



ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA (ICAI)

MÁSTER EN INGENIERÍA DE TELECOMUNICACIÓN

Aplicación de la realidad aumentada a la minifábrica ICAI

Autor: Carlos Alonso Calleja

Director: José Antonio Rodríguez Mondéjar

Madrid

Julio 2019

Declaro, bajo mi responsabilidad, que el Proyecto presentado con el título

Aplicación de la realidad aumentada a la minifábrica ICAI

en la ETS de Ingeniería - ICAI de la Universidad Pontificia Comillas en el

curso académico 2018/19 es de mi autoría, original e inédito y

no ha sido presentado con anterioridad a otros efectos.

El Proyecto no es plagio de otro, ni total ni parcialmente y la información que ha sido

tomada de otros documentos está debidamente referenciada.

Fdo.: Carlos Alonso Calleja

Fecha: 12/07/2019

Autorizada la entrega del proyecto

EL DIRECTOR DEL PROYECTO

Fdo.: José Antonio Rodríguez Mondéjar

Fecha: 12/07/2019

AUTORIZACIÓN PARA LA DIGITALIZACIÓN, DEPÓSITO Y DIVULGACIÓN EN RED DE PROYECTOS FIN DE GRADO, FIN DE MÁSTER, TESIS O MEMORIAS DE BACHILLERATO

1º. Declaración de la autoría y acreditación de la misma.

El autor D. Carlos Alonso Calleja DECLARA ser el titular de los derechos de propiedad intelectual de la obra: Aplicación de la Realidad Aumentada a la minifabrica ICAI, que ésta es una obra original, y que ostenta la condición de autor en el sentido que otorga la Ley de Propiedad Intelectual.

2º. Objeto y fines de la cesión.

Con el fin de dar la máxima difusión a la obra citada a través del Repositorio institucional de la Universidad, el autor **CEDE** a la Universidad Pontificia Comillas, de forma gratuita y no exclusiva, por el máximo plazo legal y con ámbito universal, los derechos de digitalización, de archivo, de reproducción, de distribución y de comunicación pública, incluido el derecho de puesta a disposición electrónica, tal y como se describen en la Ley de Propiedad Intelectual. El derecho de transformación se cede a los únicos efectos de lo dispuesto en la letra a) del apartado siguiente.

3º. Condiciones de la cesión y acceso

Sin perjuicio de la titularidad de la obra, que sigue correspondiendo a su autor, la cesión de derechos contemplada en esta licencia habilita para:

- a) Transformarla con el fin de adaptarla a cualquier tecnología que permita incorporarla a internet y hacerla accesible; incorporar metadatos para realizar el registro de la obra e incorporar “marcas de agua” o cualquier otro sistema de seguridad o de protección.
- b) Reproducirla en un soporte digital para su incorporación a una base de datos electrónica, incluyendo el derecho de reproducir y almacenar la obra en servidores, a los efectos de garantizar su seguridad, conservación y preservar el formato.
- c) Comunicarla, por defecto, a través de un archivo institucional abierto, accesible de modo libre y gratuito a través de internet.
- d) Cualquier otra forma de acceso (restringido, embargado, cerrado) deberá solicitarse expresamente y obedecer a causas justificadas.
- e) Asignar por defecto a estos trabajos una licencia Creative Commons.
- f) Asignar por defecto a estos trabajos un HANDLE (URL *persistente*).

4º. Derechos del autor.

El autor, en tanto que titular de una obra tiene derecho a:

- a) Que la Universidad identifique claramente su nombre como autor de la misma
- b) Comunicar y dar publicidad a la obra en la versión que ceda y en otras posteriores a través de cualquier medio.
- c) Solicitar la retirada de la obra del repositorio por causa justificada.
- d) Recibir notificación fehaciente de cualquier reclamación que puedan formular terceras personas en relación con la obra y, en particular, de reclamaciones relativas a los derechos de propiedad intelectual sobre ella.

5º. Deberes del autor.

El autor se compromete a:

- a) Garantizar que el compromiso que adquiere mediante el presente escrito no infringe ningún derecho de terceros, ya sean de propiedad industrial, intelectual o cualquier otro.
- b) Garantizar que el contenido de las obras no atenta contra los derechos al honor, a la intimidad y a la imagen de terceros.
- c) Asumir toda reclamación o responsabilidad, incluyendo las indemnizaciones por daños, que pudieran ejercitarse contra la Universidad por terceros que vieran infringidos sus derechos e intereses a causa de la cesión.
- d) Asumir la responsabilidad en el caso de que las instituciones fueran condenadas por infracción

de derechos derivada de las obras objeto de la cesión.

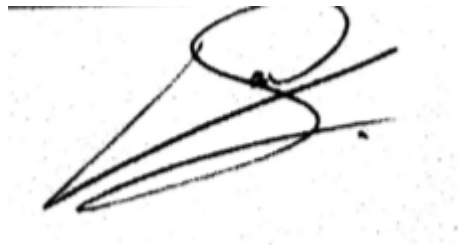
6º. Fines y funcionamiento del Repositorio Institucional.

La obra se pondrá a disposición de los usuarios para que hagan de ella un uso justo y respetuoso con los derechos del autor, según lo permitido por la legislación aplicable, y con fines de estudio, investigación, o cualquier otro fin lícito. Con dicha finalidad, la Universidad asume los siguientes deberes y se reserva las siguientes facultades:

- La Universidad informará a los usuarios del archivo sobre los usos permitidos, y no garantiza ni asume responsabilidad alguna por otras formas en que los usuarios hagan un uso posterior de las obras no conforme con la legislación vigente. El uso posterior, más allá de la copia privada, requerirá que se cite la fuente y se reconozca la autoría, que no se obtenga beneficio comercial, y que no se realicen obras derivadas.
- La Universidad no revisará el contenido de las obras, que en todo caso permanecerá bajo la responsabilidad exclusiva del autor y no estará obligada a ejercitar acciones legales en nombre del autor en el supuesto de infracciones a derechos de propiedad intelectual derivados del depósito y archivo de las obras. El autor renuncia a cualquier reclamación frente a la Universidad por las formas no ajustadas a la legislación vigente en que los usuarios hagan uso de las obras.
- La Universidad adoptará las medidas necesarias para la preservación de la obra en un futuro.
- La Universidad se reserva la facultad de retirar la obra, previa notificación al autor, en supuestos suficientemente justificados, o en caso de reclamaciones de terceros.

Madrid, a 12.de Julio de 2019

ACEPTA



Fdo Carlos Alonso Calleja

Motivos para solicitar el acceso restringido, cerrado o embargado del trabajo en el Repositorio Institucional:



ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA (ICAI)

MÁSTER EN INGENIERÍA DE TELECOMUNICACIÓN

Aplicación de la realidad aumentada a la minifábrica ICAI

Autor: Carlos Alonso Calleja

Director: José Antonio Rodríguez Mondéjar

Madrid

Julio 2019

Agradecimientos

A mi familia por darme toda la educación, valores y afecto que me han ayudado a ser quién soy hoy en día.

A mis compañeros de máster quienes me acogieron como una gran familia cuando llegué a Madrid hace un par de años.

A todos los profesores que he podido conocer a lo largo de este máster en ICAI los cuales me han brindado siempre todo tipo de apoyo siendo pilares fundamentales en la evolución en el aprendizaje que he experimentado en estos últimos dos años.

A José Antonio Rodríguez Mondéjar por confiar en mi para llevar a cabo este ambicioso proyecto y apoyarme siempre que lo he necesitado.

Finalmente, quiero acordarme de todas las personas que han pasado por mi vida y que siempre han creído en mi de una manera o de otra haciendo suyos mis éxitos, pero más aún los fracasos que también han existido.

Muchas gracias a todos.

APLICACIÓN DE LA REALIDAD AUMENTADA A LA MINIFABRICA ICAI

Autor: Alonso Calleja, Carlos.

Director: Rodríguez Mondéjar, José Antonio.

Entidad Colaboradora: ICAI – Universidad Pontificia Comillas

RESUMEN DEL PROYECTO

En este proyecto se plantea el desarrollo de una aplicación de realidad aumentada para poder visualizar los datos asociados a algunos componentes de la minifábrica ICAI. Estos datos serán tanto estáticos provenientes del datasheet como dinámicos para realizar una monitorización en tiempo real de lo que está pasando por el componente analizado.

Palabras clave: Realidad aumentada, OPC UA, Real Time, Industria 4.0

1. Introducción

La industria 4.0 es una gran herramienta de cara a la monitorización de la industria y por supuesto de las mejoras en eficiencia de la misma. Sin embargo, se detecta que muchas veces no es sencillo identificar los errores dentro de un entorno industrial y además esto mezclado con la falta de preparación de los operarios en muchas ocasiones produce grandes retrasos en el ritmo de manufactura.

Por ello nace este proyecto con el fin de aprovechar el potencial de la Industria 4.0 junto con la realidad aumentada, de manera que ambas tecnologías aprovechen sus sinergias para crear un proyecto que ayude a la monitorización de la industria de una manera sencilla y muy visual con la cual personal menos cualificado podrá detectar de una manera rápida y sencilla el componente que está visualizando a través de su dispositivo con cámara. Tras la detección, el usuario podrá indagar en la información oculta del dispositivo tanto a nivel de la información constante existente en la hoja de características del mismo, como en el valor de las variables dinámicas asociadas al funcionamiento en tiempo real de dicho componente.

2. Definición del proyecto

El proyecto consiste en realizar una aplicación de realidad aumentada mediante Unity que sea capaz apoyada por Vuforia de detectar un target adherido a un componente del laboratorio de ICAI y asociarlo a un componente existente en nuestra base de datos.

De esta manera dependiendo de que botón pulse el usuario la aplicación mostrará por pantalla los datos técnicos asociados al componente o mostrará el valor en tiempo real de la variable que posee ese componente cuando está funcionando; como puede ser un sensor, cuya variable nos dirá si está detectando o no el paso de un objeto frente a él.

El proyecto es el resultado de la integración de dos proyectos bien diferenciados como son la conexión entre el laboratorio de la universidad con la nube a través del protocolo OPC UA y los servicios proporcionados por Bluemix y la Aplicación de Realidad Aumentada que debe realizar la detección y las consultas a la base de datos NoSql para poder visualizar por pantalla los datos que correspondan al componente en cuestión.

3. Descripción del sistema

Como hemos comentado anteriormente el proyecto se basa en la integración de dos proyectos diferenciados que tienen la nube como punto de unión para la visualización de la información. Es necesario constatar la importancia del direccionamiento IP para acceder a la IP del router desde la nube, además de la utilización del protocolo de comunicación industrial OPC UA para poder establecer la conexión entre el servidor alojado en el PLC y el cliente alojado en la aplicación Node Red desarrollada en Bluemix. Una vez diseñado todo esto nuestra aplicación mediante consultas http conseguirá los archivos json correspondientes a los componentes detectados para mostrar por pantalla en caso de que el usuario así lo requiera. Esta arquitectura de sistema se podrá ver en la Ilustración 1 de una manera simplificada.

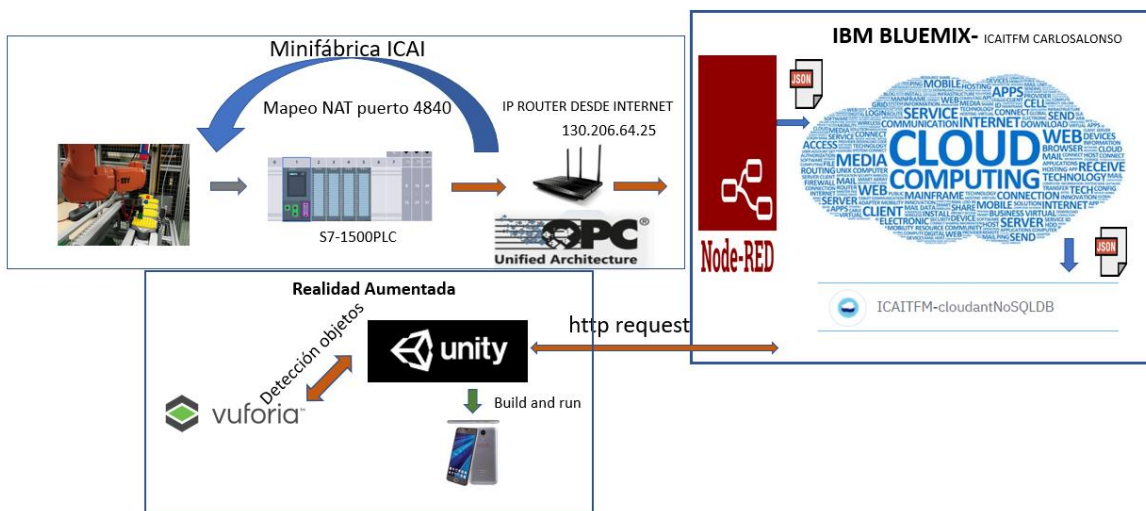


Ilustración 1 – Esquema arquitectura del proyecto

4. Resultados

Los resultados fueron muy positivos ya que se consiguieron todos los objetivos previamente asignados cuando se determinó el alcance del proyecto. Se consiguió visualizar los datos asociados al componente detectado tanto los estáticos que fue la parte más sencilla puesto que no intervenía para nada la conexión con el laboratorio como los datos dinámicos. En esto último se observó cómo se la aplicación era capaz de reflejar los cambios que sufría la variable que estaba observando en tiempo real. Para este fin se implementó una petición http la cual devolvía siempre el ultimo json que había llegado a la base de datos ya que este reflejaría el valor actual que tomaba la variable de estudio.

Por tanto, se consiguió conectar la Industria 4.0 con la realidad aumentada mediante los servicios proporcionados por Bluemix, el protocolo OPC UA y por supuesto las herramientas de realidad aumentada ya mencionadas anteriormente.

Como se puede observar en las imágenes adjuntas, la aplicación es capaz de identificar el componente que tienen delante y además en el caso en el que, en este caso, el sensor Omron, esté detectando el estado de la variable pasará a True corroborando el correcto funcionamiento del sensor, y su conexión con el PLC de control.

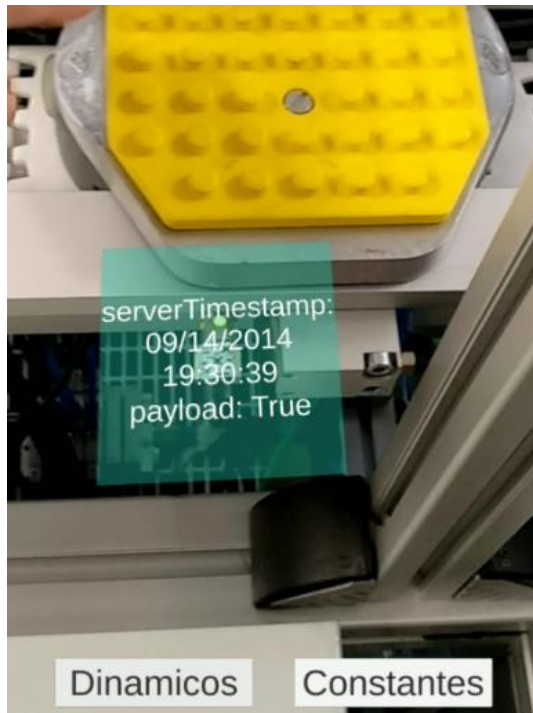


Ilustración 2 y 3 – Resultado obtenido tras detección de un componente y pulsado para visualizar datos dinámicos

5. Conclusiones

El producto de este trabajo fue una aplicación con potencial para el uso académico e industrial que facilita el trabajo a todo aquel que quiera realizar una monitorización de una planta industrial o simplemente se esté iniciando en el entorno industrial y quiera aprender de una manera fácil, visual y rápida.

La potencia del proyecto implementado como base para futuros desarrollos es incalculable teniendo en cuenta la multitud de servicios que proporciona la nube y las mejores a nivel de realidad aumentada que pueden realizarse.

Por tanto, se concluye que el producto desarrollado ha sido satisfactorio y supondrá una importante mejora en la eficiencia de la minifábrica existente en ICAI siendo útil para personal con mucha formación y también para todo aquel que no tenga formación relacionada, y quiera, mediante una forma visual y sencilla obtener datos en tiempo real de las características de un sensor y como está funcionando en ese instante.

APPLICATION OF AUGMENTED REALITY TO THE SMART FACTORY OF ICAI

Author: Alonso Calleja, Carlos.

Supervisor: Rodríguez Mondéjar, José Antonio.

Collaborating Entity: ICAI – Universidad Pontificia Comillas

ABSTRACT

This project considers the development of an augmented reality application to visualize the data associated with some components of the ICAI mini-factory. This data will be both statistic and dynamic data to perform a real-time monitoring of what is happening through the analyzed component.

Keywords: Augmented Reality, OPC UA, Real Time, Industry 4.0

1.Introduction

Industry 4.0 is a great tool for the monitoring of the industry and of course the improvements in the efficiency of it. However, it is often detected that it is not easy to identify errors within an industrial environment and this, in addition to the lack of preparation of the workers, often causes large delays in the manufacturing rate.

That is why this project was born in order to take advantage of the potential of Industry 4.0 together with augmented reality, so that both technologies take advantage of their synergies to create a project that helps monitor the industry in a simple and very visual way. Thanks to this project less qualified staff can quickly and easily detect the component you are viewing through your camera device. After detection, the user will be able to inquire into the hidden information of the device both at the level of the constant information existing in the data sheet thereof, and in the value of the dynamic variables associated with the real-time operation of the component.

2.Project definition

The project consists in developing an application of augmented reality using Unity that is able to be supported by Vuforia to detect a target adhered to a component of the ICAI laboratory and associate it with an existing component in our database.

In this way, depending on which button the user presses, the application will show on the screen the technical data associated with the component or show the value in real time of the variable that this component has when it is working; for example a sensor, whose variable will tell us if it is detecting or not the passage of an object in front of it.

The project is the result of the integration of two well differentiated projects such as the connection between the university laboratory and the cloud through the OPC UA protocol and the services provided by Bluemix and the Augmented Reality Application that must perform the detection and the queries to the database NoSql, to be able to visualize by screen the data that correspond to the component in question.

3. System description

As we have previously commented, the project is based on the integration of two differentiated projects that have the cloud as a point of union for the visualization of information. It is necessary to verify the importance of IP addressing to access the IP of the router from the cloud, in addition to the use of industrial communication protocol OPC UA to establish the connection between the server hosted in the PLC and the client hosted in the Node application Network developed in Bluemix. Once all this has been designed, our application through http queries will get the json files corresponding to the components detected to be displayed on screen in case the user so requires. This system architecture can be seen in Figure 1 in a simplified manner.

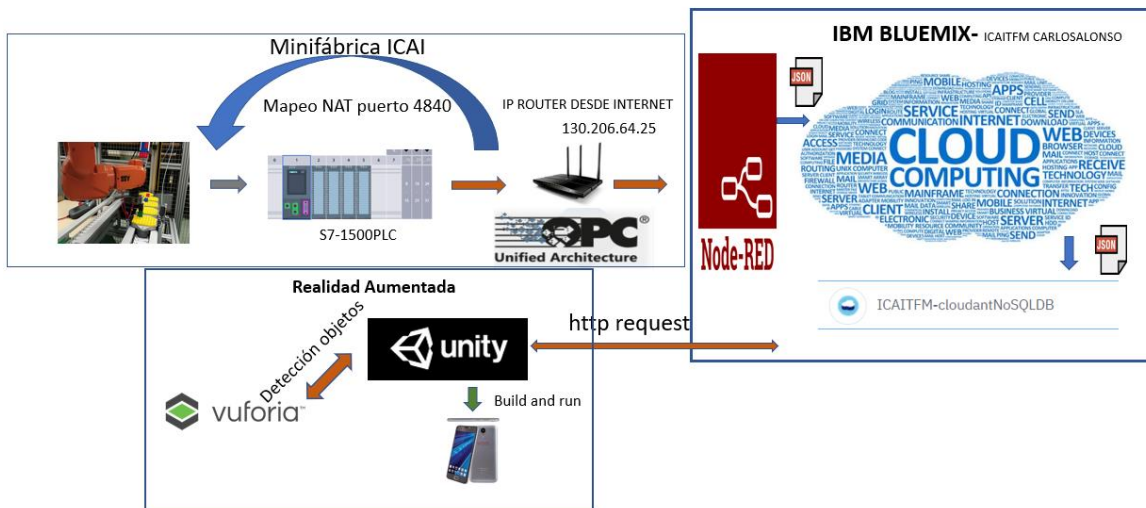


Figure 3 –Scheme architecture of the project

4. Results

The results were very positive since all previously assigned objectives were achieved when the scope of the project was determined. It was possible to visualize the data associated with the detected component, both the static ones, which was the simplest part since the connection with the laboratory and the dynamic data did not intervene at all. In the latter, it was observed how the application was able to reflect the changes suffered by the variable that was observing in real time. For this purpose, an http request was implemented, which always returned the last json that had arrived at the database since it would reflect the current value that the study variable took.

Therefore, it was possible to connect Industry 4.0 with augmented reality through the services provided by Bluemix, the OPC UA protocol and, of course, the augmented reality tools mentioned above.

As can be seen in the attached images, the application is able to identify the component in front of it and also in the case in which, in this case, the Omron sensor, is detecting

the status of the variable will pass True corroborating the correct sensor operation, and its connection to the control PLC.



Figures 4 y 3 –Result obtained after detecting a component and pressing to display dynamic data

5. Conclusions

The product of this work was an application with a lot of potential for academic and industrial use that makes it easier for anyone who wants to monitor an industrial plant or simply is starting in the industrial environment and want to learn in an easy way, visual and fast.

The power of the project implemented as a basis for future developments is incalculable taking into account the multitude of services provided by the cloud and the best at the level of augmented reality that can be realized.

Therefore, it is concluded that the product developed has been satisfactory and will imply a significant improvement in the efficiency of the existing mini-factory in ICAI being useful for highly trained staff and also for anyone who has no related training, and wants, through a visual and simple way, to obtain real-time data of the characteristics of a sensor and how it is working at that moment.

Índice de la memoria

Contenido

<i>Índice de la memoria</i>	<i>I</i>
<i>Capítulo 1. Introducción</i>	<i>5</i>
<i>Capítulo 2. Descripción de las Tecnologías</i>	<i>6</i>
2.1 La Industria 4.0 y el Protocolo OPC UA.....	6
2.2 Almacenamiento en la Nube	10
2.3 IBM CLOUD.....	11
2.4 Vuforia	13
2.5 UNITY	14
<i>Capítulo 3. Estado del Arte</i>	<i>16</i>
<i>Capítulo 4. Definición del trabajo</i>	<i>20</i>
4.1 Motivación del Proyecto	20
4.2 Objetivos	20
4.3 Metodología.....	21
<i>Capítulo 5. Implementación de la solución</i>	<i>22</i>
5.1 Infraestructura de la minifábrica	24
5.2 Implementación de la conexión con IBM	28
5.2.1 Desarrollo Node Red.....	30
5.2.2 Almacenamiento en Cloudant NoSql DB.....	35
5.3 Creación de Aplicación de Realidad Aumentada.....	37
5.3.1 Desarrollo Detección con Vuforia	37
5.3.2 Desarrollo Front End.....	40
5.3.3 Implementación de las funcionalidades con Unity	42
<i>Capítulo 6. Resultados obtenidos</i>	<i>52</i>
6.1 Caso real de uso.....	58
<i>Capítulo 7. Futuros Desarrollos y conclusiones</i>	<i>62</i>

Capítulo 8. Bibliografía..... 65

Figura 1. Arquitectura IoT.....	7
Figura 2. Piramide Industria 4.0	8
Figura 3. Arquitectura de Vuforia	14
Figura 4. Tecnologías de Realidad Aumentada en la actualidad.....	19
Figura 5. Planificación del Proyecto de Fin de Máster.....	21
Figura 6. Arquitectura del Proyecto Implementado	23
Figura 7. Minifábrica ICAI.....	23
Figura 8. Componente PLC Tia Portal	25
Figura 9. Activación servidor OPC UA.....	25
Figura 10. Asociación Variable a entrada digital	26
Figura 11. Entrada digital asociada a variable 1.....	26
Figura 12. Direcciones del servidor OPC UA	28
Figura 13. Enrutamiento de los datos desde los sensores a Bluemix	28
Figura 14. Recursos de almacenamiento Bluemix	29
Figura 15. Panel de control de la aplicación en Bluemix	30
Figura 16. Flujo conexión sensor-Node Red.....	31
Figura 17. Componente OPC UA en Node Red.....	32
Figura 18. Dirección de la variable en el servidor OPC.....	32
Figura 19. Demo OPC UA para acceder a la ID de las variables.....	33
Figura 20. Creación del cliente OPC UA	34
Figura 21. Punto de acceso a la DB.....	34
Figura 22. Base de datos dinámico-TFM	35
Figura 23. Json ejemplo de datos dinámicos	36
Figura 24. Entorno Unity.....	38
Figura 25. Base de datos Vuforia	39
Figura 26. Implementación escenario.....	40
Figura 27. Interfaz de Usuario-detección	41
Figura 28. Interfaz de Usuario.....	42
Figura 29. Nombre asociado al target.....	43
Figura 30. Inicialización de variables.....	45

Figura 31. Json recibido en Unity.....	46
Figura 32. Visualización cámara	53
Figura 33. Detección componente 1	54
Figura 34. Detección componente 2	54
Figura 35. Resultado datos estáticos.....	55
Figura 36. Json datos estáticos en la nube	56
Figura 37. Resultado datos dinámicos	57
Figura 38. Detección de sensor del laboratorio	58
Figura 39. Datos estáticos asociados al sensor	59
Figura 40. Datos dinámicos asociados al sensor	60
Figura 41. Sensor no detectando	61

Capítulo 1. INTRODUCCIÓN

Uno de los grandes problemas que podemos observar en la industria y en particular en los procesos de fabricación es la dificultad para identificar los errores derivados de una máquina ya que ese error suele contaminar el proceso completo por lo que hay que seguir debidamente el flujo componente a componente para poder identificar cuál es el origen.

La realidad aumentada está apareciendo en el siglo XXI como una manera óptima de asociar información no evidente con la realidad observada. Gracias a esta nueva tecnología se han desarrollado multitud de aplicaciones cuyo objetivo final, aparte de mostrar más ser más interactivas y fáciles de entender, es proporcionar mayor información al usuario de manera que se agilicen los procesos allí donde se quiera usar.

La manera óptima de utilizar esta potente forma de visualizar los datos es tratar de mezclar los datos estáticos, es decir, aquellos datos constantes que no depende del momento de la visualización con los datos dinámicos que si dependerán del momento exacto en el que se quiera obtener la información.

En este proyecto se pretende facilitar la identificación directa del error en la industria través del reconocimiento óptico. Para este fin se usará la cámara de una tablet o smartphone para identificar el componente que aparece en pantalla y asociar la información que se crea más relevante tanto estática como dinámica para poder observar posibles errores en caso de no encajar las características o el valor de las variables con los esperado en cada momento.

Capítulo 2. DESCRIPCIÓN DE LAS TECNOLOGÍAS

La tecnología que se han utilizado para desarrollar esta aplicación de realidad aumentada ha sido Unity en combinación con Vuforia para el reconocimiento visual. El lenguaje de programación usado dentro del entorno de Unity es #C.

Para el almacenamiento de datos de ha utilizado una base de datos NoSql en IBM.

2.1 LA INDUSTRIA 4.0 Y EL PROTOCOLO OPC UA

Para la parte de datos dinámicos será muy importante desarrollar el concepto de la Industria 4.0 y el Internet de las cosas para poder conectar los objetos del día a día a Internet y así obtener todo el potencial que se puede extraer de ello. Cada objeto tendrá unas características especiales y que lo diferenciarán de los demás claramente por el proceso que se debe seguir para monitorizar los datos.

En este caso concreto será muy importante la aplicación de IoT en un entorno industrial para permitir todas las funcionalidades de la Industria 4.0 ya que se necesitan los medios adecuados para permitir la conexión de los diferentes componentes de la fábrica con la nube.

Con la combinación del Internet of Things en la industria se puede obtener el Industrial Internet of Things que significará la evolución de la industria presentando unas mejores condiciones que el IoT sirviendo como un sistema que tendrá mayor fiabilidad frente a errores siendo más escalable.

La manera en la que está enfocada la implementación de la industria conectada será una funcionalidad orientada a eventos dentro de una red abierta en la cual los diferentes componentes tienen la facultad de funcionar de manera independiente.

Cuando IoT se lleva a la aplicación es importante destacar que hay diferentes capas que serán la capa física que es la que componen todos los dispositivos existentes en la red que tendrán una comunicación real time con la base de datos; la capa intermedia o de control que será la capa que comunique estos dispositivos con la capa de Internet; y finalmente, la capa superior que es la que permitirá esa interacción con los datos recibidos a través de Internet of Things enfocado al mundo industrial.

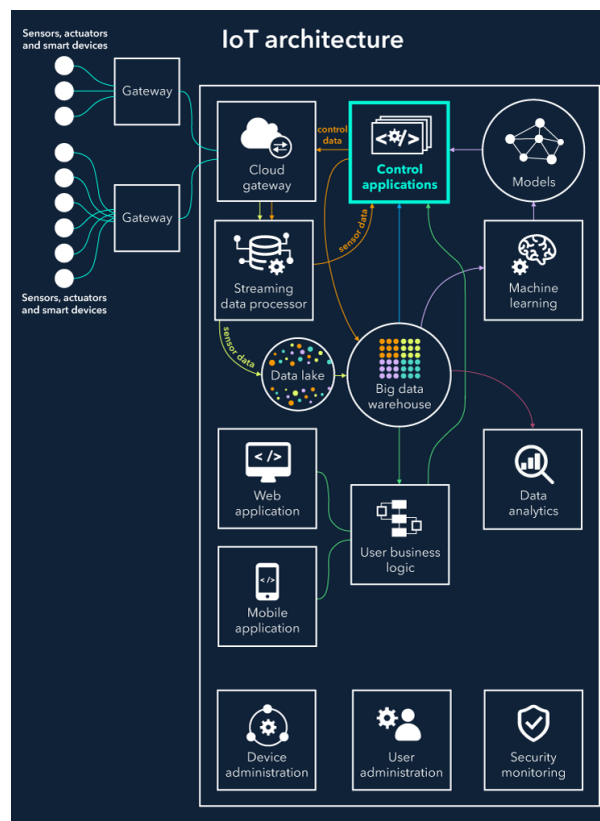


Figura 1.Arquitectura IoT

El internet de las cosas aplicado a la industria tiene como objetivo interconectar los sistemas físicos presentes en esta para que todo el sistema funcione como una orquesta sin interrupciones. Por tanto, se puede concluir que para la existencia de estos sistemas de Industria 4.0 es necesaria la existencia de los componentes físicos que puedan conectarse a Internet.

En el laboratorio presente en la universidad existe una arquitectura de comunicaciones llamada OPC-UA que establece un protocolo de comunicaciones para poder desarrollar la Industrial Internet of Things. El OPC-UA es una arquitectura libre y que sirve para muchas plataformas y además de todo esto está enfocada a los servicios lo que la hace muy escalable (SOA). La información tiene un modelo peculiar ya que utiliza diferentes nodos los cuales pueden escribir, leer y transmitir información. Este modelo fue creado como un proceso de evolución ante el protocolo OPC ya que a diferencia del anterior tiene mejoras drásticas de cara a la seguridad de la transmisión, las plataformas que soporta y su escalabilidad.

A nivel de los tipos de comunicación soportados soporta dos bastantes diferenciados entre si tanto en termino de eficiencia como de formato. Existe el protocolo SOAP a través del cual se mandan archivos en formato XML entre dos componentes siendo muy apto para el desarrollo con otras herramientas y el binario que como es lógico ofrece un mayor rendimiento al evitar algunas transformaciones de datos.

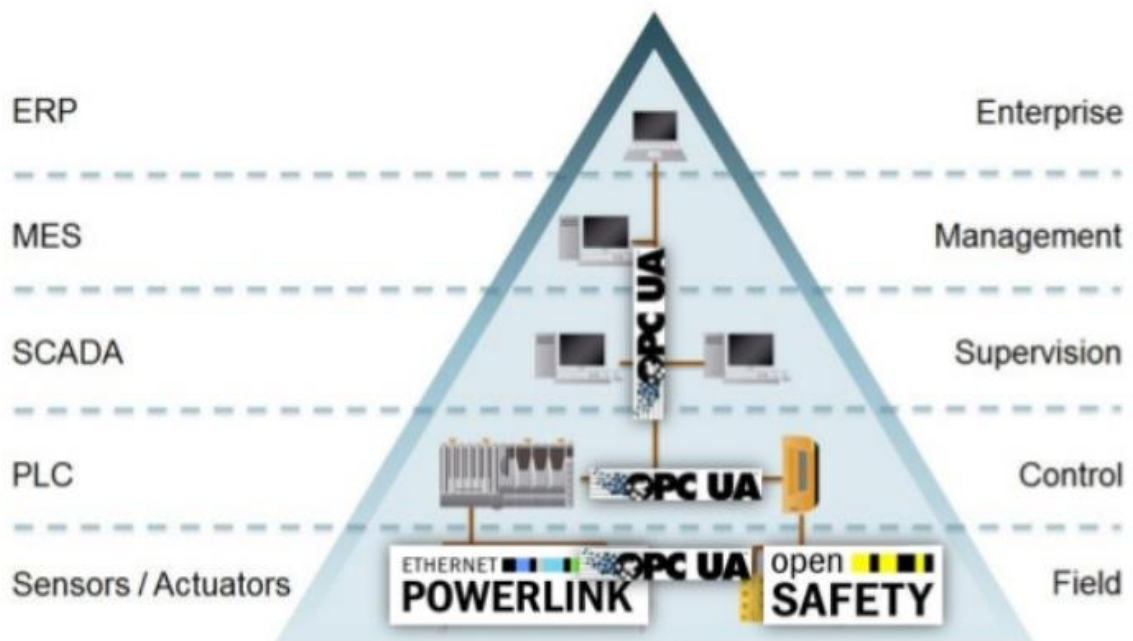


Figura 2. Pirámide Industria 4.0[15]

Como se puede observar en la figura anterior OPC-UA se utiliza para transmitir información entre multitud de componentes diferentes como son PLC, sistemas SCADA, HMI independientemente de quien haya sido el fabricante.

La pirámide de comunicaciones representa como este protocolo de comunicaciones permite dar saltos verticales entre distintos sistemas situados en medios diferentes como son el físico y la propia nube en la que se almacena la información.

2.2 ALMACENAMIENTO EN LA NUBE

La nube y como ha evolucionada tienen un gran potencial de cara a la evolución tecnológica y todas las aplicaciones que se pueden desarrollar en ella. Esta nueva infraestructura que se nos presenta hoy en día cuenta con multitud de ventajas ya que con mucha menos infraestructura y costos se pueden desarrollar aplicaciones muy potentes fácilmente escalables siendo rápidamente implementadas.

Por supuesto dentro de todo lo que es la computación en la Nube existen tecnologías muy diferenciales y formas de entender el desarrollo de aplicaciones. Dentro de esta tecnología existen 3 maneras de desarrollo principales como son:

- IaaS: Infraestructura como servicio
- PaaS: Plataforma como servicio
- SaaS: Software como servicio

Estos tres modelos de desarrollo lo que hacen es proporcionar a los usuarios tres modelos diferentes de servicios cada uno con un coste asociado debido a la complejidad del mismo.

La lógica de la computación en la nube erradica en la capacidad que tienen las empresas proveedoras de ampliar los servicios debido a los grandes datacenter con multitud de servidores que poseen pero que en el caso de la mayor parte de las compañías significarían realizar una inversión importante para auto proporcionarse este tipo de servicios además de los costes de mantenimiento recurrentes que estos sistemas tienen. Las diferencias principales entre las 3 es que mediante SaaS ya te ofrecen el entorno de usuario directamente en el propio proveedor sin tener que realizar una instalación en el dispositivo del usuario. En el caso de PaaS lo que se proporcionan son los diferentes módulos para poderlos integrar en la aplicación sin tener que realizar una programación compleja y finalmente en IaaS solo se ofrecen la parte física del servicio sobre la que se construirá la aplicación deseada.

En este trabajo que se basará en el modelo PaaS que se detallará más adelante se observa como estos módulos proporcionan APIs para desarrollar el lenguaje de programación de manera sencilla e intuitiva. Algunas de las plataformas más conocidas son Azure o IBM.

En el proyecto que aquí se está llevando a cabo hay una serie de niveles muy diferenciados en los que se evidencian las diferentes tecnologías y protocolos que se utilizan en cada ocasión. En la parte física los elementos principales son los PLCs los cuales están interconectados entre ellos siendo uno de ellos el que está conectado a Internet por el cual se establecerá la comunicación con la nube.

Como parte fundamental para representar esta interconexión entre la parte física y la parte en la nube debe destacarse el protocolo OPC UA basado en una lógica de servidor-cliente pudiendo evaluar la monitorización de los datos desde una perspectiva física fomentando un control sin necesitar Internet y también monitorizando estos datos desde IBM aprovechando todas las posibles herramientas que otorga Bluemix. La gran ventaja de utilizar los PLCs con el protocolo OPC UA es que se crea una arquitectura común para todos los componentes y no se debe individualizar el desarrollo para cada componente.

La elección de Bluemix se ha hecho debido a los servicios que otorga y a la facilidad de implementación que son mejores que las ofrecidas por Azure.

2.3 IBM CLOUD

En este apartado se desarrollará la explicación sobre que es IBM Cloud y porque se ha usado para este proyecto. Lo primero que hay que decir es que como comentamos anteriormente Bluemix sigue la estrategia (PaaS) por lo que ofrece numerosos tipos de servicios muy variados como son bases de datos, APIs para el análisis de datos, monitorización de variables, desarrollo para móvil, y por supuesto el entorno propicio para el desarrollo de aplicaciones en la nube y su gestión siendo el lenguaje de programación muy variado para no suponer ningún inconveniente al desarrollador.

La tecnología utilizada por IBM para conseguir esto es la llamada Cloud Foundry que es la infraestructura cloud que soporta el entorno de desarrollo Bluemix. La idea con la que surgió Bluemix es buscar el mayor balance escalabilidad/accesibilidad de manera que vaya incorporando el mayor número de servicios atendiendo a las necesidades de los usuarios.

Los servicios que Bluemix recomienda con respecto al desarrollo de IoT y la Industria 4.0 son principalmente WIoT (Watson Internet of Thing Platform) y por supuesto la base de datos que ofrece Bluemix para el almacenaje en cloud que es Cloudant NoSql DataBase.

El servicio Watson es muy importante si se quieren monitorizar los datos de los dispositivos teniendo la capacidad de extraer información adicional gracias al histórico que este servicio tendrá de cada dispositivo. Bluemix hace de Watson una herramienta muy potente si se combina con el resto de los servicios de análisis de datos y además proporciona un desarrollo muy simple debido a que Bluemix proporciona las APIs necesarias para no tener que desarrollar excesivo lenguaje de programación.

La manera de conectar los dispositivos con Bluemix es el software NODE-RED que es un programa que da acceso de manera sencilla a los dispositivos del laboratorio desde Bluemix además de poder conectarlo con el resto de los servicios que Bluemix proporciona. Para su implementación NODE-RED utiliza Node.js que es un lenguaje sobre el que se compila Javascript y además tiene una interfaz muy interactiva en la que se diseñan flujos con paquetes a través de los cuales se consigue una implementación muy sencilla de la programación.

Como hemos comentado anteriormente los datos son almacenados en la base de datos que nos proporciona Bluemix que a diferencia de lo que históricamente se ha entendido como bases de datos, es decir una gran tabla de almacenaje que evolucionó en las bases de datos relacionales las cuales podían almacenar los datos en pequeñas tablas de almacenaje que se relacionan entre ellas para poder acceder de una manera rápida y dinámica a los datos.

Sin embargo, partiendo de este concepto y viendo la poca flexibilidad que ofrecen este tipo de bases de datos frente al Internet of Things y la Industria 4.0 se han creado las bases de datos no relacionales ya que ofrecen una mayor flexibilidad para poder añadir datos

reduciendo todos los parámetros de la propia tabla y así propiciando que esta sea mucho más escalable.

La base de datos IBM Cloudant NoSql es una base de datos no relacional que almacena los datos en formato JSON y los va apilando según van llegando. Este nuevo formato no tiene esquemas y es muy fácil de interpretar por aquellas aplicaciones que han las llamadas oportunas.

2.4 VUFORIA

Vuforia es una de las herramientas principales que se han usado en el desarrollo de esta aplicación de realidad aumentada cuya función principal es ser capaz de identificar el target de detección ya que estará alojado en la correspondiente base de datos del desarrollador de la aplicación en el propio sistema Vuforia.

Una aplicación realizada con Vuforia utiliza la pantalla del dispositivo que se pretende utilizar como una lente en la que se mezclan los elementos del mundo virtual con los del mundo real.

- La experiencia ofrecida por el sistema Vuforia son las siguientes:
- Reconocimiento de texto.
- Reconocimiento de imágenes.
- Rastreo fijo ya que el target no se va a perder tan fácilmente una vez haya sido detectado.
- Detección instantánea.

Los elementos que componen un sistema Vuforia son los siguientes:

- Cámara: Para captar la imagen y trackearla con la base de datos.
- Base de datos: Se crea utilizando el Target Manager, pudiendo estar alojada en la nube o en local.
- Target: Son los objetivos usados por el rastreador que sirven para reconocer un objeto del mundo real, como pueden ser los image targets o los Word targets.

- Tracker: Analiza los frames que recibe la cámara con la intención de encontrar las similitudes con los targets de la base de datos.

Vuforia SDK

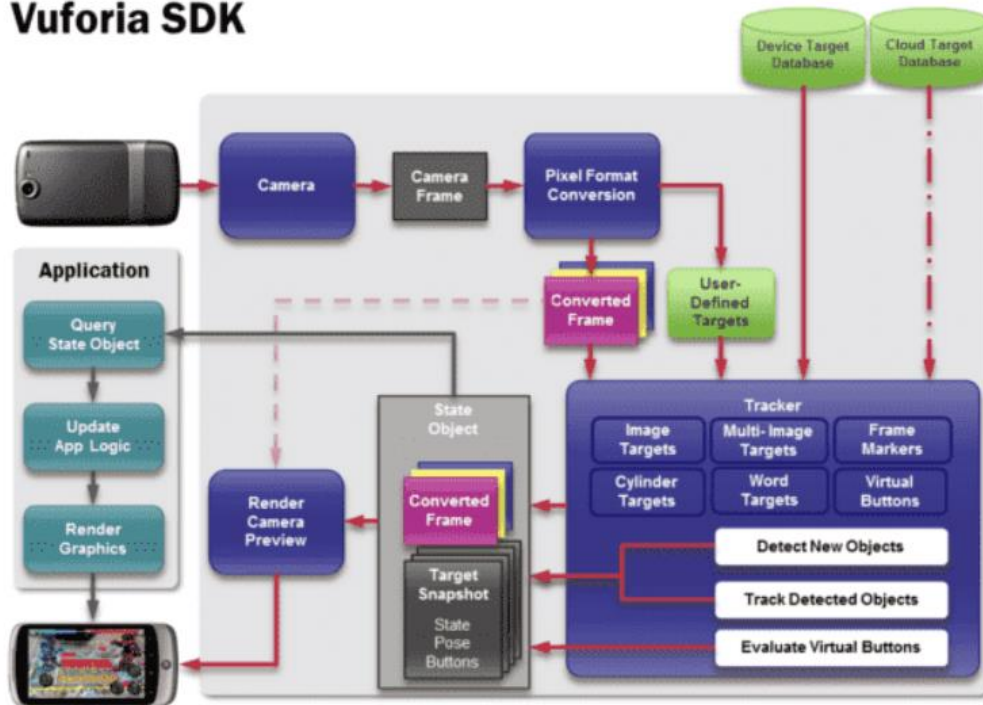


Figura 3. Arquitectura de Vuforia[5]

El modo de funcionamiento es la captación de una escena en vivo a través de la cámara. El SDK de Vuforia captará los frames de la imagen capturada por la cámara y cambia la resolución para ser captada instantáneamente por el tracker. Tras esto, busca coincidencias en la base de datos para más tarde renderizar algún contenido virtual que combinará los elementos virtuales con los elementos de la realidad observada.

2.5 UNITY

Unity es una herramienta que por encima de todo está enfocada al desarrollo de videojuegos en diferentes plataformas mediante el uso de un editor y scripting. Afortunadamente es una tecnología abierta a todos los públicos de manera que cualquiera puede acceder a las ventajas

que proporciona. Unity3D destaca por un editor muy visual y sencillo en el que con muy pocos clicks podemos crear e importar modelados en 3 dimensiones, texturas y sonidos además de incluir herramientas de desarrollo en C y Javascript con las cuales podremos extender las funcionalidades de la propia plataforma de edición que nos proporciona Unity3D.

Otra de las ventajas que tiene Unity es que posee una tienda interna como puede ocurrir en el caso de AppleStore donde hay multitud de recursos gratuitos y otros de pago extendiendo la herramienta mediante Plugins internos de la propia plataforma.

Para este caso en concreto se integrará la tecnología Vuforia dentro de la plataforma Unity con el fin de proporcionar la funcionalidad de detección de objetos siendo Unity el encargado de construir la aplicación sobre la que se montará esta funcionalidad extendiendo con un desarrollo en #C las funcionalidades de este para así poder comunicar nuestra aplicación con las bases de datos que se utilicen.

Actualmente a través de Unity se han creado la mitad de los juegos del mundo.

Capítulo 3. ESTADO DEL ARTE

La realidad aumentada, aunque tan repetida últimamente es una tecnología relativamente nueva y tiene como función principal aprovechar todo el potencial de la mezcla entre la realidad observada con una realidad virtual creada para esa ocasión concreta. El propósito final de esta tecnología es dotar de información extra a lo que podemos percibir con nuestros sentidos.

Este estudio sobre el estado del arte de esta tecnología esta enfocado en las aplicaciones industriales para lo que se ha utilizado ya que una de sus aplicaciones más conocidas es la de los videojuegos, pero no es el propósito de este trabajo a no ser que alguno de los mismo haya usado una tecnología innovadora que haya ayudado a posteriores desarrollos industriales.

Uno de los puntos en los que más se ha avanzado a la hora de construir realidad aumentada son los sensores que sirven para captar la misma incluidas las cámaras presentes en los dispositivos multimedia.

Dentro de la detección se han utilizado varias tecnologías a lo largo de la historia que pasamos a describir a continuación:

-Tecnología de seguimiento basada en sensores: Era la fundamental hace más de una década con unos sistemas de detección basados en Wifi, bluetooth, infrarrojos. Hoy en día estas técnicas se combinan con el visionado para mejorar más aun las prestaciones de las mismas.

Como principal inconveniente se observa la necesidad de infraestructuras de conexión para este tipo de redes además del factor de que funcionan mejor en interiores a no ser que esten apoyados por sistemas GPS.

-Tecnología basada en visión por computador: Se basa en la capacidad para la captación de imágenes por medio de los dispositivos actuales. Dentro de esta tecnología se extraen dos subsectores:

- Basado en marcadores: Código de barras, Qr.... Se reconocen fácilmente y permiten el calculo de la distancia y el ángulo con respecto a ellos, pero tienen como inconveniente principal que cualquier obstáculo intermedio perjudicaría a la identificación por lo que se requiere una imagen nítida del objeto a detectar. Esta es una técnica optima para los escenarios que cumplan los requisitos ya que no se requiere una gran capacidad de procesado.
- Reconocimiento sin marcadores: Todos aquellos tipos de reconocimiento basados en modelos en los cuales se definen los puntos de mayor importancia y a partir de ahí se construye el escenario en el que se renderizarán los datos. Aunque los entornos no tengan que estar previamente preparados si que deben tener unas características concretas para poder realizar el reconocimiento.

Otra opción para la detección sin marcadores es realizar este reconocimiento siguiendo los patrones naturales de la propia imagen buscando toda aquella estructura física que pueda ser identificable por el algoritmo de detección.

Y por supuesto en la actualidad hay numerosas tecnologías híbridas que utilizan varios de estos métodos para poder realizar una identificación más precisa y sobre todo más redundante ante errores que puedan aparecer.

Profundizando en las técnicas de visualización cabe destacar dos principalmente que son aquellas que constan de una pantalla transparente sobre la que se muestra la información virtual y las opacas en las que la realidad observada es simulada o grabada, y sobre ella se renderiza esa información virtual que crea la realidad aumentada.

A continuación se mostrarán todas las tecnologías existentes actualmente acorde a un estudio desarrollado por el Gobierno de Aragón[13]:

Producto	Tipo de Licencia	Plataforma	Características	Descripción
ALVAR	Libre, Comercial	Android, iOS, Windows, Flash	Seguimiento basado en marcadores y sin marcadores	Librería software para crear aplicaciones de RA y RV. Desarrollada por el Instituto Técnico de Investigación VTT (VTT Technical Research Centre of Finland).
ARLab	Libre, Comercial	Android, iOS	GPS, sensors (IMU Sensors), búsqueda visual	ARLab ofrece un amplio portfolo de soluciones tecnológicas para RA.
ARmedia	Libre, Comercial SDK	Android, iOS, Windows, Flash	Seguimiento basado en marcadores	La plataforma ARmedia es un framework de desarrollo estructurado y modula que incluye distintos módulos software. Este framework es independiente del motor de seguimiento en tiempo real y del motor de renderización.
Arpa	Libre, Comercial SDK	Android, iOS, Windows	Seguimiento basado en marcadores y sin marcadores, GPS, Sensores (IMU sensors), tracking facial y por infrarrojos, y renderización en tiempo real	Arpa Solutions es una compañía líder en el desarrollo de productos y aplicaciones de realidad aumentada a través de su plataforma propietaria ARPA AR.
ARToolkit	Open Source, Comercial SDK	Android, iOS, Linux, OSX, Windows	Seguimiento basado en marcadores y sin marcadores	ARToolkit es una plataforma de Realidad Aumentada que está disponible para múltiples sistemas operativos: iOS, Android, Linux, Windows y Mac OS.
ArUco	Open Source	Linux, OSX, Windows	Marcadores	Librería para aplicaciones de RA basada en OpenCV
Aurasma	Libre, Comercial SDK	Android, iOS	Solución sin marcadores que se basan en las características naturales de la imagen o el objeto (bordes, esquinas o texturas), una técnica conocida como NFT (Natural Feature Tracking)	Es una solución de HP que incorpora reconocimiento automático de imágenes
BazAR	Open Source	Linux, OSX, Windows	Soluciones sin marcadores basadas en Natural Feature Tracking	BazAR es una librería de visión por computador basada en la detección de características de la imagen y su posterior matching. En particular, es adecuada para detectar y registrar estructuras planares en imágenes.
Beyond Reality Face	Comercial SDK	Flash	Tracking facial	Beyond Reality Face Nxt es una solución de tracking facial para desarrolladores y usuarios que proporciona una API disponible para todas las plataformas.
Catchoom	Libre, Comercial SDK	Android, iOS	Búsqueda Visual (VisualSearch)	Esta herramienta, licenciada por la Empresa Catchoom Technologies, ofrece la posibilidad de conectar aplicaciones móviles con los servicios en la nube de CraftAR.
DAQRI	Libre, Comercial SDK	Android, iOS	Búsqueda visual, ContentAPI y características naturales (NaturalFeature)	DAQRI es una plataforma RA basada en visión que ofrece soluciones de visualización e interactivas en 4D apoyadas por un sistema de datos en la nube.
Designers ARToolkit (DART)	Libre pero con el código fuente cerrado	OSX, Windows	Marcadores, ContentAPI y TrackerInterface	DART es un conjunto de herramientas de software que permite diseñar e implementar aplicaciones y experiencias de RA de forma rápida.

IN2AR	Libre, Comercial SDK	Flash, iOS, Android	Características naturales (NaturalFeature)	Imágenes usando webcams y cámaras del móvil. La información de posicionamiento se puede usar para incluir objetos 3D o vídeos sobre la imagen y crear de esta forma aplicaciones o juegos de RA controlados por movimiento.
Instant Reality	Libre, Comercial SDK	Android, iOS, Linux, OSX, Windows	Marcadores, NaturalFeature, GPS, Sensores IMU, Tracking facial, VisualSearch, ContentAPI, SLAM, TrackerInterface	Es un framework para sistemas de realidad mixta que presenta interfaces para que los desarrolladores accedan a unos componentes y puedan realizar aplicaciones de RA/RV. Este sistema ha sido desarrollado por Fraunhofer IGD y ZGDV en cooperación con otros socios industriales.
Koozyt	Comercial SDK	Android, iOS	Marcadores	Fundada por miembros de los laboratorios de Sony Computer Science que desarrollaron la tecnología "PlaceEngine" en julio de 2007. Esta tecnología conecta el mundo real y el virtual poniendo el énfasis en el comportamiento humano.
Layar	Libre, Comercial SDK	Android, iOS	NaturalFeature, GPS, Sensores IMU, VisualSearch, ContentAPI	Layar permite a publicadores, anunciantes y marcas crear folletos, tarjetas con contenidos interactivos de RA sin necesidad de hacer desarrollos o instalar software.
Mixare	Open Source	Android, iOS	GPS	Mixare (mix AR Engine) es un browser de RA libre y de código abierto que está disponible para Android e iPhone.
OpenSpace3D	Open Source	Linux, Windows	Marcadores	OpenSpace3D es una plataforma de código abierto para desarrollar proyectos de RA y RV. Su objetivo es democratizar las aplicaciones 3D en tiempo real y proporcionar herramientas para creativos.
Rox Odometry SDK	Libre, Comercial SDK	Android, iOS, Linux, OSX, Windows	Marcadores, NaturalFeature	Permite construir aplicaciones identificando con la cámara del dispositivo objetos pre-grabados y obtener de forma exacta su posición y orientación relativa en tiempo real.
SSTT	Código fuente cerrado	Android, iOS, Windows Mobile, Linux, OSX, Windows	Marcadores, NaturalFeature	SSTT Bounce es un browser de RA que usa técnicas de tracking basadas en características naturales de la imagen.
Total Immersion	Libre, Comercial SDK	Android, iOS, Windows, Flash	Marcadores, NaturalFeature, Tracking facial	Total Immersion ofrece una plataforma comercial de RA que integra gráficos 3D interactivos en tiempo real dentro del flujo de vídeo en vivo.
UART	Open Source	iOS, OSX, Windows	Marcadores	Unity AR Toolkit (UART) es un set de plugins para el motor Unity que permite a los usuarios desarrollar y desplegar aplicaciones de RA.
Vuforia	Libre, Comercial SDK	Android, iOS	Marcadores, NaturalFeature, VisualSearch	Vuforia es una plataforma de software que permite desarrollar aplicaciones de RA para móviles y tabletas.
Wikitude	Libre, Comercial SDK	Android, iOS, BlackBerry OS	GPS, Sensores IMU, ContentAPI	Wikitude es una solución completa de RA que incluye reconocimiento de imágenes, tracking, renderización de modelos 3D, etc.
Win AR	Libre, Comercial SDK	Windows	NaturalFeature	WinAR es una plataforma de desarrollo basada en Windows para realizar aplicaciones de RA. Proporciona procesamiento de imágenes, algoritmos de seguimiento y de registro.
Yvision	Libre, Comercial SDK	Android, iOS, Windows Mobile, OSX, Windows	Marcadores	YVision es un framework que permite prototipado rápido y desarrollo de aplicaciones. Integra visión por computador, renderizado en tiempo real, simulación de físicas, RA, inteligencia artificial, etc.
Zapcode Creator	Comercial SDK	Android, iOS	Marcadores	Zapcode Creator son herramientas de creación de contenidos para crear experiencias de RA.

Figura 4. Tecnologías de Realidad Aumentada en la actualidad

Profundizando en las aplicaciones de realidad aumentada aplicadas en la industria que es la base de interés de este proyecto se observan las principales ventajas que se extraen del uso de la realidad aumentada como son la inclusión de datos digitales sobre los componentes de gran utilidad para operarios menos expertos, guiado interactivo de las funcionalidades, adición de comentarios y etiquetas sobre un modelo real, y la posibilidad de gestión en tiempo real sobre las tareas que se han ido realizando.

Capítulo 4. DEFINICIÓN DEL TRABAJO

4.1 MOTIVACIÓN DEL PROYECTO

La motivación que se extrae de este proyecto es la de mezclar lo mejor de la Industria 4.0 y la realidad aumentada por lo que pretende llevar a la realidad todo el potencial de ambas tecnologías con el fin de mejorar la eficiencia de las plantas industriales y facilitar el trabajo a los trabajadores que están encargados de supervisarlas.

Este proyecto debe ser una base para futuros desarrollos en la industria comenzando desde el pequeño laboratorio de ICAI pero siendo expandible rápidamente a cualquier tipo de industria que cuente con las tecnologías adecuadas.

4.2 OBJETIVOS

El objetivo de este proyecto es el desarrollo de una aplicación o prototipo inicial para poder generar una visualización clara de las propiedades de la minifábrica ICAI de una manera sencilla y rápida. El proyecto está dividido en dos proyectos independientes que finalmente forman el objetivo principal. Estos dos proyectos son la conexión de las variables en tiempo real desde la minifábrica ICAI a una base de datos en la nube y la creación de una aplicación que sea capaz de visualizar los objetos a identificar en el laboratorio y nos muestre por pantalla los datos que pertenecen a ese objeto y están subidos a esa base de datos. Cabe destacar que la base de datos constará de los datos que se actualizan en tiempo real y los datos estáticos de cada objeto provenientes del datasheet los cuales serán elegidos dependiendo de las necesidades del usuario o incluso en caso de ser necesario accediendo directamente al datasheet completo del componente.

4.3 METODOLOGÍA

La metodología llevada a cabo en este proyecto ha sido la que se muestra a continuación en el diagrama temporal con un margen de 1 mes hasta la entrega final de este proyecto en Julio de 2019.



Figura 5. Planificación del Proyecto de Fin de Máster

Capítulo 5. IMPLEMENTACIÓN DE LA SOLUCIÓN

Al encontrarnos ante un proyecto integrador de varios proyectos subyacentes será necesario explicar cada pequeña fase de manera independiente para posteriormente poder entender las conexiones entre estas partes y como quedaría el proyecto final.

Como es evidente por lo explicado anteriormente es muy importante ser conscientes de que este proyecto tiene como objetivo integrar el mundo físico, que es el que percibimos con nuestros sentidos con el mundo virtual el cual se encuentra oculto y tiene un acceso más lento y complejo.

En cuanto a la minifábrica de ICAI y su posterior conexión con el Cloud deberemos conocer cómo desarrollar programas en TIA PORTAL para la programación de los PLCs y además será necesario tener conocimientos del protocolo OPC UA de manera que se puedan establecer conexiones utilizando dicho protocolo con la base de datos alojada en IBM Cloud.

Para ello deberemos conocer cómo se debe desarrollar una aplicación en NODE-RED para poder recibir toda la información que transmiten los PLCs y hacerla interpretable por la Nube. Por sencillez este desarrollo se realizó sobre la plataforma Bluemix por lo que fue necesario entender los servicios que esta plataforma presta y como poder crear relaciones entre ellos. La potencia de este proyecto no solo reside en el prototipo final creado, sino que también está en la facilidad que podrá tener el personal industrial al que se le presuponen mayores conocimientos para poder cambiar la aplicación y hacerla evolucionar consiguiendo todo su potencial tanto en términos de eficiencia como de escalabilidad.

Además de todo lo anterior la base del producto desarrollado erradica en Unity y Vuforia siendo ambos programas la base para la implementación de la realidad aumentada que se buscaba en este proyecto y demás siendo capaces de hacer que esta aplicación ofrezca la información real-time que pretende desde un principio mediante la realización de peticiones

http como se ve en la siguiente imagen que representa la arquitectura sin entrar en profundidad.

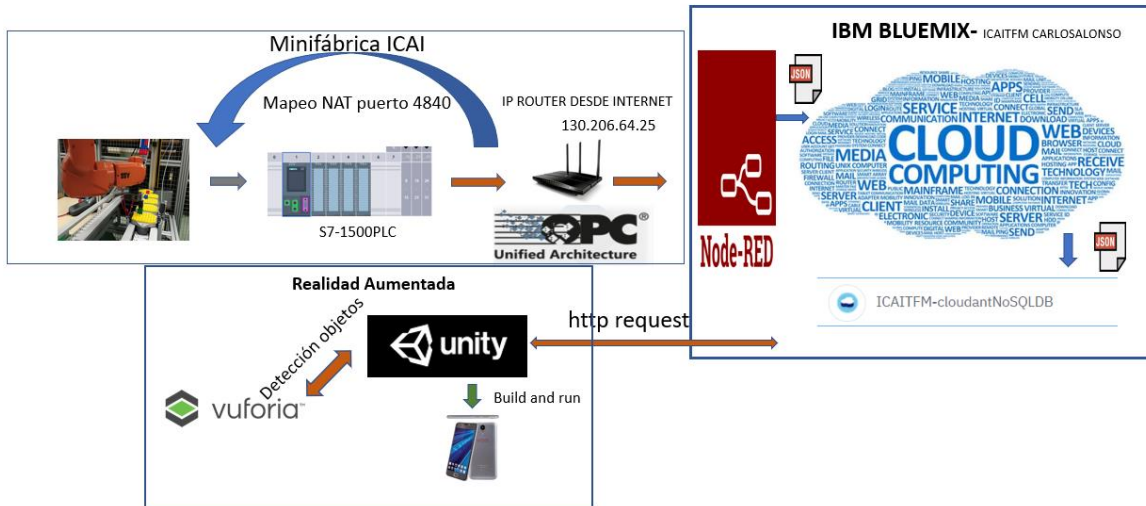


Figura 6. Arquitectura del Proyecto Implementado

A continuación, se muestra la imagen de la planta del laboratorio que se pretende monitorizar y extraer la información mediante la solución implementada.



Figura 7. Minifábrica ICAI

5.1 INFRAESTRUCTURA DE LA MINIFÁBRICA

Es importante reseñar que en esta parte del desarrollo lo que se implementa es el programa en TIA Portal para conseguir que a través del protocolo OPC UA se puedan comunicar los dispositivos con la nube para así poder almacenar las variables que posteriormente serán recogidas por nuestra aplicación a través de las pertinentes peticiones como hemos visto anteriormente. Para observar la exportación de variables a la nube no es necesario generar un programa complejo en TIA Portal puesto que el objetivo de este proyecto no tiene nada que ver con observar el comportamiento de un programa determinada sino con ser capaz de monitorizar las variables que se quiera desde la nube y como consecuencia desde nuestra aplicación de realidad aumentada.

Es por ese motivo que lo único que haremos de manera inicial es conectar una variable a un interruptor del laboratorio para poder observar como dependiendo de cómo este el interruptor en nuestra aplicación se verá uno u otro resultado. Si se consigue esto el desarrollo para el resto de las variables será exactamente igual cambiando la dirección de acceso que debe tomar OPC UA para acceder a la variable en concreto.

Lo primero que debemos hacer es crear la variable y activar el servidor OPC UA en nuestro programa además de hacer la variable visible para el mismo de tal manera que cada vez que esta variable sufra un cambio, todos los clientes que estén suscritos al servidor OPC UA recibirán un mensaje con el cambio para poder actualizar el estado de dicha variable.

Para ello ayudándonos de los conocimientos adquiridos en la asignatura Aplicaciones Industriales de las Comunicaciones creamos un proyecto en TIA Portal asignando el PLC presente en el laboratorio como podemos observar en la siguiente imagen.



Figura 8. Componente PLC Tia Portal

En este proyecto lo que haremos será poner a funcionar el servidor OPC UA y haciendo la variable1 que será nuestra variable de test visible a dicho protocolo como veremos en las siguientes imágenes.

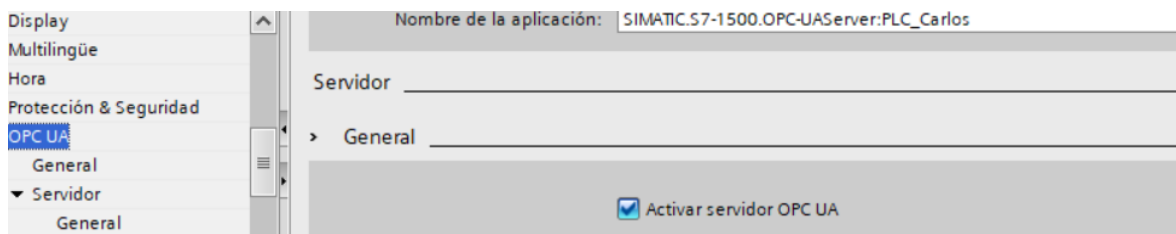


Figura 9. Activación servidor OPC UA

Variables PLC						
	Nombre	Tabla de variables	Tipo de datos	Dirección	Rema...	Accesible desde HMI/OPC UA
1	variable1	Tabla de variables e..	Bool	%I.2	<input type="checkbox"/>	<input checked="" type="checkbox"/>

Figura 10. Asociación Variable a entrada digital

Esta variable será de tipo Booleana por lo tanto lo que esperaremos recibir en nuestra base de datos será un True o un False pero eso es algo que veremos más adelante. Es una entrada digital con dos únicas posiciones como se señala en la siguiente imagen.

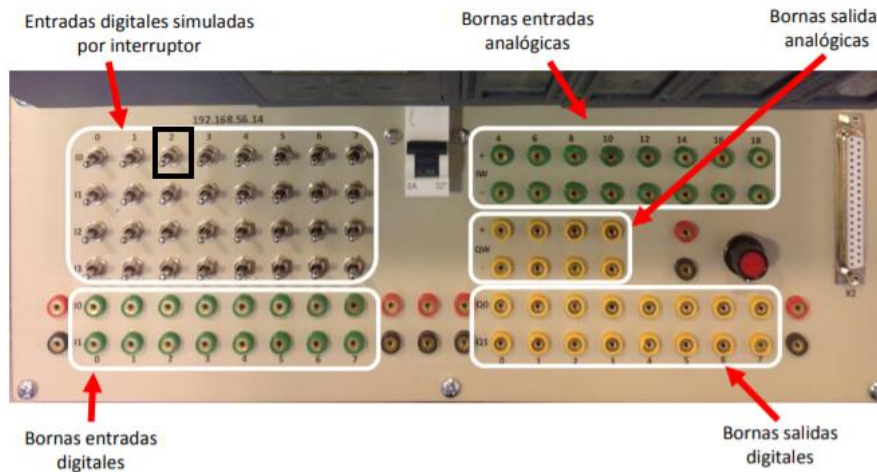


Figura 11. Entrada digital asociada a variable 1

Esta clase de protocolo puede estar orientado y es lo más normal hoy en día al traspaso de información y su posterior monitorización dentro del entorno industrial sin embargo en este caso concreto el PLC que está controlando nuestro programa debe estar conectado a Internet para que desde Bluemix se puedan recibir los datos ya que el cliente deberá estar allí alojado.

Para este apartado será fundamental el enrutamiento para poder realizar la conexión desde Bluemix al laboratorio, concretamente al PLC que tiene acceso a internet.

Como es lógico el acceso a internet se crea mediante un router el cual tendrá una red interna como la que se puede observar en un domicilio particular cuando hay conectados una serie de dispositivos diferentes al mismo router. Lo que se deduce de esta es que existen dos redes cuyo punto de separación es el propio router y estas son la red interna y la red externa. Para poder realizar una aplicación como la que estamos implementando será necesario el conocimiento de ambas redes por poder direccionar la información correctamente entre la red física (los PLCs del laboratorio) y el cloud.

En la instalación que nos encontramos cada PLC posee una IP Fija dentro de la red interna del router de la universidad por lo que el siguiente paso es poder asignar a través de NAT para mapear los puertos del PLC con los puertos del router. Otro detalle importante es la necesidad de que el router tenga en Internet una dirección IP fija como es afortunadamente en el caso de la universidad ya que si no habría que contratarla o en su lugar un servidor DNS dinámico para poder reflejar las actualizaciones en la IP con el sobrecoste que esto significa.

Para el caso que nos atañe se observa perfectamente como el PLC del laboratorio que está conectado a internet tiene una IP local ya que en caso de no ser así sería imposible poder usar el protocolo OPC UA que no permite otro tipo de configuraciones.

A continuación, se podrán observar los distintos caminos y enrutamiento que sufrirá la información para poder conectarse desde el PLC 11 presente en el laboratorio hasta nuestra aplicación ejecutándose en Bluemix. Es importante observar cómo además de las IP fijas se mapearán los puertos del PLC con el router a través de NAT para hacerlo accesible desde internet.

En la imagen que vemos a continuación se ve claramente el puerto del PLC por el que se debe enrutar la conexión.

Accesibilidad del servidor	
Direcciones de servidor	
Dirección	
opc.tcp://192.168.56.11:4840	
opc.tcp://172.20.0.20:4840	

Figura 12. Direcciones del servidor OPC UA

Y por supuesto aquí se reflejará la arquitectura de comunicaciones para poder realizar la conexión entre ambos medios.

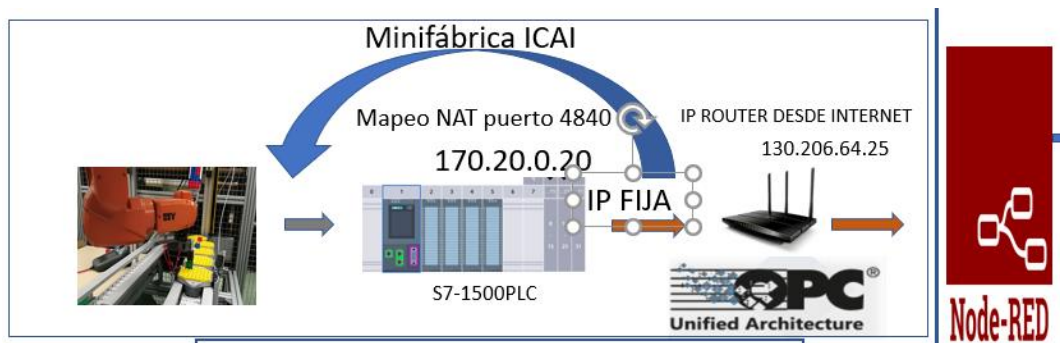


Figura 13. Enrutamiento de los datos desde los sensores a Bluemix

5.2 IMPLEMENTACIÓN DE LA CONEXIÓN CON IBM

En esta parte del desarrollo se creará una aplicación en Bluemix a través de NODE RED para conectarse a nuestro servidor OPC UA. A continuación, se creará una conexión con el servicio de almacenamiento de IBM para poder guardar el histórico de cambios que ha sufrido la variable además de la información estática que estará asociada a cada componente.

El servicio en concreto con el que estará conectada nuestra aplicación en NODE RED será el servicio Cloudant NoSql y si se pretende realizar una monitorización online sobre la evolución de las variables se deberá conectar también con el servicio Watson Internet of Things.

Lo primero es familiarizarse con el entorno Bluemix para entender cómo se realizan este tipo de conexiones como las que estamos describiendo anteriormente. Bluemix tiene un catalogo de servicios a través del cual debemos aquellos servicios que nos interesen para nuestro desarrollo que en este caso simplemente queremos que sean Node Red que será donde se ejecute nuestra aplicación y IBM Cloudant NoSql que será donde se almacene el valor de nuestras variables.

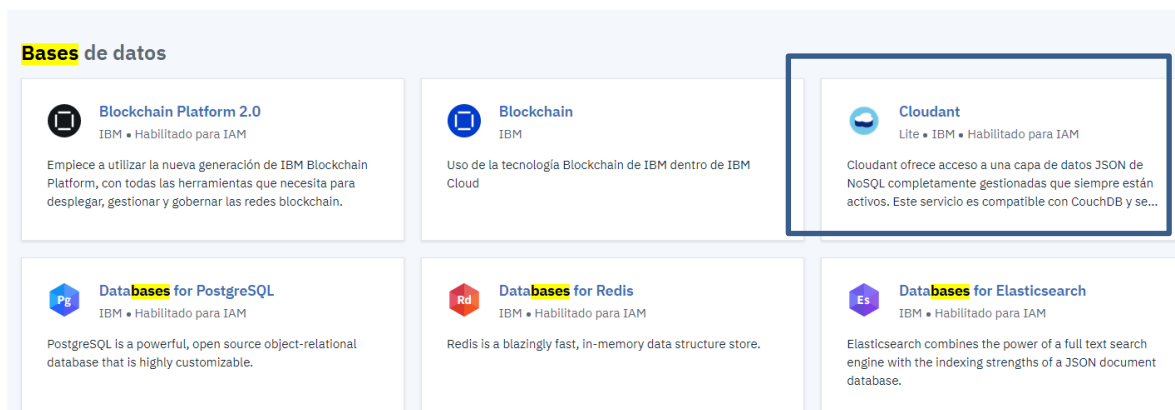


Figura 14. Recursos de almacenamiento Bluemix

Tras crear la aplicación seleccionando Node Red como entorno de desarrollo accediendo al panel tenemos el control sobre nuestra aplicación y los servicios que esta tiene asignados.

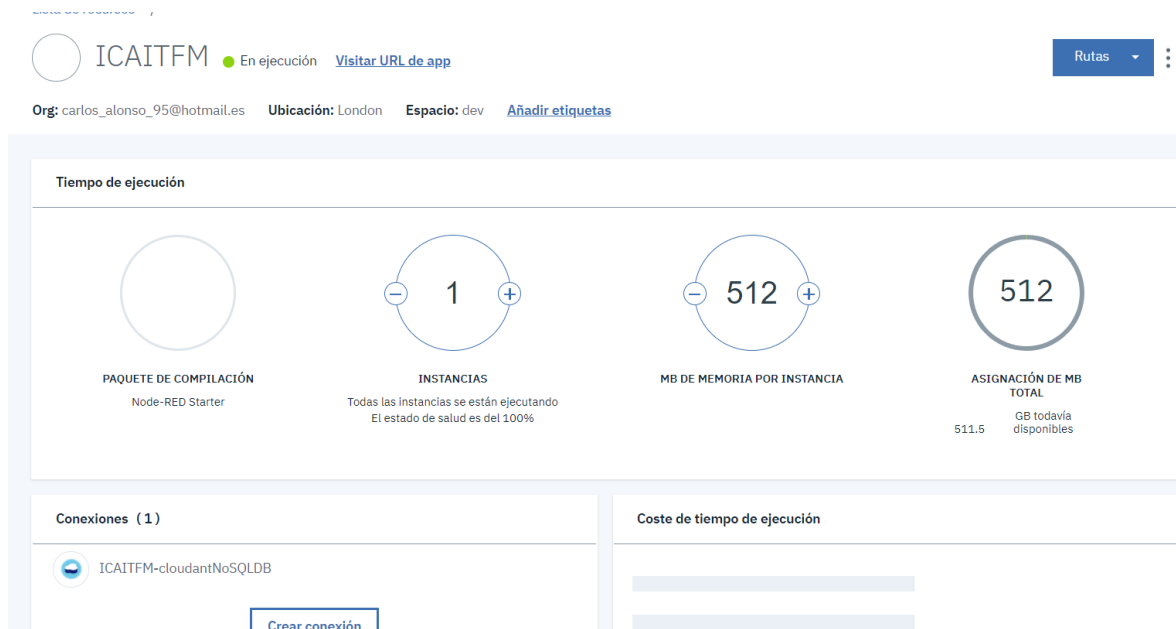


Figura 15. Panel de control de la aplicación en Bluemix

5.2.1 DESARROLLO NODE RED

Lo primero que debemos hacer para poder desarrollar una conexión OPC UA con Node red es descargar las correspondientes librerías que OPC UA tiene para posibilitar la conexión entre Node y los PLCs. Esta librería es publica creada por los diferentes desarrolladores para poder generar un cliente OPC UA que está conectado y por tanto suscrito a todos los cambios que se produzcan en las variables. Cada vez que haya un cambio el servidor mandará el nuevo valor tomado por la variable al cliente suscrito.

El diseño de la aplicación en NODE RED se basa en la implementación de flujos. Al tratarse de variables diferentes que están asociadas a diferentes componentes se organizaran en flujos diferentes que enviarán los datos a diferentes bases de datos como se explicó en la parte de desarrollo de la aplicación en Unity de manera que se facilite lo máximo posible el acceso a través de nuestra aplicación.

En este caso se procederá a presentar el ejemplo para un Flow, es decir para la conexión de una sola variable ya que en caso de querer monitorizar más habrá simplemente que repetir el proceso.

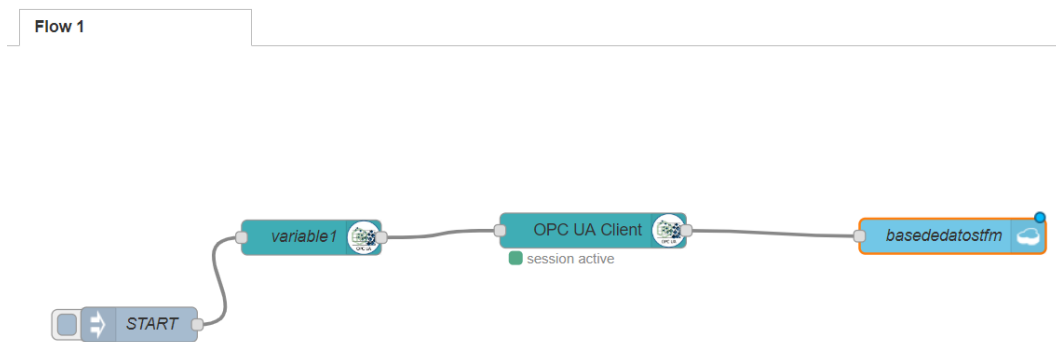


Figura 16. Flujo conexión sensor-Node Red

Los colores de los diferentes nodos que podemos observar en este Flow no son aleatorios por lo que intrínsecamente se les deduce un significado. Un botón Start grisáceo, los nodos verdes que son aquellos que pertenecen al protocolo OPC UA el nodo azul claro que pertenece a los nodos de almacenamiento en este caso la base de datos Cloudant NoSql.

El nodo Start como es lógico es el que inicia la conexión, en este caso la subscripción de la aplicación al servidor OPC UA. Al pulsar sobre él a aplicación comienza a recibir datos del servidor.

El siguiente paso será insertar el ítem al que se quiere subscribir la aplicación que será de la que dependan los mensajes que vayan llegando a nuestro cliente como por ejemplo la variable 1.

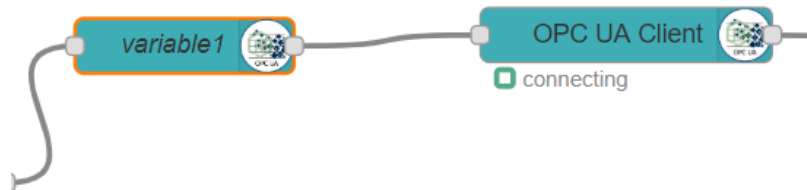


Figura 17. Componente OPC UA en Node Red

La variable1 es el ítem que se inserta al cliente OPC. Para configurar la conexión no es necesario solo nombrar a la variable correctamente, sino que lo fundamental es saber su ID dentro del servidor OPC UA para poder acceder a ella correctamente.

La imagen muestra una ventana de configuración titulada 'Edit OpcUa-Item node'. En la parte superior hay tres botones: 'Delete', 'Cancel' y 'Done'. Debajo, se encuentra una sección 'Properties' con los siguientes campos:

- Item:** ns=3;s="variable1"
- Type:** Boolean (con un menú desplegable)
- Value:** (campo de texto vacío)
- Name:** variable1

Figura 18. Dirección de la variable en el servidor OPC

Para conocer su ID lo que se hace es utilizar una DEMO que nos proporciona OPC para poder saber cómo se distribuyen las variables dentro del servidor. El link a esto demo estará en la bibliografía, pero aquí se observa cómo se obtendría esta ID.

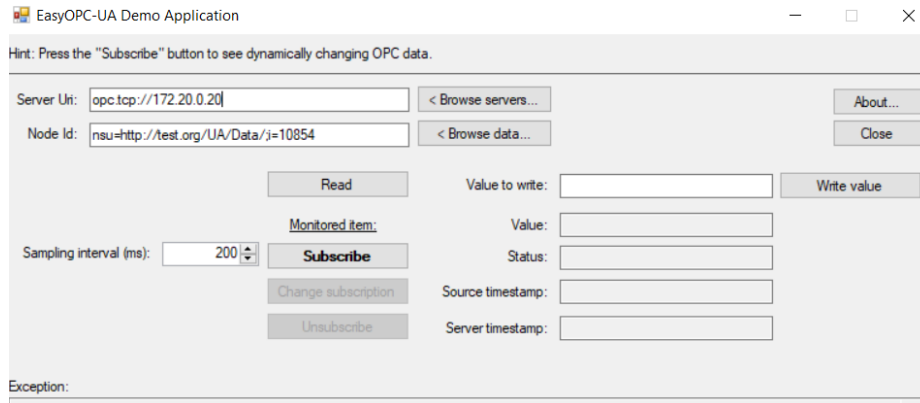


Figura 19. Demo OPC UA para acceder a la ID de las variables

A través de esta interfaz conociendo la dirección correcta se accede al servidor que maneja nuestro PLC y así se podrán ver todas las variables que hay almacenadas en él. Solo será necesario identificar la variable 1 que en este caso es ns=3.

A continuación, se ve como se configura el cliente para comprobar el valor de la variable 1 cada 500 ms. En este caso lo que se ve como endpoint es la dirección del router del PLC vista desde Internet acompañado del puerto de comunicaciones que veríamos anteriormente 4840.

The screenshot shows a configuration window titled 'Properties' with a gear icon and a document icon. It contains several fields for configuring an OPC UA client:

- Endpoint:** A dropdown menu showing 'opc.tcp://130.206.64.25:4840' with a pencil icon to its right.
- Action:** A dropdown menu showing 'SUBSCRIBE'.
- Interval:** A text input field with '500' and a dropdown menu with 'millisecond(s)'.
- Certificate:** A dropdown menu showing 'None, no certificate used'.
- Local file:** A text input field containing '../pki/server_certificate.pem'.
- Name:** An empty text input field.

Figura 20. Creación del cliente OPC UA

Y finalmente el ultimo nodo señala a que base de datos se va a conectar nuestro programa y la operación que se quiere ejecutar que en este caso es el Insert. Como es lógico la base de datos a la que debemos se puede llamar de cualquier manera, pero el servicio que se va a usar es el que anteriormente lamamos ICAITFM-cloudantNoSQLDB.

The screenshot shows a configuration window titled 'Properties' with a gear icon and a document icon. It contains several fields for configuring a database connection:

- Service:** A dropdown menu showing 'ICAITFM-cloudantNoSQLDB'.
- Database:** A text input field containing 'tfm'.
- Operation:** A dropdown menu showing 'insert'.
- Only store msg.payload object?:** A checkbox that is currently unchecked.
- Name:** A text input field containing 'basededatosfm'.

Figura 21. Punto de acceso a la DB

5.2.2 ALMACENAMIENTO EN CLOUDANT NOSQL DB

El servicio del que nos vamos a ayudar para almacenar toda esta información en tiempo real es una base de datos no relacional. La aplicación Node Red generará tantas bases de datos como variables queramos almacenar lo que será muy útil para luego acceder a ellas a través de nuestra aplicación en Unity.

Esta base de datos generada puede ser personalizada por el consumidor sin ningún problema mediante la generación de documentos de diseño.

Como se puede observar se generó la base de datos TFM como habíamos ordenado desde nuestra aplicación en Node Red con diferentes documentos cada uno con el valor de la variable 1 que se generó en cada momento y con el timestamp en el que fue generado.

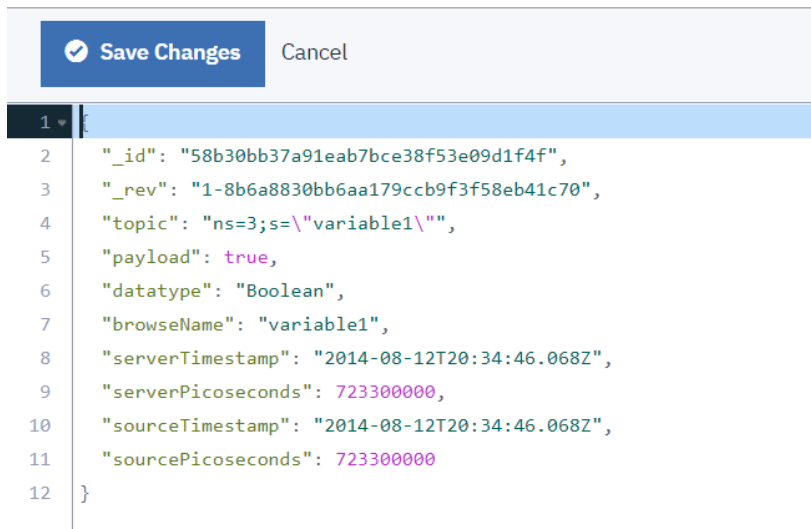


id	key	value
58b30bb37a91eab7bce38f53e09d1f4f	58b30bb37a91eab7bce38f53e09d1f4f	{ "rev": "1-8b6a8830bb6aa179ccb9f3f58"
7f2da468b9fe4a43600af21944a67ace	7f2da468b9fe4a43600af21944a67ace	{ "rev": "1-2e9e29cc9c10dbd77740b1b0"
f948a37f5a1badf27953e736ccc7b787	f948a37f5a1badf27953e736ccc7b787	{ "rev": "1-154bf9142e8f2cdca3b2d0f1b"

Figura 22. Base de datos dinámico-TFM

Y como se puede observar según fuimos variando el interruptor asociado a nuestra variable se generaron diferentes archivos Json con sus características en las destacan una ID diferencial además de un valor de la variable 1 monitorizado en tiempo real.

tfm > 58b30bb37a91eab7bce38f53e09d1f4f



```
1 {  
2   "_id": "58b30bb37a91eab7bce38f53e09d1f4f",  
3   "_rev": "1-8b6a8830bb6aa179ccb9f3f58eb41c70",  
4   "topic": "ns=3;s=\\\"variable1\\\"",  
5   "payload": true,  
6   "datatype": "Boolean",  
7   "browseName": "variable1",  
8   "serverTimestamp": "2014-08-12T20:34:46.068Z",  
9   "serverPicoseconds": 72330000,  
10  "sourceTimestamp": "2014-08-12T20:34:46.068Z",  
11  "sourcePicoseconds": 72330000  
12 }
```

Figura 23. Json ejemplo de datos dinámicos

Desde la propia base de datos se pueden realizar queries y muchas más cosas como una parte analítica y de monitorización del histórico que puede ser muy potente y aún más si se combina con otros servicios como Watson, pero en este caso lo que nos interesaba simplemente era conseguir la transferencia de datos entre el mundo físico y el mundo cloud y como aquí se puede resumir es algo que se ha conseguido.

5.3 CREACIÓN DE APLICACIÓN DE REALIDAD AUMENTADA

En este apartado se va a explicar cómo se desarrolló la aplicación en Unity y Vuforia para obtener el prototipo final el cual es capaz de detectar componentes del laboratorio gracias a una pegatina visual que tienen sobre ellos y acceder a una base de datos alojada en cloud para poder mostrar por pantalla los datos asociados a ese componente.

Primero es importante destacar algunas de las ventajas que nos proporciona Unity3D por las cuales decidimos usarlo como son su potencia para entornos en 3D, la sencilla interfaz que ofrece para la creación de escenarios sencillos como el que atañe a este proyecto. Además de todo esto es importante señalar la cantidad de assets que hay en la tienda unity muchos de ellos gratuitos con los cuales se pueden añadir funcionalidades concretas o incluso proyectos concretos que se quieran integrar.

Además, permite rápidamente compilar el proyecto diseñado para la plataforma que elijamos sin tener que hacer nada especial en el código. Para este caso concreto el entorno utilizado será Android para poder visualizarlo desde un móvil cualquiera o una Tablet que son los dispositivos que más frecuentemente se utilizan en el mundo real.

5.3.1 DESARROLLO DETECCIÓN CON VUFORIA

Lo primero es presentar el entorno de desarrollo de la aplicación que como habíamos dicho es muy visual y se puede diseñar las diferentes interfaces que queremos visualizar por parte de los usuarios y las imágenes que serán utilizadas para la detección.

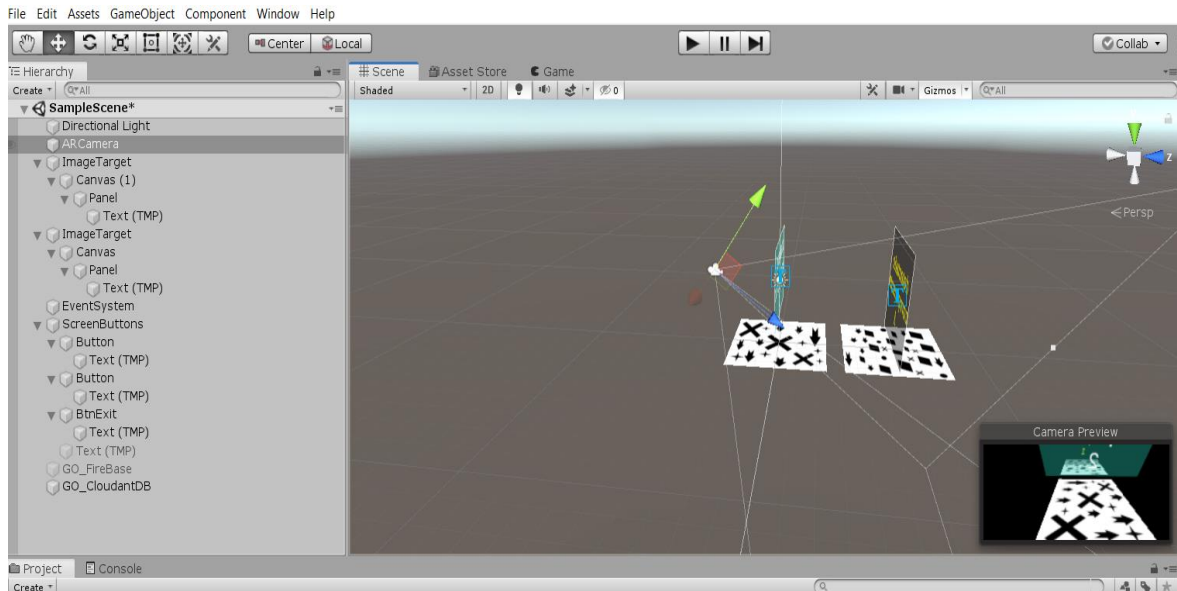


Figura 24. Entorno Unity

En la imagen anterior se pueden apreciar los diferentes componentes que componen el desarrollo del escenario sin entrar aun en la parte asociada a la programación que definirá el backend de nuestra aplicación.

Lo primero que debemos elegir son las imágenes que nuestra aplicación va a tener que detectar e interpretar para poder asociar a ellas la información del objeto deseado. Esta parte viene definida por la aplicación Vuforia que será la librería encargada de poder realizar esta detección de los datos. Esta tecnología fue explicada en la parte de descripción de tecnologías y aquí lo que haremos será explicar cómo se utilizó en nuestro proyecto.

Lo primero que debemos saber es que Vuforia tienen un algoritmo interno el cual es capaz de identificar estas imágenes ya que se conecta con las imágenes de la base de datos a la que esta enlazada la aplicación.

TFMDEFINITIVO [Edit Name](#)
 Type: Device

Targets (2)

[Add Target](#) [Download Database \(All\)](#)

Target Name	Type	Rating	Status	Date Modified
<input type="checkbox"/> IMAGEN2	Single Image	★★★★★	Active	May 15, 2019 15:36
<input type="checkbox"/> IMAGEN1	Single Image	★★★★★	Active	May 15, 2019 15:28

Figura 25. Base de datos Vuforia

En la imagen anterior se observan dos imágenes o targets que nuestra aplicación deberá ser capaz de detectar. Como se observa la base de datos es muy fácilmente escalable por lo que se pueden añadir tantas imágenes como se quiera dependiendo de los componentes del laboratorio que se quieran identificar.

Como se puede observar cada imagen viene con un rating asociado a ella a través del cual el algoritmo nos está dando información de que targets van a ser más fácilmente detectables y cuáles no. Esta característica de la base de datos es muy útil ya que en condiciones de ruido como puede ser un entorno industrial será importante utilizar targets lo más resistentes a los efectos negativos provocados por este. En esta ocasión se eligieron targets claramente diferenciales, negro sobre blanco con siluetas muy diferenciales; característica que venía recomendada de antemano por la propia herramienta Vuforia.

Siguiendo con Vuforia será importante importar las pertinentes librerías en Unity para dotar a la aplicación del potencial ofrecido por la realidad aumentada. En nuestro caso para el caso de uso de dos targets después de cargar todo el SDK Vuforia llegó el momento de posicionar la cámara en el punto pertinente para que pudiera captar de la mejor manera posible los objetivos y de alguna manera fuese más fácil la detección imaginando las situaciones normales de enfoque.

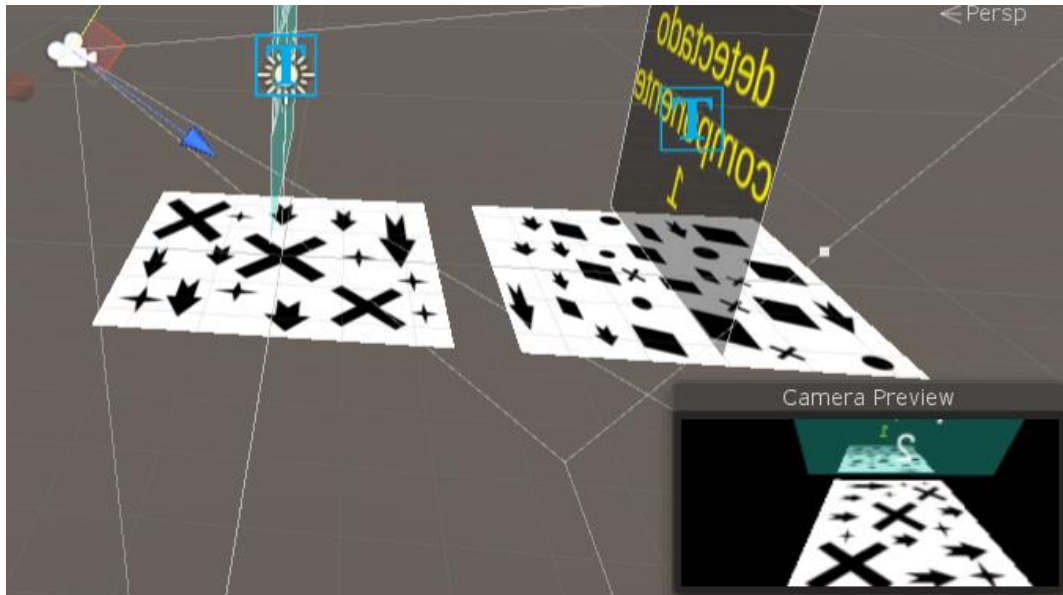


Figura 26. Implementación escenario

Como se puede observar en la anterior imagen existe un recuadro llamado cámara preview que simula lo que vería la cámara de nuestro dispositivo en un caso de uso real por tanto como se puede observar se eligió un angular correcto para poder ver sendos targets sin ninguna dificultad por parte de la cámara. Pero primero como es lógico vamos a introducir la tecnología de Vuforia y porque vamos a usarla en esta aplicación de realidad aumentada.

Una parte importante para poder realizar estos pasos sin problema será enlazar la aplicación de Unity con el software de vuforia añadiendo la licencia correspondiente al propietario de las bases de datos.

5.3.2 DESARROLLO FRONT END

Para saber que Vuforia está funcionando bien independientemente de cómo lo esté haciendo Unity se añaden unos paneles de texto Canvas sobre los targets para que cuando la cámara detecte sendos objetivos muestre por pantalla algún mensaje de que está detectando el componente que debe detectar. Esto, aunque redundante, es importante para saber si el desarrollo falla debido a la programación en Unity o simplemente no está detectando por algún error de configuración del escenario mediante Vuforia.



Figura 27. Interfaz de Usuario-detección

Sobre el escenario se diseñará la interfaz de usuario a través de la cual se seleccionará la tipología de datos que se quiere observar. Unity como se ha comentado en este proyecto nos da una interfaz muy sencilla por la cual podemos añadir botones cuyas funcionalidades serán programadas más adelante que persistirán constantes cuando el usuario inicie su aplicación.

Al ser esta una aplicación enfocada a un uso sencillo, aunque con un backend grande por detrás se enfocó la interfaz de la manera más sencilla posible para agilizar el proceso de interacción con los componentes que es lo que realmente se busca.

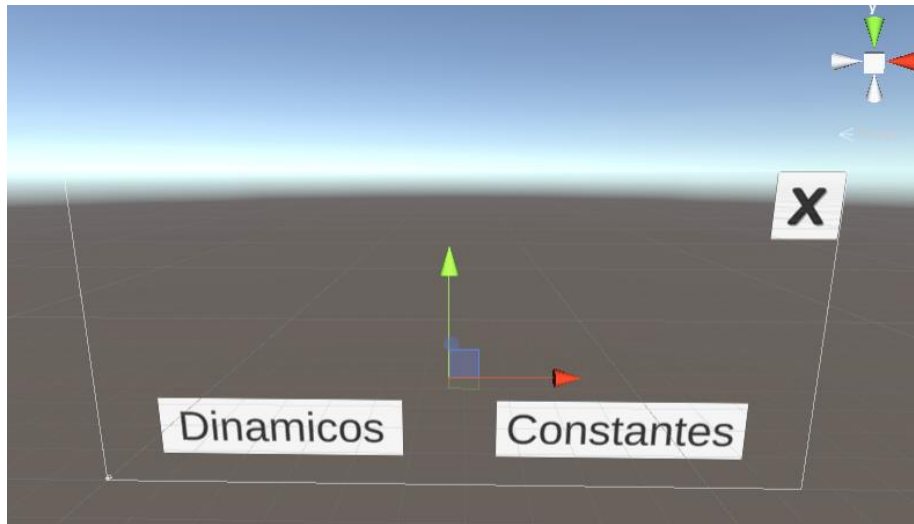


Figura 28. Interfaz de Usuario

5.3.3 IMPLENTACIÓN DE LAS FUNCIONALIDADES CON UNITY

En este apartado se desarrollarán las diferentes funcionalidades que queremos para nuestra aplicación intentado implementarlas de la manera más eficiente posible. El lenguaje de programación utilizado por Unity es #C y el soporte utilizado para configurar el código es Microsoft Visual Studio.

En este primer script de programación se definen las funcionalidades asociadas a los tres botones de la interfaz de usuario proporcionada por nuestra aplicación. Lo interesante aquí es observar como cada imagen tiene asociada un texto TextMeshProGUI como veremos en la siguiente imagen que cuando es detectada se pasa cuando pulsamos el botón como argumento para poder llamar a la función encargada de conectarse con la base de datos en Bluemix.

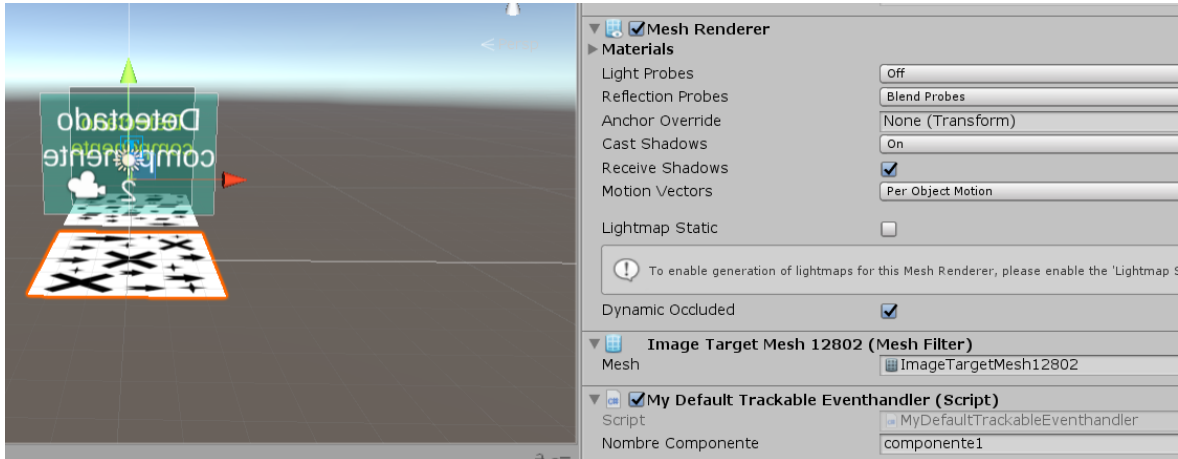


Figura 29. Nombre asociado al target

Como es lógico cada botón tendrá asociada una función diferente ya que los datos que se deben recoger de manera dinámica no tienen nada que ver con los datos que van a estar alojados de manera estática por lo que la implementación de las funciones debe ser diferente.

```
using TMPro;
using UnityEngine;

public class MenuOptions : MonoBehaviour
{
    //public Manager_DB Database;
    public CloudantDBclient BlueMix;

    //Parametros de la imagen activada por la camara
    public TextMeshProUGUI texto;
    public string nombrecomponente = "";

    public void exitapp()
    {
        Application.Quit();
    }

    public void PressButtonDinamicos()
    {
        if (texto != null)
        {

```

```

        BlueMix.GetComponent<CloudantDBclient>().Get_Dinamicos_Datos(texto);
    }
}

public void PressButtonConstantes()
{
    if (texto != null)
    {

```

```

        BlueMix.GetComponent<CloudantDBclient>().Get_Estaticos_Datos(texto,nombrecomponente)
;
    }
}
}

```

Como vemos este script cuando se pulsa alguno de los botones llama a la función get.datos ya sean dinámicos o estáticos para que esta de alguna manera consiga acceder a la base de datos donde están alojados los datos y sea capaz de coger solo los datos asociados al componente que estamos detectando.

```

public void Get_Estaticos_Datos(TextMeshProUGUI texto, string componente)
{
    //base request builder for all requests containing user name and database
name
    handler = new HttpClientHandler { Credentials = new NetworkCredential(user,
password) };

    using (var client = CreateHttpClient(handler, user, componente))
    {
        texto.text = GetLastOne_Estaticos(client, user, componente);
    }
}

/// <summary>
///
/// </summary>
/// <param name="texto"></param>
public bool Get_Dinamicos_Datos(TextMeshProUGUI texto)
{
    //base request builder for all requests containing user name and database
name
    handler = new HttpClientHandler { Credentials = new
NetworkCredential(user, password) };

    using (var client = CreateHttpClient(handler, user, DB_Dinamicos))
    {
        texto.text = GetLastOne_Dinamicos(client, user, DB_Dinamicos);
    }
}

```

```
    return true;  
}
```

En el script que podemos ver que es una pequeña parte del código, se identifican claramente las funciones a las que anteriormente estábamos llamando. Se crea un cliente http para poder hacer las peticiones contra la base de datos como es lógico, pero sin embargo se llama a otra función llamada `GetLastOne_Tipología` de datos que comentaremos a continuación en la cuál se verá claramente el sentido de que en el caso de dinámicos sea booleana. Es importante recalcar que las variables que se están pasando se han inicializado desde Unity, aunque podrían haberse hecho desde código, pero la ventaja que nos da estos es que cuando la aplicación se está ejecutando se puede cambiar en tiempo real el valor de las variables por si queremos acceder a otra información de la base de datos.

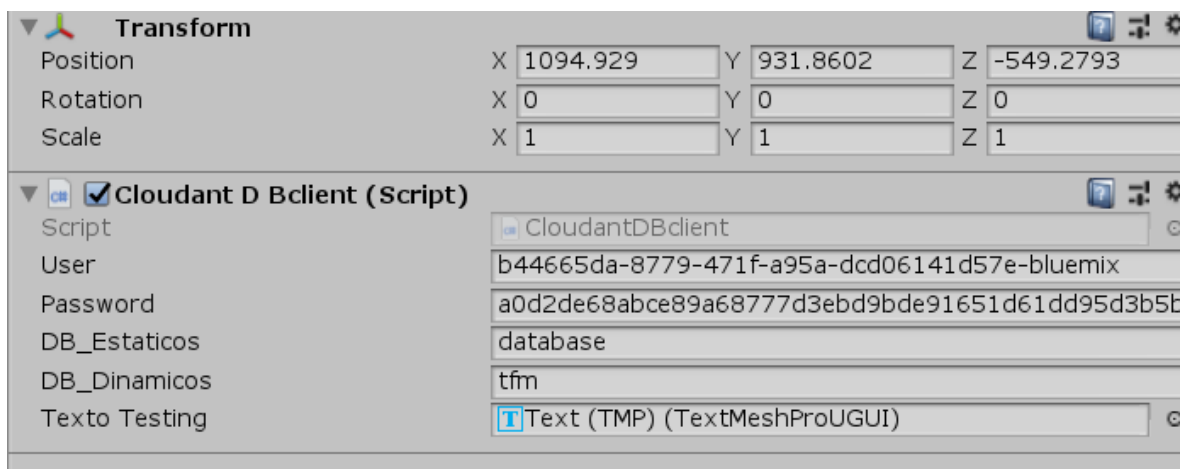


Figura 30. Inicialización de variables

Como se observa la parte dinámica va directamente a la base de datos `tfm` ya que para este trabajo con la monitorización de una única variable asociada a un interruptor, pero el sentido que tiene esto es poder asociar cada variable a un componente por lo que entonces lo que se pasaría como argumento sería lo mismo que estamos pasando en las llamadas estáticas pero con algún añadido para redireccionar la petición a la base de datos dinámica específica de

este componente como por ejemplo sería componente1_dinamicos haciendo un pequeño cambio en el código.

Lo más interesante del desarrollo es ver como se ha conseguido obtener la información que queremos de la variable ya que el Json que nos devuelve tras hacer la petición http es el siguiente:

```
1 {
2   "total_rows": 3,
3   "offset": 0,
4   "rows":
5   [
6     {
7       "id": "58b30bb37a91eab7bce38f53e09d1f4f",
8       "key": "58b30bb37a91eab7bce38f53e09d1f4f",
9       "value": { "rev": "1-8b6a8830bb6aa179ccb9f3f58eb41c70" },
10      "doc": {
11        "_id": "58b30bb37a91eab7bce38f53e09d1f4f",
12        "_rev": "1-8b6a8830bb6aa179ccb9f3f58eb41c70",
13        "topic": "ns=3;s=variable1",
14        "payload": true,
15        "datatype": "Boolean",
16        "browseName": "variable1",
17        "serverTimestamp": "2014-08-12T20:34:46.068Z",
18        "serverPicoseconds": 723300000,
19        "sourceTimestamp": "2014-08-12T20:34:46.068Z",
20        "sourcePicoseconds": 723300000
21      }
22    }
23  ]
24 }
```

Figura 31. Json recibido en Unity

Y lo que nos interesa mostrar por pantalla es a priori solamente el valor del payload de la variable (True/False) y quizás el tiempo en el que ha sucedido el cambio para observar si el valor de la variable es el correcto, pero no por el azar sino porque ha habido un suceso reciente que ha hecho que esto sea así. Para esto lo que se hace es obtener el Json y separarlo en filas cogiendo solo la parte doc que es la que nos interesa y dentro de la cual seleccionaremos los campos que deseamos mostrar por pantalla.

```
private static string GetLastOne_Dinamicos(HttpClient client, string user, string database)
{
```

```
using (var streamreader = new
StreamReader(client.GetStreamAsync(string.Format("https://{0}.cloudant.com/{1}/_a
ll_docs?include_docs=true", user, database)).Result))
{
    JToken LastDoc = null;
    DateTime ServerTimeStamp = new DateTime();

    var responseContent = (JsonObject)JToken.ReadFrom(new
JsonTextReader(streamreader));

    JArray rows = (JArray)responseContent.SelectToken("rows");

    //recorre todos los documentos obtenidos en la consulta.
    foreach (JToken JsonToken in rows)
    {
        var aux = (JToken)JsonToken.SelectToken("doc");

        DateTime time =
Convert.ToDateTime(aux["serverTimestamp"].Value<string>());

        //comparamos la fecha del documento actual con el anterior
almacenado. El mayor es el más reciente.
        if (time > ServerTimeStamp)
        {
            ServerTimeStamp = time;
            LastDoc = aux;
        }
    }

    string typevalue = "";

    //obtenemos la información del documento más reciente
    if (LastDoc != null)
    {
        typevalue = "serverTimestamp: " +
LastDoc["serverTimestamp"].Value<string>() + "\r\n" +
            "payload: " + LastDoc["payload"].Value<string>();
    }

    return typevalue;
}
}
```

Lo más importante del anterior script es resaltar como en la petición que se hace http a través de la URL y después se recorren todos los documentos obtenidos en la consulta con el fin de quedarnos con el último de ellos que será el que tenga el valor de la variable en el momento en el que nos encontramos. Después lo que hacemos es recorrer el json con el que nos hemos quedado tras comparar sus ServerTimestamps y quedarnos con el Token 'doc' para así poder seleccionar los campos dentro de él que nos interesa mostrar por pantalla. En el caso de los datos estáticos el proceso es el mismo con la diferencia de que solo habrá alojado un Json por lo que no sería necesario hacer la lógica para quedarnos con el último existente.

```
private void PedirDatosDinamicos()
{
    //call bluemix with nombre componente, textbox
    if ((texto != null) &
        (BlueMix.GetComponent<CloudantDBclient>().Get_Dinamicos_Datos(texto)))
    {
        //volvemos a llamar a este metodo de forma recursiva con una espera
        //asignada que es el segundo parametro de Invoke
        Invoke("PedirDatosDinamicos", 1f);
    }
}
```

Para evitar tener que pulsar el botón de datos dinámicos para que se actualice el valor de la variable creamos un bucle para que una vez que sea pulse el botón cada segundo el programa internamente vuelve a realizar una llamada para poder actualizar el valor. Se asignó un valor de 1 segundo ya que se observó como con menos tiempo no era suficiente para recibir un true tras la llamada al servidor donde esta alojada la base de datos lo que explica que la función de llamada fuese booleana.

Como último detalle, pero fundamental para el desarrollo de la aplicación es necesario saber que para que todo esto funcione se debe programar una arquitectura enfocada a eventos para Vuforia para poder mostrar sobre el texto que se ve cuando se detecta una imagen todos estos datos que queremos provenientes de la base de datos. Por este fin siguiendo las recomendaciones de las librerías de vuforia se procedió a programar el siguiente script para poder realizar la aplicación deseada.

```
using TMPro;
using UnityEngine;
using Vuforia;
// evento que se dispara cuando se detecta una imagen
public class MyDefaultTrackableEventHandler : MonoBehaviour, ITrackableEventHandler
{
    //public Manager_DB Database;

    public string NombreComponente;

    private TrackableBehaviour mTrackableBehaviour;

    void Start()
    {
        mTrackableBehaviour = GetComponent<TrackableBehaviour>();
        if (mTrackableBehaviour)
        {
            mTrackableBehaviour.RegisterTrackableEventHandler(this);
        }
    }

    public void OnTrackableStateChanged(TrackableBehaviour.Status previousStatus,
    TrackableBehaviour.Status newStatus)
    {
        if (newStatus == TrackableBehaviour.Status.DETECTED ||
            newStatus == TrackableBehaviour.Status.TRACKED ||
            newStatus == TrackableBehaviour.Status.EXTENDED_TRACKED)
        {
            //mostrar botones

            //asignar textbox para escribir los datos

            GameObject.Find("ScreenButtons").GetComponent<MenuOptions>().nombrecomponente =
            NombreComponente;
```

```
transform.GetComponentInChildren<TextMeshProUGUI>().text = "";//  
borramos  
GameObject.Find("ScreenButtons").GetComponent<MenuOptions>().texto =  
transform.GetComponentInChildren<TextMeshProUGUI>();// pasas cajas vacía  
  
    }  
    else  
    {  
        // Stop audio when target is lost  
        GameObject.Find("ScreenButtons").GetComponent<MenuOptions>().texto =  
null;  
    }  
}
```

Lo más importante de este último script es observar como en el momento de detección se pasa el nombre del componente como argumento cuando llamamos a la función Menu.Options para que se guarde en esa variable el componente detectado que será la clave para poder acceder a la base de datos correcta tanto refiriéndonos a datos dinámicos como hablando de datos estáticos. Como se puede observar debemos dejar el recuadro de texto que se monta sobre el componente detectado vacío para pasárselo a la función principal y que lo rellene con los datos apropiados de los componentes que debemos analizar.

Capítulo 6. RESULTADOS OBTENIDOS

Como resultados principales del proyecto debemos comenzar recalcando que se han conseguido los resultados principales del proyecto ya que cuando se establecieron los objetivos y el alcance se establecieron los siguientes puntos a enumerar los cuales se han cumplido todos uno por uno con el inconveniente de que no se ha llegado a monitorizar todas las variables que se querían del laboratorio. Por lo que podemos concluir que el proyecto está correctamente desarrollado, pero no se ha completado con los casos de usos reales que se pretendía por lo que toda la infraestructura y programación esta desarrollada y haría falta aplicar los casos de uso que se pretendía dentro de la minifábrica ICAI.

Lo que se ha hecho es usar una variable asociada a un interruptor para comprobar que todo funciona correctamente y dejar todo preparado para añadir las variables que se quiera al elegir los componentes concretos de estudio. Por tanto, las imágenes que se van a observa en este último apartado reflejan el funcionamiento de la aplicación para un componente x al que se le asocia una variable en tiempo real x cuyo significado no es más que poder observar que la aplicación muestra lo que queremos y los cambios en tiempo real que sufre la variable asociada a ese componente.

El proyecto se debe pensar como un proyecto de integración de dos proyectos diferentes como son la conexión a través del protocolo OPC UA y la nube usando los servicios que bluemix deja a nuestro alcance y una aplicación de realidad aumentada que consiga detectar los componentes en tiempo real y ofrecernos los datos tanto constantes que habrán sido subidos previamente a la nube como los dinámico que se estarán cagando siguiendo un protocolo de eventos OPC UA cada vez que las variables a las que estemos suscritas sufran algún tipo de cambio. El proyecto requiere aunar conocimientos de programación con el conocimiento industrial para poder conocer y crear esos proyectos con sus protocolos de comunicaciones asociados.

A continuación, se mostrarán una serie de capturas de lo que el usuario percibiría desde la pantalla de su móvil desde un inicio hasta el momento de mostrar los datos. Lo primero es ver la interfaz tan sencilla que se diseñó para dicha aplicación con los botones y sus estructuras como fueron diseñada en Unity.



Figura 32. Visualización cámara

El siguiente ocurre de manera automática sin pulsar ningún botón ya que la aplicación está diseñada para que detecte el componente en cuanto la cámara comience a captar los píxeles del target que lleva adherido. Esto pese a ser tan simple puede ser útil para usuarios sin experiencia en el mundo industrial que puedan identificar rápidamente que es el componente que tienen delante de ellos.

En este caso al estar tratando con dos componentes imaginarios se puede observar como dependiendo de cuál sea la imagen de fondo la aplicación nos muestra por pantalla uno u otro mensaje de detección.



Figura 33. Detección componente 1



Figura 34. Detección componente 2

Una vez detectado viene la parte más interesante de la aplicación ya que llega el momento de pulsar uno u otro botón dependiendo de qué datos se quieran ver por pantalla en ese preciso instante.

Las capturas para evitar redundancia se han hecho con un solo objeto para poder observar que la aplicación acude a la base de datos correcto y muestra por pantalla la información deseada.

Se puede observar como si pulsamos sobre datos estáticos cuando la aplicación está detectando el componente nos muestra los datos correspondientes a la base de datos componente 2 donde hay un json preconfigurado.

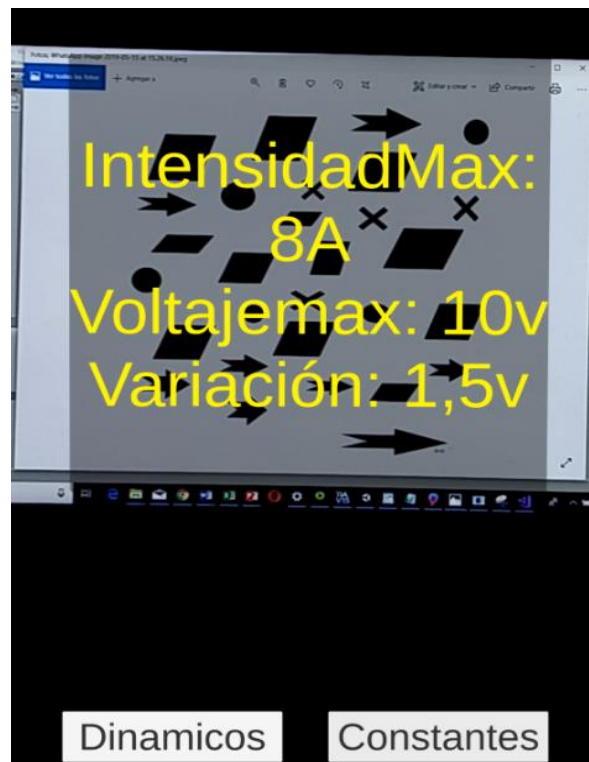
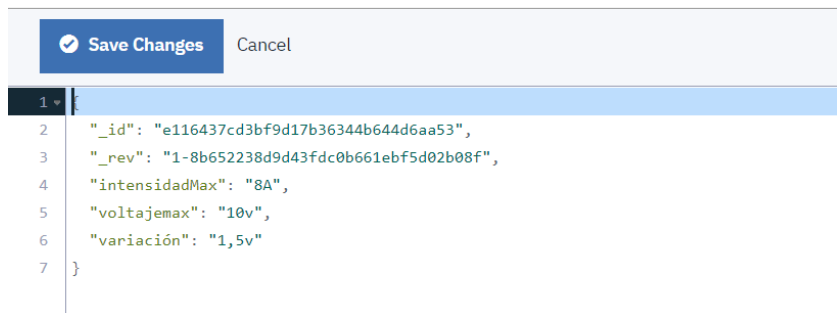


Figura 35. Resultado datos estáticos

Y aquí se observa como efectivamente esta información se corresponde con la información alojada en la base de datos componente 2:

componente2 > e116437cd3bf9d17b36344b644d6aa53



```
1 {  
2   "_id": "e116437cd3bf9d17b36344b644d6aa53",  
3   "_rev": "1-8b652238d9d43fdc0b661ebf5d02b08f",  
4   "intensidadMax": "8A",  
5   "voltajemax": "10v",  
6   "variación": "1,5v"  
7 }
```

Figura 36. Json datos estáticos en la nube

Por último, queda visualizar la parte estática la cual como hemos comentado anteriormente está hecha para una variable de entrada asociada a un interruptor del laboratorio pero que se pretende en posteriores desarrollos asociarla al componente x que sea analizado. Acorde con nuestra implementación se observa como se ha capturado el ultimo Json que es el que muestra el último valor tomado por la variable por lo que la aplicación ya estaría terminada.

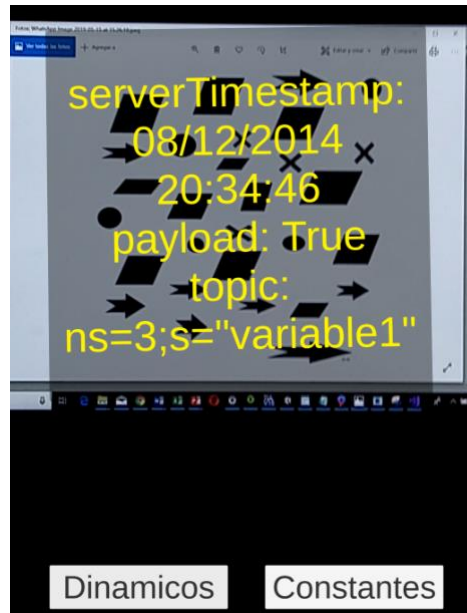


Figura 37. Resultado datos dinámicos

Como conclusiones podemos decir que ha resultado muy interesante observar cómo se puede obtener a través del OPC UA que es un protocolo enfocado a las comunicaciones internas dentro de una red una aplicación externa de realidad aumentada que se sirve de él para dar un acceso rápido y sencillo a la información oculta o menos aparente de los componentes que se estudian y pretenden monitorizar.

Al utilizar OPC UA en lugar de protocolos de comunicaciones a más bajo nivel se ha ahorrado mucha complejidad ya que se evidencia que los PLCs y el protocolo anteriormente nombrado tienen suficiente lógica como para conectarse directamente a Internet. En cuanto al entorno Bluemix se puede deducir que ha sido el más adecuado puesto que es el que mayores facilidades nos da y además contiene un mayor número de servicios para poder desarrollar evoluciones futuras a la propia aplicación.

En el caso de la realidad aumentada se considera que las tecnologías elegidas son aptas para este fin ya que su integración es muy sencilla y las posibilidades que Unity nos da a través de su Asset Store para poder modificar cualquier cosa en el futuro son increíbles como por

ejemplo poder integrar otras bases de datos como firebase de manera sencilla y rápida como ya se hizo en este proyecto cuando solo se había contemplado la aplicación para la visualización de datos estáticos.

6.1 CASO REAL DE USO

Finalmente para dar por terminado el proyecto consiguiendo todos los objetivos que se habían propuesto se mostrarán los resultados que se obtuvieron cuando la variable1 se asoció a uno de los sensores del laboratorio de la minifábrica ICAI en el que se puede ver perfectamente como se consiguió identificar dicho sensor, además de sus características tanto estáticas pertenecientes a su hoja de características como las dinámicas que se actualizaban en tiempo real dependiendo del estado del sensor.



Figura 38. Detección de sensor del laboratorio

En la figura 38 podemos ver como efectivamente la aplicación es capaz de identificar el código superpuesto al sensor y rápidamente identifica de que sensor se trata ya que esta información esta enlazada dentro de la base de datos.

En este momento tocaba probar las funcionalidades creadas por la aplicación separadas claramente entre los datos dinámicos y estáticos asociados al propio sensor.

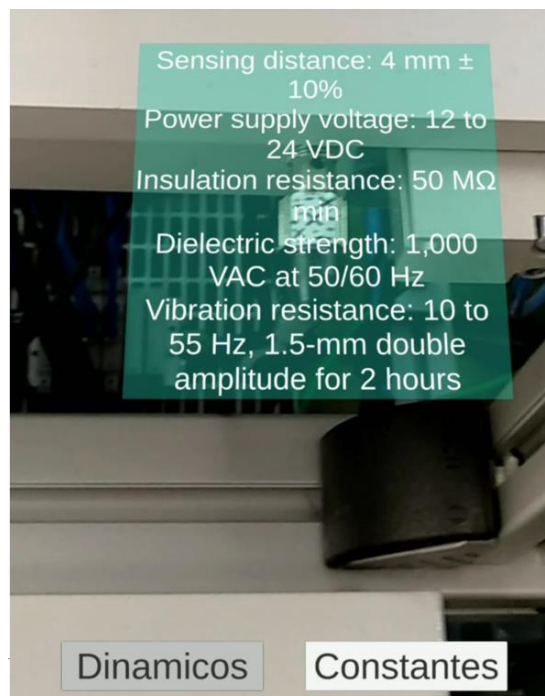


Figura 39. Datos estáticos asociados al sensor

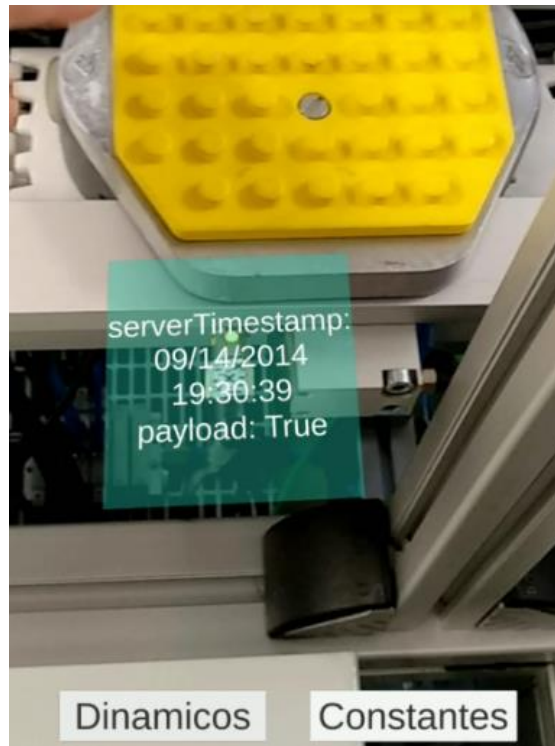


Figura 40. Datos dinámicos asociados al sensor

Como se puede observar se consiguieron todas las funcionalidades deseadas ya que para el sensor de estudio la aplicación es capaz de mostrar los datos estáticos asociados al mismo además de el estado de la variable que permite informarnos de si el sensor este o no detectando. En la figura 40 podemos ver como la variable esta en el estado True ya que en ese momento hay una pieza metálica delante del sensor. A continuación, veremos como cuando esto no ocurre el estado de la variable es False si todo esta funcionando correctamente y la conexión al PLC es la correcta.



Figura 41. Sensor no detectando

Se observa como efectivamente cuando no hay ningún componente delante del sensor la variable pasa al estado False lo que nos dice que el sensor no detecta ninguna actividad.

Para que todo esto suceda la aplicación en Bluemix debe estar activada, es decir, el proyecto en Node Red debe estar encendido escuchando los cambios que puedan suceder en la variable para insertarlos en la base de datos de la aplicación.

Capítulo 7. FUTUROS DESARROLLOS Y

CONCLUSIONES

Como conclusión primero de todo es importante reseñar que el proyecto ha conseguido los objetivos iniciales que se había marcado y ha conseguido dotar a la minifábrica de ICAI de una aplicación de realidad aumentada con la que se pueden controlar en tiempo real el estado de los sensores del laboratorio.

A través de esta aplicación desarrollada en tiempo real y de una manera sencilla podemos identificar cuando un sensor está funcionando mal y saber en dónde erradica el problema de la cadena de montaje. Cualquier persona sin necesidad de tener un mínimo de conocimientos podrá saber cuál es el sensor que tiene delante y adentrarse en su hoja de características sin necesidad de tener que realizar una búsqueda en la web lo que mejorará la eficiencia en tiempo y en resultados.

Además de esto cuando alguien realice pruebas en el laboratorio con cualquier proyecto y algo no funciona podrá utilizar esta aplicación, enfocando componente a componente de manera que pueda saber en tiempo real donde esta el problema sin tener que utilizar material rudimentario como voltímetros midiendo cada salida de cada componente.

La base de datos del proyecto consta del sensor de estudio, pero es fácilmente escalable para añadir tantos sensores como se quiera y cualquier componente que pueda tener un motivo de estudio o monitorización.

El proyecto que aquí ha sido desarrollado, aunque muy potente y suponiendo un gran avance hacia la industria 4.0 vista desde un prisma de sencillez y eficiencia también supone un importante punto de partida para futuros desarrollos con un gran número de posibilidades.

Desde la perspectiva de que este proyecto sea la base para futuros desarrollos se vislumbran multitud de posibilidades de acuerdo a todos los servicios ofrecidos por las plataformas situadas en la nube y en concreto bluemix que es la que aquí se utiliza.

Esta aplicación busca la visualización de los datos a través de la realidad aumentada pero además de esto podría añadir servicios extra predictivos, de carácter analítico o que buscarán una mejora en la eficiencia temporal o energética. Para este tipo de fines se presenta Watson Internet of Things como un servicio de desarrollo que permite extraer todo el valor añadido de los datos que estamos obteniendo a través de la aplicación y el protocolo de conexiones que hemos desarrollado.

Otro posible desarrollo futuro es hacer que todo el laboratorio esté conectado a la nube exportando datos continuamente para que a través de esta aplicación teniendo los PLCs encendidos podamos monitorizar cualquier cosa e incluso servir como actuador en algún proyecto y poder cambiar el valor de las variables de entrada a través de la propia aplicación.

Es conveniente hacer hincapié en que este tipo de tecnologías están avanzando rápidamente y la industria 4.0 ya se está convirtiendo en Industria 4.1 debido al gran número de avances que actualmente se están produciendo por lo que sería interesante que los futuros desarrollos en cuanto a esta aplicación se hagan en poco tiempo ya que si no puede quedar totalmente obsoleta en cuanto a las tecnologías existentes.

Con respecto a la aplicación desarrollada en Unity se podría mejorar la eficiencia de la misma haciéndola asíncrona ya que actualmente tiene que esperar un tiempo determinado a recibir los datos del servidor http y en ese tiempo la aplicación está totalmente parada por lo que se podría aprovechar ese tiempo para realizar cualquier tipo de tarea.

Esta aplicación consiste en una sola pestaña de visualización pero se podrían añadir más pestañas con un menú inicial para poder observar todas las acciones anteriormente comentadas que se realizarían con los servicios Bluemix de manera que el proyecto fuese una aplicación que monitoriza completamente el laboratorio con todo tipo de análisis que

facilitarían muchos trabajos a alto nivel y permitirían al profesor tener un control inmediato de lo que está pasando en el por ejemplo cuando se está realizando un examen de simulación en TIA Portal.

Por supuesto el proyecto no tiene por qué quedarse en el entorno académico, sino que puede ser extensible a la industria como manera de monitorizar fábricas al completo que estén controladas por PLCs que tengan cierta inteligencia.

Capítulo 8. BIBLIOGRAFÍA

- [1] <https://www.codeproject.com/Articles/1105449/Getting-Started-with-Cloudant>
- [2] <https://github.com/cloudant/haengematte/tree/master/c%23dotNet-45>
- [3] <https://github.com/achalshant/Cloudant-Test-C-Sharp/blob/master/CloudantTest/Program.cs>
- [4] <https://docs.unity3d.com/Manual/JSONSerialization.html>
- [5] <https://www.desarrollolibre.net/blog/android/realidad-aumentada-con-vuforia#.XQKug4gzZPY>
- [6] <https://docs.unity3d.com/es/2018.1/Manual/vuforia-sdk-overview.html>
- [7] <http://www.javierlosadasanchez.com/como-desarrollar-apps-con-vuforia-y-unity/>
- [8] <https://www.opiron.com/2018/06/18/que-es-opc-ua/>
- [9] <https://flows.nodered.org/node/node-red-contrib-opcua>
- [10] <https://www.opiron.com/2019/04/22/del-plc-a-la-nube-4-requisitos-cumplidos-con-opc-ua/>
- [11] <https://stackoverflow.com/questions/28347727/available-opc-ua-servers-on-the-internet/34671281>

[12] <https://unity3d.com/es/unity>

[13] <https://www.aragon.es/documents/20127/674325/Estado%20del%20arte%20de%20Realidad%20Aumentada.pdf/f51f996d-eca5-5de4-6d07-8324ee629902>

[14] <https://larraioz.com/articulos/opc-desde-el-clasico-al-nuevo-opc-ua>

[15] https://www.slideshare.net/stephane_potier/open-standards-key-to-success-for-industry-40-40582587