, que deben estar en extensión .obj, para ello se crea una carpeta en el proyecto de Unity dentro de Assets con el nombre de "Objetos3d_Avisos", los objetos se visualizarán como en la Figura 34.

Figura 34

Objetos 3D de avisos de estados de sensores



Nota. Se observa que se tiene en total doce objetos 3D, tres por cada sensor.

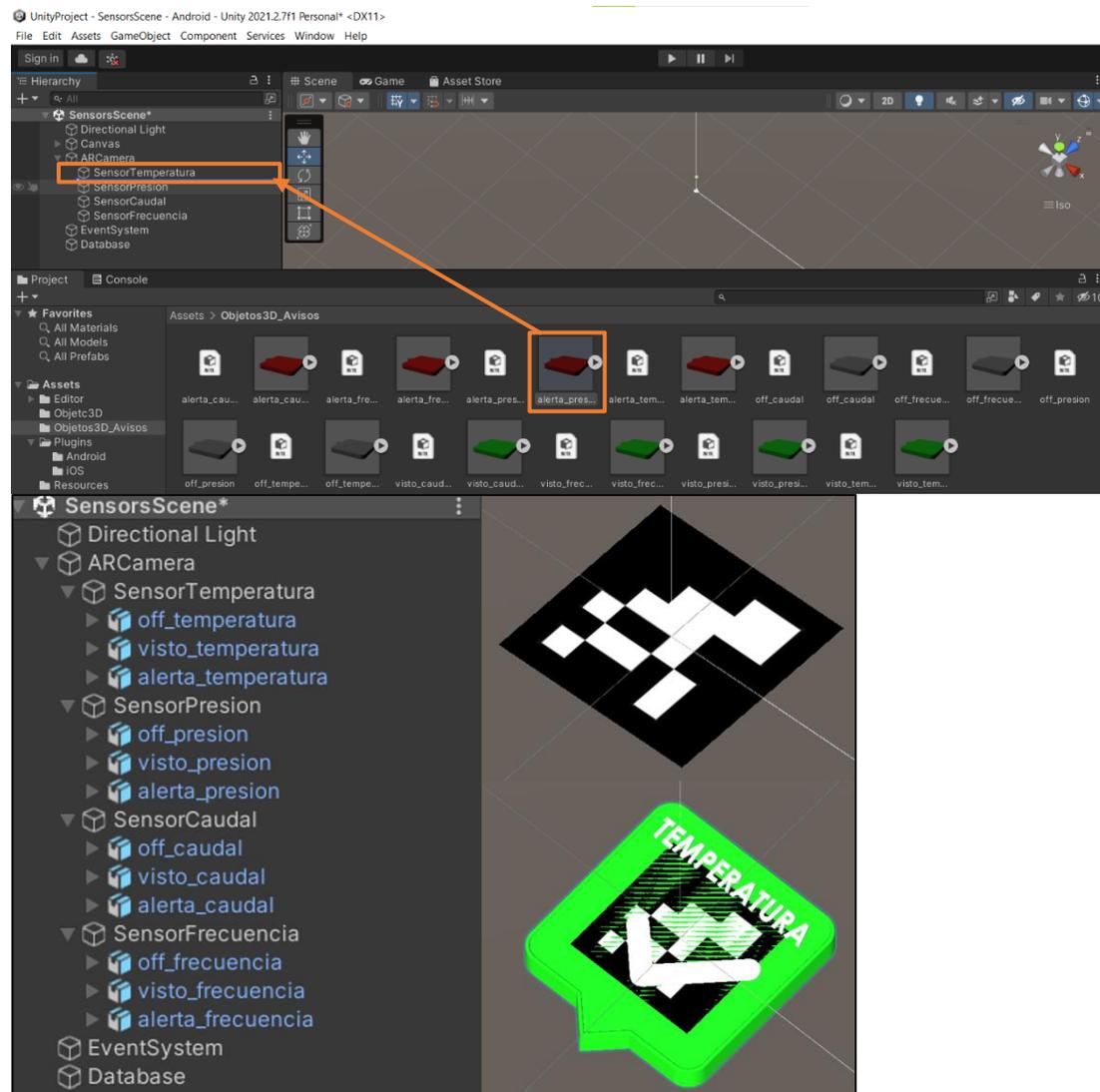
Luego se arrastra cada objeto 3D dentro de los Image Target correspondientes a cada sensor, como se muestra en la Figura 35.

Debido a que se tiene los estados para cada sensor, se debe decidir cuál de los estados debe activar para ciertas condiciones de los sensores, por ende, se creará un objeto de Unity llamado Database, el cual contendrá un script en donde se asignarán las condiciones de activado de cada objeto 3D para cada sensor.

En el script Database se declaran 12 GameObject correspondientes a cada señal de banco de pruebas de caudal, como se muestra en la Figura 36.

Figura 35

Objetos 3D dentro de las Image Target



Nota. Se observa que cada dentro de cada *Image Target* hay tres objetos 3D

Figura 36

Declaración de *GameObjects* para cada sensor

```
Script de Unity (1 referencia de recurso) | 0 referencias  
public class Database : MonoBehaviour  
{  
    public GameObject temp0, temp1, temp2;  
    public GameObject caudal0, caudal1, caudal2;  
    public GameObject frec0, frec1, frec2;  
    public GameObject pres0, pres1, pres2;  
}
```

Nota. Se observa tres *GameObject* correspondientes a cada estado de los sensores

Además, se declaran funciones que se activan y desactivan cada vez que detectan cada marcador. En la Figura 37, se muestra las funciones y condiciones correspondientes a la señal de presión.

Estas funciones se activan y desactivan con los eventos que se generan por default en cada *Image Target*, que son Objetos encontrados o perdidos, como se muestra en la Figura 38.

Figura 37

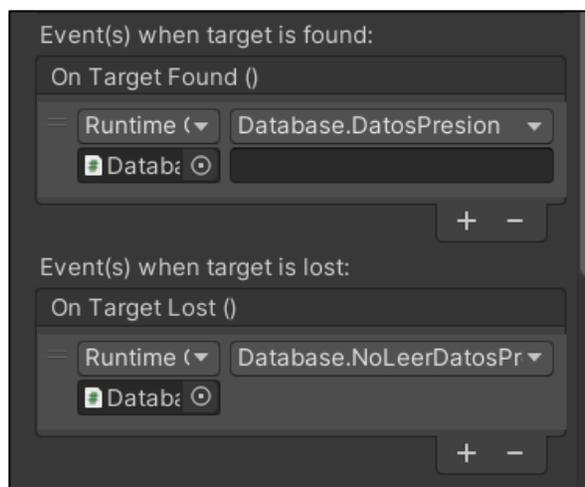
Programación para activar o desactivar objetos 3D de presión

```
101 void NoLeerDatosPresion()  
102 {  
103     pres0.SetActive(false);  
104     pres1.SetActive(false);  
105     pres2.SetActive(false);  
106 }  
0 referencias  
107 void DatosPresion(string datoP)  
108 {  
109  
110     if (datoP == "OFF")  
111     {  
112         pres0.SetActive(true);  
113         pres1.SetActive(false);  
114         pres2.SetActive(false);  
115     }  
116     else if (datoP == "ON")  
117     {  
118         pres0.SetActive(false);  
119         pres1.SetActive(true);  
120         pres2.SetActive(false);  
121     }  
122     else if (datoP == "FAIL")  
123     {  
124         pres0.SetActive(false);  
125         pres1.SetActive(false);  
126         pres2.SetActive(true);  
127     }  
128     text.text = "Presion";  
129 }
```

Nota. Dentro de cada condicional *if* se observa que debe estar activo solo un estado por sensor

Figura 38

Asignación de funciones de activación y desactivación con Image Target

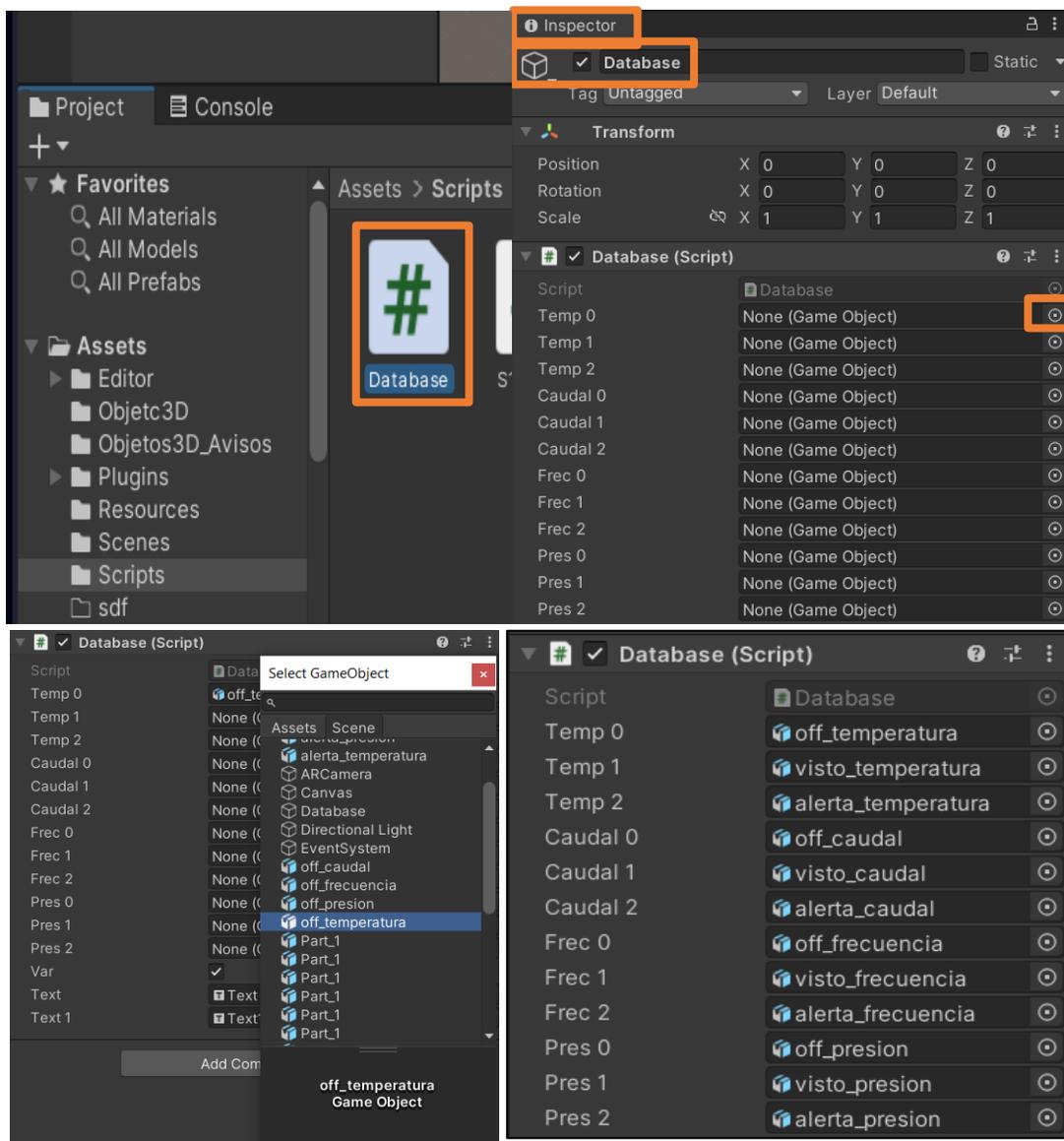


Nota. Cuando la cámara detecte el *Image Target* activará la función *DatosPresion* y cuando no detecte se activará la función *NoLeerDatosPresion*

A continuación, se debe asignar cada objeto 3D en el objeto Database, para ello se selecciona el objeto *Database*, posteriormente, en la parte de *Inspector* del editor de Unity, se puede observar los 12 GameObject que se crearon en el script, dando clic en ícono en forma de círculo que se encuentra ubicado en la derecha de cada GameObject se puede seleccionar uno de la lista según corresponda a cada estado del sensor correspondiente, como se muestra en la Figura 39.

Figura 39

Asignación de objetos 3D al Script Database



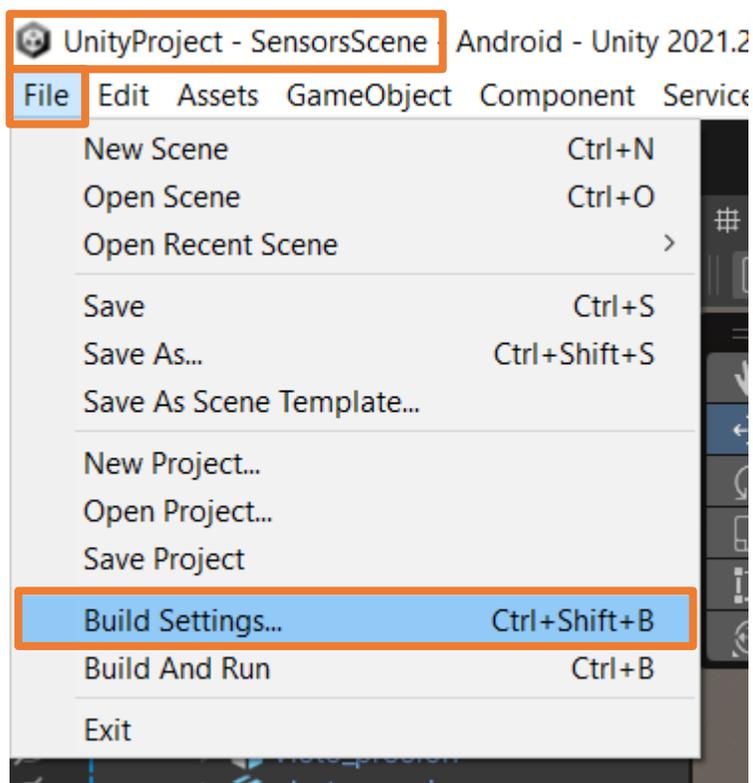
Nota. Se asignan en el orden como se muestra en la Figura 39

Exportación de Herramienta AR como librería Unity

Una vez que la herramienta de Realidad Aumentada está finalizada, se procede a exportar para su posterior implementación en la aplicación Android, para ello se presiona en la barra superior de Unity *File-Build Settings*, como se indica en la Figura 40.

Figura 40

Acceso a la configuración compilación para exportar



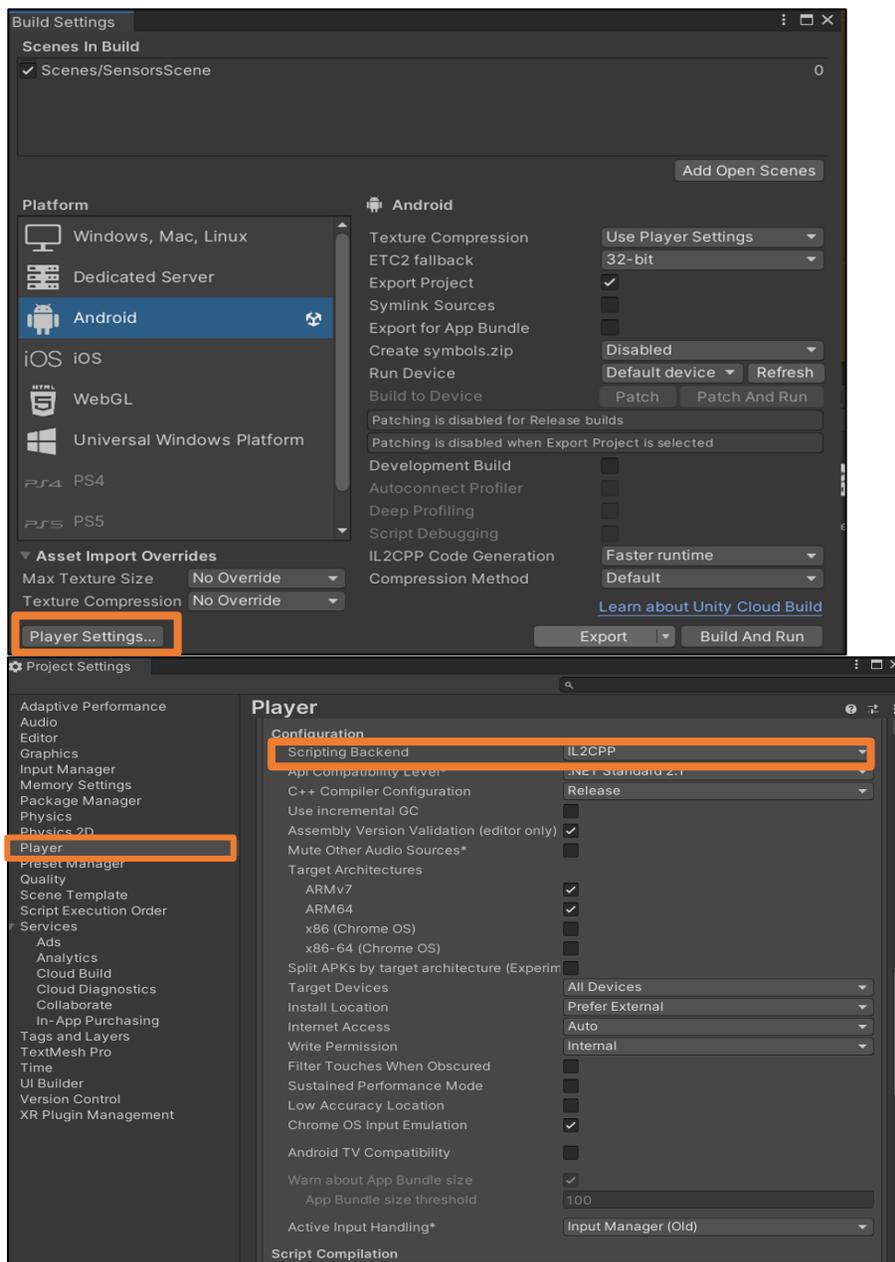
Nota. La opción "Build Settings..." permite configurar y preparar el proyecto para la generación de una compilación o ejecutable.

Es necesario activar la opción *IL2CPP* en la configuración de *Player Settings...*, como se muestra en la Figura 41, para poder interactuar con el script Database en la aplicación android.

Por último, se da clic en *Export* y se selecciona una carpeta en la cual está la librería de Unity, en este proyecto se llamará *androidBuild*, como se indica en la Figura 42.

Figura 41

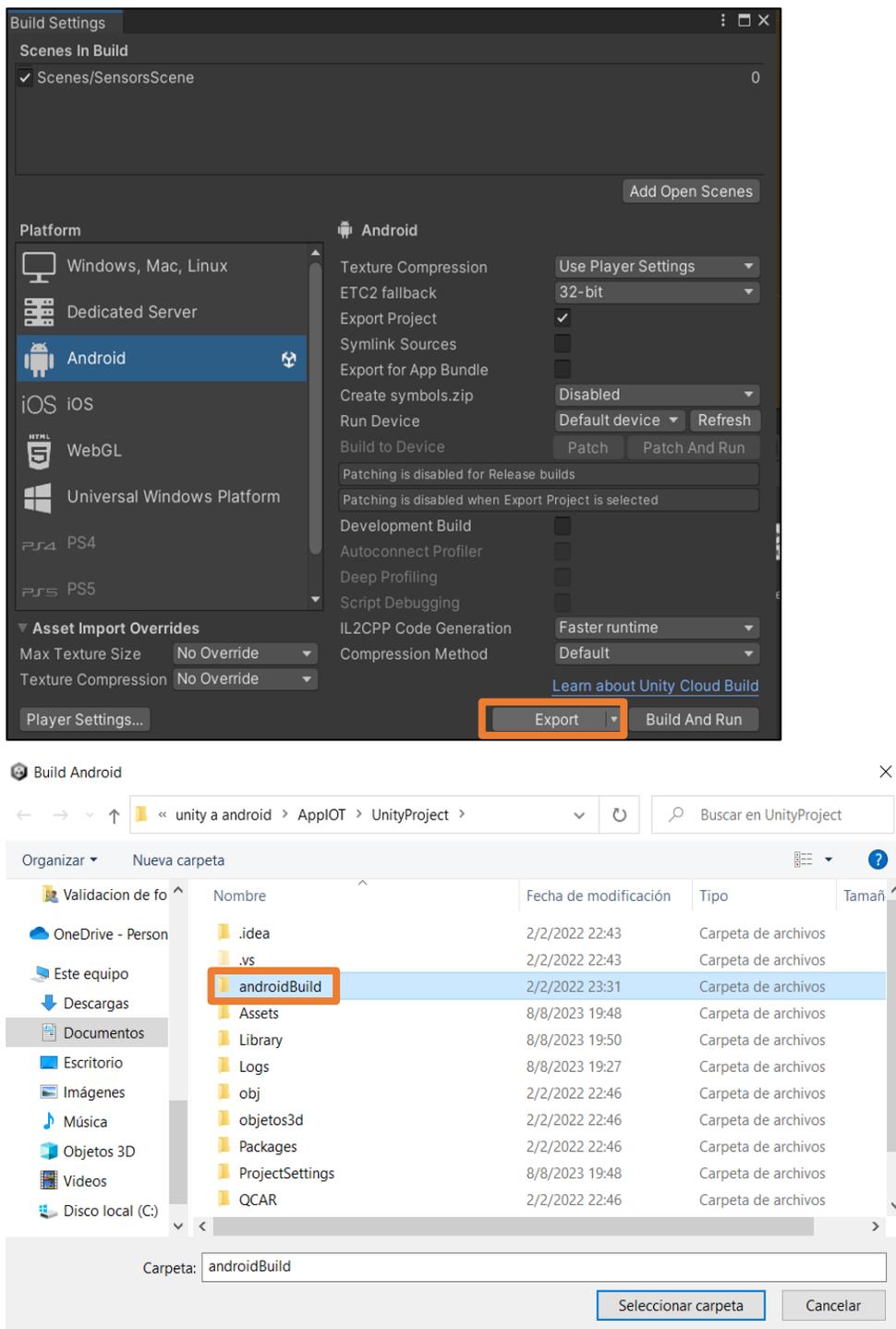
Build Settings de proyecto Unity



Nota. La opción " IL2CPP " convierte el código C# utilizado en el proyecto de Unity en código nativo de la plataforma de destino.

Figura 42

Exportación de proyecto Unity



Nota. La carpeta *androidBuild* se encuentra dentro de proyecto de Unity

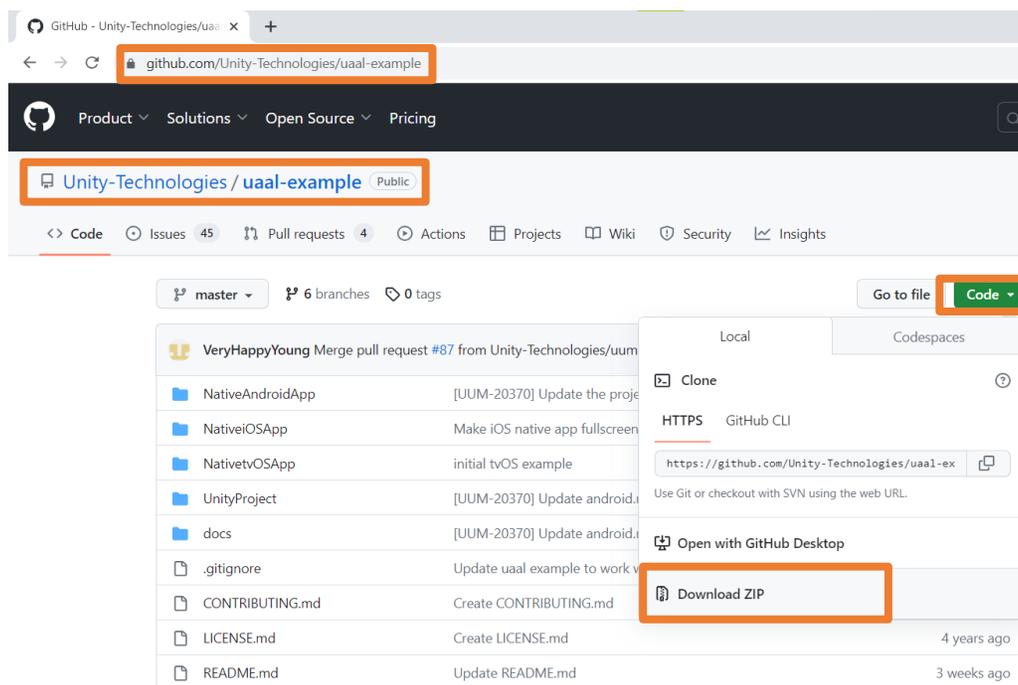
Importación de librería Unity correspondiente a la herramienta AR

Se carga un proyecto de GitHub de Unity Technologies (2019) que trata de la integración de Unity a Android como una biblioteca, el cual se encuentra al público en el perfil de Unity Technologies en la página de GitHub.

Para ello se debe descargar el proyecto *uaal-example* el cual se encuentra en el sitio web de GitHub/Unity-Technologies, dentro de ese proyecto se da clic en el botón *Code* y posteriormente en *Download ZIP*, así como se indica en la Figura 43.

Figura 43

Proyecto de Unity para exportar como librería



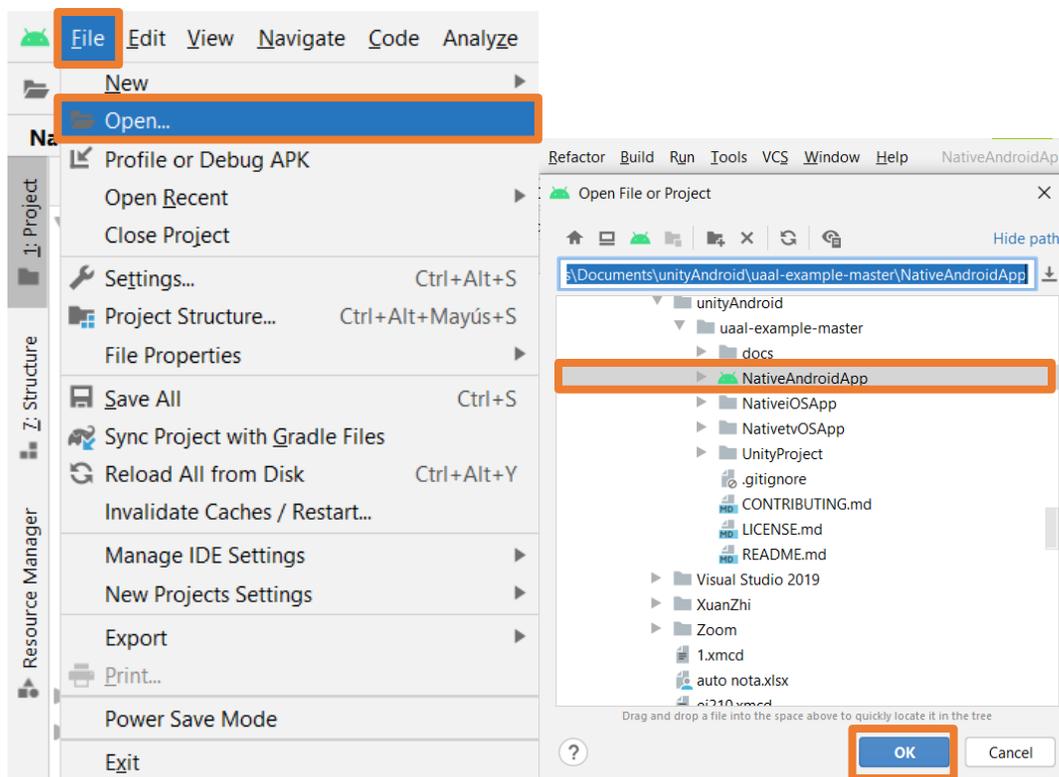
Nota. El proyecto es ejecutable a partir de la versión 2019.3.0a2 de Unity

Dicho proyecto se abre en la plataforma de Android Studio, en consecuencia, se abre la plataforma de Android Studio, luego se da clic en el menú *File*, se escoge la opción *Open...*, se

selección la carpeta *NativeAndroidApp* y se da clic en el botón *OK*, como se muestra en la Figura 44.

Figura 44

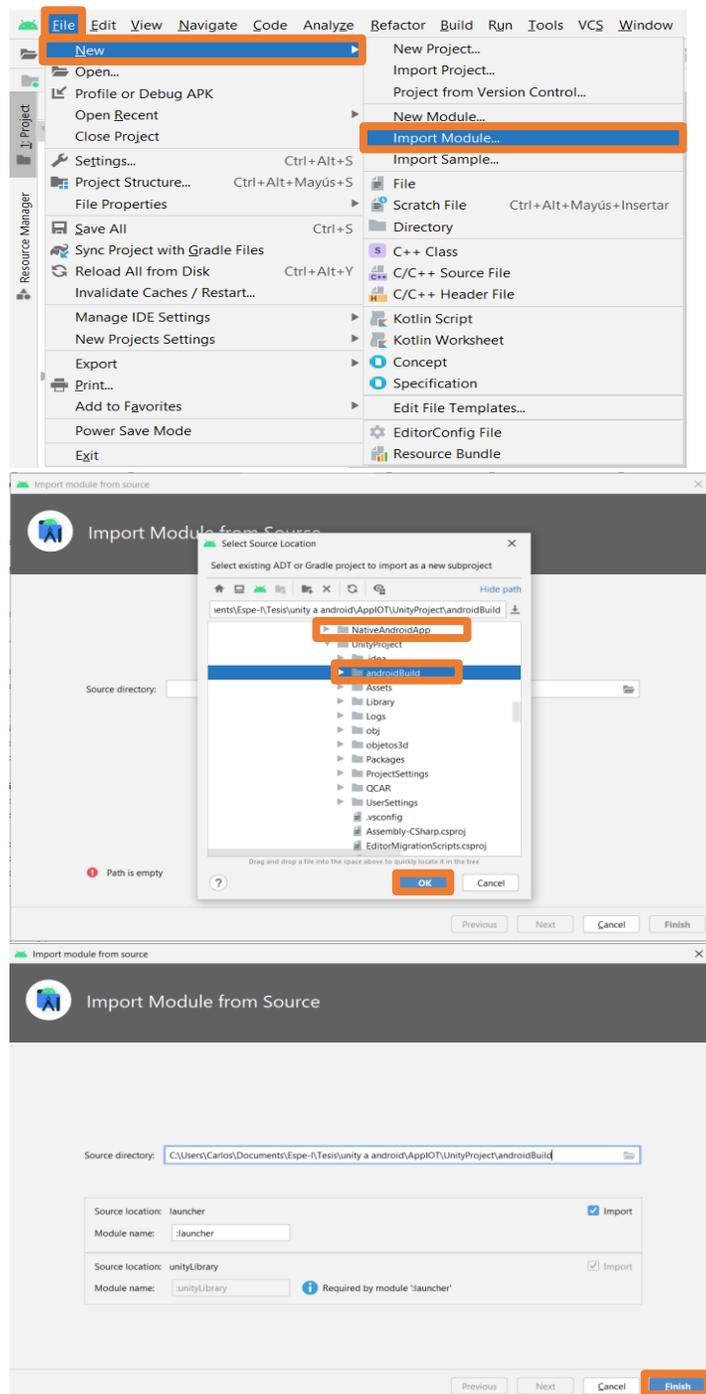
Pasos para abrir el proyecto en Android Studio



Posteriormente, se importa la herramienta AR creada en Unity, en la barra superior de Android Studio, se da clic el menú *File*, se escoge en la opción *New*, se selección la opción *Import Module...*, luego en la ventana que se abre se selecciona la carpeta donde se guardó el proyecto de Unity creado como herramienta AR, se escoge la carpeta *androidBuild*, se da clic en el botón *OK* y en el botón *Finish*, como se muestra en la Figura 45.

Figura 45

Pasos para la importación de librería Unity



Nota. Al importar la librería de Unity como módulo permite acceder a sus funciones desde código nativo de android

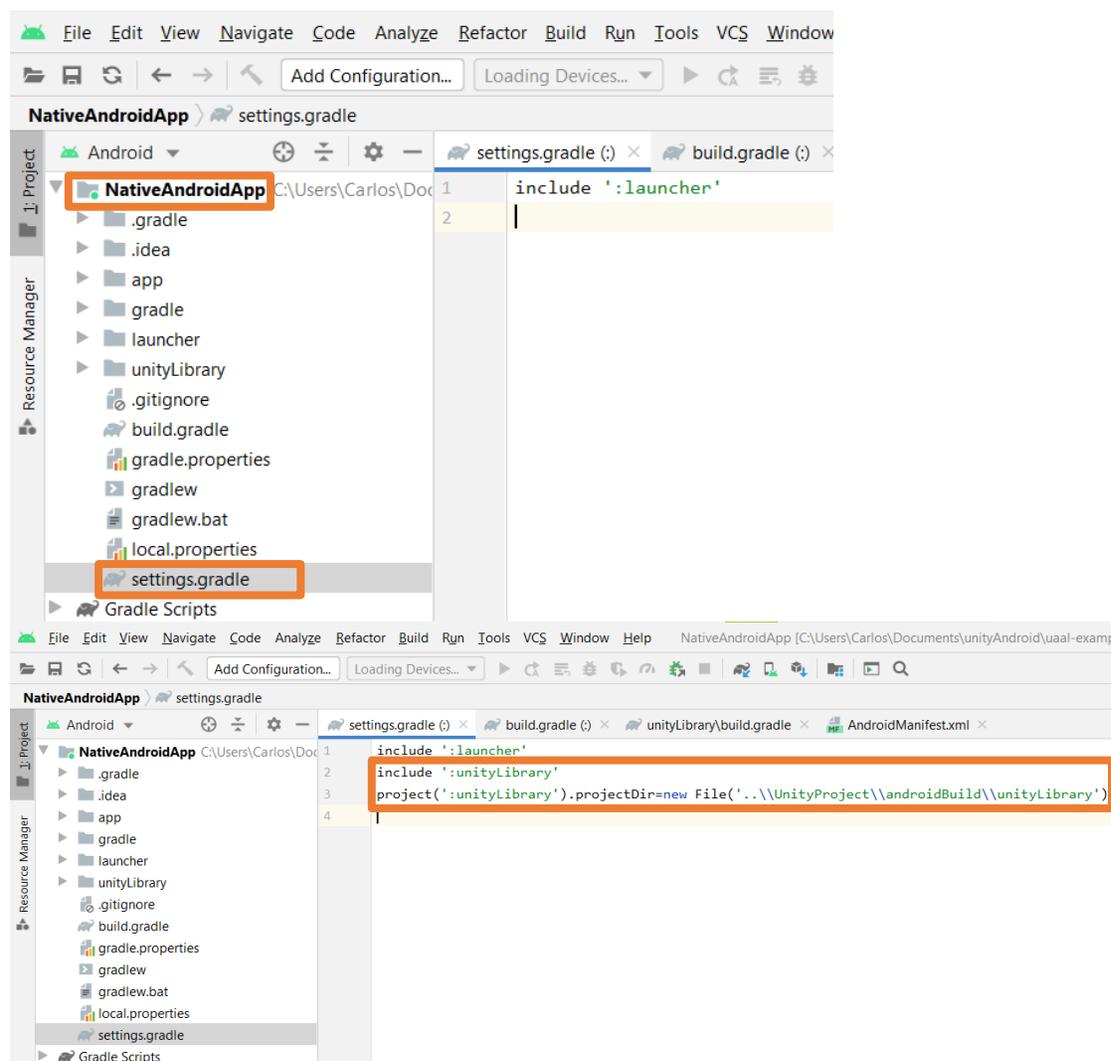
Dirigirse al archivo *settings.graddle*, agregar las 2 siguientes líneas de código:

- `include ':unityLibrary'`
- `project(':unityLibrary').projectDir=new File('..\UnityProject\androidBuild\unityLibrary')`

la dirección del archivo *settings.graddle*, y el resultado de como quedan las líneas de código ingresado se muestran en la Figura 46.

Figura 46

Configuración del archivo settings.graddle

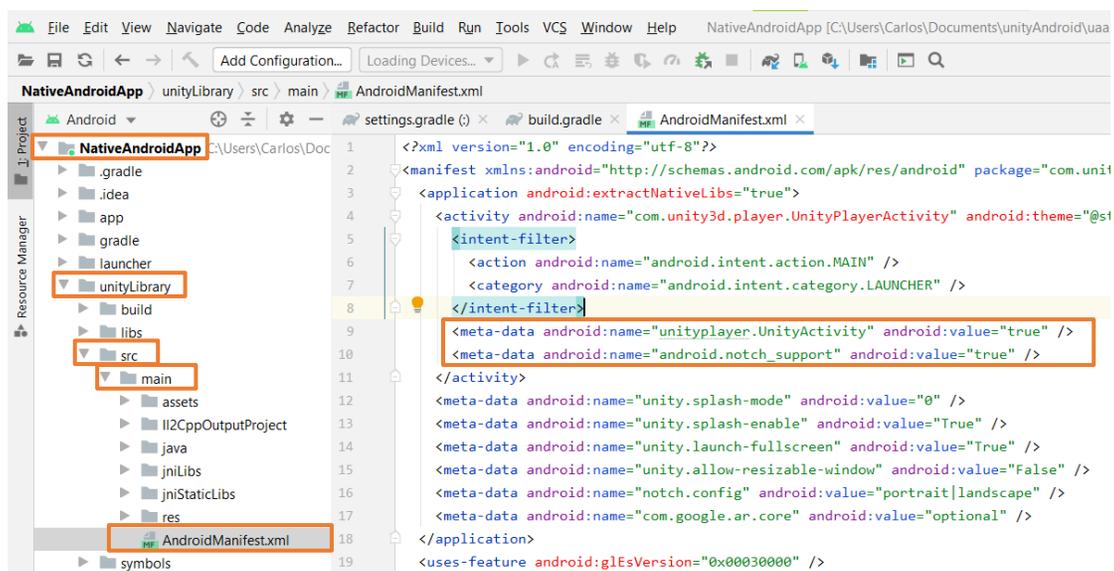


Nota. Las líneas de código agregadas permiten detectar al proyecto de Unity como librería

Luego, se debe abrir el archivo AndroidManifest.xml para eliminar unas líneas de código, el archivo debe eliminarse y se encuentra en la carpeta NativeAndroidApp/unityLibrary/src/main y las líneas de código a eliminar son las que se muestran en la Figura 47.

Figura 47

Corrección en la estructura de AndroidManifest



Nota. Las líneas de código mostradas se eliminan para evitar que se creen dos aplicaciones

Posteriormente, se modifican las siguientes líneas del archivo build.gradle de unityLibrary para corregir problemas con la herramienta AR:

```
apply plugin: 'com.android.application'
```

a

```
apply plugin: 'com.android.library'
```

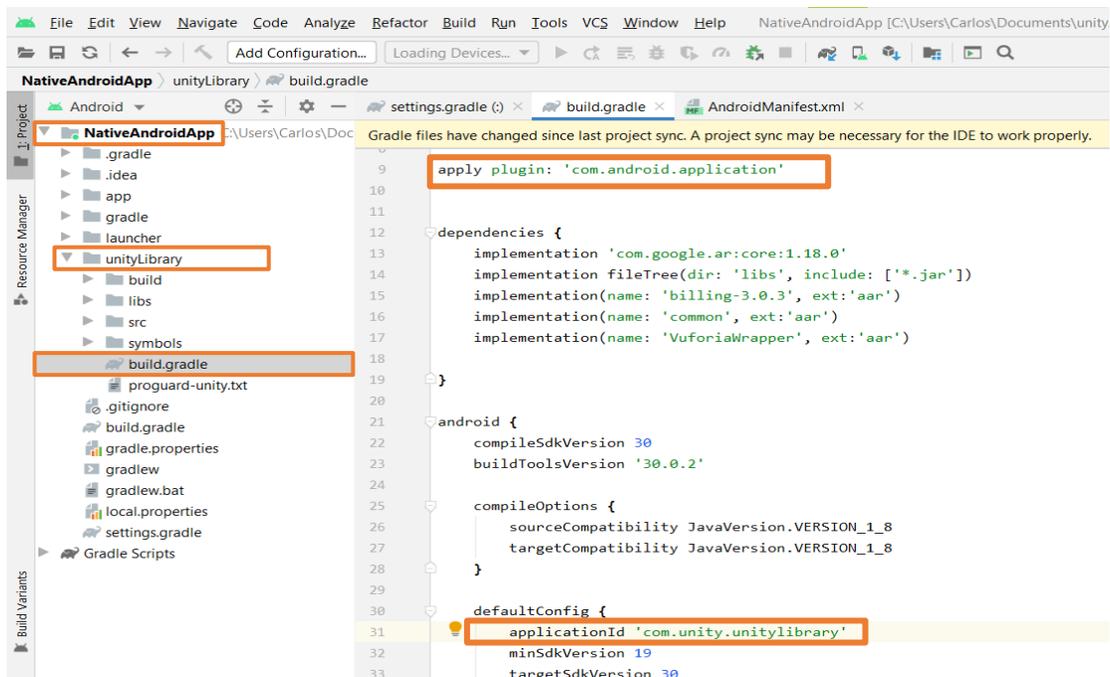
Luego se elimina la línea de código:

```
applicationId 'com.unity.unitylibrary'
```

En la Figura 48, se muestra la ubicación de archivo build.gradle, las líneas a ser modificadas.

Figura 48

Corrección de código de build.gradle



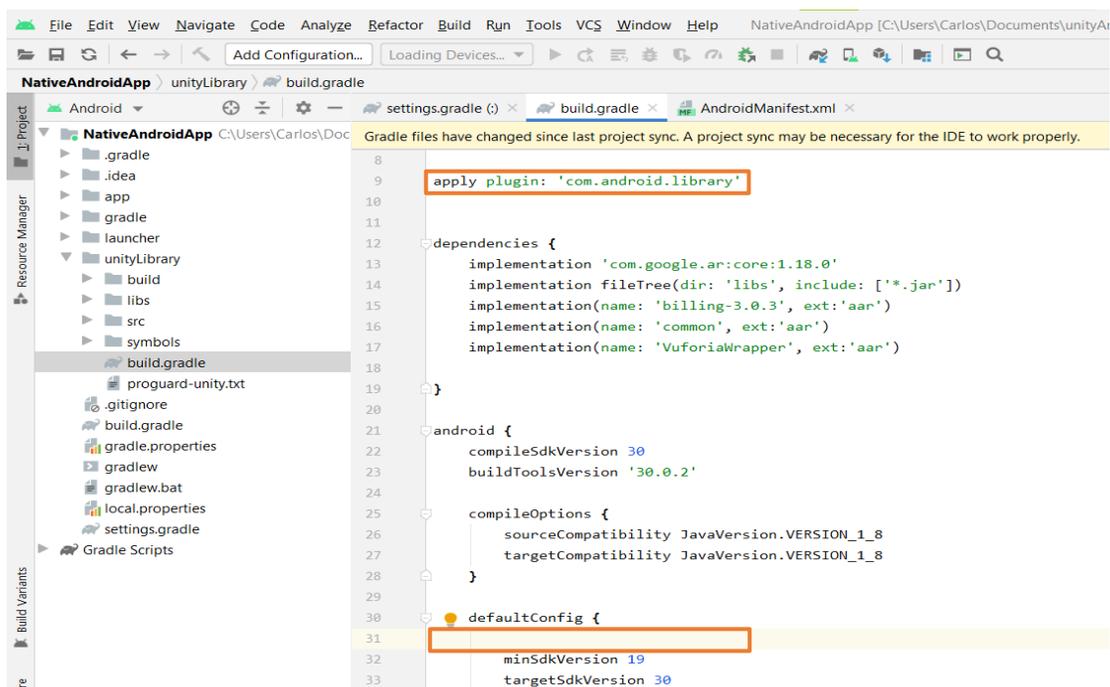
```
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help NativeAndroidApp [C:\Users\Carlos\Documents\unityA
Add Configuration... Loading Devices...
NativeAndroidApp > unityLibrary > build.gradle
Project: NativeAndroidApp C:\Users\Carlos\Doc
Resource Manager: NativeAndroidApp
  .gradle
  .idea
  app
  gradle
  launcher
  unityLibrary
    build
    libs
    src
    symbols
  build.gradle
  proguard-unity.txt
  .gitignore
  build.gradle
  gradle.properties
  gradlew
  gradlew.bat
  local.properties
  settings.gradle
  Gradle Scripts
Build Variants:
Gradle files have changed since last project sync. A project sync may be necessary for the IDE to work properly.
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
apply plugin: 'com.android.application'

dependencies {
    implementation 'com.google.ar:core:1.18.0'
    implementation fileTree(dir: 'libs', include: ['*.jar'])
    implementation(name: 'billing-3.0.3', ext:'aar')
    implementation(name: 'common', ext:'aar')
    implementation(name: 'VuforiaWrapper', ext:'aar')
}

android {
    compileSdkVersion 30
    buildToolsVersion '30.0.2'

    compileOptions {
        sourceCompatibility JavaVersion.VERSION_1_8
        targetCompatibility JavaVersion.VERSION_1_8
    }

    defaultConfig {
        applicationId 'com.unity.unitylibrary'
        minSdkVersion 19
        targetSdkVersion 30
    }
}
```



```
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help NativeAndroidApp [C:\Users\Carlos\Documents\unityA
Add Configuration... Loading Devices...
NativeAndroidApp > unityLibrary > build.gradle
Project: NativeAndroidApp C:\Users\Carlos\Doc
Resource Manager: NativeAndroidApp
  .gradle
  .idea
  app
  gradle
  launcher
  unityLibrary
    build
    libs
    src
    symbols
  build.gradle
  proguard-unity.txt
  .gitignore
  build.gradle
  gradle.properties
  gradlew
  gradlew.bat
  local.properties
  settings.gradle
  Gradle Scripts
Build Variants:
Gradle files have changed since last project sync. A project sync may be necessary for the IDE to work properly.
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
apply plugin: 'com.android.library'

dependencies {
    implementation 'com.google.ar:core:1.18.0'
    implementation fileTree(dir: 'libs', include: ['*.jar'])
    implementation(name: 'billing-3.0.3', ext:'aar')
    implementation(name: 'common', ext:'aar')
    implementation(name: 'VuforiaWrapper', ext:'aar')
}

android {
    compileSdkVersion 30
    buildToolsVersion '30.0.2'

    compileOptions {
        sourceCompatibility JavaVersion.VERSION_1_8
        targetCompatibility JavaVersion.VERSION_1_8
    }

    defaultConfig {
        minSdkVersion 19
        targetSdkVersion 30
    }
}
```

Nota. Al agregar y eliminar las líneas de código como en la Figura 48, se está ajustando la configuración de compilación de Android Studio para que sea compatible con la biblioteca de Unity que se está importando.

Al terminar de realizar los cambios de la estructura del código, el proyecto debe sincronizarse, por ello emite una opción en la parte superior del código para sincronizar, como se muestra en la Figura 49, se da clic en *Sync Now*, para terminar de sincronizar el proyecto.

Figura 49

Sincronizar estructura de herramienta AR



Diseño de la aplicación Android

Partiendo del proyecto de GitHub de Unity Technologies (2019) abierto anteriormente en la plataforma Android, se diseñan las pantallas de la interfaz, para lo cual, se requiere programar en Java y Kotlin. Se generan 6 interfaces que son: Splash Screen, Pantalla de Autenticación, Pantalla de Inicio, Pantalla de Monitoreo en tiempo real, Pantalla de Registros y Pantalla de Asistente con Realidad Aumentada.

Splash Screen

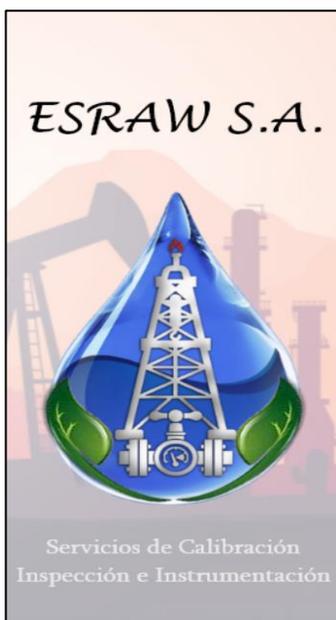
Es una pantalla de espera, mientras se inicia la aplicación en la cual se muestra el logo y nombre de la empresa ESRW S.A. como se muestra en la Figura 50, para la programación

se necesita crear la imagen necesaria, en varios tamaños para las resoluciones de los diferentes móviles android.

El diseño de la pantalla de splash screen se realizó en Adobe XD y se exportó en varios tamaños para varios tipos de densidades de imágenes. Las densidades de las imágenes se clasifican según sus dpi (*puntos por pulgada*), para anydpi o lpdi~120 dpi, mdpi~160 dpi, hdpi~240 dpi, xhdpi~320 dpi, xxhdpi~480 dpi y xxxhdpi~640 dpi. En la Figura 51, se muestra cómo se exporta las imágenes en formato .png y las opciones para escoger varios tipos de densidades.

Figura 50

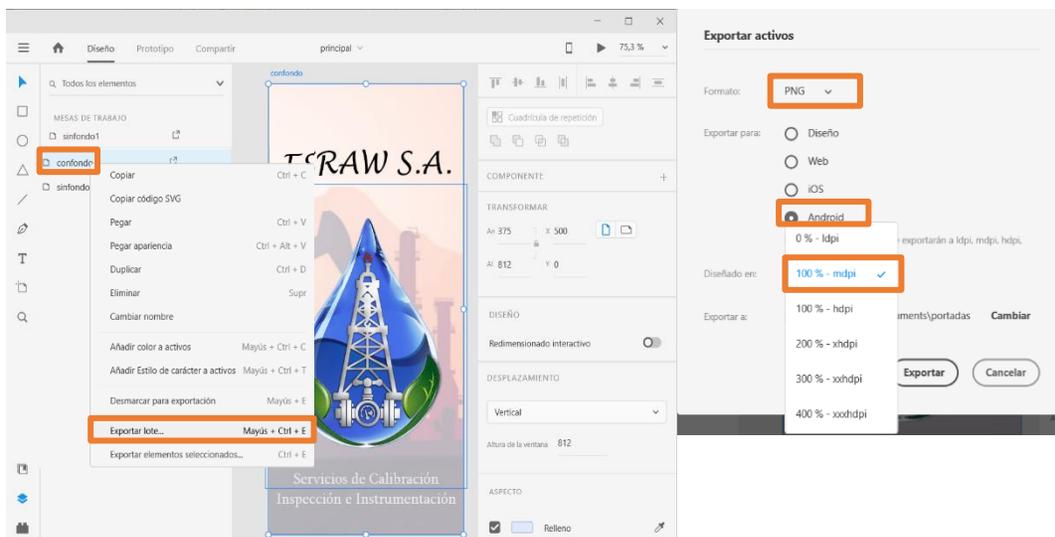
Pantalla de Splash Screen



Nota. La imagen del splash screen muestra el nombre y el logo de la empresa ESRAW S.A.

Figura 51

Exportar imágenes de Adobe XD con diferentes densidades



Las imágenes deberán ser cargadas dentro de la carpeta del proyecto, en la dirección NativeAndroidApp\app\src\main\res, en cada una de las carpetas, como se muestra en la Figura 52.

Figura 52

Dirección de las carpetas a cargar las imágenes de splash screen

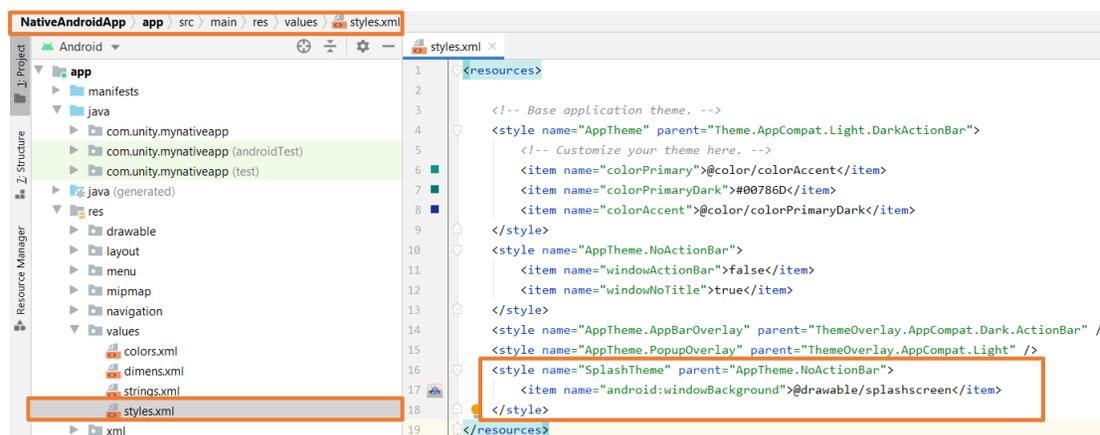
NativeAndroidApp\app\src\main\res\

Nombre	Fecha de modificación	Tipo	Tamaño
drawable	21/2/2021 15:07	Carpeta de archivos	
drawable-ldpi	23/2/2021 0:15	Carpeta de archivos	
drawable-v24	17/2/2021 21:08	Carpeta de archivos	
layout	19/7/2022 14:06	Carpeta de archivos	
menu	10/2/2022 15:05	Carpeta de archivos	
mipmap-hdpi	23/2/2021 0:15	Carpeta de archivos	
mipmap-ldpi	27/9/2019 4:20	Carpeta de archivos	
mipmap-mdpi	23/2/2021 0:15	Carpeta de archivos	
mipmap-xhdpi	23/2/2021 0:15	Carpeta de archivos	
mipmap-xxhdpi	23/2/2021 0:15	Carpeta de archivos	
mipmap-xxxhdpi	23/2/2021 0:16	Carpeta de archivos	
navigation	22/2/2021 23:26	Carpeta de archivos	
values	28/7/2023 2:51	Carpeta de archivos	
values-v23	9/8/2023 15:05	Carpeta de archivos	
xml	6/4/2021 15:41	Carpeta de archivos	

Luego, se abre el archivo `styles.xml` en la plataforma Android Studio, que se encuentra en la dirección `NativeAndroidApp/app/src/main/res/values`, para ingresar las siguientes líneas de código que se encuentran señaladas, como se muestra en la Figura 53, las cuales indican que la imagen se cargará al iniciar la aplicación. El archivo `styles.xml` es el encargado de definir el color y el tamaño de fuente, el color de fondo, y demás atributos de la apariencia de la aplicación Android, en este caso indica que se abrirá una imagen para realizar un splash screen.

Figura 53

Líneas de código para Splash Screen



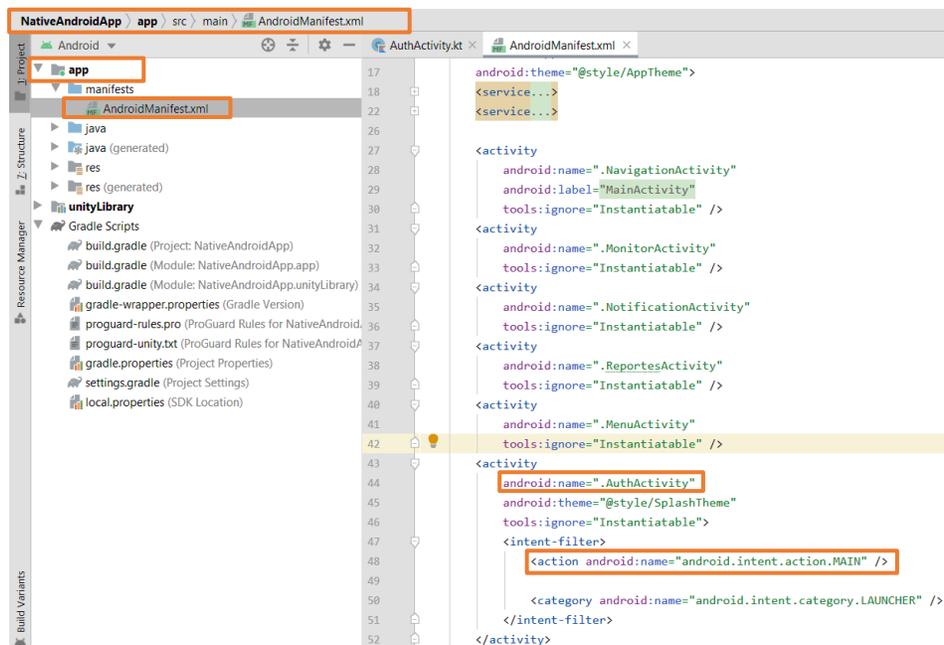
Nota. El splash screen se agrega en el archivo `styles.xml` dado que es el encargado de definir los estilos visuales y la apariencia de los componentes aplicación android

Se debe iniciar el splash screen en la primera pantalla o *activity* que se ejecuta al inicio de aplicación Android.

Para saber cuál es la primera pantalla que se inicia se debe revisar el archivo `AndroidManifest.xml` de la app, que se encuentra en la dirección `NativeAndroidApp/app/src/main`, una vez abierto el archivo se debe revisar el código que contenga el comando de pantalla inicial que es `android.intent.action.MAIN`, en este caso es `AuthActivity`, como se indica en la Figura 54.

Figura 54

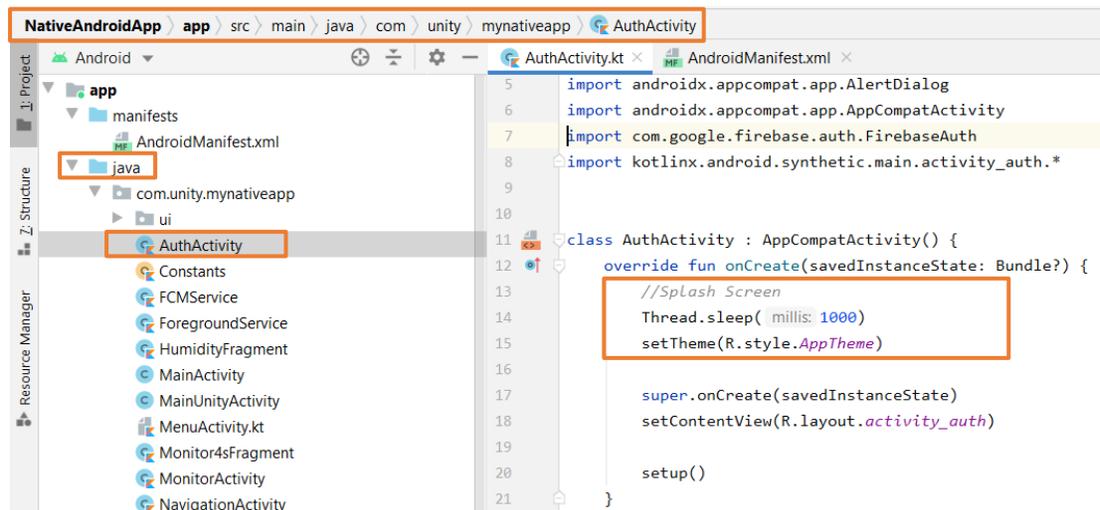
Identificación de la pantalla inicial



Después se abre el archivo *AuthActivity*, que se encuentra en la dirección `NativeAndroidApp/app/src/main/java/com/unity/mynativeapp/` y se añade las líneas de código que se señalan en la Figura 55, dentro para hacer el efecto de carga durante 1 segundo.

Figura 55

Inicialización de Splash Screen



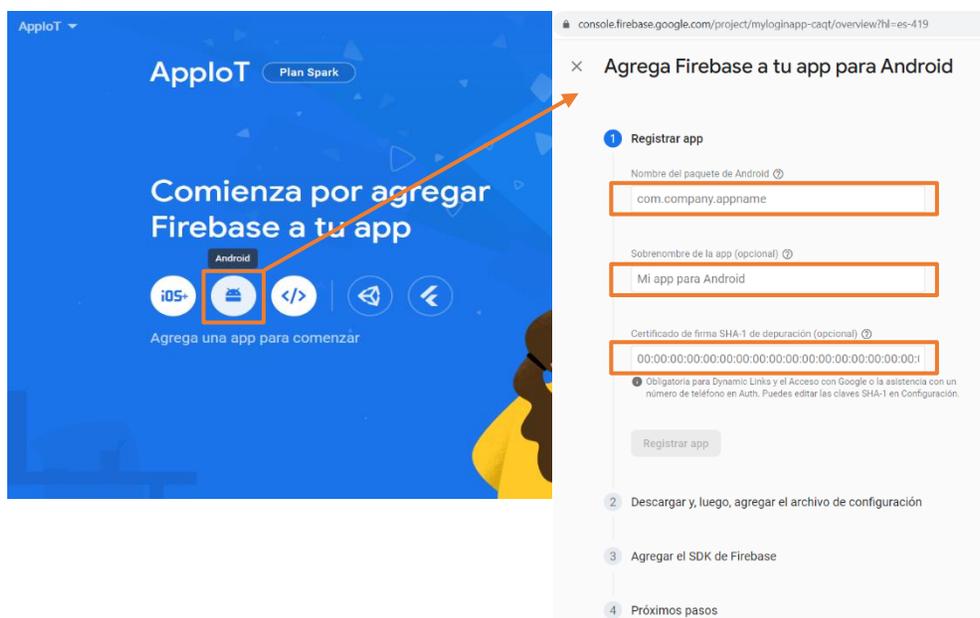
Conexión de Firebase con Android Studio

Para acceder a los servicios de Firebase con la aplicación android se debe establecer una conexión a través del sitio web de Firebase y la plataforma Android Studio.

Primero se abre la consola de Firebase dirigiéndose al sitio web <https://console.firebase.google.com/>, se ingresa el usuario registrado en este caso el correo de Gmail como se realizó en la creación de base de datos Firebase, en la pantalla que aparece se da clic el ícono de android y se debe llenar los espacios que se indican en la Figura 56.

Figura 56

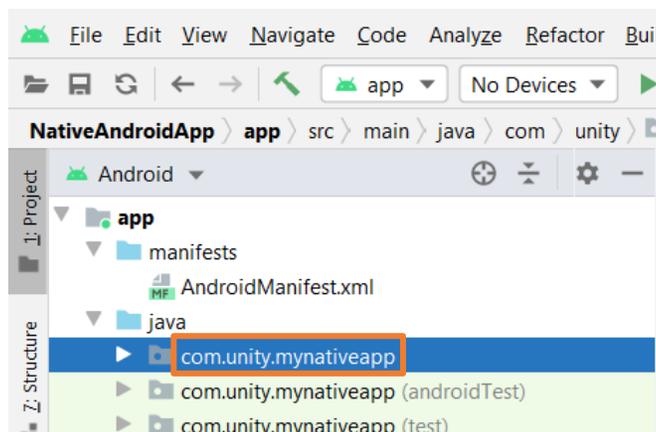
Primera parte para agregar el servicio Firebase a Android



El primer cuadro a rellenar debe llevar el nombre de paquete de Android , el cual se encuentra en el lado izquierdo de la plataforma Android estudio en la zona Project, el nombre es el que se muestra en la Figura 57.

Figura 57

Identificación del nombre de proyecto

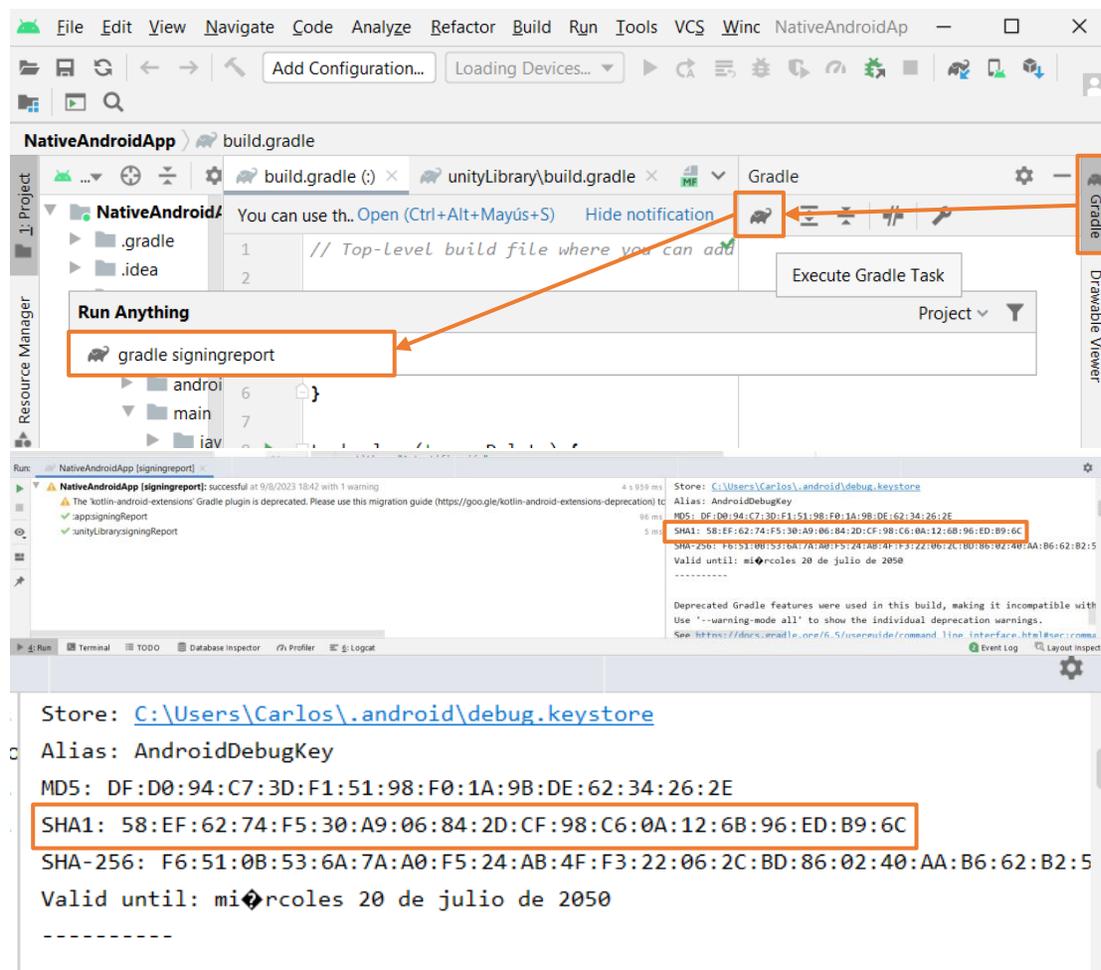


El segundo cuadro a llenar se pone el nombre de proyecto que se desee, en este caso se eligió “MyloginApp”.

El tercer recuadro a llenar el certificado de firma o huella digital SHA-1 de depuración, para encontrar dicha firma hay que dirigirse a la zona derecha de Android Studio, presionar en Gradle, luego en el ícono de elefante llamado “Execute Gradle Task”, después, ingresar el comando “gradle signingreport” que ejecutará una acción la cual dará como resultado la huella digital SHA1 de depuración, los pasos antes mencionados se muestran en la Figura 58.

Figura 58

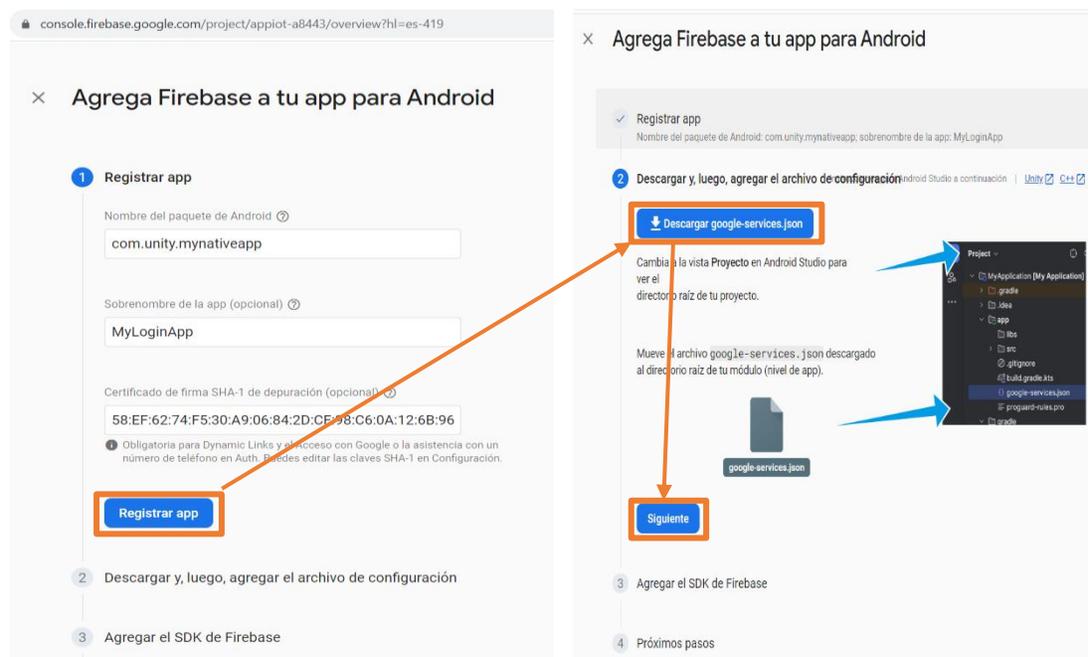
Pasos para hallar la huella digital SHA1 de depuración



Luego de llenar los 3 recuadros con la información correspondiente, se presiona el botón “Registrar app”, aparecerá otra ventana, en la cual se debe dar clic en “Descargar google-services.json” y en “Siguiente”, así como se muestra en la Figura 59.

Figura 59

Segunda parte para agregar el servicio Firebase

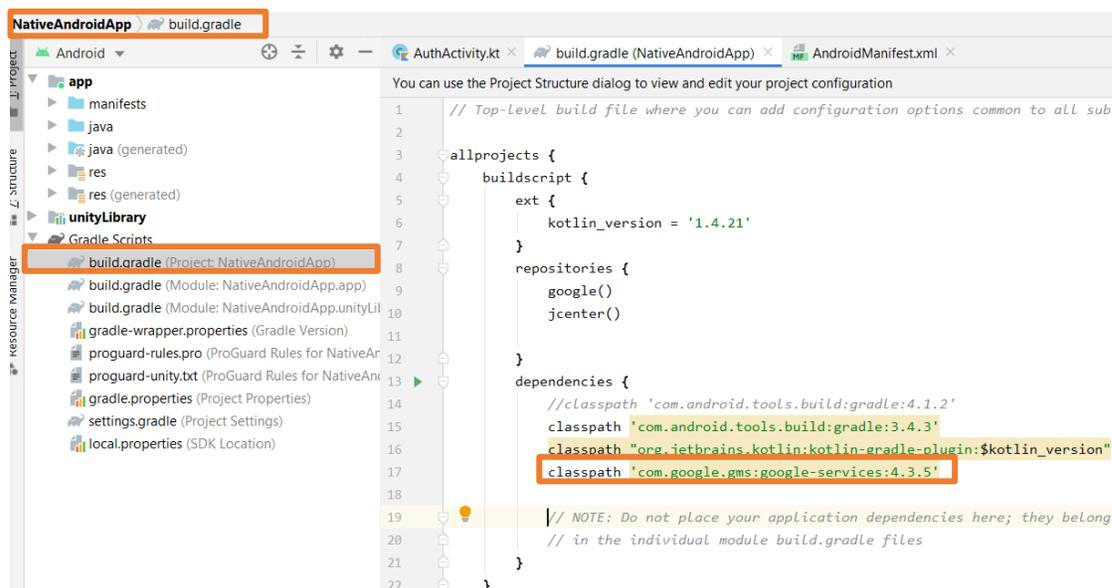


Una vez descargado el archivo “google-services.json”, se debe pegar en la carpeta app del proyecto, en la dirección “uaal-example-master\NativeAndroidApp\app”.

A continuación, se agrega una línea de código que es una dependencia en el archivo “build.gradle(Project: NativeAndroidApp)”, para poder acceder al servicio de Firebase. En la Figura 60, se indica la línea de código que se añade y la dirección de archivo.

Figura 60

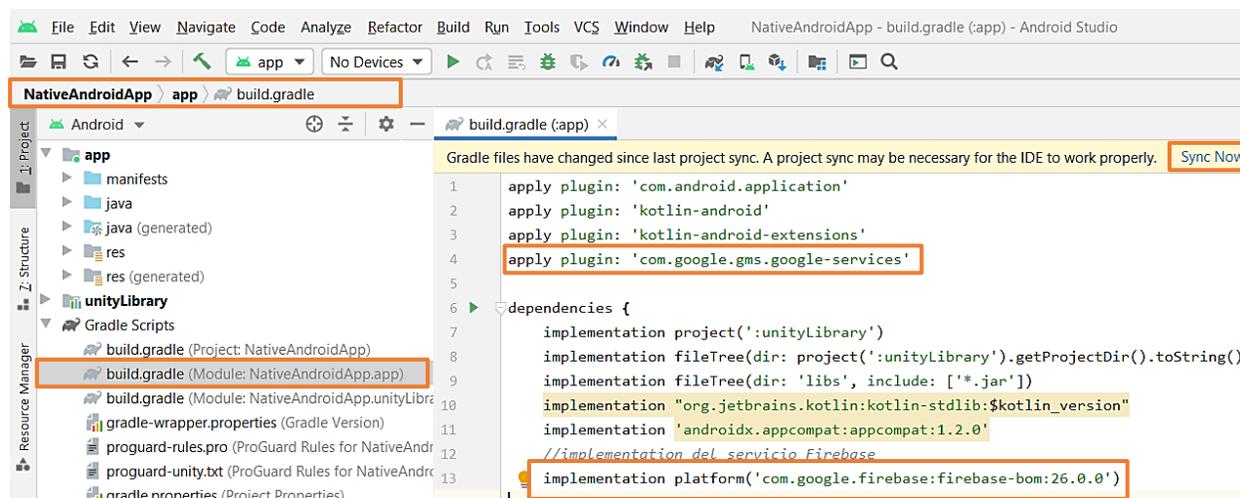
Dependencia en “build.gradle Project” para servicio Firebase



Luego, se agrega 2 líneas de código que son un complemento o plugin y una dependencia en el archivo “build.gradle(Module: NativeAndroidApp)”, para poder acceder al servicio de Firebase. En la Figura 61, se indica las líneas de código que se añaden y la dirección de archivo.

Figura 61

Plugin y dependencia en “build.gradle Module” para servicio Firebase



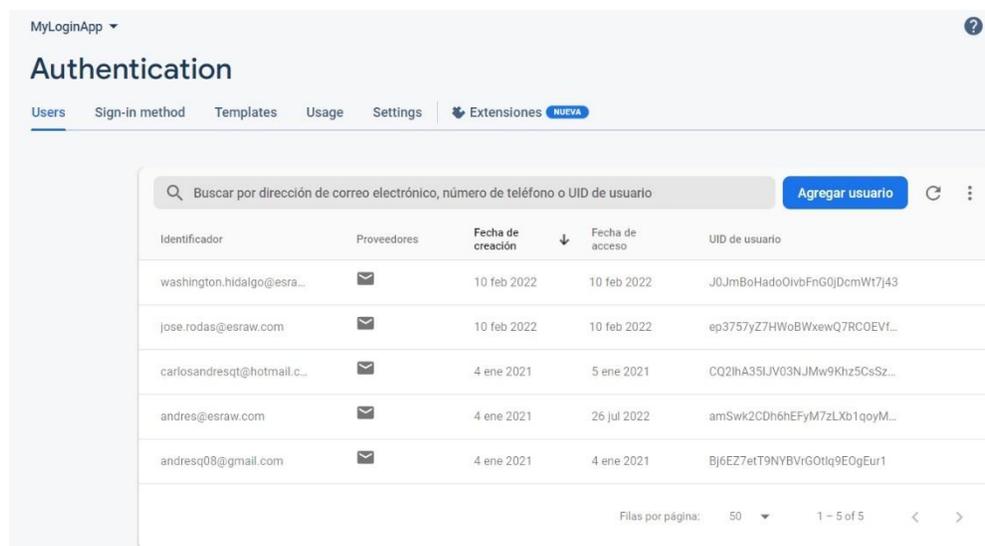
Después de agregar el complemento y dependencias necesarias, se sincroniza el proyecto de Android con archivos de Gradle, dando clic en *Sync Now*, como se indica en la parte derecha de la Figura 61.

Autenticación

La aplicación tendrá una autenticación de usuario, solo válido para los miembros de la empresa ESRW S.A., estos usuarios son proporcionados por la misma empresa y se autentica mediante base de datos de Firebase, en la Figura 62 se muestra la pestaña de autenticación.

Figura 62

Autenticación con Firebase para ESRAW S.A.



The screenshot shows the Firebase Authentication console interface. At the top, there is a search bar with the text "Buscar por dirección de correo electrónico, número de teléfono o UID de usuario" and a blue button labeled "Agregar usuario". Below the search bar is a table with the following columns: "Identificador", "Proveedores", "Fecha de creación", "Fecha de acceso", and "UID de usuario". The table contains five rows of user data. At the bottom of the table, there is a pagination control showing "Filas por página: 50" and "1 - 5 of 5".

Identificador	Proveedores	Fecha de creación	Fecha de acceso	UID de usuario
washington.hidalgo@esra...	✉	10 feb 2022	10 feb 2022	J0JmBoHadoOivbFnG0jDcmWt7j43
jose.rodas@esraw.com	✉	10 feb 2022	10 feb 2022	ep3757yZ7HWoBWxewQ7RC0EVf...
carlosandresqt@hotmail.c...	✉	4 ene 2021	5 ene 2021	CQ2lhA35lJV03NjMw9Khz5CsSz...
andres@esraw.com	✉	4 ene 2021	26 jul 2022	amSww2CDh6hEFyM7zLXb1qoyM...
andresq08@gmail.com	✉	4 ene 2021	4 ene 2021	Bj6EZ7etT9NYBvRG0tlq9E0gEur1

Nota. En la pestaña *Users* del servicio de Autenticación de Firebase se puede administrar los usuarios.

Corresponde al archivo `AuthActivity`, el cual contendrá una función que acceda a Firebase y que autentifique si el usuario es correcto o no. En la Figura 63, se muestra la dirección del archivo `AuthActivity` y la codificación para autenticar a los usuarios que ingresen en la aplicación.

Figura 63

Codificación para autenticación de Usuarios de ESRAW S.A.



```

23 public fun setup(){
24     title = "Autenticación"
25
26     loginButton.setOnClickListener { it:View?
27         if (emailEditText.text.isNotEmpty() && passwordEditText.text.isNotEmpty()){
28
29             FirebaseAuth.getInstance()
30                 .signInWithEmailAndPassword(emailEditText.text.toString(),
31                     passwordEditText.text.toString()).addOnCompleteListener{ it:Task<AuthR
32                 if(it.isSuccessful){
33                     showMenu( email: it.result?.user?.email ?: "", ProviderType.BASIC)
34                     ForegroundService.startService( context: this, message: "Monitoreando")
35                 }else{
36                     showAlert()
37                 }
38             }
39         }
40     }
41 }

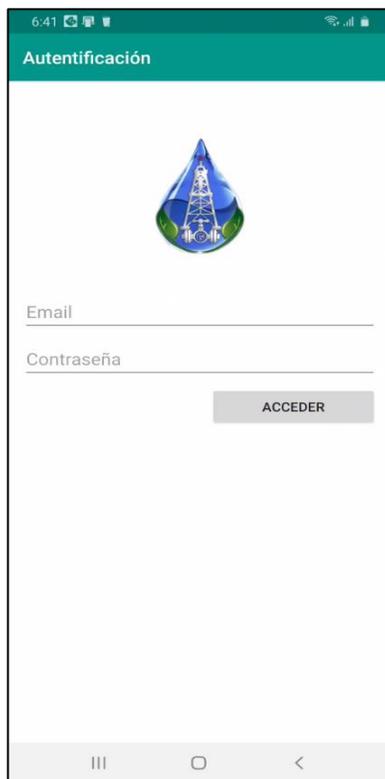
```

Nota. El código intenta autenticar al usuario mediante Firebase Authentication utilizando un correo electrónico y contraseña proporcionados. Si la autenticación tiene éxito, se muestra la pantalla de Menú y se inicia un servicio en primer plano para emitir notificaciones, si la autenticación falla, se muestra una alerta.

En la Figura 64, se muestra la pantalla de Autenticación, con el logo de la empresa ESRAW S.A., dos casilleros de texto correspondientes al email empresarial y contraseña proporcionada por la empresa, además de un botón de acceder para autenticar los datos insertados en los casilleros de texto.

Figura 64

Pantalla de Autenticación



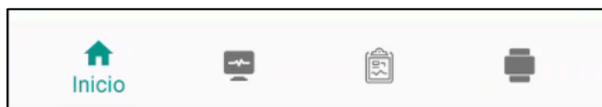
Nota. En la pantalla de Autenticación se inicia con un correo registrado y una contraseña, como se indica en el Manual de Usuario (Anexo 2).

Barra navegación

Para navegar entre las pestañas de la aplicación Android se necesita de una barra de navegación para poder acceder a las pantallas Menú, Monitoreo en Tiempo Real, Registro de Datos y a la herramienta AR, total son 4 ítems, como se muestra en la Figura 65, la primera desde la izquierda es el ícono de inicio el cual muestra la pestaña de pantalla de Menú, el siguiente ícono se asemeja a un monitor esta entrelazado con la pestaña de monitoreo, el siguiente ícono que se asemeja a una hoja de papel, los dirige a la pestaña de registro de datos y último es el ícono que permite abrir la herramienta AR.

Figura 65

Barra de navegación



Nota. La barra de navegación se encuentra en la parte inferior de la aplicación.

En la Figura 66, se muestra la codificación de la barra de navegación.

Figura 66

Codificación de barra de navegación.

```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android"
      xmlns:app="http://schemas.android.com/apk/res-auto">

  <item
    android:id="@+id/navigation_home"
    android:icon="@drawable/ic_home_black_24dp"
    android:title="@string/title_home" />

  <item
    android:id="@+id/navigation_monitoreo"
    android:icon="@drawable/ic_monitor"
    android:title="@string/title_monitoreo" />

  <item
    android:id="@+id/navigation_registro"
    android:icon="@drawable/ic_report"
    android:title="@string/title_registro" />

  <item
    android:id="@+id/navigation_ra"
    android:icon="@drawable/ic_ra"
    android:title="Asistencia con Realidad Aumentada" />

</menu>
```

Nota. Cada <item> representa un elemento de la barra de navegación con un ícono y un título asociado.

La función “setupNavigation()” que se muestra en la Figura 67, se incluyen en las 4 pantallas (Menú, Monitoreo en Tiempo Real, Registro de Datos y Herramienta AR) principales

que permiten navegar entre pantallas al presionar cada uno de los íconos de la barra de navegación.

Figura 67

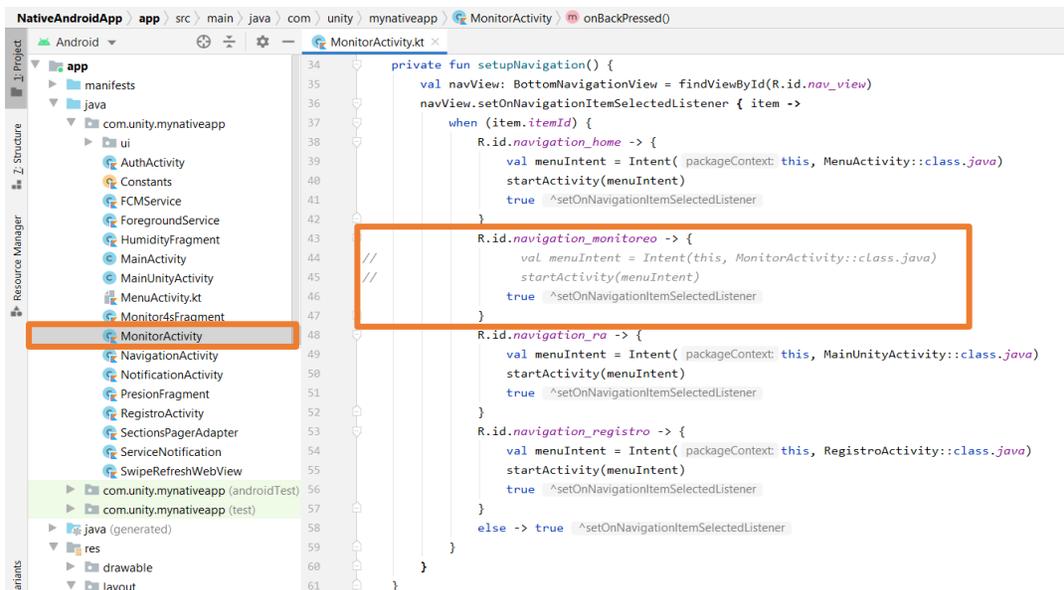
Codificación de la función navegar entre pantallas desde la barra de navegación

```
private fun setupNavigation() {
    val navView: BottomNavigationView = findViewById(R.id.nav_view)
    navView.setOnNavigationItemSelectedListener { item ->
        when (item.itemId) {
            R.id.navigation_home -> {
                val menuIntent = Intent( packageContext: this, MenuActivity::class.java)
                startActivity(menuIntent)
                true ^setOnNavigationItemSelectedListener
            }
            R.id.navigation_monitoreo -> {
                val menuIntent = Intent( packageContext: this, MonitorActivity::class.java)
                startActivity(menuIntent)
                true ^setOnNavigationItemSelectedListener
            }
            R.id.navigation_ra -> {
                val menuIntent = Intent( packageContext: this, MainUnityActivity::class.java)
                startActivity(menuIntent)
                true ^setOnNavigationItemSelectedListener
            }
            R.id.navigation_registro -> {
                val menuIntent = Intent( packageContext: this, RegistroActivity::class.java)
                startActivity(menuIntent)
                true ^setOnNavigationItemSelectedListener
            }
            else -> true ^setOnNavigationItemSelectedListener
        }
    }
}
```

En la Figura 68, se muestra cómo se implementó la codificación para navegar entre pantallas desde la barra de navegación con una modificación la cual está señalada en un recuadro. La modificación en el código se realizó para que el caso cuando se presiona el ícono de Monitoreo, no se vuelva abrir la pantalla de monitoreo, si no que se mantenga sin ninguna acción; para los demás íconos si pueden abrir las demás pantallas.

Figura 68

Implementación de función de navegación entre pantallas opción Monitor



Nota. Cuando se presione en el ícono de monitor y se encuentre activa la pantalla de *Monitoreo en tiempo real* pantalla se mantendrá sin ninguna novedad, pero en caso que se presionen los otros 3 íconos, la aplicación se dirigirá a la pantalla de su respectivo ícono presionado.

Pantalla Menú

Una vez autenticado el usuario accederá a una pantalla menú, que muestra el logo de la empresa ESRW S.A. seguido de una breve descripción de los servicios que ofrece la empresa, además de un botón *CERRAR SESIÓN* y dejar de monitorear en segundo plano, como se muestra en la Figura 69.

Figura 69*Pantalla Menú****Pantalla Monitoreo en Tiempo Real***

Para el monitoreo en tiempo real es necesario acceder a la base de datos, que se encuentra en Firebase, para lo cual es necesario codificar en la actividad Monitor4sFragment, como se presenta en la Figura 70, donde se toma los valores de la base de datos que se encuentra en formato float y los transforma en string para después mostrar en la interfaz.

Primero se declaran las variables que inicialicen la base de datos, luego se accede a los valores de la base de datos y se capturan en los datos almacenados en ese momento con la función child y con la extensión apropiada en la cual se encuentran los valores enviados por la tarjeta UNO WiFi R3 ATmega328P ESP8266.

Figura 70

Codificación para obtener valores de la base de datos

```

28 }
29
30 override fun onActivityCreated(savedInstanceState: Bundle?) {
31     super.onActivityCreated(savedInstanceState)
32
33     val database = Firebase.database
34     val myRef = database.reference
35
36     myRef.addValueEventListener(object : ValueEventListener {
37         @SuppressWarnings("SetTextI18n")
38         override fun onDataChange(dataSnapshot: DataSnapshot) {
39             val temperaturav: String = dataSnapshot.child( path: "Estaciones/B").value.toString()
40             val presionv: String = dataSnapshot.child( path: "Estaciones/C").value.toString()
41             val flujov: String = dataSnapshot.child( path: "Estaciones/D").value.toString()
42             val frecuenciav: String = dataSnapshot.child( path: "Estaciones/E").value.toString()
43
44             temperatureTextView.text = temperaturav + " C°"
45             presionTextView.text = presionv + " psi"
46             flujoTextView.text = flujov + " gal/min"
47             turbinaTextView.text = frecuenciav + " Hz"
48         }
49     })
50
51     override fun onCancelled(error: DatabaseError) {
52         Log.w( tag: "TAG", msg: "Failed to read value.", error.toException())
53     }
54 }
55

```

Nota. Este código establece una función que lee la base de datos de Firebase. Cada vez que los datos en la base de datos cambian, la función “onDataChange” se ejecuta, obteniendo los valores de diferentes nodos y actualizando las vistas en la interfaz de usuario con esos valores.

En la Figura 71, se muestra la interfaz de *Monitoreo en tiempo real*, en la parte superior está el título de la pestaña “Monitoreo del Banco de Calibración”, después la pantalla se divide en cuatro partes, en las cuales se distribuyen las diferentes magnitudes monitoreadas.

Figura 71

Pantalla de Monitoreo en tiempo real



Nota. Los valores de temperatura se presentan en grados Celsius, los valores de presión en psi, los valores de frecuencia en Hertz y los valores de caudal en galones por minuto.

Interfaz de registro de datos

Para generar los registros de datos se utiliza Looker Studio, para acceder a Looker Studio se ingresa al sitio web <https://lookerstudio.google.com>, desde un navegador con internet, para crear un registro de datos se necesita una cuenta Gmail, en la Figura 72 se muestra como la presentación del registro desde la página de Looker Studio.

Looker Studio se utiliza como una herramienta en web para realizar informes, por lo cual la interfaz debe mostrar la página web del informe que genera Looker Studio, se codifica una WebView para acceder a la página web como si fuera un navegador, se aplica la codificación mostrada en la Figura 73, con la dirección URL

“https://lookerstudio.google.com/u/0/reporting/12d3bb11-54fd-46f4-b53e-d24fac31998e/page/RR2sB”, la dirección URL mencionada anteriormente contiene el registro de datos de Google Looker Studio.

Figura 72

Registro de Datos en Looker Studio

Hora	Presión	Temperatura	Flujo	Frecuencia	Fecha
1. 6:42:48 p. m.	44.65	0.95	85.79	22.48	20220922..
2. 12:29:02 p. m.	25.3	5.5	171.58	41.9	20220805..
3. 8:08:11 a. m.	25.3	5.5	171.58	41.9	20220806..
4. 9:14:48 p. m.	25.3	5.5	171.58	41.9	20220779..
5. 12:42:25 a. m.	25.3	5.5	171.58	41.9	20220801..
6. 9:46:49 a. m.	25.3	5.5	171.58	41.9	20220807..
7. 9:25:50 a. m.	25.3	5.5	171.58	41.9	20220729..
8. 8:24:50 p. m.	25.3	5.5	171.58	41.9	20220806..
9. 3:56:49 p. m.	25.3	5.5	171.58	41.9	20220727..
10. 4:42:48 a. m.	25.3	5.5	171.58	41.9	20220803..
11. 1:51:51 a. m.	25.3	5.5	171.58	41.9	20220727..
12. 7:31:48 p. m.	25.3	5.5	171.58	41.9	20220829..
13. 11:38:12 p. m.	25.3	5.5	171.58	41.9	20220807..

Figura 73

Codificación para WebView

```

35
36 //Refresh
37 swipeRefreshLayout.setOnRefreshListener {
38     webView.reload()
39 }
40 swipeRefreshLayout.isEnabled = false
41 webView.settings.javaScriptEnabled = true
42 webView.settings.domStorageEnabled = true
43
44 //WebView
45 webView.webChromeClient = object : WebChromeClient() {
46
47     webView.webViewClient = object : WebViewClient() {
48         override fun shouldOverrideUrlLoading(view: WebView?, url: String?): Boolean {
49             return false
50         }
51         override fun onPageStarted(view: WebView?, url: String?, favicon: Bitmap?) {
52             super.onPageStarted(view, url, favicon)
53             swipeRefreshLayout.isRefreshing = true
54         }
55         override fun onPageFinished(view: WebView?, url: String?) {
56             super.onPageFinished(view, url)
57             swipeRefreshLayout.isRefreshing = false
58         }
59     }
60
61     webView.refreshStateListener = { it: Boolean
62         swipeRefreshLayout.isEnabled = it
63     }
64
65     webView.loadUrl(BASE_URL)
66 }

```

Adicional en la codificación se agrega la función *swipeRefresh*, en caso de requerir actualizar los datos de los registros.

En la Figura 74, se muestra la interfaz de registro de datos, generado en Google Looker Studio.

Figura 74

Interfaz de Registro de Datos



A continuación, se ingresa a Google Scripts a través de su sitio web <https://www.google.com/script/start/>, se inicia un Script dado clic en Start Scripting, se crea un nuevo proyecto, se genera un nuevo script el cual se programará para obtener los datos y enviarlos a una hoja de cálculo de Google.

En la Figura 75, se muestra cómo se establece la comunicación entre la base de datos en Firebase y el Script.

En la Figura 76, se presenta la codificación de ingresar los valores a almacenar en la hoja de cálculo de Google.

Luego, se obtiene un objeto tipo JSON de la base de datos, que se lee dependiendo la ubicación de cada valor, como se muestra en la Figura 76, después, se envía los datos a una hoja de cálculo de Google en la que se almacena la información de los sensores del banco de pruebas, como se muestra en la Figura 77.

Figura 75

Codificación para obtener información de la base de Datos de Firebase

```
function obtenerDatos1() {
  // Dirección de la base de datos RealTime de Firebase
  var bd = "myloginapp-caqt.firebaseio.com" ;
  // Secreto de la base de datos Firebase
  var secreto = 'WBWK69LIDV1BwEQxk6bBxPtPOFswuehhIbj18poy';
  // Ruta de la base de datos RealTime
  var ruta = "Estaciones";

  var options = { method:'get', contentType: 'application/json' }; //opciones de la llamada a fetch
  var firebaseURL = 'https://'+bd+'/'+ruta+'.json?auth='+secreto; // construyo la ruta

  var resultado = UrlFetchApp.fetch(firebaseURL, options);

  // Hoja de cálculo activa
  var ss = SpreadsheetApp.getActive();
  var sheet = ss.getSheetByName("portafolio")

  var ultimalinea = sheet.getLastRow(); // obtiene la última línea

  // Fechas y horas actuales.
  var fecha=Utilities.formatDate(new Date, "GMT-5", "yyyy/MM/dd");
  var d = new Date();
  var hora = d.toLocaleTimeString();
}
```

Figura 76

Envío de datos de Firebase a la hoja de cálculo de Google

```
// Objeto JSON según el resultado

var e = JSON.parse(resultado);
//var e = importDate();
//Logger.log(e["A"]);
//if(e["A"]>0){
if(true){

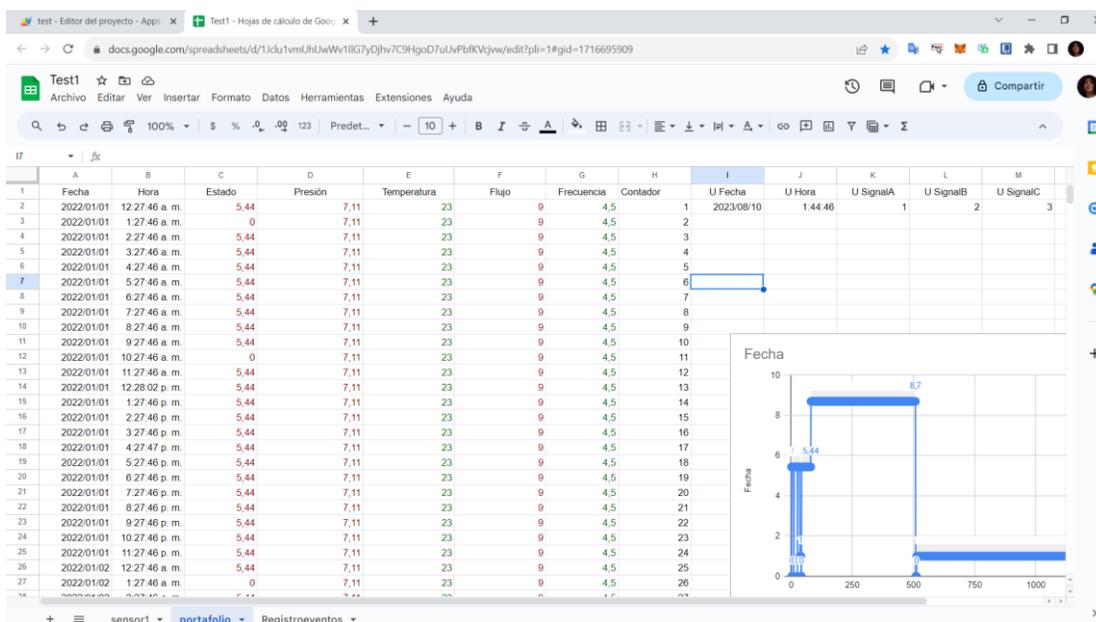
for (clave in e) {
  var columna = clave.charCodeAt(0)-62; // columna de cada valor
  //Logger.log(columna);
  var valorActual = sheet.getRange(2,columna).getValue();
  if (true) { // Si los valores son distintos
    // 1) Cambio fecha y hora en la fila dos
    sheet.getRange(2,9).setValue(fecha);
    sheet.getRange(2,10).setValue(hora);
    // 2) Cambio valor actual en la fila dos
    sheet.getRange(2,columna+8).setValue((e[clave]));
    // 3) Pongo fecha y hora en la última línea
    sheet.getRange(ultimaleña+1,1).setValue(fecha);
    sheet.getRange(ultimaleña+1,2).setValue(hora);
    // 4) Escribo en la última línea un registro de lo que acontece
    segunActivo1(clave,e[clave],ultimaleña+1,columna,sheet);
  }else{
    blanco(ultimaleña+1,columna,sheet);
  }

  // Logging de los datos...
  Logger.log(e);
  Logger.log(clave);
  //Logger.log(e[clave]);
}

}
```

Figura 77

Implementación de datos en Hoja de Cálculo de Google



Notificaciones y eventos

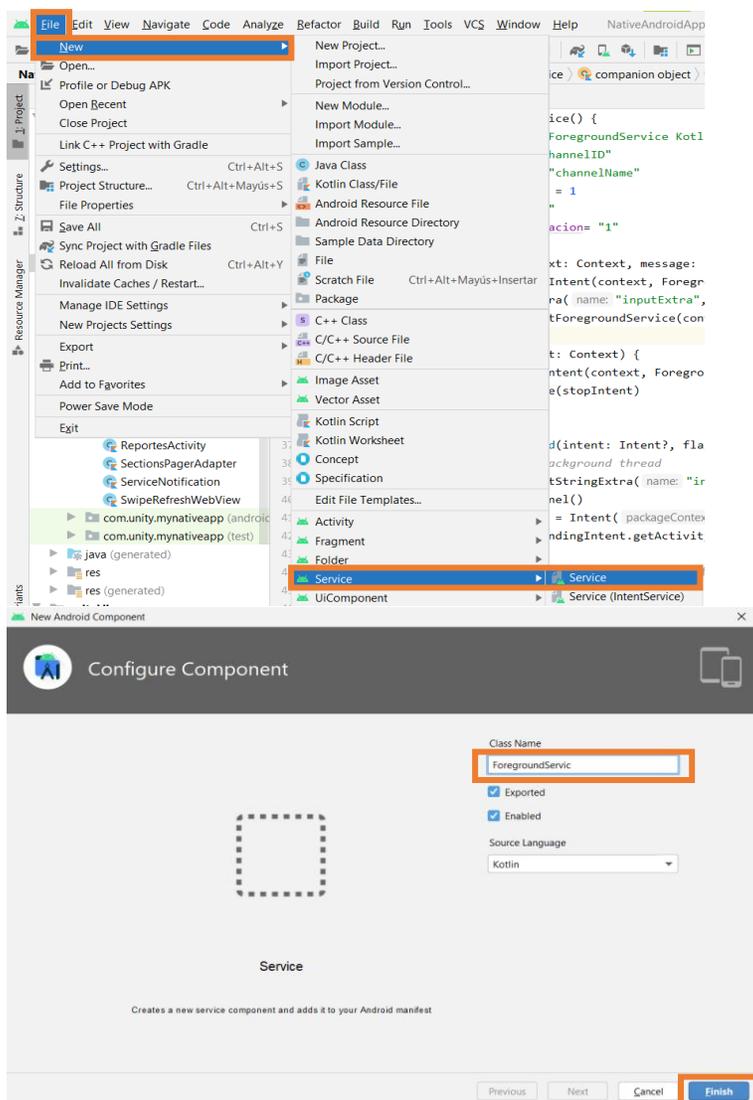
Las notificaciones se generan con un servicio que se inicializa cuando el usuario ha sido autenticado. Mediante un servicio llamado *Foreground*, se mantiene en ejecutándose en segundo plano hasta que el usuario cierre su sesión; caso contrario, aunque el usuario cierre la aplicación o esté en otras aplicaciones, este servicio se seguirá ejecutando y enviando notificaciones al usuario sobre el estado del banco de calibración de caudal.

Para crear el servicio anteriormente mencionado, se da clic en el menú *File* de la plataforma Android Studio, luego clic en *New*, en el menú *Servicio* se escoge la opción *Servicio*, así como se muestra en la Figura 78, se le asigna un nombre al servicio y se da clic en el botón *Finish*.

En la Figura 79, se muestra las funciones que inicializan el servicio *Foreground* cuando el usuario pase de la actividad *Autenticación*.

Figura 78

Pasos para crear un servicio en Android Studio



Nota. El servicio Foreground se utilizará para garantizar que las notificaciones se entreguen de manera confiable y oportuna, independientemente de si la aplicación está actualmente visible para el usuario

Figura 79

Codificación de *ForegroundService*

```

class ForegroundService : Service() {
    private val CHANNEL_ID = "ForegroundService Kotlin"
    private val channelId = "channelID"
    private val channelName = "channelName"
    private val notificationId = 1
    private val TAG = "service"
    private var MensajeNotificacion = "1"
    companion object {
        fun startService(context: Context, message: String) {
            val startIntent = Intent(context, ForegroundService::class.java)
            startIntent.putExtra(name: "inputExtra", message)
            ContextCompat.startForegroundService(context, startIntent)
        }
        fun stopService(context: Context) {
            val stopIntent = Intent(context, ForegroundService::class.java)
            context.stopService(stopIntent)
        }
    }
    override fun onStartCommand(intent: Intent?, flags: Int, startId: Int): Int {
        //do heavy work on a background thread
        val input = intent?.getStringExtra(name: "inputExtra")
        createNotificationChannel()
        val notificationIntent = Intent(packageContext: this, AuthActivity::class.java)
        val pendingIntent = PendingIntent.getActivity(
            context: this,
            requestCode: 0, notificationIntent, flags: 0
        )
        val notification = NotificationCompat.Builder(context: this, CHANNEL_ID)
            .setContentTitle("Banco de Pruebas - Caudal")
            .setContentText(MensajeNotificacion)
            .setSmallIcon(R.drawable.ic_monitor)
            .setPriority(NotificationCompat.PRIORITY_MAX)
            .setContentIntent(pendingIntent)
            .build()
        startForeground(id: 1, notification)
    }
}

```

Nota. El servicio en *Foreground* es programado para mostrar una notificación persistente incluso cuando la aplicación no está en primer plano.

Interfaz de AR

Cuando se inicia la interfaz de AR, se deben enviar los datos de los sensores al módulo AR de Unity, para ello se accede a la base de datos de Firebase y al script Database de la librería de Unity, mediante el código mostrado en la Figura 80, es cual es implementado en el archivo “MainUnityActivity” que se encuentra en la dirección “NativeAndroidApp/app/src/main/java/com/unity/mynativeapp”.

Figura 80

Comunicación entre Unity y Android

```
mUnityPlayer.UnitySendMessage( s: "Database", s1: "DatosTemp",DatoT);
mUnityPlayer.UnitySendMessage( s: "Database", s1: "DatosPresion",DatoP);
mUnityPlayer.UnitySendMessage( s: "Database", s1: "DatosCaudal",DatoC);
mUnityPlayer.UnitySendMessage( s: "Database", s1: "DatosFrec",DatoF);
```

Con una serie de condicionales se designa el estado del sensor, como se muestra en la

Figura 81.

Figura 81

Asignación de estados de los sensores

```
public void estadoSensores(){
    if (fDatoC==0 && fDatoT==0 && fDatoF==0 && fDatoP==0){
        DatoE="OFF";
        DatoP="OFF";
        DatoT="OFF";
        DatoC="OFF";
        DatoF="OFF";
    }else{
        if(fDatoP>1){
            DatoP="ON";
        }else if(fDatoP<=1){
            DatoP="FAIL";
        }

        if(fDatoT>15 && fDatoT<=30){
            DatoT="ON";
        }else {
            DatoT="FAIL";
        }

        if(fDatoC>=0){
            DatoC="ON";
        }else{
            DatoC="FAIL";
        }

        if(fDatoF>=0 && fDatoF<=61){
            DatoF="ON";
        }else{
            DatoF="FAIL";
        }
    }
}
```

Capítulo IV

Pruebas y resultados

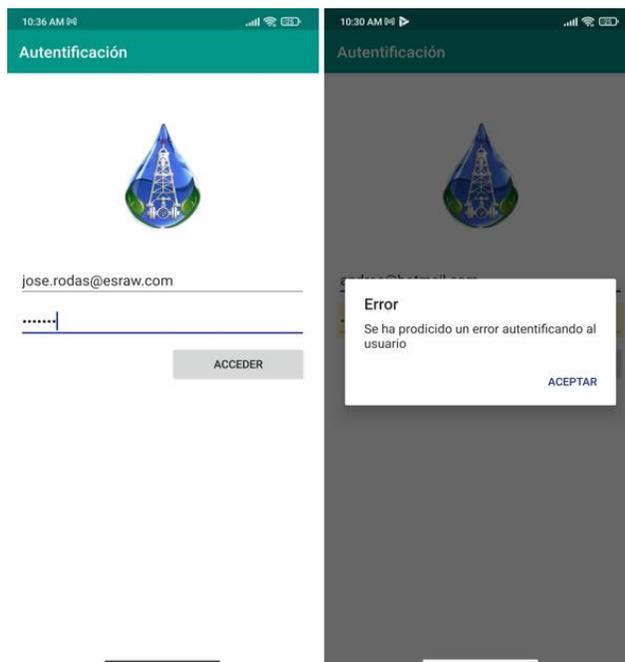
En este capítulo se presentará las pruebas y resultados del sistema IoT con realidad aumentada para establecer veracidad del monitoreo en tiempo real, reconocimiento de los estados de los sensores y la verificación de los informes generados; con el fin de validar la hipótesis del proyecto.

Pruebas de Autenticación de Usuario

En la Figura 82, se utilizó un usuario no registrado por la empresa en la base de datos, por ende, la aplicación muestra un mensaje de error en el que le indica que tiene problemas en la autenticación, mientras que en la Figura 83, se ingresa un usuario registrado y accede a la pantalla de Menú de la aplicación.

Figura 82

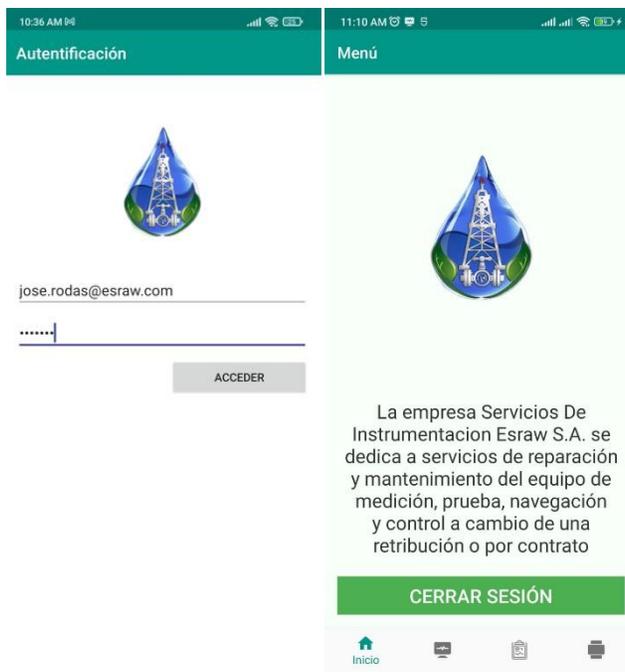
Ingreso de usuario erróneo



Nota. Si se ingresa un correo o contraseña incorrecta se emitirá un mensaje de error de autenticación

Figura 83

Autenticación de Usuario exitosa



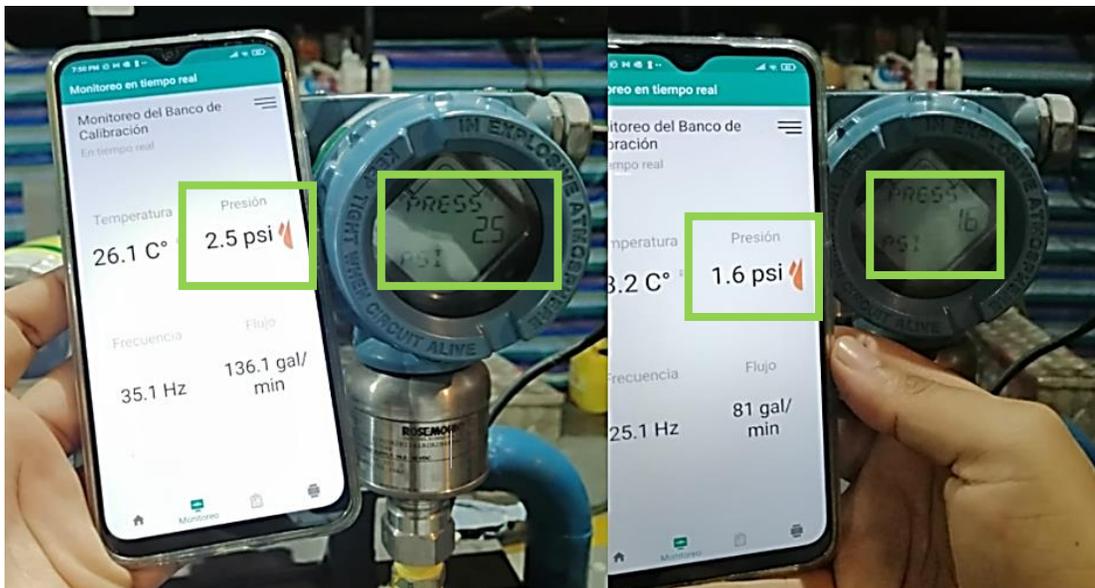
Nota. Si se ingresa correctamente, se abrirá la pantalla menú y se mostrará la barra de navegación

Pruebas de Monitoreo en Tiempo Real

Se comparará los datos de la aplicación y los marcados en el sensor. La primera magnitud a revisar es presión, en la Figura 84 se presentan los valores medidos de presión, la segunda magnitud a revisar es caudal; se puede observar los valores medidos de temperatura en la Figura 85, se pueden observar los valores medidos de temperatura en la Figura 86, en la Figura 87 se pueden observar los valores de la magnitud de frecuencia.

Figura 84

Comparación de valor de magnitud de presión



Nota. Se están comparando los valores de presión medidos por el sensor en el banco de pruebas con los valores mostrados en la aplicación Android

Como se puede observar en la Tabla 4, el valor medido en el sensor de presión y el valor de la aplicación android son los mismos, obteniendo un error absoluto entre 0 y -0.1 psi aproximadamente.

Tabla 4

Comparación de mediciones de Presión

# Prueba	Sensor Presión	App Android	Error absoluto
1	2.5	2.5	0.0
2	1.6	1.6	0.0
3	1.7	1.6	-0.1
4	1.6	1.5	-0.1
5	2.0	1.9	-0.1

# Prueba	Sensor Presión	App Android	Error absoluto
6	2.3	2.2	-0.1
7	2.2	2.2	0.0
8	2.2	2.2	0.0
9	2.4	2.4	0.0
10	2.4	2.3	-0.1

Nota. Valores expresados en psi

Según la Tabla 5, el valor medido en el sensor de caudal y el valor de la aplicación android son similares, con un error menor a 0.22 gal/min.

Figura 85

Comparación de valor de magnitud Caudal



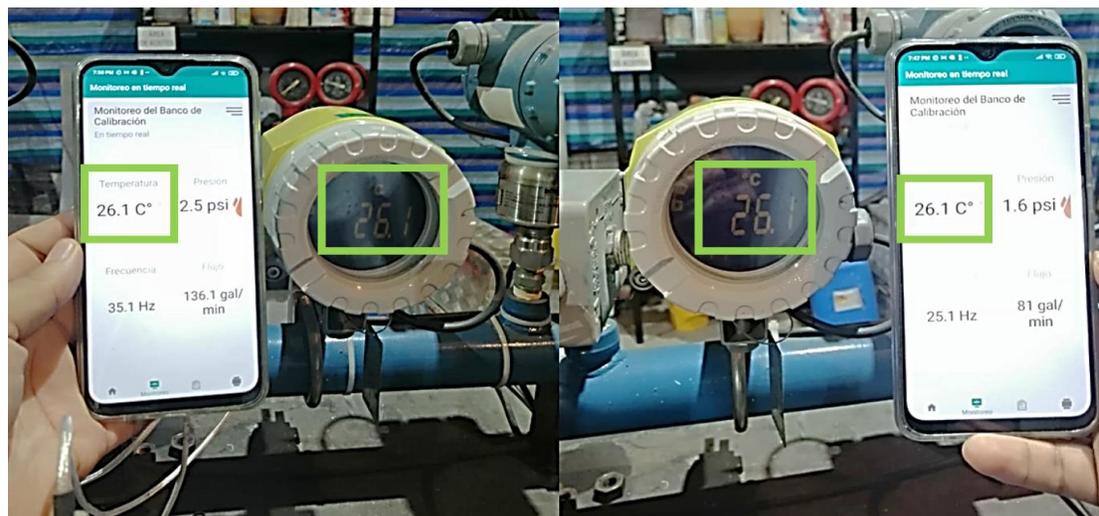
Nota. Se están comparando los valores de caudal medidos por el sensor en el banco de pruebas con los valores mostrados en la aplicación Android

Tabla 5*Comparación de mediciones de Caudal*

# Prueba	Sensor Caudal	App Android	Error absoluto
1	135.03	136.1	0.07
2	81.182	81.0	-0.182
3	90.123	90.0	-0.123
4	98.145	98.0	-0.145
5	100.45	100.3	-0.15
6	112.18	112.2	0.02
7	120.74	120.8	0.06
8	129.78	130.0	0.22
9	100.45	100.5	0.05
10	90.647	90.6	-0.047

Nota. Valores expresados en gal/min

De acuerdo a la Tabla 6, el valor medido en el sensor de temperatura y el valor de la aplicación android son similares, con un error aproximadamente entre 0 a -0.1 °C.

Figura 86*Comparación de valor de magnitud Temperatura*

Nota. Se están comparando los valores de temperatura medidos por el sensor en el banco de pruebas con los valores mostrados en la aplicación Android

Tabla 6

Comparación de mediciones de Temperatura

# Prueba	Sensor Temperatura	App Android	Error
1	26.1	26.1	0.0
2	26.1	26.1	0.0
3	26.0	26.0	0.0
4	25.9	25.8	-0.1
5	25.9	25.9	0.0
6	25.8	25.8	0.0
7	25.6	25.5	-0.1
8	26.1	26.0	-0.1
9	26.1	26.0	-0.1
10	26.1	26.0	-0.1

Nota. Valores expresados en °C

Según la Tabla 7, el valor medido en el sensor de frecuencia y el valor de la aplicación android son similares, con un error menor a 0,1 Hz.

Figura 87

Comparación de valor de magnitud Frecuencia



Nota. Se están comparando los valores de frecuencia medidos por el sensor en el banco de pruebas con los valores mostrados en la aplicación Android

Tabla 7

Comparación de mediciones de Frecuencia

# Prueba	Sensor Frecuencia	App Android	Error
1	25.05	25.1	0.05
2	35.05	35.1	0.05
3	18.00	18.1	0.1
4	23.05	23.1	0.05
5	25.15	25.2	0.05
6	30.45	30.5	0.05
7	32.80	32.9	0.1
8	30.10	30.2	0.1
9	27.40	27.5	0.1
10	23.50	23.6	0.1

Nota. Valores expresados en Hz

Pruebas de Informes Generados

Se analizará 2 registros de datos primarios obtenidos durante el procedimiento de validación en la empresa ESRAW S.A.

Los datos primarios son los valores de los sensores que son registrados durante una calibración en el banco de pruebas. En la hoja de datos primarios, se miden 5 puntos debido a que la empresa ESRAW S.A. se basa en las guías del DKD (Deutscher Kalibrierdienst – Servicio Alemán de Calibración) para la realización de sus calibraciones. Estos 5 puntos son designados en base al rango del equipo a medir, se realiza 1 ciclo de medición (ascenso y descenso).

Para este caso el equipo a calibrar es una turbina, por ende, la magnitud principal será el flujo de líquido(caudal) para diferentes frecuencias, las magnitudes de presión y temperatura son complementarias para el cálculo de la incertidumbre que genera el laboratorio.

En la Figura 88, se presenta el primer registro a analizar, es un registro de datos primarios para banco de pruebas de caudal realizado el día 3 de febrero de 2022, por el técnico José Rodas, el equipo verificado es una turbina de marca STAUFF.

En la Figura 89, se muestra el registro de datos del banco de calibración el día 03 de febrero de 2022, en la Tabla 8, se realiza la comparación de datos en frecuencia y se determina el error existente entre las dos mediciones.

Tabla 8

Comparación de datos de Registro en Frecuencia – Prueba 1

Ítems	Ciclo 1 – Ascenso			Ciclo 1 - Descenso		
	Técnico J.R.	Registro IoT	Error Absoluto	Técnico W.H.	Registro IoT	Error Absoluto
1	0,00	0	0	0	0	0
2	20,01	20,1	0,09	20,01	20,1	0,09
3	30,00	30,1	0,1	30	30,1	0,1

Ítems	Ciclo 1 - Ascenso			Ciclo 1 - Ascenso		
	Técnico J.R.	Registro IoT	Error Absoluto	Técnico W.H.	Registro IoT	Error Absoluto
4	40,03	40,1	0,07	40,03	40,1	0,07
5	50,07	51	0,93	50,07	50,9	0,83

Nota. Valores expresados en Hz

En la Tabla 9, se realiza la comparación de datos en presión y se calcula la diferencia entre las dos mediciones.

Tabla 9

Comparación de datos de Registro en Presión – Prueba 1

Ítems	Ciclo 1 - Ascenso			Ciclo 1 - Descenso		
	Técnico J.R.	Registro IoT	Error Absoluto	Técnico W.H.	Registro IoT	Error Absoluto
1	2,5	2,5	0	2,5	2,5	0
2	3	3	0	3	3	0
3	3	3	0	3	3	0
4	3	3	0	3	3	0
5	3	3	0	3	3	0

Nota. Valores expresados en psi

En la Tabla 10, se realiza la comparación de datos en temperatura y se determina la diferencia entre las dos mediciones.

Tabla 10*Comparación de datos de Registro en Temperatura – Prueba 1*

Ítems	Ciclo 1 - Ascenso			Ciclo 1 - Descenso		
	Técnico J.R.	Registro IoT	Error Absoluto	Técnico W.H.	Registro IoT	Error Absoluto
1	24	23,9	-0,1	23,9	23,9	0
2	23,9	23,9	0	23,9	23,9	0
3	23,9	23,9	0	23,9	23,9	0
4	23,9	23,9	0	23,9	23,9	0
5	23,9	23,9	0	23,9	23,9	0

Nota. Valores expresados en °C

En la Tabla 11, se analiza la diferencia de datos en caudal determinando el error entre las dos mediciones.

Tabla 11*Comparación de datos de Registro en Caudal – Prueba 1*

Ítems	Ciclo 1 - Ascenso			Ciclo 1 - Descenso		
	Técnico J.R.	Registro IoT	Error Absoluto	Técnico W.H.	Registro IoT	Error Absoluto
1	0	0	0	0	0	0
2	97,12	99,25	2,13	98,52	99,54	1,02
3	149,782	150,66	0,878	149,821	151,15	1,329
4	199,113	201,56	2,447	200,703	201,45	0,747
5	248,238	250,8	2,562	249,121	251,14	2,019

Nota. Valores expresados en gal/min

Figura 88

Registro de datos primarios de ESRAW S.A. para prueba 1

DATOS PRIMARIOS PARA BANCO DE PRUEBAS DE CAUDAL		Identificación: FTM-033 Revisión: 002 Aprobación: 2021-07-16 Página: 1 de 1					
SERVICIOS DE INSTRUMENTACIÓN ESRAW S.A.							
1.- DATOS DEL SOLICITANTE:							
CLIENTE: <u>ESRAW S.A</u>	TÉCNICOS: <u>JOSE RODAS</u>	FECHA RECEPCIÓN: _____					
RIG: _____	LUGAR CALIB: <u>TALLER</u>	DESCRIPCIÓN ADE: <u>VALIDACIÓN</u>					
OPERADORA: _____	UBICACIÓN IBP: <u>BODEGA</u>	PROCEDIMIENTO: <u>PTM-007</u>					
DIRECCIÓN: _____	LOCACIÓN: _____	CÓDIGO ÚNICO: <u>ASC-0277-A1E</u>					
2.- DATOS TÉCNICOS DE EQUIPOS:							
La evaluación del factor de riesgo en la trazabilidad con la relación de exactitud (TAR), de recomendaciones informacionales el patrón de trabajo debe ser 4 veces más exacto que el IBP.							
EQUIPO PATRÓN PRESIÓN		INSTRUMENTO BAJO PRUEBA					
Fabricante: <u>ROSEMOUNT CORP</u>	_____	Fabricante: <u>STAUFF</u>	_____				
Modelo: <u>VRS 100</u>	_____	Modelo: <u>SP5-BD500</u>	_____				
Rango: <u>300 gal/min</u>	_____	No. Serie: <u>SNP2513100-A</u>	_____				
Exactitud: <u>0,5 %</u>	_____	Resolución: _____	_____				
Resolución: _____	_____	Exactitud: <u>2 %</u>	_____				
No. Serie: <u>E3-1841498</u>	_____	Rango Presión: _____	_____				
Trazabilidad: <u>L.M.E.L 20215 DEF</u>	_____	Rango Eléctrico: _____	_____				
Incertidumbre: <u>0,9 gal/min</u>	_____	Coefficiente Temp: _____	_____				
Fecha Calibra: <u>2021-09-15</u>	_____	Datos Extras: _____	_____				
3.- DATOS AMBIENTALES & OTROS:							
Temp. In: <u>28,4</u>	Temp. Out: <u>27,8</u>	Tipo Fluido: <u>AGUA</u>	Inicio Calibración: <u>2022-02-03</u>				
R.H. In: <u>52,2</u>	R.H. Out: <u>51,3</u>	Altitud: <u>253</u>	Final Calibración: <u>2022-02-03</u>				
P. Al. In: <u>979,5</u>	P. Al. Out: <u>979,5</u>	Δ Altura: _____	Próxima Calibración: _____				
4.- PROCEDIMIENTO PARA TOMA DE DATOS:							
La calibración se realiza por medición del banco de pruebas para turbinas, aplicando el procedimiento antes mencionado, se debe cumplir los siguientes detalles: Puntos de calibración según su exactitud: (2) dos ciclos (Exact: > 0.1 % - 5 Puntos); 1 ciclo: (1 Ascenso & 1 Descenso).							
5.- PROCESO Y TOMA DE DATOS:							
Lecturas No.	Equipo Patrón - Esraw Frecuencia Hz	I.B.P. 1er Ciclo - ASCENSO			I.B.P. 1er Ciclo - DESCENSO		
		Presión psi	Temperatura °C	Caudal (gal/min)	Presión psi	Temperatura °C	Caudal (gal/min)
		1	0	2,5	24,0	0,00	2,5
2	20,01	3,0	23,9	97,120	3,0	23,9	98,520
3	30,00	3,0	23,9	149,782	3,0	23,9	149,821
4	40,03	3,0	23,9	199,113	3,0	23,9	200,703
5	50,07	3,0	23,9	248,238	3,0	23,9	248,121
6							
7							
8							
9							
10							
11							

Figura 89

Registro de Datos del Banco de Calibración del 03/02/2022



En la Figura 90, se presenta el segundo registro a analizar, es un registro de datos primarios para banco de pruebas de caudal realizado el día jueves 4 de febrero de 2022 por el técnico Washington Hidalgo, el equipo verificado es una turbina de marca STAUFF.

En la Figura 91, se muestra el registro de datos del banco de calibración el día 04 de febrero de 2022.

En la Tabla 12, se realiza la comparación de datos en frecuencia y se determina el error existente entre las dos mediciones.

Tabla 12*Comparación de datos de Registro en Frecuencia – Prueba 2*

Ítems	Ciclo 1 - Ascenso			Ciclo 1 - Descenso		
	Técnico W.H.	Registro IoT	Error	Técnico W.H.	Registro IoT	Error
1	0	0	0	0	0	0
2	15,02	15,1	0,08	15,02	15,1	0,08
3	20,07	20,1	0,03	20,07	20,1	0,03
4	25,03	25,1	0,07	25,03	25,1	0,07
5	30,05	30,1	0,05	30,05	30,1	0,05

Nota. Valores expresados en Hz

En la Tabla 13, se realiza la comparación de datos en presión y se calcula la diferencia entre las dos mediciones.

Tabla 13*Comparación de datos de Registro en Presión – Prueba 2*

Ítems	Ciclo 1 - Ascenso			Ciclo 1 - Descenso		
	Técnico W.H.	Registro IoT	Error	Técnico W.H.	Registro IoT	Error
1	2,5	2,5	0	2,5	2,5	0
2	3	3	0	3	3	0
3	3	3	0	3	3	0
4	3	3	0	3	3	0
5	3	3	0	3	3	0

Nota. Valores expresados en psi

En la Tabla 14, se realiza la comparación de datos en temperatura y se determina la diferencia entre las dos mediciones.

Tabla 14*Comparación de datos de Registro en Temperatura - Prueba 2*

Ítems	Ciclo 1 - Ascenso			Ciclo 1 - Descenso		
	Técnico W.H.	Registro IoT	Error	Técnico W.H.	Registro IoT	Error
1	24	23,9	-0,1	23,9	23,9	0
2	23,9	23,9	0	23,9	23,9	0
3	23,9	23,9	0	23,9	23,9	0
4	23,9	23,9	0	23,9	23,9	0
5	23,9	23,9	0	23,9	23,9	0

Nota. Valores expresados en °C

En la Tabla 15, se analiza la diferencia de datos en caudal determinando el error entre las dos mediciones.

Tabla 15*Comparación de datos de Registro en Caudal - Prueba 2*

Ítems	Ciclo 1 - Ascenso			Ciclo 1 - Descenso		
	Técnico W.H.	Registro IoT	Error	Técnico W.H.	Registro IoT	Error
1	0	0	0	0	0	0
2	74,928	75,21	0,282	75,138	75,29	0,152
3	100,921	101,2	0,279	101,098	101,34	0,242
4	126,107	126,3	0,193	126,13	126,41	0,28
5	151,02	151,23	0,21	151,2	151,4	0,2

Nota. Valores expresados en gal/min

Figura 90

Registro de datos primarios de ESRAW S.A. para prueba 2

DATOS PRIMARIOS PARA BANCO DE PRUEBAS DE CAUDAL
SERVICIOS DE INSTRUMENTACIÓN ESRAW S.A.

Identificación: FTM-033
 Revisión: 002
 Aprobación: 2021-07-16
 Página: 1 de 1

1.- DATOS DEL SOLICITANTE:

CUENTE: ESRAW SA TÉCNICOS: W. Hidalgo FECHA RECEPCIÓN: _____
 RIG: _____ LUGAR CALIB: Taller DESCRIPCIÓN ADC.: Validación
 OPERADORA: _____ UBICACIÓN IBP: Bodega PROCEDIMIENTO: PTM-007
 DIRECCIÓN: _____ LOCALCIÓN: _____ CÓDIGO ÚNICO: ASC-0280-ATE-ESRAW

2.- DATOS TÉCNICOS DE EQUIPOS:

La evaluación del factor de riesgo en la trazabilidad con la relación de exactitud (TAR), de recomendaciones internacionales el patrón de trabajo debe ser 4 veces más exacto que el IBP.

EQUIPO PATRÓN PRESIÓN	INSTRUMENTO BAJO PRUEBA
Fabricante: <u>ROSEMOUNT CORP.</u>	Fabricante: <u>STAUFF</u>
Modelo: <u>VRS-100</u>	Modelo: <u>SPG-B03000</u>
Rango: <u>300 gall/min</u>	No. Serie: <u>SNP2813009-A</u>
Exactitud: <u>0,5%</u>	Resolución: _____
Resolución: _____	Exactitud: <u>2%</u>
No. Serie: <u>E3-1841498</u>	Rango Presión: _____
Trazabilidad: <u>LMEL2021SDER</u>	Rango Eléctrico: _____
Incertidumbre: <u>0,9 gall/min</u>	Coefficiente Temp: _____
Fecha Calibra: <u>2021-09-15</u>	Datos Extras: _____

3.- DATOS AMBIENTALES & OTROS:

Temp. In: 27,2 Temp. Out: 28,0 Tipo Fluido: Agua Inicio Calibración: 2022-02-04
 R.H. In: 48,7 R.H. Out: 50,2 Altitud: 258 Final Calibración: 2022-02-04
 P. Al. In: 980,1 P. Al. Out: 980,1 Δ Altura: _____ Próxima Calibración: _____

4.- PROCEDIMIENTO PARA TOMA DE DATOS:

La calibración se realiza por medición del banco de pruebas para turbinas, aplicando el procedimiento antes mencionado, se debe cumplir los siguientes detalles:
 Puntos de calibración según su exactitud: (2) dos ciclos (Exactitud > 0.1% - 5 Puntos); 1 ciclo: (1 Ascenso & 1 Descenso).

5.- PROCESO Y TOMA DE DATOS:

Lecturas No.	Equipo Patrón - Eswaw Eq. Frecuencia Hz	L.B.P. 1er Ciclo - ASCENSO			L.B.P. 1er Ciclo - DESCENSO		
		Presión psi	Temperatura °C	Caudal (gall/min)	Presión psi	Temperatura °C	Caudal (gall/min)
1	0	2,5	24,0	0,00	2,5	23,9	0,00
2	15,02	3,0	29,0	74,988	3,0	23,9	75,138
3	20,07	3,0	23,9	100,921	3,0	23,9	101,098
4	25,03	3,0	23,9	126,107	3,0	23,9	126,130
5	30,05	3,0	23,9	151,020	3,0	23,9	151,200
6	—						
7	—						
8	—						
9	—						
10	—						
11	—						

FIRMA DE RESPONSABILIDAD

SERVICIO DE INSTRUMENTACIÓN ESRAW S.A.
 RUC: 2290324893001
 LABORATORIO DE CALIBRACION

6.- OBSERVACIONES:

Copyright Eswaw © 2021 Dirección: Ángel de Ugarte N13-08 y Antonio Calera El Coca Orellana Ecuador (+593) 994497269 / (+593) 997938203 www.esraw.com

Figura 91

Registro de Datos del Banco de Calibración del 04/02/2022



Prueba de generación de Notificaciones

En la Figura 92, se presenta la interfaz con la notificación de inicio de calibración dado que la aplicación le indica que se está calibrando, siempre y cuando se pase de la interfaz de autenticación.

Figura 92*Notificación Inicial de encendido*

En la Figura 93, se presenta la interfaz de inicio con la notificación de apagado del banco de calibración.

Figura 93*Notificación de Apagado del Banco de Calibración*

En la Figura 94, se presenta la interfaz de inicio con la notificación de Estado de Alerta de sensores y además también se muestra la notificación de alerta cuando esta inicia la herramienta AR.

Figura 94

Notificación de estado de alerta



En la Tabla 16, se muestra si las notificaciones se cumplen o no, realizadas en 10 pruebas.

Tabla 16*Prueba de notificaciones*

# Prueba	Notificación de Inicio	Notificación de Apagado	Notificación de Apagado inesperado	Notificación de alerta de sensor
1	Cumple	Cumple	Cumple	Cumple
2	Cumple	Cumple	No cumple	Cumple
3	Cumple	Cumple	Cumple	Cumple
4	Cumple	Cumple	Cumple	Cumple
5	Cumple	Cumple	No cumple	Cumple

Prueba de reconocimiento del estado del sensor

En la Figura 95, se presenta la herramienta de Realidad Aumentada detectando el marcador de frecuencia y mostrando el aviso de sensor de frecuencia en estado de apagado (en la Figura además se indica en un cuadro que el led del variador de frecuencia está apagado).

Figura 95

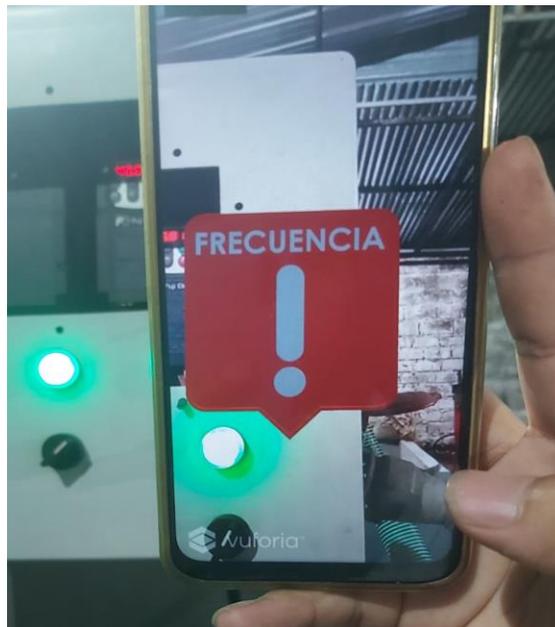
Herramienta AR detectando el sensor en estado de apagado



En la Figura 96, se presenta la herramienta de Realidad Aumentada detectando el marcador de frecuencia y mostrando el aviso de sensor de frecuencia en estado de apagado (en la Figura además se indica en un cuadrado que el led del variador de frecuencia esta apagado).

Figura 96

Herramienta AR detectando el sensor en estado de alerta



En la Figura 97, se presenta la herramienta de Realidad Aumentada detectando el marcador de frecuencia y mostrando el aviso de sensor de frecuencia en estado de apagado (en la Figura además se indica en un cuadrado que el led del variador de frecuencia esta apagado).

Figura 97

Herramienta AR detectando el sensor en estado de encendido



Validación de la hipótesis

Para la comprobación de la hipótesis, se utilizaron los resultados de las Tablas 11 y 15 los cuales se muestran en la Tabla 17, correspondientes a las mediciones del sensor de caudal dado que es la magnitud principal a analizar en un banco de pruebas de medidores de caudal y que es la magnitud que presentó mayor variación entre las medidas de los datos primarios y los registros de la aplicación Android, se les aplicará el método de estadístico T-Student.

La hipótesis planteada en el proyecto es:

¿La implementación del Sistema Internet de las Cosas con realidad aumentada permitirá la digitalización, monitoreo y generación de eventos de un banco de calibración de medidores de caudal de la empresa ESRW S.A.?

Las variables de la hipótesis son:

- Variable independiente: Sistema de Internet de las Cosas con realidad aumentada.

- Variable dependiente: Digitalización, monitoreo y generación de eventos de un banco de calibración de medidores de caudal.

Con esta hipótesis se planteará una hipótesis nula y una hipótesis alternativa:

- **Hipótesis nula (H_0):** La implementación del Sistema Internet de las Cosas con realidad aumentada permitirá la digitalización, monitoreo y generación de eventos de un banco de calibración de medidores de caudal de la empresa ESRAW S.A.
- **Hipótesis alternativa (H_1):** La implementación del Sistema Internet de las Cosas con realidad aumentada no permitirá la digitalización, monitoreo y generación de eventos de un banco de calibración de medidores de caudal de la empresa ESRAW S.A.

Tabla 17

Mediciones del sensor de caudal

Pruebas	Datos Primarios	App IoT	Técnico
1	0	0	J. Rodas
2	97,12	99,25	J. Rodas
3	149,782	150,66	J. Rodas
4	199,113	201,56	J. Rodas
5	248,238	250,8	J. Rodas
6	0	0	J. Rodas
7	98,52	99,54	J. Rodas
8	149,821	151,15	J. Rodas
9	200,703	201,45	J. Rodas
10	249,121	251,14	J. Rodas
11	0	0	W. Hidalgo
12	74,928	75,21	W. Hidalgo

13	100,921	101,2	W. Hidalgo
14	126,107	126,3	W. Hidalgo
15	151,02	151,23	W. Hidalgo
16	0	0	W. Hidalgo
17	75,138	75,29	W. Hidalgo
18	101,098	101,34	W. Hidalgo
19	126,13	126,41	W. Hidalgo
20	0	0	J. Rodas

Nota. Valores expresados en gal/min

Se utilizan las condiciones presentadas en la Tabla 18, para validar las hipótesis nula o alternativa.

Tabla 18

Condiciones presentadas para realizar la validación por el método de T-Student

Hipótesis	Condición de validación
$H_0: \mu \geq \mu_0$	$t \leq -t_\alpha$: Se rechaza H_0
$H_1: \mu < \mu_0$	
Cola a la izquierda	Caso contrario, se rechaza
$H_0: \mu \leq \mu_0$	$t \geq t_\alpha$: Se rechaza H_0
$H_1: \mu > \mu_0$	
Cola a la derecha	Caso contrario, se acepta
$H_0: \mu = \mu_0$	$t \leq -t_{\alpha/2}$ o $t \geq t_{\alpha/2}$: Se rechaza H_0
$H_1: \mu \neq \mu_0$	
Dos colas	Caso contrario, se acepta

Donde, t es el valor de la distribución calculado y t_α es el valor de distribución Tabla 19.

Tabla 19

Tabla de distribución *t* de Student

	$\alpha/2$					
	0,0005	0,001	0,005	0,01	0,025	0,05
36	3,582	3,333	2,719	2,434	2,028	1,688
37	3,574	3,326	2,715	2,431	2,026	1,687
38	3,566	3,319	2,712	2,429	2,024	1,686
39	3,558	3,313	2,708	2,426	2,023	1,685
40	3,551	3,307	2,704	2,423	2,021	1,684
41	3,544	3,301	2,701	2,421	2,020	1,683
42	3,538	3,296	2,698	2,418	2,018	1,682
43	3,532	3,291	2,695	2,416	2,017	1,681
44	3,526	3,286	2,692	2,414	2,015	1,680
45	3,520	3,281	2,690	2,412	2,014	1,679
46	3,515	3,277	2,687	2,410	2,013	1,679
47	3,510	3,273	2,685	2,408	2,012	1,678
48	3,505	3,269	2,682	2,407	2,011	1,677
49	3,500	3,265	2,680	2,405	2,010	1,677
50	3,496	3,261	2,678	2,403	2,009	1,676
51	3,492	3,258	2,676	2,402	2,008	1,675
52	3,488	3,255	2,674	2,400	2,007	1,675
53	3,484	3,251	2,672	2,399	2,006	1,674
54	3,480	3,248	2,670	2,397	2,005	1,674
55	3,476	3,245	2,668	2,396	2,004	1,673
56	3,473	3,242	2,667	2,395	2,003	1,673
57	3,470	3,239	2,665	2,394	2,002	1,672
58	3,466	3,237	2,663	2,392	2,002	1,672
59	3,463	3,234	2,662	2,391	2,001	1,671
60	3,460	3,232	2,660	2,390	2,000	1,671
120	3,373	3,160	2,617	2,358	1,980	1,658
∞	3,300	3,098	2,581	2,330	1,962	1,646
α	0,001	0,002	0,01	0,02	0,05	0,1

Nota. Tomado de (UPV, 2018)

Con un porcentaje de aceptabilidad esperado de:

$$\mu_0 = 95\%$$

Cálculo del estadístico t:

- Nivel de significancia (α): 0.05
- Media de mediciones de datos primarios (\bar{x}_1): 114.9 gal/min
- Media de mediciones App IoT (\bar{x}_2): 115.7 gal/min
- Desviación estándar de mediciones de datos primarios (s_1): 77.0
- Desviación estándar de mediciones App IoT (s_2): 77.6
- Tamaño de mediciones de datos primarios (n_1): 20
- Tamaño de mediciones de App IoT (n_2): 20

Luego, calcula el estadístico t utilizando la fórmula de la ecuación (1):

$$t = \frac{(\tilde{x}_1 - \tilde{x}_2)}{\sqrt{\left(\frac{s_1^2}{n_1}\right) + \left(\frac{s_2^2}{n_2}\right)}} \quad (1)$$

$$t = \frac{(114.9 - 115.7)}{\sqrt{\left(\frac{77.0^2}{20}\right) + \left(\frac{77.6^2}{20}\right)}}$$

$$t = -0.03$$

Se determina los grados de libertad con la ecuación (2):

$$gl = n_1 + n_2 - 2 \quad (2)$$

$$gl = 20 + 20 - 2$$

$$gl = 38$$

Utilizando la tabla de distribución t de Student de dos colas para encontrar el valor crítico correspondiente a los grados de libertad y nivel de significancia deseado obtenemos el siguiente valor de t.

Valor t_α de Tabla 19 es 1.686 y el valor t calculado es -0.03, para efectos de comparación se utiliza el valor absoluto $t=0.03$ por lo tanto

$$t \geq t_{\alpha/2}$$

$$0.03 \leq 1.686$$

Por lo tanto, aceptamos la hipótesis nula, la cual pronuncia que la implementación del Sistema Internet de las Cosas con realidad aumentada permitirá la digitalización, monitoreo y generación de eventos de un banco de calibración de medidores de caudal de la empresa ESRAW S.A.

En este caso, la hipótesis que se plantea respaldar es la nula, dado que la condición de aceptación es que los valores medidos por la aplicación IoT y los valores de los datos primarios no difieren, si se da el caso, se considera que la digitalización de datos se realizó correctamente.

Alcances

El sistema IoT desarrollado e implementado monitorea en tiempo real del banco de calibración, lo que es de gran ayuda para los usuarios que no se encuentren en el laboratorio de la empresa, a fin de tener en cuenta y supervisar las actividades que se encuentran realizando el técnico en el proceso de calibración.

La aplicación móvil es sencilla y factible, proporciona información necesaria al usuario. Su función es monitorear los datos en tiempo real y mediante realidad aumentada puede observar el estado de los sensores.

Limitaciones

El sistema es dependiente de conexión a internet, por el hecho que es un sistema de monitoreo en tiempo real.

Capítulo V

Conclusiones y Recomendaciones

Conclusiones

- El proyecto se finalizó con el diseño e implementación del sistema IoT con realidad aumentada para la empresa ESRAW S.A. ubicada en la ciudad de El Coca, en la provincia de Orellana, la comunicación es vía WiFi con acceso a internet, es independiente de un computador, mediante el análisis de resultados se concluyó que el sistema IoT monitoriza el banco de pruebas de caudal en tiempo real.
- Para el diseño de la aplicación móvil se partió utilizo lenguaje nativo Kotlin de Android para acceder a las funciones y permisos de los dispositivos Android, para poder activar los servicios en segundo plano mientras la aplicación no se esté usando y mantener así el monitoreo constante.
- La función de monitoreo utiliza la base de datos de Firebase de Google, la cual se encuentra constantemente recibiendo datos enviados por una placa UNO WIFI R3 ATmega328P ESP8266.
- Se implementó realidad aumentada importando una librería creada en el motor de videojuegos Unity, la cual accede a los datos de la base de datos mediante comunicación enviada desde la aplicación Android a un complemento de Unity creado con código C#, esta librería importada muestra el estado de los sensores del banco de pruebas de caudal con la utilización de la cámara del dispositivo android, enfocando a los marcadores designados para cada sensor.
- Las notificaciones y eventos se pueden observar una vez autenticado el usuario, además se muestra notificaciones flotantes de los eventos más

relevantes dado que la aplicación móvil posee un servicio en segundo plano codificado en lenguaje nativo de Android.

- La utilización de los recursos y servicios online de Google permitió que el sistema no necesite de pago de licencias y se mantenga libre de costos adicionales dado que se ocupa un mínimo de la capacidad que Google ofrece en su plan gratuito, además de que ser servicios muy seguros y fiables por la cantidad de servidores que poseen.
- Las lecturas mostradas en tiempo real de los sensores con las almacenadas en los registros son similares dado que se ocupan de la misma base de datos de Firebase, con relación a lo registrado por el técnico en las diferentes pruebas se puede observar que existe un error mínimo en el sensor de frecuencia de 1Hz, en el sensor de temperatura menor a 1°C, en el sensor de presión el error es casi nulo y en el sensor de caudal el error es menor a de 2,1 gal/min.
- La tecnología IoT proporciona información en tiempo real de los valores de los sensores del banco de pruebas de caudal, desde cualquier parte del mundo que posea conectividad a Internet.
- La implementación económica de este proyecto fue accesible, debido a que todos los softwares del sistema se ocuparon con licencia de acceso libre, el valor implementar los componentes electrónicos con una base elaborada por impresión 3D es de un valor menor a los 200 dólares.

Recomendaciones

- El sistema IoT debe tener buena conexión WiFi con conectividad a internet para el funcionamiento óptimo del mismo.
- Leer el manual de especificaciones de cada sensor para asegurar su correcto manejo y no sobrepasar la capacidad de medición.
- Para las conexiones a la placa de adquisición de datos, es importante conocer el modo de salida del sensor del banco de pruebas para evitar cortocircuitos en el sistema, dado que son salidas de baja corriente directa o en tipo Hart.
- Aplicar filtros tanto físicos como lógicos para evitar ruido en las mediciones, mejorará la calidad y la precisión de los datos recopilados. Esta estrategia de mitigación del ruido contribuirá de manera significativa a la fiabilidad de las mediciones y al éxito general del proyecto.
- Actualizar una vez cada seis meses la librería de Arduino FirebaseHttpClient dado que Firebase y Google actualizan sus direcciones 1 vez al año, y el fingerprint se modifica en la librería.
- Utilizar AR en aplicaciones industriales para optimizar los procesos al proporcionar información en tiempo real directamente a los operadores tiene un impacto positivo en la eficiencia y la toma de decisiones.
- El dispositivo tecnológico en el que se instale la aplicación se recomienda deba tener una cámara posterior de 8 MP, un sistema android con versión 5 en adelante y una capacidad de memoria libre 200 MB para garantizar una experiencia de usuario fluida y la plena funcionalidad de la aplicación.
- El ancho de banda de la red WiFi con internet debe tener de capacidad 1 MB/s para una actualización de información eficiente.

- La adopción de las tecnologías de este proyecto en aplicaciones industriales representa un paso audaz hacia la transformación digital y puede allanar el camino para una mayor excelencia operativa en un mundo empresarial en constante evolución.
- Facilitar y promocionar el manual de usuario previo a la utilización del sistema para fomentar el uso óptimo y mejor experiencia del usuario.

Bibliografía

Abud Figueroa, M. A. (2012). Modelo de Objetos de Aprendizaje co. *Revista Internacional de la Educación en Ingeniería*, 1-7.

Afnan Uda, M. N., Jambek, A. B., Hashim, U., Uda, M. N., & Bahrin, M. A. (2020). Development of Internet of Things (IOT) Based Electronic Reader for Medical Diagnostic System. *IOP Conference Series: Materials Science and Engineering* 743 012020.

Alonso, V., Dacal-Nieto, A., Barreto, L., António, A., & Rivero, E. (09 de Marzo de 2020). *Implicaciones de la industria 4.0 en la metrología de visión artificial: una descripción general*. Obtenido de ScienceDirect:
<https://www.sciencedirect.com/science/article/pii/S2351978919311072#cebibsec1>

Apple Inc. (2020). *ARKit: Augmented Reality for iOS*. Obtenido de
<https://developer.apple.com/augmented-reality/arkit/>

Apriani, D., Aan, M., & Saputra, W. (2022). Data Visualization Using Google Data Studio. *International Journal of Cyber and IT Service Management (JCITSM)*, 2(1), 11-19.

Ardito, L., Coppola, R., Malnati, G., & Torchiano, M. (2020). Effectiveness of Kotlin vs. Java in Android App Development Tasks. *Information and Software Technology*, 106374.

- Basagain, X., Olabe, M., Espinosa, K., & Roueche, C. (2014). Realidad Aumentada en la Educación: una tecnología emergente. *ESIGELEC*, 1.
- Borgia, E. (2014). The Internet of Things vision: Key Features, Applications and Open Issues. *Computer Communications*, 54, 1-31.
- Chen, Y., Wang, Q., Chen, H., Song, X., Tang, H., & Tian, M. (2019). An overview of augmented reality technology. *Journal of Physics: Conference Series, Volume 1237, Issue 2*.
- Daponte, P., De Vito, L., Picariello, F., & Riccio, M. (2014). State of the art and future developments of the Augmented Reality for measurement applications. *Measurement*, 57, 53-70.
- Desai, O., Desai, S., DE, M., & Tikhe, K. S. (2020). Realidad aumentada: una nueva forma de ver el mundo. *REVISTA INTERNACIONAL DE INVESTIGACIÓN Y TECNOLOGÍA EN INGENIERÍA (IJERT)*, Volumen 09, Número 03.
- Dey, A., Billinghamurst, M., Lindeman, R. W., & Swan, J. E. (2018). A systematic review of 10 years of augmented reality usability studies: 2005 to 2014. *Frontiers in robotics and AI*, 5, 37.
- Donayre Monteza, F. (Junio de 2017). *Análisis y diseño de un software para control y monitoreo de variables del proceso de crianza de langostino*. Obtenido de Repositorio Institucional de la Universidad de Piura: <https://pirhua.udep.edu.pe/handle/11042/3067>
- Durnev, A. (11 de 05 de 2017). *Essentials of Augmented Reality Software Development Under Android Platform*. Obtenido de <https://core.ac.uk/download/pdf/237084886.pdf>
- Ekanayake, L. J., Ihalage, D., & Abyesundara, S. P. (2021). Performance Evaluation of Google Spreadsheet over RDBMS through Cloud Scripting Algorithms. *International Conference*

on Computer Communication and Informatics (ICCCI). Obtenido de
doi:10.1109/iccci50826

Endress Hauser. (2020). *iTEMP TMT162*. Obtenido de Transmisor de temperatura de campo:
<https://www.lasc.endress.com/es/instrumentacion-campo/medicion-temperatura/transmisor-de-campo>

Endress Hauser. (2020). *Proline Promag 10W*. Obtenido de Caudalímetro electromagnético:
<https://www.lasc.endress.com/es/instrumentacion-campo/medicion-caudal/caudalimetro-electromagnetico>

ESRAW. (2018). *Reseña de la empresa Esraw S.A.* Obtenido de <https://www.esraw.com>

Fernández, Á. (31 de Octubre de 2013). *Playstation*. Obtenido de
<https://blog.es.playstation.com/2013/10/31/wonderbook-el-libro-de-las-pociones-llega-a-ps3-el-mes-que-viene/>

Firestore Documentation. (18 de Diciembre de 2020). *Firestore Realtime Database*. Obtenido de
Firestore: <https://firebase.google.com/docs/database?hl=es-419>

Fonović, D., Sirotić, Z., Tanković, N., & Sovilj, S. (2021). Low-Power Wireless IoT System for
Indoor Environment Real-Time Monitoring and Alerting. *44th International Convention on
Information, Communication and Electronic Technology (MIPRO)*, 1654-1658.

Google Developers. (2020). *Firestore*. Obtenido de Google Developers:
<https://developers.google.com/learn/topics/firebase>

Gubbi, J., Buyya, R., Marusic, S., & Palaniswami, M. (2013). Internet of Things (IoT): A vision,
architectural elements, and future directions. *Future Generation Computer Systems* 29,
1645-1660.

- Guinard, D., & Trifa, V. (2009). *Towards the Web of Things: Web Mashups for Embedded Devices*. Obtenido de https://www.researchgate.net/publication/319393385_Towards_the_Web_of_Things_Web_Mashups_for_Embedded_Devices
- Iglesias, M. (2022). *TOLERANCIAS GEOMETRICAS GD&T GEOMETRIC DIMENSIONING AND TOLERANCING : basado en asme y14. 5-2018*. Buenos Aires: edUTecNe.
- Kalachev, O. (2021). *Online ArUco markers generator*. Obtenido de <https://chev.me/arucogen/>
- Klein, G., & Murray, D. (2007). Parallel Tracking and Mapping for Small AR Workspaces. *2007 6th IEEE and ACM International Symposium on Mixed and Augmented Reality*, 225-234.
- Madakam, S., Ramaswamy, R., & Tripathi, S. (2015). Internet of Things (IoT): A Literature Review. *Journal of Computer and Communications*, 3, 164-173.
- Mishra, D., Buyya, R., Mohapatra, P., & Patnaik, S. (2019). *Intelligent and Cloud Computing. Smart Innovation, Systems and Technologies*. Singapur: Springer Singapur.
- Moroney, L. (2017). *The Definitive Guide to Firebase: Build Android Apps on Google's Mobile Platform*. Seattle: Springer Science.
- Mourtzis, D., Vlachou, E., Zogopoulos, V., & Fotini, X. (2017). *Programación integrada de producción y mantenimiento a través del monitoreo de máquinas y la realidad aumentada: un enfoque de la Industria 4.0*. Obtenido de HAL-Inria: <https://hal.inria.fr/hal-01666188/document>
- Mucchetti, M. (2020). *BigQuery for Data Warehousing*. Santa Mónica, CA, USA.: Apress.
- Naik, P. G., & Oza, K. S. (2021). *Migrating from Java to Kotlin*. Chandigarh: International Institute of Organized Research (I2OR).

- Nizam, M. A., Alamri, A., Alotaibi, M., & Alharthi, R. (2020). *Performance Analysis of Google Firebase Cloud Computing Platform*. Obtenido de International Journal of Advanced Science and Technology, 29(9s), 116-123.: <https://doi.org/10.14257/ijast.2020.29.9s.11>
- Nunberg, G. (2012). *The Advent of the Internet*: 12th April, Courses.
- OILM. (1986). *Verification equipment for national metrology services*. OIML P 4 Edition.
- Oktaviyana, E. D., Ronaldo, D., & Arifin, M. (2019). APLIKASI BOOKING KOST BERBASIS ANDROID DI KOTA PALANGKA RAYA. *Jurnal Teknologi Informasi (JTI)*, Vol. 13 No. 2, 1-11.
- Proaño, C. (02 de 08 de 2017). *Crece la importancia de Internet de las Cosas en las empresas*. Obtenido de COMPUTERWORLD: <http://www.computerworld.es/tendencias/crece-la-importancia-de-internet-de-las-cosas-en-las-empresas>
- Putri, O. E. (Mayo de 2022). *Pengembangan Website Marketplace Branded Stuff Menjadi Aplikasi Mobile dengan Menggunakan Android Studio*. Obtenido de ResearchGate: https://www.researchgate.net/publication/360791300_Pengembangan_Website_Marketplace_Branded_Stuff_Menjadi_Aplikasi_Mobile_dengan_Menggunakan_Android_Studio
- Radenković, M., Tubić, A., Stojanović, V., Petrovic, N., & Nejkovic, V. (2021). *Adopting Google Apps Script for Development of Fitness Applications for Different Purposes*. Obtenido de https://www.researchgate.net/publication/347848942_Adopting_Google_Apps_Script_for_Development_of_Fitness_Applications_for_Different_Purposes
- RobotDyn. (2017). *UNO + WiFi R3 ATmega328P + ESP8266*. Obtenido de RobotDyn: <https://robotdyn.com/uno-wifi-r3-atmega328p-esp8266-32mb-flash-usb-ttl-ch340g-micro-usb.html>

- Rodríguez Sote, J. L., López Londoño, A., Vega Botero, C. A., & Flórez Hurtado, R. D. (2017). *Sistema de monitoreo y control remoto usando IOT para un regulador de presión*.
Obtenido de www.redalyc.org/: www.redalyc.org/pdf/849/84955649011.pdf
- Rosemount. (Abril de 2020). *Transmisor de presión Rosemount 3051*. Obtenido de Rosemount 3051: <https://www.emerson.com/documents/automation/hoja-de-datos-del-producto-rosemount-3051-transmisor-de-presi%F3n-es-es-73192.pdf>
- Salazar, J., & Silvestre, S. (2016). *Internet de las cosas*. TechPedia.
- Sanz, R. L., & Galán López, R. (2014). *Introducción a la movilidad: 4G/LTE y el desarrollo de aplicaciones Android*. Dextra Editorial. Obtenido de <https://elibro.net/es/lc/espe/titulos/43939>
- Siltanen, S. (2012). Theory and applications of marker-based augmented reality. *VTT SCIENCE* 3. Obtenido de <https://www.vttresearch.com/sites/default/files/pdf/science/2012/S3.pdf>
- UIT-T. (15 de 06 de 2012). *Overview of the Internet of things*. Obtenido de International Telecommunication Union: <https://www.itu.int/rec/T-REC-Y.2060-201206-l/en>
- Unity Technologies. (2019). *Unity as a Library integration example to iOS and Android*. Obtenido de GitHub: <https://github.com/Unity-Technologies/uaal-example>
- UPV. (2018). *Distribución t de Student*. Obtenido de Universidad Politécnica de Valencia: <https://estdg.blogs.upv.es/files/2018/01/Tabla-T-Student-2.pdf>
- Valqui, A., Csaburi, G., & Suaznabar, C. (Octubre de 2019). *Desafíos de la transformación digital para la metrología de América Latina y el Caribe*. Obtenido de IDB - Banco Interamericano de Desarrollo: https://publications.iadb.org/publications/spanish/document/Metrología_4.0_Desafíos_de_la_transformación_digital_para_la_metrología_de_América_Latina_y_el_Caribe_es.pdf

Vermesan, O., & Friess, P. (2016). *Internet of Things Connecting the Physical, Digital and Virtual Worlds*. Denmark: River Publishers.

Vuforia. (2023). *Vuforia Developer Portal*. Obtenido de <https://developer.vuforia.com/>

Wikitude GmbH. (2023). *Wikitude SDK: Augmented Reality for Developers*. Obtenido de <https://www.wikitude.com/developer/>

Wu, Y., Wang, P., & Sun, C. (2021). Design and implementation of OBU terminal for Vehicle Ad-hoc Network. *Journal Of Physics: Conference Series*, 1920(1), 012016. doi: 10.1088/1742-6596/1920/1/012016.

Yuen, S. C., Yaoyuneyong, G., & Johnson, E. (2011). Augmented Reality: An Overview and Five Directions for AR in Education. *Journal of Educational Technology Development and Exchange (JETDE)*, Vol. 4 : Iss. 1 , Article 11.

Zarate Nava, M. R., Mendoza González, C. F., & Aguilar Galicia. (3 de Diciembre de 2013). *Marcadores para la Realidad Aumentada para fines educativos*. Obtenido de ReCIBE. Revista electrónica de Computación, Informática, Biomédica y Electrónica, (3): <https://www.redalyc.org/pdf/5122/512251564004.pdf>

Zitelia. (2019). *La monitorización de procesos industriales y la transformación digital de la industria*. Obtenido de www.zitelia.com: <https://www.zitelia.com/monitorizacion-procesos-industriales-transformacion-digital-industria/>

Anexos