



HoloScada: Realidad aumentada en la Industria 4.0

Grado en Ingeniería Informática

Trabajo Fin de Grado

Autor:

Ramsés Duidar Moreno

Tutor/es:

Félix Jesús García Clemente



HoloScada

Realidad aumentada en la Industria 4.0

Autor

Ramsés Duidar Moreno

Tutor/es

Félix Jesús García Clemente

Departamento de Ingeniería y Tecnología de Computadores



Grado en Ingeniería Informática



Murcia, 2 de Junio de 2024

Preámbulo

Este proyecto se ha desarrollado en colaboración con Binarii, una empresa líder en innovación y adopción de nuevas tecnologías para la transformación digital de industrias hacia el paradigma de la "Smart Factory". En este contexto, el proyecto incorpora tecnologías cognitivas, utilizando específicamente las gafas de realidad mixta HoloLens de Microsoft como herramientas clave para avanzar hacia este futuro industrial.

Un Sistema de Control y Adquisición de Datos (SCADA) es fundamental en la automatización industrial, permitiendo la supervisión, control y adquisición de datos críticos de procesos a través de interfaces intuitivas. Tradicionalmente, estas interfaces son pantallas de grandes dimensiones ubicadas en oficinas aisladas de la planta de producción. Este proyecto propone un sistema que no solo mejora la eficiencia en la gestión de operaciones complejas sino que también proporciona una capa vital de inteligencia operativa y toma de decisiones basada en datos en tiempo real.

Todo el código del proyecto se encuentra disponible en el siguiente repositorio: <https://github.com/ramsesduidar/TFG-Hololens.git>

Resumen

Desde la primera revolución industrial, la industria ha estado en continua evolución, cambiando por completo la forma en la que producimos, fabricamos y gestionamos recursos.

Tras el surgimiento de la cuarta revolución industrial, la realidad aumentada ha ido cobrando cada vez más presencia en el sector redefiniendo los paradigmas tradicionales de la industria y demostrando ser una tecnología con un elevado potencial.

La realidad aumentada permite ofrecer una capa de información digital sobre el entorno físico, proporcionando soluciones innovadoras que aumentan la eficiencia y eficacia de los procesos de formación, fabricación, mantenimiento y control de planta y procesos de inspección.

La idea principal es mostrar los datos de manera más intuitiva y añadir información adicional para facilitar el trabajo de los operarios en planta. Básicamente, se trata de proyectar hologramas que aporten información relevante y guíen en la resolución de problemas.

Aunque este enfoque ya proporciona una herramienta potente para la resolución de problemas y optimización de los procesos, teniendo guías en tiempo real que van más allá de un manual o una serie de imágenes, esto relega a la realidad aumentada a comportarse como un elemento pasivo que solo aporta información, sin permitir una interacción directa con el mundo real. En este proyecto intentaremos ir un paso más allá, permitiendo a los usuarios interactuar con los procesos industriales a través del uso de la realidad aumentada.

Este Trabajo Final de Grado se centrará en el desarrollo de una aplicación para la monitorización y control de procesos industriales haciendo uso de la realidad aumentada. El objetivo principal es crear una interfaz de usuario intuitiva y eficiente que permita a los operarios supervisar e interactuar con los procesos industriales en tiempo real.

A lo largo del documento, se expondrán los principales casos de uso en los que interviene y mejora la realidad aumentada, se explicarán sus beneficios y se presentarán algunas soluciones desarrolladas por otras empresas, para entender mejor como esta

tecnología se integra a día de hoy en la industria.

Finalmente, se explicará como se llevará a cabo el desarrollo de nuestra aplicación, se expondrán los objetivos y evaluará el producto final.

Extended Abstract

La industria ha experimentado una fascinante evolución a lo largo de las décadas, desde sus raíces en la Revolución Industrial con la invención de la máquina de vapor hasta la era contemporánea de la Industria 4.0. Cada fase de este desarrollo ha estado marcada por avances significativos en tecnología, transformando radicalmente la forma en la que producimos, fabricamos y gestionamos recursos.

Tras el surgimiento de la cuarta revolución industrial, la realidad aumentada ha ido cobrando cada vez más presencia en el sector redefiniendo los paradigmas tradicionales de la industria, siendo una tecnología con un alto potencial.

En la actualidad, los procesos industriales están en su mayoría automatizados, gracias al uso de PLC y sistemas SCADA. Los PLC (controlador lógico programable), son pequeños ordenadores utilizados para la automatización de procesos industriales. Estos están diseñados para responder ante señales de entrada y producir señales de salida. Estas señales de salida son las que manipulan el comportamiento de las máquinas, como encender un motor o abrir una válvula.

Por otro lado, los sistema SCADA (Control Supervisor y Adquisición de Datos) son utilizados para la monitorización de los procesos industriales mostrando información relevante sobre el proceso y notificando ante posibles errores. Estos sistemas muestran la información a través de un HMI (interfaz Humano - Máquina) para que un operario pueda controlar y gestionar el proceso.

Sin embargo estas interfaces suelen presentar dificultades a la hora de interpretarlas e interactuar con ellas, lo que se traduce en tiempo de formación para los operarios. Además, la información que pueden mostrar es limitada y en ocasiones los mensajes de error no pueden ser los suficientemente específicos o claros, lo que puede derivar en dificultades para entender que es lo que está fallando.

Esta tecnología ha ido desarrollándose y evolucionando con el tiempo, presentando interfaces cada vez más detalladas y con mensajes más específicos. Sin embargo, esto también se traduce en diagramas más complejos con los problemas mencionados anteriormente que esto trae consigo.

En este contexto, la realidad aumentada (RA) emerge como una herramienta que

redefine los paradigmas tradicionales de la industria. La RA no solo ofrece una capa de información digital sobre el entorno físico, sino que también se integra de manera cohesiva en la Industria 4.0, proporcionando soluciones innovadoras para la formación, fabricación, mantenimiento y control de planta, así como en procesos de inspección. En el ámbito industrial, a menudo se le denomina Realidad Aumentada Industrial (IAR).

La idea suele ser sencilla: imprimir hologramas que aporten información relevante y guíe en la resolución de problemas, mostrando los datos de una manera más intuitiva y añadiendo información adicional que facilita el trabajo de los operarios en planta.

El uso de la realidad aumentada permite ofrecer guías más precisas y fáciles de entender para un humano e incluso permite a los operarios de menor experiencia resolver y entender los problemas más complejos.

A continuación, vamos a presentar los principales casos de uso de la realidad aumentada en la industria, demostrando cómo esta tecnología transforma los procesos y prácticas tradicionales, proporcionando soluciones innovadoras y mejorando la eficiencia operativa en diversos contextos industriales:

Entrenamiento y aprendizaje: Uno de los casos de uso principales es el de poder entrenar a los operarios en el desarrollo de distintos procesos y aportar guías para que puedan desarrollar cualquier operación por primera vez. Con el uso de la realidad aumentada se pueden crear tutoriales interactivos en tiempo real, que guíen a los trabajadores paso a paso, similar a los que se pueden encontrar en los videojuegos.

Esto no solo reduce el tiempo que los expertos deben dedicar a la formación de nuevos empleados, sino que también aumenta la productividad. Al seguir estas instrucciones detalladas, se garantiza la eficacia de las operaciones, siempre y cuando los pasos estén correctamente programados.

Fabricación: La realidad aumentada también es fundamental en la fabricación, proporcionando guías detalladas para el montaje de diversos dispositivos y máquinas. Aunque este uso podría considerarse parte del entrenamiento, en este contexto se enfoca específicamente en instrucciones precisas para ensamblar componentes, asegurando la correcta ejecución de cada paso del proceso.

Asistencia remota: Otro de los usos de la realidad aumentada es el de poder solicitar la asistencia de un profesional, compartiendo con él lo que se está viendo para poder explicar mejor la situación. Pero no se trata sólo de hacer una videollamada; gracias al uso de la realidad aumentada, el experto desde el otro lado puede interactuar directamente con la aplicación, señalando objetos y proporcionando indicaciones que se materializan como hologramas en el entorno del usuario, mejorando así la eficacia de la consulta.

Con esto no sólo se consigue mejorar la productividad y el aprendizaje de los trabajadores sino que también se consigue mantener a los expertos accesibles desde cualquier sitio aunque no estén cerca del lugar de trabajo.

Mantenimiento y control de planta: Por supuesto, la realidad aumentada no sólo se usa para la formación de futuros expertos, sino que también se usa en el ámbito del mantenimiento y control de los recursos y procesos industriales.

Con el uso de la realidad aumentada, un operario puede acercarse a una máquina y visualizar mediante hologramas información detallada sobre su estado e incluso recibir notificaciones acerca de errores. La principal ventaja frente a la opción tradicional es la de poder obtener una visión más clara del estado actual y poder recibir indicaciones más precisas, como por ejemplo en el caso de los errores, ya que a veces visualizar esta información en los SCADA suele ser un poco tedioso e incluso puede requerir algo de formación para comprender el diagrama en su totalidad.

Con la RA obtenemos información más precisa e intuitiva pudiendo estar en frente de la máquina observándola a la vez que consultamos los datos.

Inspección: Este apartado se refiere a la inspección de los productos durante todo su ciclo de vida, desde que llega una nueva pieza para reemplazar otra más antigua hasta inspecciones regulares de mantenimiento. En todos estos casos la RA permite mejorar estos procesos de inspección ofreciendo información más precisa y completa frente a manuales o libros de requisitos que pueden ser ambiguos o requerir alta experiencia para poder aplicar en un caso real.

Un encargado de mantenimiento puede acercarse a examinar el equipo y ver claramente si algo está fallando gracias a mensajes en tiempo real u observar posibles defectos de fabricación comparando el equipo con un modelo 3D de una versión perfectamente construida. También podría disponer de una guía más interactiva que le indique todos los pasos que debe de seguir para asegurarse que no se le olvida revisar ningún elemento y así evitar futuros errores.

La realidad aumentada ha demostrado ser una herramienta poderosa en el ámbito industrial, aportando numerosos beneficios que impactan positivamente en diversos aspectos operativos. Gracias a la RA, es posible mejorar significativamente la productividad, reducir errores, y optimizar procesos, lo que se traduce en un aumento general de la eficiencia y la calidad del trabajo. Los principales beneficios que la RA aporta en el entorno industrial son los siguientes:

Mejora de la productividad: La RA transforma la capacitación y el aprendizaje, permitiendo la creación de tutoriales interactivos en tiempo real. Esto acelera la formación de operarios, reduciendo así el tiempo que los expertos deben invertir en enseñar, lo que se traduce en un aumento significativo de la productividad.

Eficiencia y eficacia: En el ámbito de la fabricación, las guías de montaje en realidad aumentada optimizan los procesos de ensamblaje al proporcionar instrucciones visuales precisas. Esto no solo acelera la producción, sino que también minimiza errores y mejora la calidad del producto final.

Prevención de errores: En el mantenimiento y control de planta, la RA ofrece una visión detallada del estado de las máquinas mediante hologramas. Esto permite a los operarios identificar problemas potenciales y recibir notificaciones de errores en tiempo real, evitando fallas costosas y mejorando la eficacia de las operaciones.

Reducción de costes y puesta en marcha: La asistencia remota no solo simplifica la resolución de problemas, sino que también reduce los costes asociados con desplazamientos de expertos. La capacidad de recibir instrucciones interactivas a distancia agiliza las operaciones y disminuye los tiempos de inactividad. Además, una tasa de errores menor también se traduce en menores costes.

Optimización de inspecciones: La RA facilita las inspecciones a lo largo del ciclo de vida de los productos, desde la recepción de nuevas piezas hasta las revisiones regulares de mantenimiento. Proporciona información detallada y precisa, permitiendo a los encargados de mantenimiento identificar rápidamente defectos, comparar con modelos 3D y seguir guías interactivas para garantizar una inspección exhaustiva y evitar futuros errores.

En resumen, la realidad aumentada no sólo transforma los procesos industriales, sino que también impulsa una serie de beneficios clave que van desde la formación eficiente hasta la prevención proactiva de problemas, mejorando la productividad y la rentabilidad en la industria.

Todo lo expuesto demuestra como la realidad aumentada es capaz de proporcionar una herramienta potente para la resolución de problemas y optimización de los procesos, teniendo guías en tiempo real que van más allá de un manual o una serie de imágenes. Sin embargo, los casos de uso presentados anteriormente relegan al uso de la realidad aumentada a comportarse como un elemento pasivo que solo aporta información, sin permitir una interacción directa con el mundo real. En este proyecto intentaremos ir un paso más allá, permitiendo a los usuarios interactuar con los procesos industriales a través del uso de la realidad aumentada, integrando por completo esta tecnología en la industria.

Con interacción, nos referimos a ser capaces no sólo de monitorizar, sino también de manipular los procesos. Esto va desde poner en marcha o parar una máquina hasta introducir parámetros de entrada para ciertos procesos, como establecer el porcentaje de llenado en un tanque de agua, proporcionando así una experiencia más inmersiva y completa a los usuarios, siendo capaces de poder operar haciendo uso de un único

dispositivo.

Sin embargo, aunque hoy en día el desarrollo de aplicaciones de realidad aumentada es relativamente accesible, este tipo de aplicaciones presentan una serie de dificultades. Primero de todo, es fundamental poder reconocer los objetos para visualizar los datos y objetos virtuales adecuados. Además, en nuestro caso de uso, también es necesario interactuar con estos objetos virtuales, como en el caso de pulsar un botón, y evitar que el usuario tenga las manos ocupadas sosteniendo el dispositivo para apuntar a los objetos reales.

Aquí es donde entra en juego el concepto realidad mixta, donde los objetos digitales coexisten e interactúan con el entorno físico en tiempo real, permitiendo al usuario manipular ambos mundos con sus manos sin necesidad de tocar una pantalla.

Para que un operario pueda acercarse a una máquina estropeada y recibir indicaciones en tiempo real sobre qué hacer, necesitamos que tenga las manos libres y pueda ver los objetos virtuales con sus propios ojos, no a través de una pantalla. Además, para permitir la navegación y confirmación de pasos en un proceso (similar a un tutorial de un videojuego), es esencial interactuar directamente con los objetos virtuales usando las manos. Estas necesidades se pueden satisfacer utilizando hardware específico para realidad mixta, como las Hololens 2 desarrolladas por Microsoft. Estas gafas de realidad mixta proyectan hologramas en el espacio con los que el usuario puede interactuar directamente, gracias a su capacidad para reconocer y seguir el movimiento de las manos y los ojos.

Por otro lado, también es necesario poder recoger información de las máquinas y sensores para procesarla y mostrarla al usuario, por lo que será necesario desarrollar una solución que nos permita recoger los datos y simplifique el desarrollo de la aplicación, permitiéndonos centrarnos en la interfaz y experiencia de usuario.

Otro punto muy importante también es el de crear los modelos 3D de los objetos y máquinas que quieran representarse mediante hologramas, con el objetivo de poder crear una experiencia lo más precisa y completa posible.

Este Trabajo Final de Grado se centrará en el desarrollo de una aplicación para la monitorización y control de procesos industriales haciendo uso de la realidad aumentada. El objetivo principal es crear una interfaz de usuario intuitiva y eficiente que permita a los operarios supervisar e interactuar con los procesos industriales en tiempo real. Para esto, se crearán dos escenarios (uno virtual y otro haciendo uso de una maqueta) que simulen algún proceso industrial y se construirá una aplicación para los Hololens 2 que permita reconocer cada escena y presentar los objetos adecuados al usuario.

También, haremos uso de códigos QR para ser capaces de detectar las diferentes escenas y actuar como corresponda ante cada una de ellas. En el caso de la maqueta, además, intentaremos dibujar hologramas en el entorno que aporten algún tipo de información sobre el proceso.

Para abordar el problema de la comunicación y recolección de los datos de los PLC, utilizaremos los de la marca Siemens y haremos uso de una librería desarrollada para C# que permite la lectura y escritura de valores en PLC de dicha marca. Sin embargo, para el futuro tendremos que intentar una solución que nos permita comunicarnos con distintos tipos de PLC sin aumentar demasiado la complejidad de la aplicación. Debido a esto intentamos desarrollar una arquitectura para la aplicación lo más limpia posible donde el núcleo de la aplicación sea lo más independiente posible de los módulos relacionados con la comunicación con los PLC y el reconocimiento de las escenas y objetos.

A lo largo del documento se expondrán los principales casos de uso en los que interviene y mejora la realidad aumentada, se explicarán sus beneficios y se presentarán algunas soluciones desarrolladas por otras empresas, para entender mejor como esta tecnología se integra a día de hoy en la industria. Luego se expondrán los objetivos y se explicará la metodología que se seguirá durante el desarrollo, como se resolverá cada aspecto del desarrollo.

Finalmente se evaluará el producto final y discutiremos sobre los aspectos en los que podría mejorar la aplicación e introduciremos posibles alternativas que resuelvan de mejor manera los problemas tratados en el desarrollo, principalmente sobre el reconocimiento de objetos y la recolección de los datos del proceso, sirviendo así de guía para los próximos pasos del proyecto.

Extended Abstract

The industry has undergone a fascinating evolution over the decades, from its roots in the Industrial Revolution with the invention of the steam engine to the contemporary era of Industry 4.0. Each phase of this development has been marked by significant technological advancements, radically transforming the way we produce, manufacture, and manage resources.

Following the emergence of the fourth industrial revolution, augmented reality has been gaining increasing prominence in the sector, redefining the traditional paradigms of the industry, this being a technology with high potential.

Currently, industrial processes are largely automated, thanks to the use of PLCs and SCADA systems.

PLCs (Programmable Logic Controllers) are small computers used for the automation of industrial processes. They are designed to respond to input signals and produce output signals. These output signals control the behavior of machines, such as turning on a motor or opening a valve.

On the other hand, SCADA (Supervisory Control and Data Acquisition) systems are used to monitor industrial processes, displaying relevant information about the process and alerting operators to potential errors. These systems present information through an HMI (Human-Machine Interface) so that an operator can control and manage the process.

However, these interfaces often present challenges in interpretation and interaction, leading to training time for operators. Additionally, the information they can display is limited, and sometimes error messages are not specific or clear enough, which can make it difficult to understand what is failing.

This technology has developed and evolved over time, offering increasingly detailed interfaces with more specific messages. However, this also results in more complex diagrams, which bring the aforementioned challenges.

In this context, Augmented Reality (AR) emerges as a tool that redefines traditional industry paradigms. AR not only overlays digital information on the physical environ-

ment but also integrates seamlessly into Industry 4.0, providing innovative solutions for training, manufacturing, plant maintenance and control, as well as inspection processes. In the industrial field, it is often referred to as Industrial Augmented Reality (IAR).

The concept is usually straightforward: projecting holograms that provide relevant information and guide problem-solving by displaying data more intuitively and adding supplementary information that facilitates the work of plant operators.

The use of augmented reality allows for the provision of more precise and easier-to-understand guides for humans, enabling even less experienced operators to resolve and comprehend more complex problems.

Below, we will present the main use cases of augmented reality in the industry, demonstrating how this technology transforms traditional processes and practices, providing innovative solutions and improving operational efficiency in various industrial contexts:

Training and learning: One of the main use cases is the ability to train operators in various processes and provide guides for performing any operation for the first time. Augmented reality can be used to create interactive real-time tutorials that guide workers step by step, similar to those found in video games.

This not only reduces the time experts must dedicate to training new employees but also increases productivity. By following these detailed instructions, the effectiveness of operations is ensured, provided that the steps are correctly programmed.

Manufacturing: Augmented reality is also crucial in manufacturing, providing detailed guides for assembling various devices and machines. Although this use could be considered part of training, in this context, it specifically focuses on precise instructions for assembling components, ensuring the correct execution of each step of the process.

Remote assistance: Another use of augmented reality is the ability to request assistance from a professional, sharing what is being seen to better explain the situation. But it's not just about making a video call; thanks to augmented reality, the expert on the other end can interact directly with the application, pointing out objects and providing instructions that materialize as holograms in the user's environment, thus improving the effectiveness of the consultation.

This not only improves productivity and the learning of workers but also keeps experts accessible from anywhere, even if they are not near the workplace.

Maintenance and plant control: Of course, augmented reality is not only used for training future experts but also in the field of maintenance and control of industrial resources and processes.

With augmented reality, an operator can approach a machine and visualize detailed information about its status through holograms, and even receive notifications about errors. The main advantage over the traditional option is the ability to obtain a clearer view of the current state and receive more precise instructions, such as in the case of errors, as visualizing this information in SCADA systems can be tedious and may even require some training to fully understand the diagram.

With AR, we obtain more precise and intuitive information, being able to be in front of the machine while observing it and consulting the data.

Inspection: This section refers to the inspection of products throughout their lifecycle, from when a new part arrives to replace an older one to regular maintenance inspections.

In all these cases, AR allows us to improve these inspection processes by offering more precise and complete information compared to manuals or requirement books that may be ambiguous or require high expertise to apply in a real case.

A maintenance manager can approach to examine the equipment and clearly see if something is wrong thanks to real-time messages or observe possible manufacturing defects by comparing the equipment with a 3D model of a perfectly built version. They could also have a more interactive guide that indicates all the steps they must follow to ensure that they do not forget to check any element and thus avoid future errors.

Augmented reality has proven to be a powerful tool in the industrial field, providing numerous benefits that positively impact various operational aspects. Thanks to AR, it is possible to significantly improve productivity, reduce errors, and optimize processes, resulting in an overall increase in efficiency and work quality. The main benefits that AR brings to the industrial environment are as follows:

Improvement of productivity: AR transforms training and learning, enabling the creation of real-time interactive tutorials. This accelerates operator training, thus reducing the time experts need to invest in teaching, resulting in a significant increase in productivity.

Efficiency and effectiveness: In the manufacturing realm, augmented reality assembly guides optimize assembly processes by providing precise visual instructions. This not only speeds up production but also minimizes errors and enhances the quality of the final product.

Error prevention: In plant maintenance and control, AR provides a detailed view of machine status through holograms. This enables operators to identify potential issues and receive real-time error notifications, preventing costly failures and enhancing operational efficiency.

Cost reduction and commissioning: Remote assistance not only simplifies problem resolution but also reduces costs associated with expert travel. The ability to receive interactive instructions remotely streamlines operations and decreases downtime. Additionally, a lower error rate also translates to reduced costs.

Inspection optimization: AR facilitates inspections throughout the product lifecycle, from receiving new parts to regular maintenance checks. It provides detailed and precise information, enabling maintenance supervisors to quickly identify defects, compare with 3D models, and follow interactive guides to ensure thorough inspections and prevent future errors.

In summary, augmented reality not only transforms industrial processes but also drives a range of key benefits from efficient training to proactive issue prevention, enhancing productivity and profitability in the industry.

All of the above demonstrates how augmented reality can provide a powerful tool for problem-solving and process optimization, offering real-time guidance beyond a manual or a series of images. However, the use cases presented above relegate augmented reality to behaving like a passive element that only provides information, without allowing direct interaction with the real world. In this project, we aim to go a step further by enabling users to interact with industrial processes through augmented reality, fully integrating this technology into the industry.

By interaction, we mean being able not only to monitor but also to manipulate processes. This ranges from starting or stopping a machine to entering input parameters for certain processes, such as setting the filling percentage in a water tank, providing a more immersive and comprehensive experience to users, enabling them to operate using a single device.

However, although the development of augmented reality applications is relatively accessible nowadays, such applications pose a series of challenges. Firstly, it is essential to recognize objects in order to display the appropriate virtual data and objects. Additionally, in our use case, it is also necessary to interact with these virtual objects, such as pressing a button, and to avoid the user having their hands occupied holding the device to aim at real objects.

This is where the concept of mixed reality comes into play, where digital objects coexist and interact with the physical environment in real time, allowing the user to

manipulate both worlds with their hands without needing to touch a screen.

For an operator to approach a malfunctioning machine and receive real-time instructions on what to do, we need them to have their hands free and be able to see the virtual objects with their own eyes, not through a screen. Furthermore, to enable navigation and confirmation of steps in a process (similar to a tutorial in a video game), it is essential to directly interact with virtual objects using hands. These needs can be met using specific hardware for mixed reality, such as the Hololens 2 developed by Microsoft. These mixed reality glasses project holograms into space with which the user can interact directly, thanks to their ability to recognize and track hand and eye movements.

On the other hand, it is also necessary to collect information from machines and sensors to process and display it to the user. Therefore, it will be necessary to develop a solution that allows us to collect data and simplifies the development of the application, allowing us to focus on the interface and user experience.

Another very important point is also the creation of 3D models of the objects and machines to be represented as holograms, with the aim of creating the most accurate and complete experience possible.

This Final Degree Project will focus on the development of an application for monitoring and controlling industrial processes using augmented reality. The main objective is to create an intuitive and efficient user interface that allows operators to supervise and interact with industrial processes in real-time. To achieve this, two scenarios will be created (one virtual and one using a physical model) to simulate an industrial process, and an application for the Hololens 2 will be built to recognize each scene and present the appropriate objects to the user.

Additionally, we will use QR codes to be able to detect different scenes and act accordingly for each of them. In the case of the physical model, we will also attempt to draw holograms in the environment that provide some type of information about the process.

To address the problem of communication and data collection from the PLCs, we will use Siemens PLCs and make use of a C# library developed for reading and writing values in PLCs of this brand. However, for the future, we will need to explore a solution that allows us to communicate with different types of PLCs without increasing the complexity of the application too much. Due to this, we aim to develop an architecture for the application as clean as possible where the core of the application is as independent as possible from the modules related to communication with the PLCs and scene and object recognition.

Throughout the document, the main use cases in which augmented reality intervenes and improves will be presented, its benefits will be explained, and some solutions developed by other companies will be presented to better understand how this technology is currently integrated into the industry. Then, the objectives will be presented, and the methodology to be followed during development, along with how each aspect of development will be addressed, will be explained.

Finally, the final product will be evaluated, and discussions will be held regarding areas where the application could be improved. Possible alternatives that address the development issues more effectively, particularly focusing on object recognition and data collection from the process, will be introduced. This will serve as a guide for the project's next steps.

Índice general

1	Introducción	1
1.1	Casos de uso de la RA en la Industria	2
1.2	Beneficios del uso de la RA	4
2	Estado del arte	7
2.1	PTC	7
2.2	Aircada Pro	9
2.3	Innovae	9
2.4	Comparativa final	11
3	Análisis de objetivos y metodología	13
3.1	Objetivos y contribuciones	13
3.2	Metodología	13
4	Diseño y resolución del trabajo realizado	15
4.1	Herramientas utilizadas	15
4.2	Resolución del trabajo	16
4.3	Diagrama de clases	35
4.4	Mejoras adicionales	36
4.4.1	Hologramas	36
4.4.2	Menú Mano	39
4.4.3	Uso de tareas	41
4.5	Ánalisis de resultados	43
5	Conclusiones y vías futuras	45
	Bibliografía	47
	Lista de Acrónimos y Abreviaturas	49

Índice de figuras

1.1	Ejemplo de escenario de entrenamiento.	3
1.2	Ejemplo del uso de la RA en fabricación.	4
1.3	Ejemplo del uso de la RA en asistencia remota.	5
1.4	Ejemplo del uso de la RA en control.	6
4.1	Añadir botón a la escena	17
4.2	Añadir OnClick al botón	17
4.3	Añadir etiqueta a la escena.	18
4.4	Dibujar objetos en QR	25
4.5	Añadir objeto al visualizer	26
4.6	Configurar la Escena para los eventos	28
4.7	Escena virtual con factory I/O	29
4.8	Configuración Escena_PLC	30
4.9	Configurar botón para que escriba en la variable Boton_Marcha	30
4.10	Ejemplos de interfaces usando MRTK3	32
4.11	Antes y después del menú	32
4.12	Establecer scala de 0.001	33
4.13	Layout del menú	34
4.14	Estructura de los botones	35
4.15	Configuracion de Tab View y Toggle Collection	35
4.16	Diagrama de clases de la aplicación	37
4.17	Menú para la maqueta	39
4.18	Holograma para la maqueta	40
4.19	Holograma para la maqueta	41
4.20	Ejemplo de menu mano	42
4.21	Holograma para la maqueta	43

Índice de tablas

2.1 Tabla comparativa de las distintas soluciones	11
---	----

Índice de Códigos

4.1	Ejemplo código para escribir en un PLC	17
4.2	Ejemplo de código para leer de un PLC	18
4.3	Clase Atributo	19
4.4	Función LeerVariables()	20
4.5	Objetos de escritura v1	21
4.6	Objeto de lectura v1	23
4.7	Código evento QRLeído	27
4.8	Código para dibujar holograma	38
4.9	Ejemplo del uso de tareas	42

1 Introducción

La industria ha experimentado una fascinante evolución a lo largo de las décadas, desde sus raíces en la Revolución Industrial con la invención de la máquina de vapor hasta la era contemporánea de la Industria 4.0 [1, 2]. Cada fase de este desarrollo ha estado marcada por avances significativos en tecnología, transformando radicalmente la forma en la que producimos, fabricamos y gestionamos recursos.

En este contexto, la realidad aumentada (RA) emerge como una herramienta que redefine los paradigmas tradicionales de la industria. La RA no solo ofrece una capa de información digital sobre el entorno físico, sino que también se integra de manera cohesiva en la Industria 4.0, proporcionando soluciones innovadoras para la formación, fabricación, mantenimiento y control de planta, así como en procesos de inspección. En el ámbito industrial, a menudo se le denomina Realidad Aumentada Industrial (IAR) [3].

El objetivo de este TFG es desarrollar un producto que demuestre el potencial de la realidad aumentada en el ámbito de la industria para mejorar la eficiencia y productividad de los procesos industriales mediante la creación de SCADA¹ virtuales.

A pesar de que hoy en día el desarrollo de aplicaciones de realidad aumentada es relativamente accesible, este proyecto presenta un gran desafío. Es fundamental reconocer el entorno para representar los objetos virtuales de manera precisa. Además, en nuestro caso de uso, es necesario interactuar con estos objetos virtuales, como pulsar un botón, y evitar que el usuario tenga las manos ocupadas sosteniendo el dispositivo para apuntar a los objetos reales.

Aquí es donde entra en juego la realidad mixta, donde los objetos digitales coexisten e interactúan con el entorno físico en tiempo real, permitiendo al usuario manipular ambos mundos con sus manos sin necesidad de tocar una pantalla.

Para que un operario pueda acercarse a una máquina estropeada y recibir indicaciones en tiempo real sobre qué hacer, necesitamos que tenga las manos libres y pueda ver los objetos virtuales con sus propios ojos, no a través de una pantalla. Además, para permitir la navegación y confirmación de pasos en un proceso (similar a un videojuego),

¹<https://www.cursosaula21.com/que-es-un-sistema-scada/>

es esencial interactuar directamente con los objetos virtuales usando las manos.

Estas necesidades se pueden satisfacer utilizando hardware específico para realidad mixta, como las Hololens 2 desarrolladas por Microsoft. Estas gafas de realidad mixta proyectan hologramas en el espacio con los que el usuario puede interactuar directamente, gracias a su capacidad para reconocer y seguir el movimiento de las manos y los ojos.

Aunque existen otras alternativas y distintos tipos de dispositivos [4], para nuestro caso necesitamos que el dispositivo soporte la interactividad con los objetos virtuales y pueda ser utilizado durante largos períodos sin causar fatiga al usuario, algo que suele ocurrir con los dispositivos de realidad virtual.

Finalmente, en un contexto industrial, también es crucial poder recoger información de las máquinas y sensores para procesarla y mostrarla al usuario. Por estas razones, es necesario crear un framework que resuelva estas necesidades y facilite el desarrollo de la aplicación, permitiéndonos centrarnos en la interfaz y experiencia de usuario.

1.1 Casos de uso de la RA en la Industria

En esta sección, se presentan los principales casos de uso de la realidad aumentada en la industria. Se destacará cómo esta tecnología está transformando los procesos y prácticas tradicionales, proporcionando soluciones innovadoras y mejorando la eficiencia operativa en diversos contextos industriales:

Entrenamiento y aprendizaje: Uno de los casos de uso principales es el de poder entrenar a los operarios en el desarrollo de distintos procesos y aportar guías para que puedan desarrollar cualquier operación por primera vez. Con el uso de la realidad aumentada se pueden crear tutoriales interactivos en tiempo real, que guían a los trabajadores paso a paso, similar a los que se pueden encontrar en los videojuegos.

Esto no solo reduce el tiempo que los expertos deben dedicar a la formación de nuevos empleados, sino que también aumenta la productividad. Al seguir estas instrucciones detalladas, se garantiza la eficacia de las operaciones, siempre y cuando los pasos estén correctamente programados.

Fabricación: La realidad aumentada también es fundamental en la fabricación, proporcionando guías detalladas para el montaje de diversos dispositivos y máquinas. Aunque este uso podría considerarse parte del entrenamiento, en este contexto se enfoca específicamente en instrucciones precisas para ensamblar componentes, asegurando la correcta ejecución de cada paso del proceso.



Figura 1.1: Ejemplo de escenario de entrenamiento.

Asistencia remota: Otro de los usos de la realidad aumentada es el de poder solicitar la asistencia de un profesional, compartiendo con él lo que se está viendo para poder explicar mejor la situación. Pero no se trata sólo de hacer una videollamada; gracias al uso de la realidad aumentada, el experto desde el otro lado puede interactuar directamente con la aplicación, señalando objetos y proporcionando indicaciones que se materializan como hologramas en el entorno del usuario, mejorando así la eficacia de la consulta.

Con esto no sólo se consigue mejorar la productividad y el aprendizaje de los trabajadores sino que también se consigue mantener a los expertos accesibles desde cualquier sitio aunque no estén cerca del lugar de trabajo.

Mantenimiento y control de planta: Por supuesto, la realidad aumentada no sólo se usa para la formación de futuros expertos, sino que también se usa en el ámbito del mantenimiento y control de los recursos y procesos industriales.

Con el uso de la realidad aumentada, un operario puede acercarse a una máquina y visualizar mediante hologramas información detallada sobre su estado e incluso recibir notificaciones acerca de errores. La principal ventaja frente a la opción tradicional es la de poder obtener una visión más clara del estado actual y poder recibir indicaciones más precisas, como por ejemplo en el caso de los errores, ya que a veces visualizar esta información en los SCADA suele ser un poco tedioso e incluso puede requerir algo de formación para comprender el diagrama en su totalidad.

Con la RA obtenemos información más precisa e intuitiva pudiendo estar en frente de la máquina observándola a la vez que consultamos los datos.



Figura 1.2: Ejemplo del uso de la RA en fabricación.

Inspección: Este apartado se refiere a la inspección² de los productos durante todo su ciclo de vida, desde que llega una nueva pieza para reemplazar otra más antigua hasta inspecciones regulares de mantenimiento. En todos estos casos la RA permite mejorar estos procesos de inspección ofreciendo información más precisa y completa frente a manuales o libros de requisitos que pueden ser ambiguos o requerir alta experiencia para poder aplicar en un caso real.

Un encargado de mantenimiento puede acercarse a examinar el equipo y ver claramente si algo está fallando gracias a mensajes en tiempo real u observar posibles defectos de fabricación comparando el equipo con un modelo 3D de una versión perfectamente construida. También podría disponer de una guía más interactiva que le indique todos los pasos que debe de seguir para asegurarse que no se le olvida revisar ningún elemento y así evitar futuros errores.

1.2 Beneficios del uso de la RA

La realidad aumentada ha demostrado ser una herramienta poderosa en el ámbito industrial, aportando numerosos beneficios que impactan positivamente en diversos aspectos operativos. Gracias a la RA, es posible mejorar significativamente la productividad, reducir errores, y optimizar procesos, lo que se traduce en un aumento general de la eficiencia y la calidad del trabajo. A continuación, se detallan los principales beneficios que la RA aporta en el entorno industrial:

²<https://www.ptc.com/en/resources/augmented-reality/ebook/inspection-use-cases>

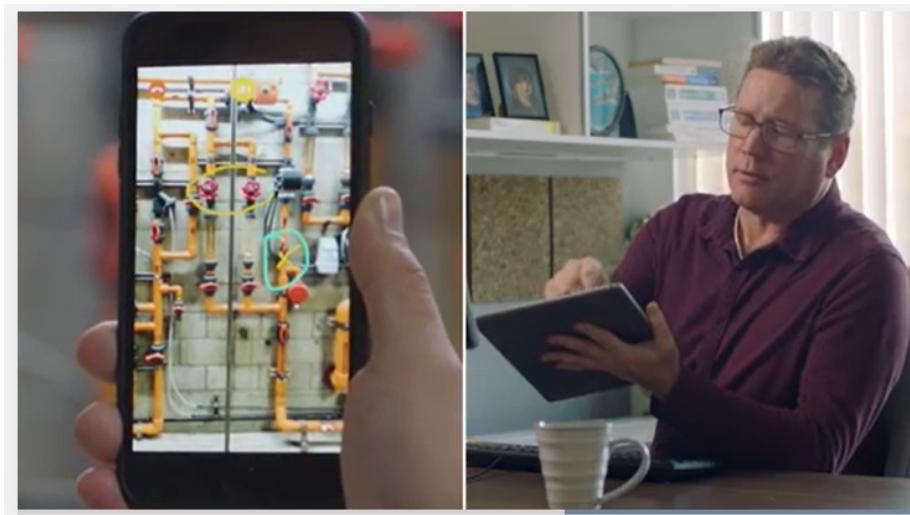


Figura 1.3: Ejemplo del uso de la RA en asistencia remota.

Mejora de la productividad: La RA transforma la capacitación y el aprendizaje, permitiendo la creación de tutoriales interactivos en tiempo real. Esto acelera la formación de operarios, reduciendo así el tiempo que los expertos deben invertir en enseñar, lo que se traduce en un aumento significativo de la productividad.

Eficiencia y eficacia: En el ámbito de la fabricación, las guías de montaje en realidad aumentada optimizan los procesos de ensamblaje al proporcionar instrucciones visuales precisas. Esto no solo acelera la producción, sino que también minimiza errores y mejora la calidad del producto final.

Prevención de errores: En el mantenimiento y control de planta, la RA ofrece una visión detallada del estado de las máquinas mediante hologramas. Esto permite a los operarios identificar problemas potenciales y recibir notificaciones de errores en tiempo real, evitando fallas costosas y mejorando la eficacia de las operaciones.

Reducción de costes y puesta en marcha: La asistencia remota no solo simplifica la resolución de problemas, sino que también reduce los costes asociados con desplazamientos de expertos. La capacidad de recibir instrucciones interactivas a distancia agiliza las operaciones y disminuye los tiempos de inactividad. Además, una tasa de errores menor también se traduce en menores costes.

Optimización de inspecciones: La RA facilita las inspecciones a lo largo del ciclo de vida de los productos, desde la recepción de nuevas piezas hasta las revisiones regulares de mantenimiento. Proporciona información detallada y precisa, permitiendo a los encargados de mantenimiento identificar rápidamente defectos, comparar con modelos 3D y seguir guías interactivas para garantizar una inspección exhaustiva y evitar futuros errores.



Figura 1.4: Ejemplo del uso de la RA en control.

En resumen, la realidad aumentada no sólo transforma los procesos industriales, sino que también impulsa una serie de beneficios clave que van desde la formación eficiente hasta la prevención proactiva de problemas, mejorando la productividad y la rentabilidad en la industria.

2 Estado del arte

A continuación, revisaremos las soluciones comerciales más relevantes que integran el uso de la realidad aumentada en la industria. Si bien existen numerosas empresas y productos en el mercado, nos centraremos en tres de los principales actores en este campo: Vuforia de PTC, AircadaPro y tres soluciones del grupo Innovae.

A través de un análisis detallado de estas soluciones, exploraremos cómo cada una de ellas aprovecha la realidad aumentada para mejorar la eficiencia, la productividad y la calidad en los procesos industriales.

2.1 PTC

La empresa PTC proporciona varias soluciones relacionadas con la industria y uno de estos es el del uso de la RA en los procesos principales. Bajo el nombre de Vuforia, esta empresa ofrece cuatro soluciones que integran la realidad aumentada¹ en la industria:

Vuforia Expert Capture: Se trata de una solución² para crear tutoriales paso a paso sobre los distintos procesos y máquinas de la industria. Esto corresponde con el caso de entrenamiento y aprendizaje en el que se usa la realidad aumentada para entrenar y acelerar la formación de los trabajadores.

Vuforia Expert Capture permite incluir videos e indicaciones holográficas en tiempo real además de guiar al operario de un lugar a otro si así se requiere.

Ejemplo de uso: Una empresa de fabricación implementa Vuforia Expert Capture para entrenar a su personal en el proceso de ensamblaje de una nueva línea de productos. Los trabajadores pueden seguir un tutorial paso a paso en tiempo real mientras realizan la tarea, lo que les permite aprender de manera más efectiva y reducir el tiempo de capacitación.

Vuforia Chalk: Es una herramienta para pedir asistencia remota³. Permite comunicarse en tiempo real con hasta 5 personas compartiendo lo que el operario ve para que los otros participantes puedan dibujar indicaciones sobre el video y así

¹<https://www.ptc.com/es/technologies/augmented-reality>

²<https://www.ptc.com/es/products/vuforia/vuforia-expert-capture>

³<https://www.ptc.com/es/products/vuforia/vuforia-chalk>

poder guiar de una manera más efectiva. Como se ha explicado anteriormente, la principal ventaja sería la de evitar desplazamientos innecesarios, teniendo a los expertos accesibles desde cualquier lugar.

Ejemplo de uso: Un técnico de mantenimiento en una planta de fabricación encuentra un problema complejo en una máquina. Utiliza Vuforia Chalk para conectarse con un experto remoto que puede ver el problema a través de la cámara del teléfono del técnico. El experto utiliza las herramientas de dibujo de Vuforia Chalk para señalar y explicar visualmente cómo resolver el problema, lo que permite al técnico solucionarlo de manera rápida y efectiva sin necesidad de que el experto se desplace al sitio.

Vuforia Studio: Vuforia Studio⁴ es un software potente diseñado para crear soluciones de realidad aumentada que ayudan a entender mejor cómo operar, mantener y solucionar posibles errores en la industria. Esta herramienta permite incluir animaciones 3D y datos del negocio para enriquecer la experiencia del usuario y proporcionar información relevante de manera intuitiva.

Con Vuforia Studio, los usuarios pueden añadir modelos 3D de las máquinas, alertas, indicaciones y etiquetas con datos relevantes. Incluso, si el modelo 3D es lo suficientemente detallado e incorpora las piezas internas de la máquina, como en el caso de un motor, los usuarios pueden visualizar los componentes internos y añadir alertas para mostrar mensajes de error más precisos en caso de fallo de una pieza.

Ejemplo de uso: Una empresa de fabricación utiliza Vuforia Studio para crear una solución de realidad aumentada que permite a los operarios visualizar y comprender mejor el funcionamiento de una línea de producción compleja. Los operarios pueden interactuar con modelos 3D de las máquinas y recibir alertas en tiempo real sobre posibles problemas o tareas pendientes. Esto ayuda a mejorar la eficiencia operativa y reducir el tiempo de inactividad al permitir una respuesta más rápida a los problemas en el proceso de producción⁵.

Vuforia Engine: Vuforia Engine⁶ es una herramienta versátil que permite crear soluciones de realidad aumentada de propósito general. Utiliza visión artificial para el reconocimiento de objetos y el escaneo de áreas interiores, lo que permite crear experiencias de realidad aumentada en diversos entornos y aplicaciones.

Aunque no está específicamente diseñado para la industria, Vuforia Engine puede ser utilizado en una amplia gama de aplicaciones y podría ser usado para crear aplicaciones en el ámbito de la industria.

⁴<https://www.ptc.com/es/products/vuforia/vuforia-studio>

⁵<https://www.ptc.com/-/media/Files/PDFs/Case-Studies/Howden-vuforia-studio-case-study-Feb-2019.pdf>

⁶<https://www.ptc.com/es/products/vuforia/vuforia-engine>

2.2 Aircada Pro

Aircada Pro⁷ es una aplicación diseñada para monitorear procesos industriales y máquinas, ofreciendo información en tiempo real y utilizando inteligencia artificial para el reconocimiento de objetos.

Esta plataforma permite crear y asignar gráficos personalizados a distintas máquinas, ofreciendo funcionalidades avanzadas como la visualización de modelos 3D, configuración de alarmas, guías para procesos, entre otros.

Cuando se activa una alarma, Aircada Pro es capaz de mostrar una flecha guía que dirige al usuario hacia la máquina afectada, proporcionando instrucciones detalladas paso a paso sobre las acciones necesarias.

Además, Aircada Pro facilita la creación de mapas detallados de las instalaciones industriales, lo que permite a cualquier persona localizar objetos específicos y navegar hacia ellos de manera eficiente, lo que mejora la productividad ya que los operarios no pierden tiempo buscando o preguntando en qué dirección está cierta máquina.

Esta aplicación también ofrece funcionalidades de realidad virtual para crear réplicas virtuales de las instalaciones, que puede ser usado para la formación y capacitación del personal en un entorno simulado.

En definitiva, se trata de una solución que resuelve de manera satisfactoria los principales casos de uso vistos en el capítulo anterior.

2.3 Innovae

Se trata de una empresa⁸ especializada en realidad aumentada y virtual que integra estas tecnologías en el sector educativo, marketing y, por supuesto, en la industria. En el ámbito de la industria, ofrece tres productos para resolver tres casos de uso distintos:

1. Procesos guiados con Realidad Aumentada, SAAM: Este sería el análogo de Vuforia Expert Capture y Vuforia Studio. Las principales áreas de aplicación de SAAM⁹ son las siguientes:

Mantenimiento preventivo: Para el mantenimiento periódico de infraestructuras y equipos, SAAM da instrucciones claras de cómo llevar a cabo el mantenimiento y registra errores o fallos de manera centralizada.

⁷<https://aircada.com/>

⁸<https://www.innovae.com/>

⁹<https://www.innovae.com/saam-sistema-avanzado-asistencia-mantenimiento/>

Formación: A través de indicaciones digitales, aquellos operarios en formación pueden realizar tareas complejas con eficiencia y seguridad. Además, la actividad de cada usuario es trazable para su supervisión.

Tareas de inspección y auditoría: Para evaluar de manera eficaz el estado de las instalaciones y activos, la solución ofrece una guía integral que permite que incluso operarios noveles lleven a cabo el proceso sin fallos.

Tareas de montaje: El sistema facilita el ensamblaje de piezas de componentes con indicaciones de cada paso a tomar y asegura la conformidad con los estándares de calidad.

2. Asistencia técnica remota con Realidad Aumentada, ATR: No voy a entrar mucho en detalle ya que se trata de una aplicación muy similar a la introducida con Vuforia Chalk pero si que voy a indicar las características¹⁰ propias que introduce:

Multidispositivo: Desarrollado con tecnología web, es compatible con cualquier dispositivo (Móvil, Tablet, PC, Smart Glasses) sin necesidad de instalar ninguna aplicación.

Traducción simultánea: Incorpora un sistema avanzado de traducción en tiempo real a más de 40 idiomas.

Compartición de ficheros: Permite el envío y compartición de imágenes, vídeos y archivos en tiempo real durante la sesión de asistencia remota.

Grabación de vídeos: Facilita la grabación completa de las sesiones de asistencia remota, capturando tanto el audio como la imagen, junto con las instrucciones visuales añadidas durante la sesión.

Creación de tickets de incidencias: Permite a los técnicos generar tickets de incidencias durante la sesión, lo que facilita la gestión y seguimiento de los problemas identificados.

Generación de informes técnicos: Automatiza la creación de informes colaborativos detallados que documentan las acciones realizadas durante la asistencia remota, proporcionando una base para la resolución eficiente de problemas.

3. Formación con Realidad Virtual, SAVI: Aunque la realidad virtual no es el objetivo de este TFG, si que me pareció interesante hablar brevemente sobre esto. SAVI¹¹ ofrece una serie de entornos virtuales para formar a los trabajadores en distintos escenarios y también se puede pedir la creación de un escenario personalizado.

¹⁰<https://www.innovae.com/atr-asistencia-remota/>

¹¹<https://www.innovae.com/savi-formacion-realidad-virtual/>

Tabla 2.1: Tabla comparativa de las distintas soluciones

Solución	Caso de Uso					
	Entrenamiento	Fabricación	Asistencia Remota	Inspección	Mantenimiento y control	Interaccion
Vuforia	Si	Si	Si	Si	Si	No
AircadaPro	Si(y VR)	No	No	Si	Si	No
Innovae	Si(y VR)	Si	Si	Si	Si	No
HoloScada	No	No	No	No	Si	Si

SAVI también es compatible con experiencias room-scale, las cuales permiten a los usuarios formarse en una misma experiencia virtual de manera presencial. Algunos de los escenarios que ya incluyen van desde extinción de incendios hasta prevención de riesgos en planta.

2.4 Comparativa final

Después de analizar las distintas soluciones podemos ver como en el fondo las tres son muy similares y tratan de resolver el mismo problema. Para comparar las distintas alternativas utilizaremos los principales casos de uso como guía.

En la tabla 2.1 se presenta una comparación entre las soluciones analizadas en relación con los casos de uso expuestos anteriormente en el apartado 1.1. Aunque las tres soluciones cubren la mayoría de los casos de uso mencionados, cada una aporta distintas funcionalidades y aborda los problemas de manera diferente. Por ejemplo, Innovae incorpora traducción en la asistencia remota, mientras que AircadaPro ofrece navegación para guiar al operario hacia una alarma o lugar específico.

Aunque nuestro producto está actualmente lejos de poder competir con las soluciones expuestas en este apartado y de satisfacer plenamente alguno de los casos de uso, observamos en la tabla que podemos aportar algo que el resto de las soluciones no incorporan, y eso está relacionado con la interacción.

Todas las soluciones mostradas (y más que puedan existir que no se hayan discutido en este documento) comparten la misma idea: mostrar los datos de una manera más intuitiva y añadir información adicional para facilitar el trabajo de los operarios en planta. Básicamente, imprimir hologramas que aporten información relevante y guíe en la resolución de problemas.

Sin embargo, la mayoría de estas soluciones actúan como elementos pasivos que solo aportan información sin permitir una interacción directa con el mundo real. HoloScada va un paso más allá: no solo incluimos guías o alarmas que notifican errores, sino que

también permitimos la interacción con las máquinas a través de hologramas.

3 Análisis de objetivos y metodología

3.1 Objetivos y contribuciones

El objetivo de este TFG, como se menciona en la introducción, será el de desarrollar una serie de escenarios (dos en concreto) en los que se proporcionarán una serie de etiquetas y botones que permitan al usuario ver información relevante en tiempo real y poder interactuar con la máquina a través de la RA.

Como hemos mencionado en el apartado 4.11, el factor diferenciador de HoloScada es el de poder interactuar con las máquinas. Esto puede ser poner en marcha o apagar la máquina o establecer parámetros de entrada como el porcentaje de llenado en un tanque de agua.

Este será el principal caso de uso que se desarrollará en este TFG. Crearemos una aplicación para las Hololens 2 que demuestre como podemos interactuar con los procesos industriales a través de la RA, además de incluir información para monitorizar y comprender el proceso.

3.2 Metodología

El desarrollo va a estar orientado a resolver un caso de uso, es decir, montar dos escenarios de prueba (proceso industrial), automatizarlos y crear un SCADA con la tecnología de realidad mixta.

La metodología seguirá un enfoque iterativo e incremental, estableciendo tareas intermedias para construir poco a poco el producto final. Los pasos principales son los siguientes:

- 1. Añadir un botón a la escena y que al pulsar en él se escriba un valor en un PLC.** Usando las operaciones de escritura de la librería S7NetPlus, desarrollaremos un script que ejecute una llamada para escribir un valor arbitrario en un PLC, vinculado a un botón en la escena de Unity. Comprobaremos el valor escrito en el PLC utilizando una herramienta como TIA Portal V16.
- 2. Añadir una etiqueta y que se escriba en esta el valor de una variable del PLC.** Sería muy similar al caso anterior pero esta vez leeremos del PLC un

valor y lo mostraremos por pantalla, asignando el valor leído a un objeto etiqueta de Unity.

3. **Escanear QR.** Usando alguna librería, ser capaces de leer códigos QR y mostrar su información. Con esto ya tendríamos la capacidad de poder localizar objetos.
4. **Mostrar los elementos al escanear un QR cualquiera.** El objetivo final es el de asignar un QR a cada objeto físico real y que al leerlo se muestre su SCADA virtual propio. Para esto tenemos que ser capaces de ocultar los objetos (botón y etiqueta) por defecto y mostrarlos cuando se lea un QR.
5. **Asignar varias escenas a distintos QR y mostrar la escena correcta al visualizar su QR.** En este paso no mostraremos los objetos al leer un QR cualquiera, sino que asociaremos cada escena a un QR para que esta se muestre u oculte en función del valor del QR leído.
6. **Creación de un gemelo digital en Factory I/O.** Para simular un caso de uso se hará uso de una aplicación que contiene varias escenas virtuales. Se escojerá una escena y se programará un PLC para ejecutar el proceso industrial de la escena.
7. **Crear el SACADA virtual con todos los botones y etiquetas necesarias.** Añadir más botones y etiquetas para mostrar todos los datos relevantes y permitir al usuario un mínimo de interacción con el proceso desarrollado en el paso anterior.
8. **Mejorar la estética de la interfaz.** Rediseñaremos las escenas para cambiar los botones 3D y etiquetas por una interfaz de usuario más estética y funcional.

4 Diseño y resolución del trabajo realizado

4.1 Herramientas utilizadas

Antes de abordar la resolución del trabajo realizado vamos a introducir brevemente la principales herramientas que se usarán en el desarrollo de la aplicación:

Unity: Unity¹ es un motor de desarrollo de videojuegos y plataforma de creación de contenido interactivo en 2D y 3D. Ampliamente utilizado en la industria de la realidad aumentada (RA), proporciona un entorno robusto para el diseño y desarrollo de aplicaciones inmersivas y soporte para el uso de dispositivos de RA como las Hololens.

C#: C# es un lenguaje de programación desarrollado por Microsoft dentro de su plataforma .NET. Orientado a objetos, es el lenguaje de script utilizado para el desarrollo en Unity.

Visual Studio: Visual Studio es un entorno de desarrollo integrado (IDE) utilizado para crear aplicaciones para diversas plataformas, incluida la realidad aumentada. Ofrece herramientas avanzadas de depuración y diseño para desarrolladores e incorpora integración con el editor de Unity.

Mixed Reality ToolKit 3 (MRTK 3): MRTK 3² es un conjunto de herramientas de código abierto desarrollado por Microsoft para facilitar la creación de experiencias de realidad mixta. Ofrece componentes y scripts reutilizables para plataformas como HoloLens y dispositivos de realidad mixta de Windows.

S7NetPlus: S7NetPlus³ es una librería para el lenguaje de programación C# que facilita la comunicación con controladores Siemens S7 (PLC) en entornos de automatización industrial. Ofrece una serie de métodos para crear y comunicarse con distintos PLC.

¹<https://unity.com/>

²<https://learn.microsoft.com/es-es/windows/mixed-reality/mrtk-unity/mrtk3-overview/>

³<https://github.com/S7NetPlus/s7netplus>

QRCode: Es la librería⁴ que nos permite trabajar con lector QR, acceder a los QR leídos y suscribirnos a los eventos de detección de un QR.

Hololens 2: Dispositivo de realidad aumentada creado por Microsoft⁵. Ofrece capacidades avanzadas de visualización holográfica y será el dispositivo para el que se desarrolle la aplicación.

TIA Portal V16: El Totally Integrated Automation (TIA) Portal V16 es un entorno de ingeniería que combina diferentes herramientas para la programación y configuración de sistemas de automatización industrial, compatible con Siemens. Con esta plataforma se crearan los programas que luego se cargan en los PLC.

NetToPlcSim: NetToPlcSim⁶ es un software que permite conectar a la red un PLC simulado mediante S7-Plcsim. Con esto podemos usar los PLC virtuales de TIA Portal para hacer pruebas.

Factory I/O: Factory I/O⁷ es una simulación de fábrica en 3D diseñada para aprender tecnologías de automatización. Diseñado para ser fácil de usar, permite construir rápidamente una fábrica virtual utilizando una selección de piezas industriales comunes.

4.2 Resolución del trabajo

Una vez introducidas las distintas herramientas/tecnologías que vamos a usar para el desarrollo del proyecto, pasaremos a exponer como se han resuelto los distintos objetivos/pasos que hemos comentado en el capítulo anterior:

Tarea 1: Escribir un valor en un PLC

El primer paso será el de añadir un botón a la escena. En la parte izquierda de la ventana, iremos a la sección "Packages" y seleccionamos *MRTK UX Components -> Button -> Prefabs* y arrastramos a la escena el objeto llamado *PressableButton* (ver figura 4.1).

Para añadir la interactividad tenemos primero que crear un script en el que definir el comportamiento y añadir una función que haga lo que queremos. En el menú superior *Assets -> Create -> C# Script*, indicamos el nombre y lo editamos desde Visual Studio o cualquier otro editor. Para vincular el script al botón, seleccionaremos el botón en la escena y en el menú de la derecha buscaremos la opción "Add Component", escribimos

⁴<https://www.nuget.org/Packages/Microsoft.MixedReality.QR>

⁵<https://www.microsoft.com/es-es/hololens>

⁶<https://nettoplcsim.sourceforge.net/>

⁷<https://factoryio.com/>

el nombre del script y aceptamos. Luego en el mismo menú, bajo la sección Presable Button, añadiremos en la lista *OnClicked()* la función del script (ver figura 4.2). En el código 4.1 podemos ver como escribir el valor 45 en la variable del PLC alojada en el primer bloque de datos (DB1) en la dirección 2.0.

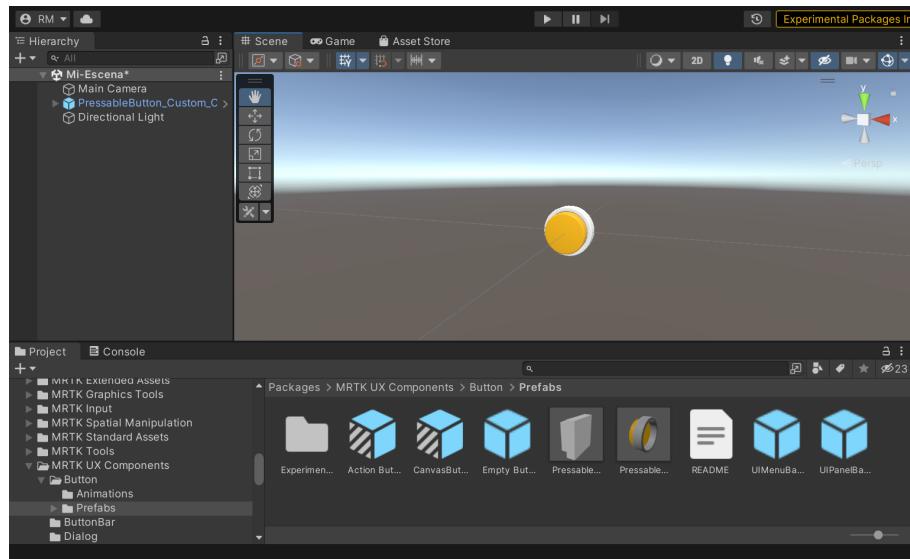


Figura 4.1: Añadir botón a la escena

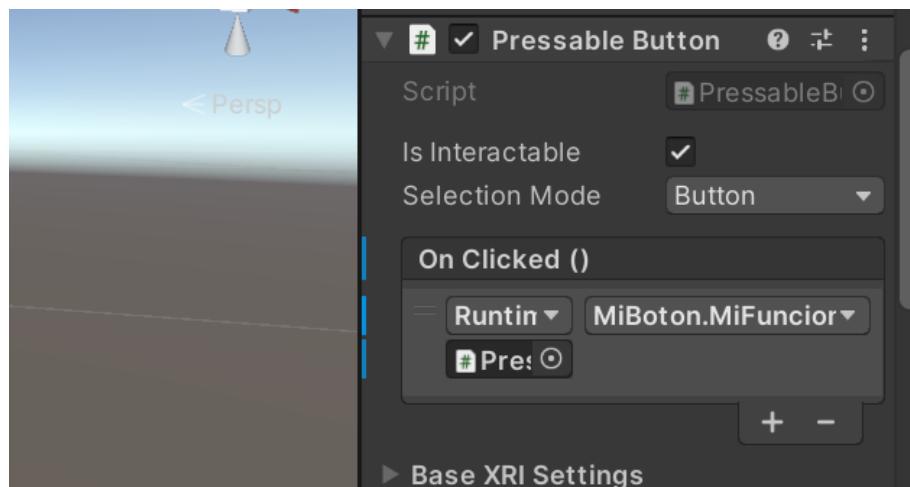


Figura 4.2: Añadir OnClick al botón

Código 4.1: Ejemplo código para escribir en un PLC

```
1 public void MiFuncion()
2 {
3     Plc plc = new Plc(CpuType.S7300, "127.0.0.1", 0, 2);
```

```

4     plc.Open();
5
6     if (plc.IsConnected)
7     {
8         plc.Write("DB1.DBW2.0", 45);
9         plc.Close();
10    }
11}

```

Tarea 2: Leer una variable del PLC

Este caso es muy similar al anterior: añadimos un objeto a la escena, creamos un script y lo asociamos al objeto. Sin embargo, en este caso añadiremos una etiqueta y la función tendrá que ejecutarse automáticamente cada cierto tiempo. La escena quedaría como se ve en la figura 4.3 y el código asociado 4.2. Véase que la función *Update()* se ha incluido en el mismo script (nombre MiBoton.cs) que la función *MiFuncion()* del código 4.1

En Unity, todas las clases que heredan de MonoBehaviour tienen la función *Update()* que se ejecuta en cada ciclo de renderizado si la clase está asociada a un GameObject, en este caso la etiqueta *TextMeshPro*. El código es muy simple: le pedimos al gameObject al que estamos asociados el objeto de tipo *TextMeshPro* (la etiqueta), leemos la variable del PLC y el valor se lo asignamos a la etiqueta. En nuestro caso el valor leído será 0 hasta que pulsemos el botón, lo que escribirá el valor 45.

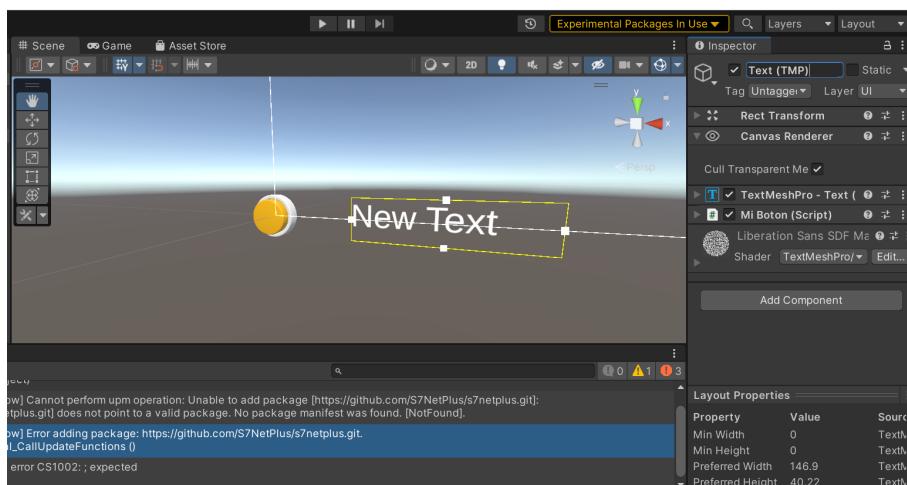


Figura 4.3: Añadir etiqueta a la escena.

Código 4.2: Ejemplo de código para leer de un PLC

```

1 void Update()
2 {

```

```

3   TextMeshProUGUI label = gameObject.GetComponent<TextMeshProUGUI>();
4
5   if (label != null)
6   {
7     Plc plc = new Plc(CpuType.S7300, "127.0.0.1", 0, 2);
8     plc.Open();
9
10    if (plc.IsConnected)
11    {
12      ushort valor = (ushort)plc.read("DB1.DBW2.0");
13      plc.Close();
14
15      Debug.Log("Leemos " + valor);
16
17      label.text = valor.ToString();
18    }
19  }
20 }
```

Modelo de clases v1

Con los dos apartados anteriores vemos la necesidad de empezar a plantear un modelo de clases que contenga la lógica necesaria para escribir y leer datos en un PLC y definir las propiedades y variables del PLC.

Para definir la dirección IP, modelo, variables, etc, tenemos las clases *Clase_PLC* y *Atributo*:

Código 4.3: Clase Atributo

```

1 public Atributo(string nombre, VarType variable, ...)
2 {
3   this.Nombre = nombre;
4   this.Tipo_Variable = variable;
5   ...
6
7   this.dataItem = new DataItem()
8   {
9     DataType = this.Tipo_Dato,
10    VarType = this.Tipo_Variable,
11    DB = this.Direccion_Tipo_Dato,
12    BitAdr = this.BitAdr,
13    Count = this.Cantidad,
14    StartByteAdr = this.StartByteAdr,
15    Value = new object()
16  };
17 }
```

Atributo: La clase **Atributo** contiene la información de una variable del PLC, como el nombre de la variable, tipo de variable (entero, cadena, entero doble...), tipo de dato (normalmente bloque de datos, DB) y la dirección. Con todo esto se crea el objeto *DataItem* (ver código 4.3) que más adelante veremos su uso.

Clase_PLC: Por otro lado, **Clase_PLC** define la dirección IP, tipo de CPU y el propio PLC de la librería S7NetPlus. Además contiene un diccionario con todos los objetos del tipo *Atributo* visto antes y como clave de cada atributo un *string*(el nombre del atributo). Además proporciona distintas funciones para iniciar y cerrar una conexión, leer todas las variables, escribir una variable, obtener el valor de una variable, añadir un atributo, etc.

En la función *LeerVariables()* (código 4.4) podemos ver el uso de crear un *DataItem* en la clase *Atributo*. La librería S7NetPlus proporciona una función para leer y escribir múltiples datos de un PLC en una sola petición capturando la información de cada variable en una clase *DataItem*. Con esto obtenemos el *DataItem* de cada *Atributo* y lo guardamos en una lista para ejecutar la operación de lectura con esta lista.

Cabe mencionar que hay un límite de variables que se pueden leer en una petición, por esto es necesario trocear la lista para hacer varias peticiones.

Código 4.4: Función LeerVariables()

```

1 public void LeerVariables()
2 {
3
4     List<DataItem> dataitems = ObtenerDataItems();
5
6     int num_items = dataitems.Count;
7
8     double num_pasos = num_items*1.0 / READ_LIMIT;
9     num_pasos = Math.Ceiling(num_pasos);
10    //Debug.Log("Haremos pasos: " + num_pasos);
11
12    StartConnection();
13
14    // Recorrer la lista en pasos
15    for (int i = 0; i < num_pasos; i++)
16    {
17        // Calcular el índice inicial y final para cada paso
18        int indiceInicio = i * READ_LIMIT;
19        int indiceFinal = Math.Min((i + 1) * READ_LIMIT, dataitems.Count);
20
21        // Acceder a los elementos en el rango actual
22        List<DataItem> elementosDelPaso = dataitems.GetRange(indiceInicio, indiceFinal - indiceInicio);
23
24        // Imprimir los elementos del paso actual
25        //Debug.Log($"Paso {i + 1}: desde " + indiceInicio + " hasta " + indiceFinal);
26        // read the list of variables
27        My_plc.ReadMultipleVars(elementosDelPaso);
28
29        //Debug.Log("");
30    }
31    // close the connection
32    EndConnection();
33}
```

Con todo esto ya tendríamos los necesarios para poder leer y escribir datos en un PLC pero todavía nos quedaría crear las clases que consuman ese PLC. Aquí entrarían las clases *Escena_PLC*, *ObjetoEscritura* y *ObjetoLectura* que ya serían clases vinculadas a Unity:

Código 4.5: Objetos de escritura v1

```

1 public abstract class ObjetoEscritura : MonoBehaviour
2 {
3     public string nombreVariable;
4     public Escena_PLC escena;
5
6     void Start()
7     {
8         if (escena == null)
9             escena = GetComponentInParent<Escena_PLC>();
10
11        PressableButton boton = GetComponent<PressableButton>();
12        if (boton.OnClicked == null)
13        {
14            throw new Exception("Onclick no asignado para el botón " + boton.name);
15        }
16    }
17
18    public Escena_PLC getEscena()
19    {
20        return this.escena;
21    }
22
23    public abstract void EscribirValor();
24}
25
26
27 public class ObjetoEscrituraBoolean : ObjetoEscritura
28{
29    // Valor a escribir
30    public bool valor;
31
32    // If true, escribir valor y luego negarlo
33    public bool conmutar;
34    public override void EscribirValor()
35    {
36
37        if (conmutar)
38        {
39            this.getEscena().EscribirVariable(this.nombreVariable, valor.ToString());
40            valor = !valor;
41        }
42        else
43        {
44            this.getEscena().EscribirVariable(this.nombreVariable, valor.ToString());
45            this.getEscena().EscribirVariable(this.nombreVariable, (!valor).ToString());
46        }
47    }
48}
```

Escena_PLC: Esta clase se encarga de inicializar los datos del PLC y servirlo a los objetos de escritura y lectura, además de almacenar datos relevantes para la

escena como el QR asociado y el objeto 3D asociado.

Al asignar esta clase a un objeto de Unity podemos desde el editor, en la sección de la derecha, inicializar todas los atributos de la clase, como la dirección IP y los atributos del PLC en una lista. En esta clase los atributos se dividen en bloques de datos, al igual que en un PLC real, para mantener un orden y hacer la lista más legible aunque como ya hemos visto *Clase_PLC* tiene un diccionario con todos los atributos.

ObjetoEscritura: Los objetos de escritura definen la lógica necesaria para realizar escrituras en un PLC. Se trata de una clase base abstracta que contiene el nombre de la variable en la que se quiere escribir y la *Escena_PLC* a la que llamar para realizar las escrituras. La función *EscribirValor()* se define abstracta para que las subclases la implementen como sea conveniente.

En el siguiente código 4.2 podemos ver la sub-clase *ObjetoEscrituraBoolean* que, como su nombre indica, escribe un booleano en el PLC. Como esta clase se usa principalmente para activar señales en el PLC debemos de escribir el valor (true por ejemplo) y negarlo inmediatamente para reiniciar la señal.

Para iniciar un motor, el PLC recibirá una señal que se representa con un bit 0 o 1, 1 indicando que recibe la señal. Pero estas señales son momentáneas, se recibe la señal y el bit comuta a 1 iniciando el motor pero cuando se deja de recibir la señal el bit vuelve a 0. Por eso, para simular este comportamiento escribimos el valor y luego el negado del valor.

ObjetoLectura: Esta clase es la que se usa para leer una variable del PLC y escribir el valor en un campo de texto. En este caso la clase no es abstracta, se puede instanciar, pero define una función que puede ser sobreescrita por subclases. En la función *Update()* se realiza la lectura de la variable y el resultado se pasa a la función *GenerarCadena(string)* que devuelve otra cadena para asignarla a la etiqueta.

Esta función *GenerarCadena()* (ver código 4.6) es la que otras clases podrán redefinir para cambiar la cadena que devuelve. Por ejemplo, si el valor que se está leyendo corresponde a un booleano devolver una cadena si el valor leído es true u otra si es false.

Cabe mencionar que la función *LeerVariable()* devuelve el valor que tiene en ese momento el objeto *DataItem* asociado. Si todavía no se ha ejecutado la función *LeerVariables()* (recordar que esta dentro de la función *Update()* en *Esce-*

na_PLC) es posible que el valor que se lea este desactualizado. Sin embargo esto no es un problema por que en todo caso el valor estará desactualizado por un frame y esto para nosotros es imperceptible. La solución ideal sería la de usar un patrón Observer como se usa más adelante para otras funciones, pero por simplicidad se decide dejarlo así, por lo menos para este TFG.

Código 4.6: Objeto de lectura v1

```

1 public class ObjetoLectura : MonoBehaviour
2 {
3     public string nombreVariable;
4     public Escena_PLC escena;
5     public TextMeshProUGUI texto;
6     public bool read_auto = true;
7
8     void Start()
9     {
10         if (escena==null)
11             escena = GetComponentInParent<Escena_PLC>();
12
13         if (texto==null)
14             texto = GetComponent<TextMeshProUGUI>();
15     }
16
17     void Update()
18     {
19         // Actualizar el texto con el valor leído en cada frame
20         if (read_auto)
21         {
22             escena.PedirVariable(nombreVariable);
23             string valor = escena.LeerVariable(nombreVariable).ToString();
24             //Debug.Log("hemos leido: " + valor + " en: " + nombreVariable);
25             texto.text = this.GenerarCadena(valor);
26
27         }
28     }
29
30 }
31
32 protected virtual string GenerarCadena(string entrada)
33 {
34     string salida = entrada;
35
36     return salida;
37 }
38 }
```

Tarea 3: Escanear QR

Para escanear códigos QR haremos uso del paquete⁸ que proporciona Microsoft para detectar códigos QR en dispositivos para Realidad Mixta de Microsoft, como en este caso lo son las Hololens 2. Esta API no solo detecta los códigos, sino que también

⁸<https://www.nuget.org/Packages/Microsoft.MixedReality.QR>

obtiene la localización, tamaño y contenido de los códigos detectados.

Microsoft ya proporciona una implementación que utiliza esta librería para escanear códigos QR y dibujar objetos en la posición del QR. Utilizaremos esta implementación⁹ en nuestro proyecto y la modificaremos según sea necesario.

La implementación, bajo el namespace **QRTracking**, contiene 4 clases principales:

QRCode: Esta clase contiene los datos del QR para crear el objeto que se va a dibujar en la escena. Entre estos datos se encuentran el identificador del código, fecha de detección, el contenido, etc.

QRCodesManager: Actúa como el listener y observer. Inicializa el lector de QRs (*QRCodeWatcher*) y recibe de este los eventos cuando se añade, actualiza o elimina un QR. Cada vez que recibe un evento este añade, actualiza o elimina el QR de un diccionario propio en el que registra todos los QR y notifica del evento al *QRCodesVisualizer*.

QRCodesVisualizer: Esta clase es la que se encarga de instanciar el objeto y posicionarlo en la escena. Guarda un diccionario con los objetos instanciados asociados al identificador de su QR y los elimina o crea según el evento que recibe del Manager.

SpatialGraphNodeTracker: Al objeto 3D que crea el *QRCodesVisualizer*, se le asigna esta clase que recibe el identificador del *SpatialGraphNode* del código QR. Con este id recuperamos el objeto *SpatialGraphNode* que contiene la posición del QR y en la función *Update()* nos encargamos de ir actualizando la posición del objeto padre para que siempre este en la misma posición que la del código QR que estamos viendo. Internamente la librería se encarga de actualizar la posición del *SpatialGraphNode* cada vez que se vuelve a leer un QR o su posición cambia con respecto a la cámara, ya sea porque esta se está moviendo o el propio QR. Esto correspondería al evento *Update*.

Con todo esto conseguimos que al escanear un QR se dibuje el objeto que elijamos, tal y como se muestra en la figura 4.4. En este caso estamos dibujando un cuadrado y unas flechas indicando el eje de coordenadas.

Tarea 4: Mostrar los elementos al escanear un QR dado

Como ya hemos visto en el apartado anterior, si queremos dibujar un objeto en la posición de cierto QR, sólo tenemos que añadirle el script de *SpatialGraphNodeTracker*

⁹<https://github.com/y1-msft/QRTracking/tree/main/SampleQRCodes/Assets/Scripts>

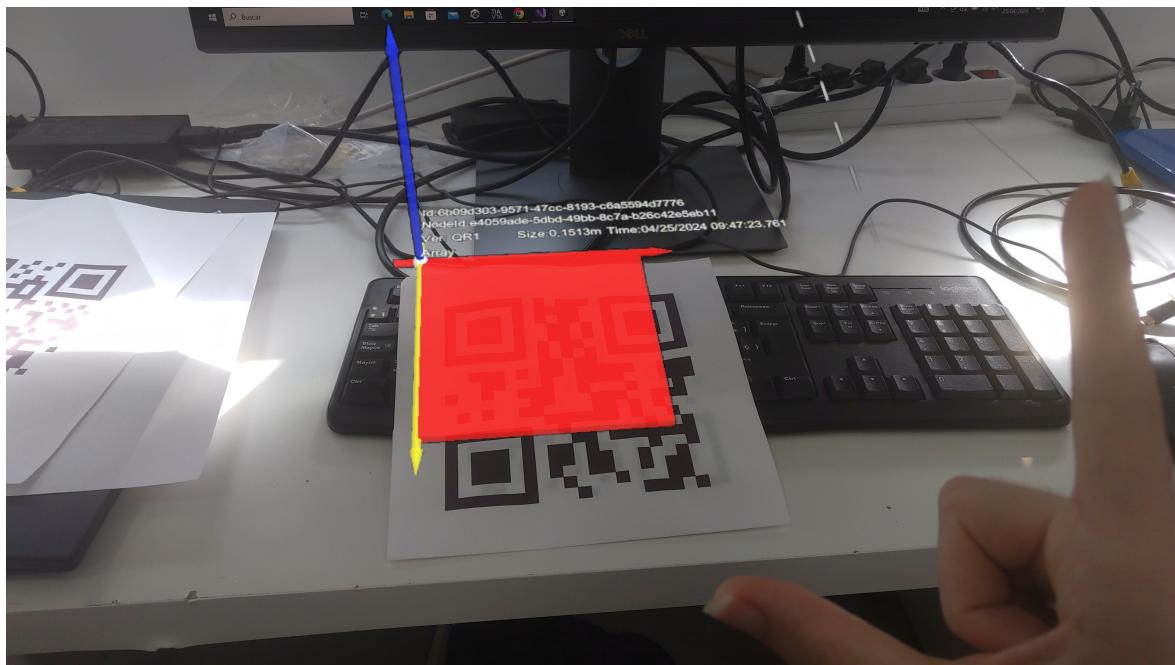


Figura 4.4: Dibujar objetos en QR

y pasarle el id del *SpatialGraphNode* de dicho código QR.

El objetivo ahora es el de añadir un objeto a la escena y ocultarlo, y sólo mostrarlo cuando se lea un QR cualquiera. Para esto podríamos modificar las operaciones que se ejecutan al recibir los eventos del *QRCodesManager* para que cuando se detecte un código se active un objeto con la llamada *this.objecto.setActive(true)*. Para esto tendríamos que añadir un nuevo atributo publico en la clase, de tipo *GameObject*, y desde el editor arrastrar el objetos que queremos activar al *QRCodesVisualizer*. Esto quedaría tal y como se muestra en la figura 4.5.

Para que todo esto funcione adecuadamente es necesario comprobar en *QRCodesVisualizer* la fecha¹⁰ de detección del QR y si es anterior a la fecha de inicio del *QRCodesManager* no hacer nada con ese QR. Esto se debe a que las Hololens 2 mantienen todos los QR leídos desde que se inicia el driver y si no hacemos este filtro, al iniciar la aplicación, el lector de QR empezará a enviar a nuestro Manager los eventos de los códigos que tiene almacenados y se empezarán a dibujar en pantalla todos estos códigos.

¹⁰<https://learn.microsoft.com/en-us/windows/mixed-reality/develop/advanced-concepts/qr-code-tracking-overview#managing-qr-code-data>

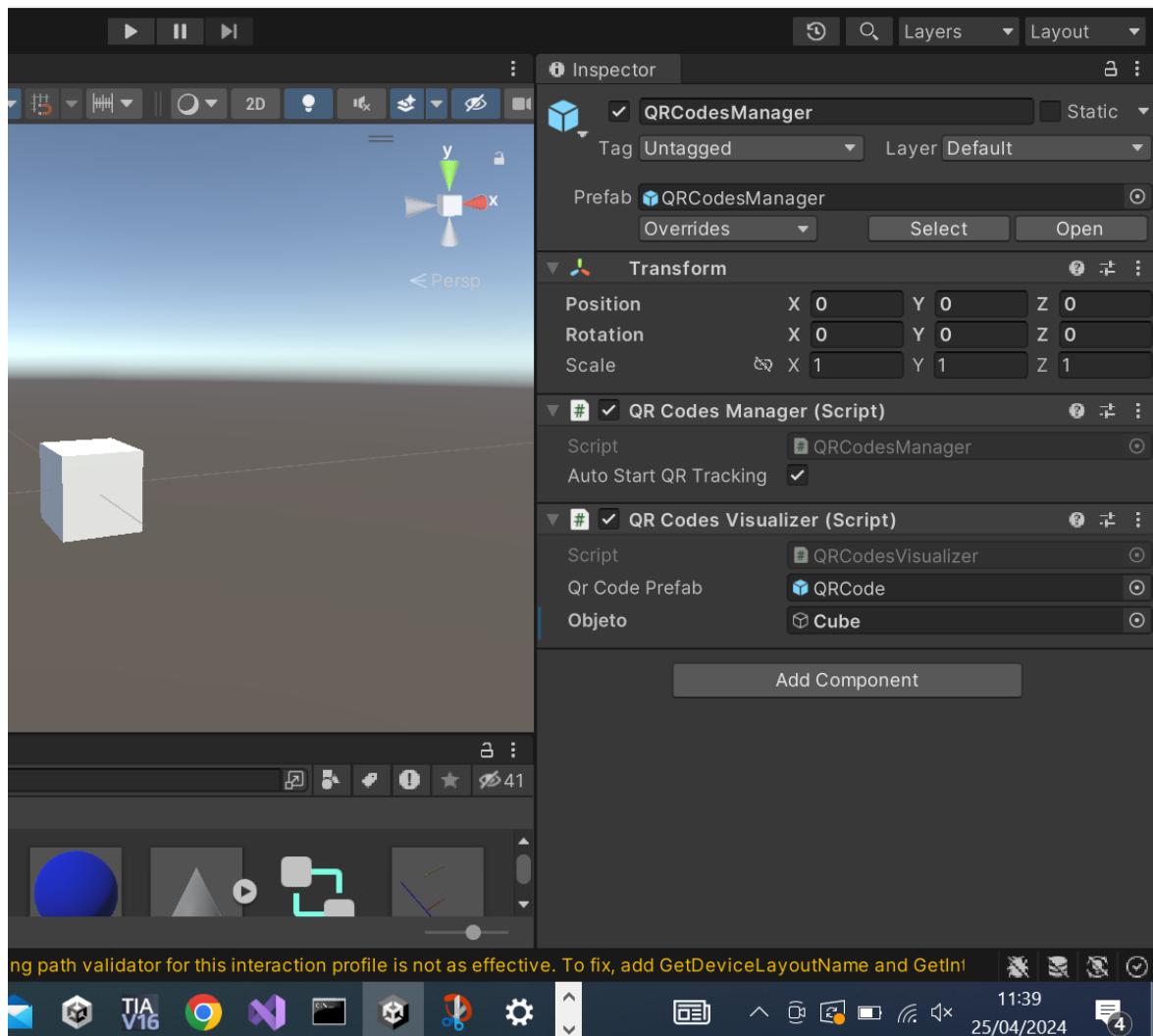


Figura 4.5: Añadir objeto al visualizer

Tarea 5: Mostrar la escena correcta al leer un QR

La solución propuesta en el apartado anterior nos vale si sólo tenemos un objeto y queremos que se active al leer cualquier QR. Sin embargo, nosotros tenemos ahora varios objetos y queremos que se muestren sólo cuando se lea cierto QR.

Para resolver esto vamos a hacer que cada vez que se lea un código, *QRCodeVisualizer* notifique un evento y desde otra clase manejarlo. Primero creamos una clase que represente el evento, en el que añadiremos el valor del QR y el id del *SpatialGraphNode* asociado (esto nos servirá para más adelante).

En la clase *QRCodeVisualizer* añadimos un nuevo atributo público del tipo *EventHand-*

ler<HoloEventArgs> para que otras clases puedan suscribirse.

Por último, creamos la clase que va a manejar estos eventos. Esta clase va a contener una lista con todos los objetos del tipo *Escena_PLC* que queremos que se activen/desactiven en función del código leído. Al recibir un evento esta clase recorrerá la lista de escenas redirigiendo el evento para que cada *Escena_PLC* se active o desactive si el valor del QR es el que esperaban.

Código 4.7: Código evento QRLeido

```

1 public class HologramasManager : MonoBehaviour
2 {
3
4     public List<Escena_PLC> escenas = new List<Escena_PLC>();
5     public QRCodeVisualizer LectorObjetos;
6
7     // Start is called before the first frame update
8     void Start()
9     {
10         LectorObjetos.eventoHolo += QRLeido;
11     }
12
13     public void QRLeido(object sender, HoloEventArgs evento)
14     {
15         escenas.ForEach(escena => {
16             escena.Activar(evento);
17         });
18     }
19 }
20
21 public class Escena_PLC : MonoBehaviour
22 {
23     ...
24     public bool Activar(HoloEventArgs evento)
25     {
26
27         if(evento.ValorQR == this.qrcode_name)
28         {
29             gameObject.SetActive(true);
30
31             return true;
32         } else
33         {
34             // si se desactiva el objeto con esto activo luego los botones no funcionan
35             if(TryGetComponent<ObjectManipulator>(out var obj))
36             {
37                 obj.enabled = false;
38             }
39             gameObject.SetActive(false);
40
41             return false;
42         }
43     }
44 }
45 }
```

Para que esto funcione adecuadamente, la escena tendría que quedar como se muestra en la figura 4.6. Con esto los objetos Canicas y FactoryI/O recibirán los eventos y

se activarán o desactivarán en función del valor que tengan asignado en su variable `qrcode_name` que se puede modificar desde el editor de Unity.

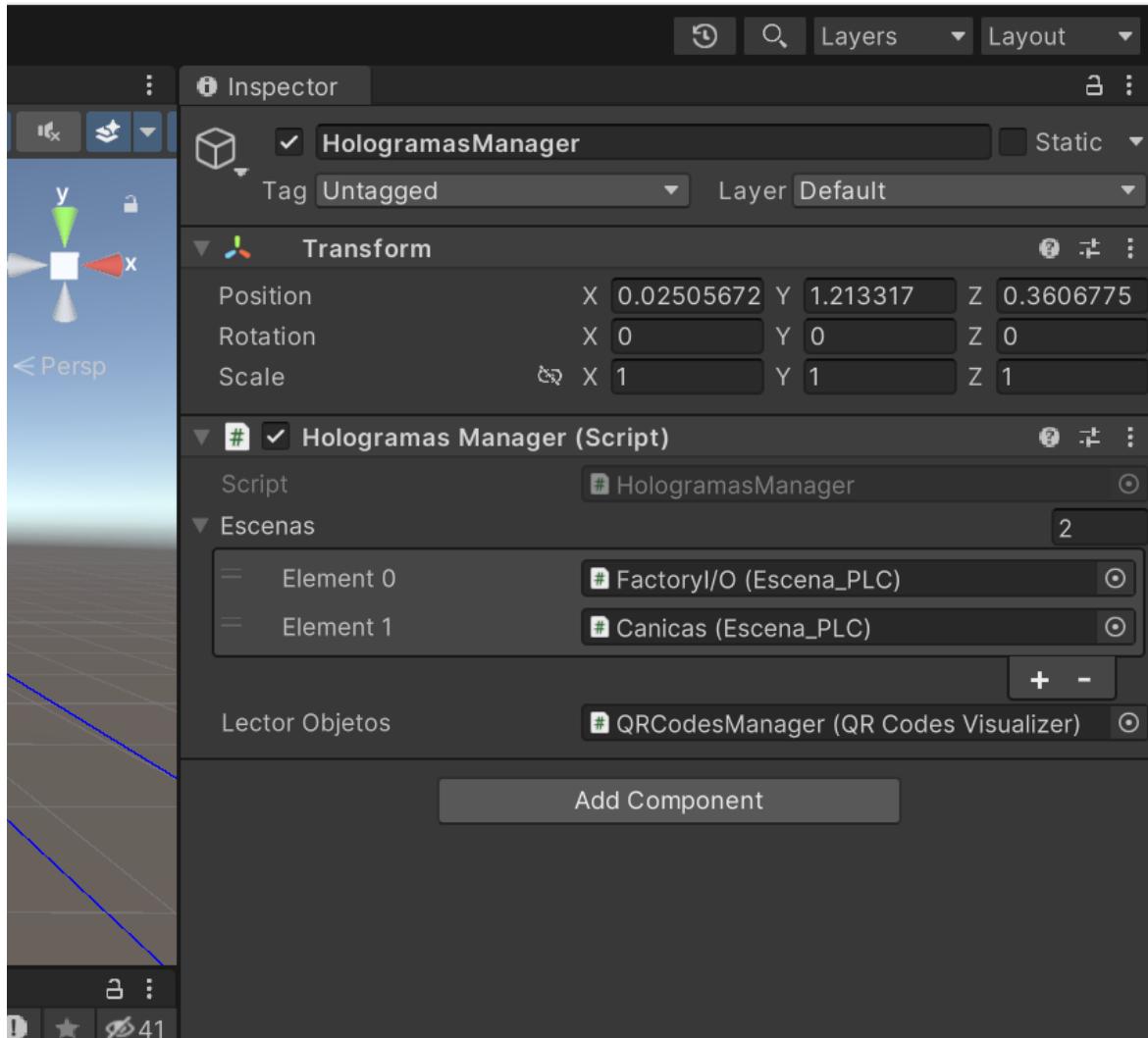


Figura 4.6: Configurar la Escena para los eventos

Tarea 6: Creación de un gemelo digital en Factory I/O

La configuración de los PLC y la preparación de la escena en Factory I/O es manejada por Binarii, en particular por un estudiante en prácticas. Aunque no entraremos en detalles sobre cómo se crea la escena en esta aplicación, se presentará el resultado final (ver figura 4.7) para comprender con qué estamos trabajando.

La escena diseñada es bastante sencilla: consiste en una máquina transportadora en

forma de "T" que redirige los palets de entrada hacia la izquierda o derecha.

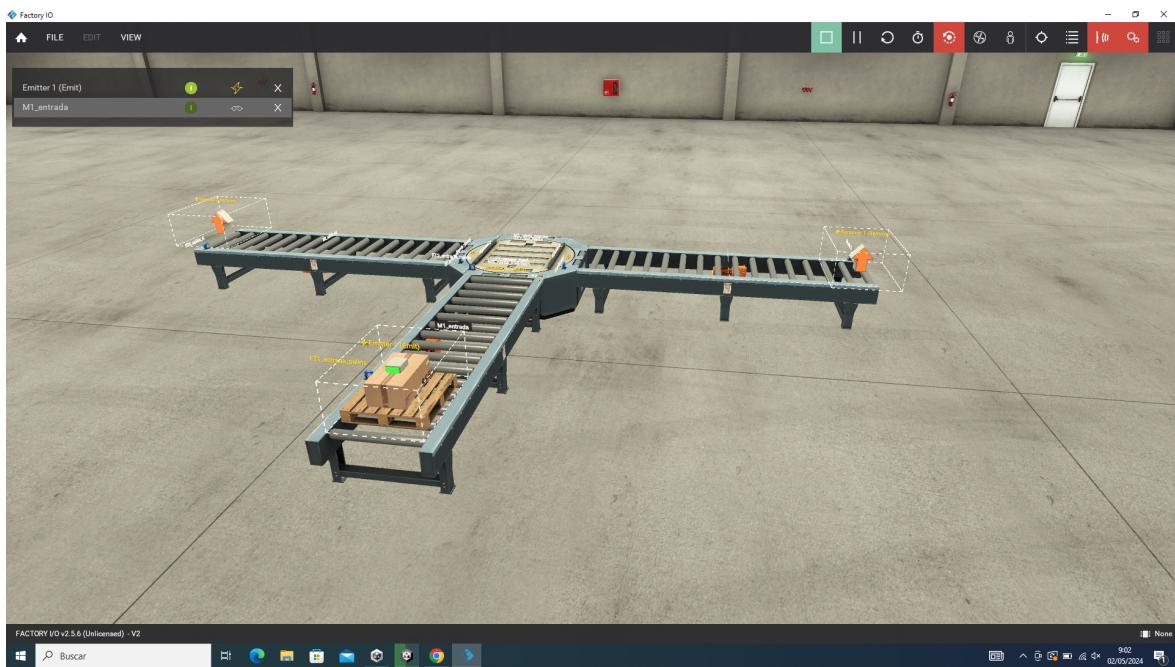


Figura 4.7: Escena virtual con factory I/O

Al tratarse de un entorno virtual, no vamos a poder dibujar hologramas anclados a la máquina, por lo que esto se abordará para la segunda escena, que se desarrollará con una maqueta.

Tarea 7: Creación del SACADA virtual

Con todo lo expuesto anteriormente, ya podríamos crear un panel de control básico para interactuar con un PLC y mostrar datos en pantalla.

Primero empezaríamos añadiendo un objeto vacío a la escena que llamaremos Panel1. En este objeto añadimos algunos botones y etiquetas para escribir datos en el PLC y mostrarlos al usuario. El siguiente paso sería el de añadir al objeto raíz la clase *Escena_PLC* y configurarla. Para esto especificamos la dirección ip, modelo, variables del PLC que queremos hacer accesibles y el valor del QR al que queremos responder.

Luego añadimos a los botones y etiquetas los objetos de lectura y escritura como corresponda (como se ve en la figura 4.9) y les asignamos el objeto raíz *Escena_PLC*, configurando el objeto como se ve en la figura 4.8

Por último, si queremos que los objetos respondan a los códigos QR, añadimos un objeto a la escena con el QRCodeManager y QRCodeVisualizer y otro con el Holo-

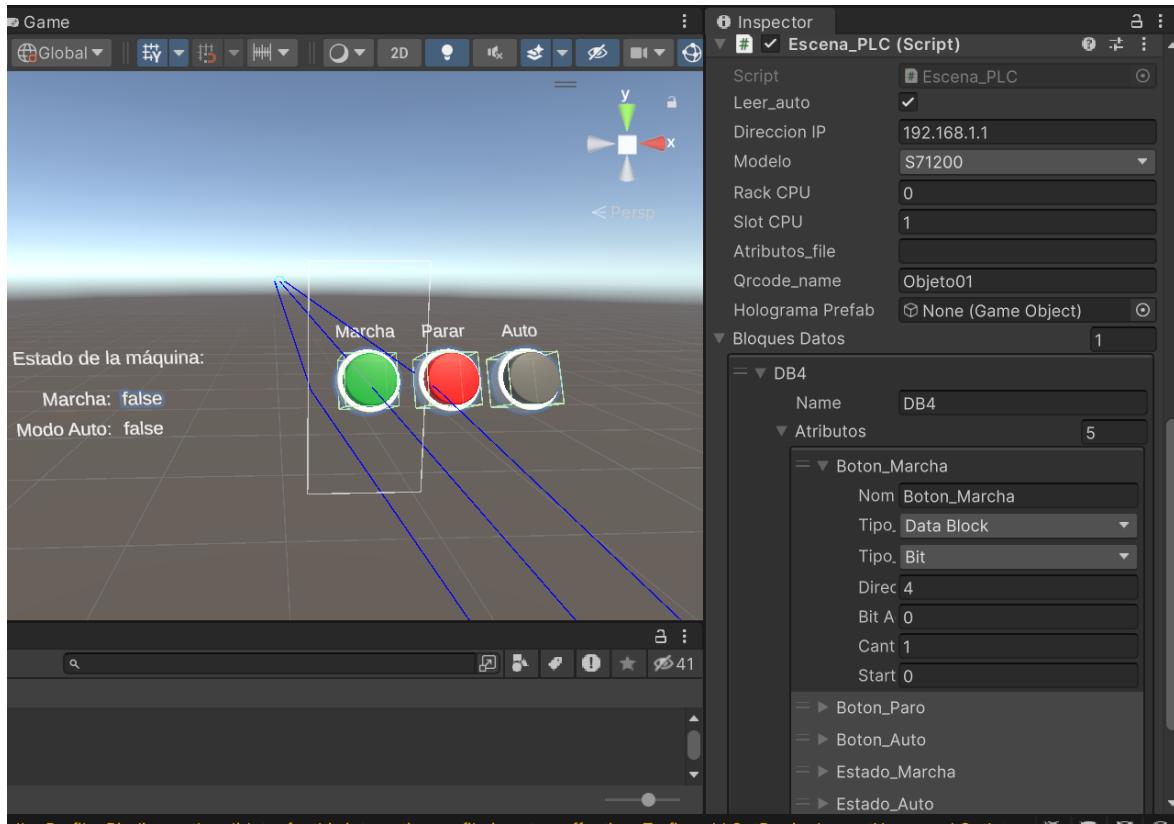


Figura 4.8: Configuración Escena_PLC

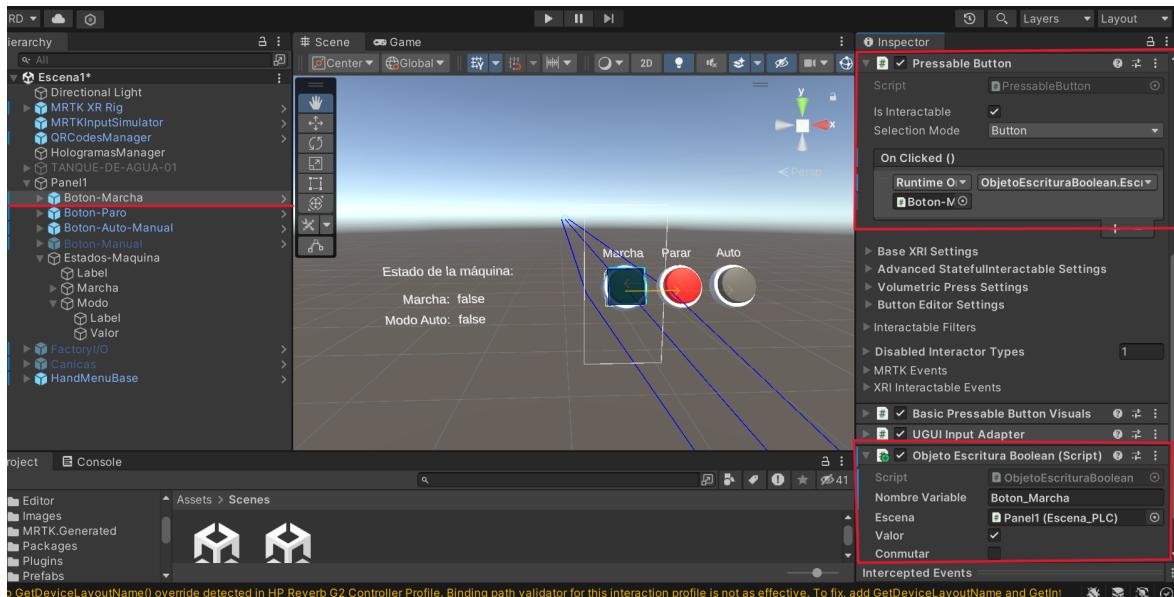


Figura 4.9: Configurar botón para que escriba en la variable Boton_Marcha

gramasManager tal y como se ve en las figuras 4.5 y 4.6 pero esta vez añadiendo en la lista de escenas nuestro objeto Panel1.

Con esto, ya tendríamos una escena funcional en la que podemos leer y escribir valores en un PLC y ocultar o activar la escena en función del QR que estemos leyendo.

Tarea 8: Mejorar la estética de la interfaz

En este último apartado vamos a discutir como mejorar las estéticas de la interfaz para presentar un producto más cuidado y profesional.

Actualmente estamos haciendo uso de los botones 3D que ofrece la librería MRTK3 y de etiquetas simples para mostrar texto. Pero con Unity se pueden crear interfaces más complejas y el objetivo es hacer uso de las herramientas que ofrece para crear un SCADA más elegante a la vista.

En nuestro caso, Microsoft lleva trabajando varios años para mejorar la experiencia de usuario en aplicaciones de realidad mixta y facilitar el trabajo a los desarrolladores para crear interfaces más ricas y expresivas. Con el lanzamiento de Mixed Reality Toolkit 3, redefinieron la librería de componentes, añadiendo mejoras y nuevas características que se integran de mejor manera con las herramientas que ofrece Unity para el desarrollo de UI's, incorporando además una lista de componentes ya fabricados proporcionando así un framework más completo para desarrollar interfaces de realidad mixta.

En la siguiente figura 4.10 tendríamos un ejemplo de lo que se puede hacer con la librería¹¹ ¹².

¹¹<https://learn.microsoft.com/es-es/windows/mixed-reality/mrtk-unity/mrtk3-uxcomponents/packages/uxcomponents/mixed-reality-design-language>

¹²<https://learn.microsoft.com/es-es/windows/mixed-reality/mrtk-unity/mrtk3-uxcomponents/packages/uxcomponents/button>



Figura 4.10: Ejemplos de interfaces usando MRTK3

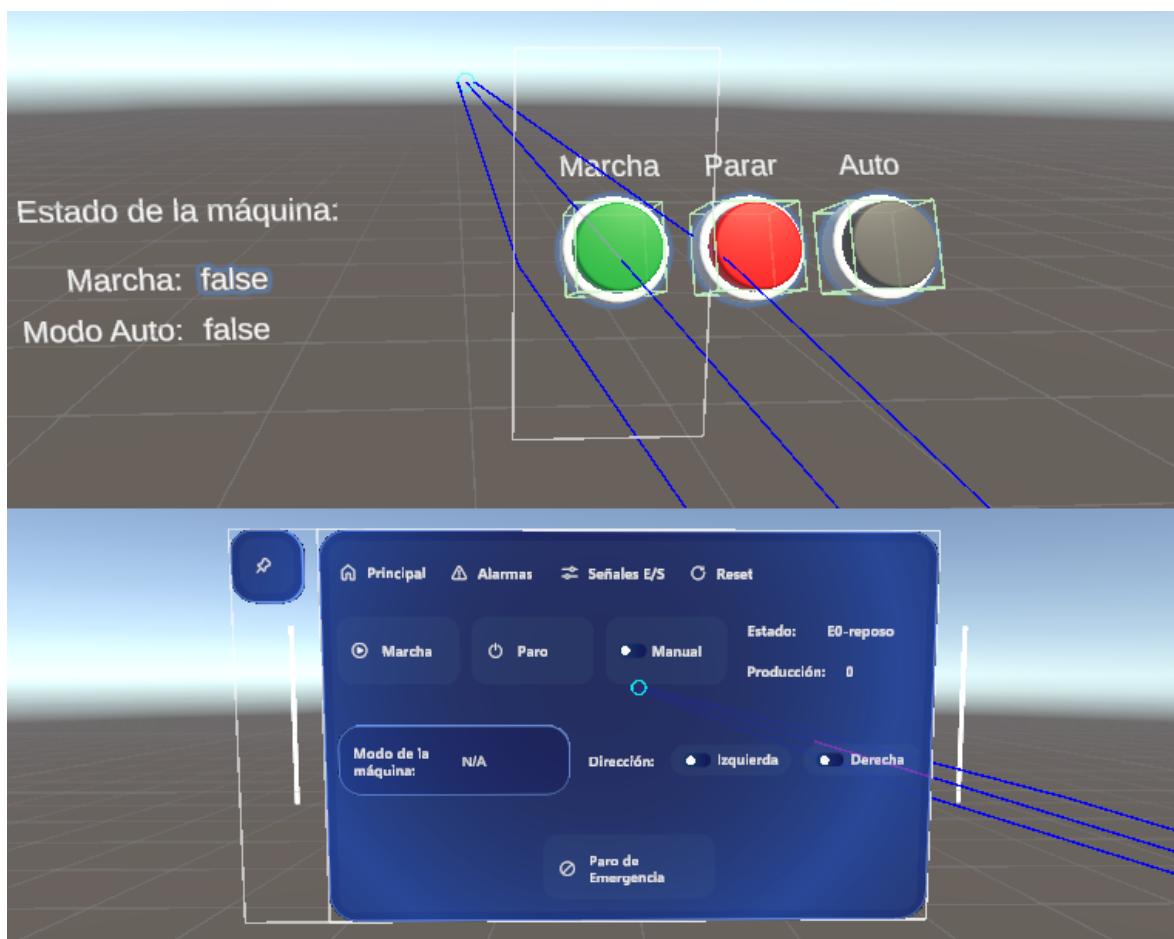


Figura 4.11: Antes y después del menú

Haciendo uso de la librería podemos ver como quedaría la comparación, en la figura 4.11, entre el primer panel que hemos creado en el apartado anterior y el nuevo haciendo uso de los componentes de MRTK3, que a parte de los visual también hemos añadido comportamiento adicional.

Para crear el nuevo panel tenemos que seguir algunas pautas y recomendaciones que marca Microsoft. Primero añadimos un objeto vacío a la escena que llamaremos FactoryI/O (desde este panel controlaremos el proceso que estamos simulando desde la aplicación del mismo nombre) y añadimos nuestra clase *Escena_PLC* (la misma que en la figura 4.8) y otras clases para añadir comportamiento adicional al panel, en este caso la clase *Radial View*. Esta clase permite que el objeto al que está vinculado siga al usuario permitiendo así que este pueda moverse por toda la escena sin perder de vista el menú.

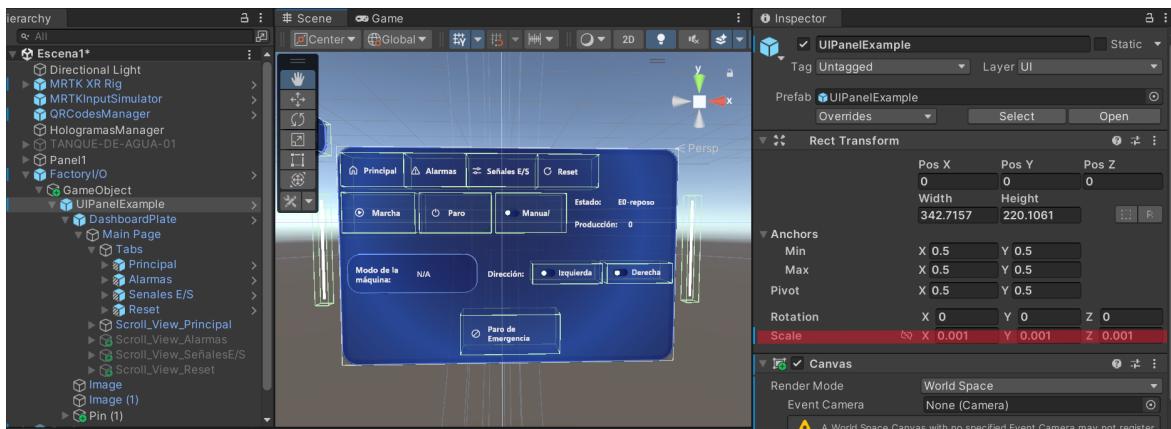


Figura 4.12: Establecer scala de 0.001

A continuación empezaríamos a añadir los componentes (botones y texto) ya creados para ir construyendo la interfaz. Para que los componentes se renderizan adecuadamente es necesario que estén contenidos en un objeto de tipo canvas de Unity. Además, Microsoft recomienda añadir el canvas dentro de un objeto que no sea un canvas para reducir redibujados y aumentar el rendimiento¹³, por lo que al objeto raíz añadiremos un objeto vacío y a este objeto el canvas. En este canvas establecemos la escala para todos los ejes a 0,001, como se ve en la figura 4.12, siguiendo las recomendaciones de la librería. Con esto estamos diciendo que una unidad de medida equivale a 1 milímetro, por lo que un objeto dentro del canvas que mida 300 ancho para el usuario serán 30cm de ancho.

¹³<https://learn.microsoft.com/en-us/shows/mixed-reality-dev-days/building-rich-ui-for-mr-in-mrtk3> minuto 31:18

Una vez hecho esto ya sólo faltaría construir la interfaz siguiendo la estructura de layouts verticales, horizontales o en grid, muy similar a como se diseñan interfaces en el mundo web o Android. El resultado se puede ver en la figura 4.13. Los objetos principales que estamos usando para componer la interfaz son botones simples (ver figura 4.14) y etiquetas de texto. En la figura 4.12 podemos ver más en detalle la jerarquía de los componentes en la que podemos ver bajo el objeto "Main Page" los objetos "Tabs" y cuatro scroll views. Haciendo uso de las clases que incorpora MRTK3 "Toggle Collection" y "Tab View" conseguimos que en función del botón que pulsemos (los que son hijos de Tabs), cambie el contenido de la página teniendo así distintas secciones.

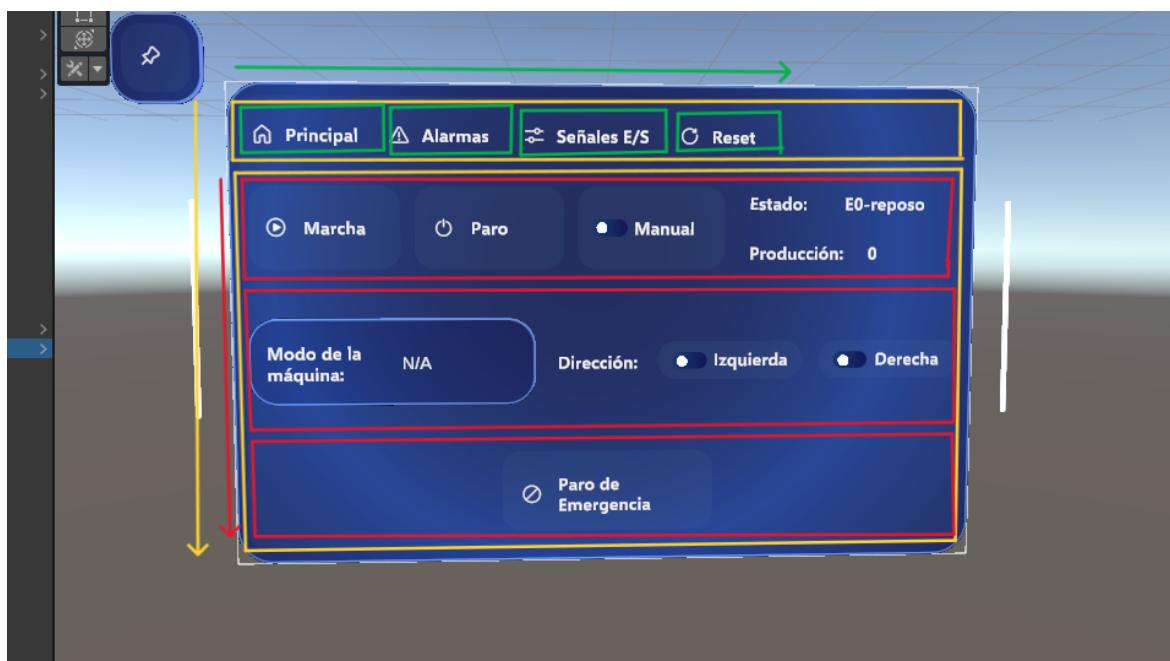


Figura 4.13: Layout del menú

Para hacer uso de estas clases sólo tenemos que añadirlas al objeto "Tabs" y vincular los botones con los scroll views tal como se muestra en la figura 4.15

Además, también hemos añadido un botón en la esquina superior izquierda que permite anclar el menú en la posición en la que se encuentra para que ya no siga al usuario.

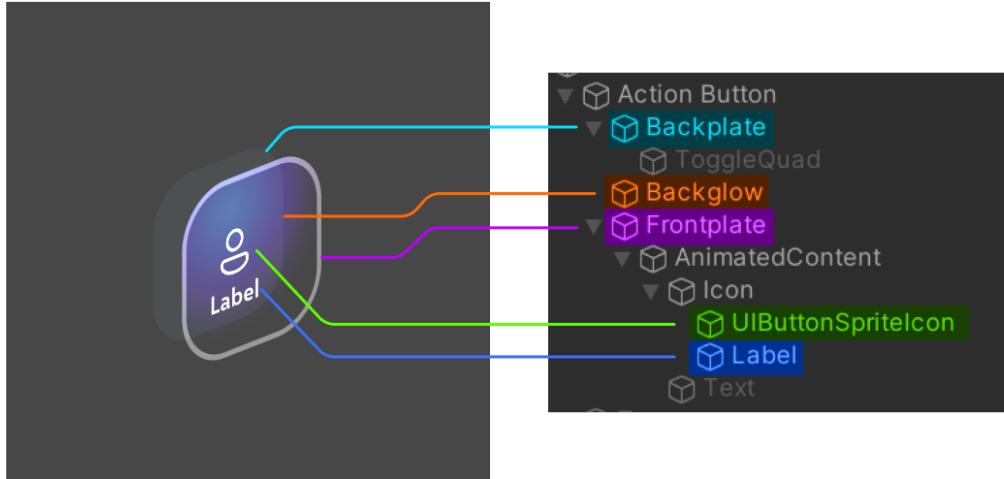


Figura 4.14: Estructura de los botones

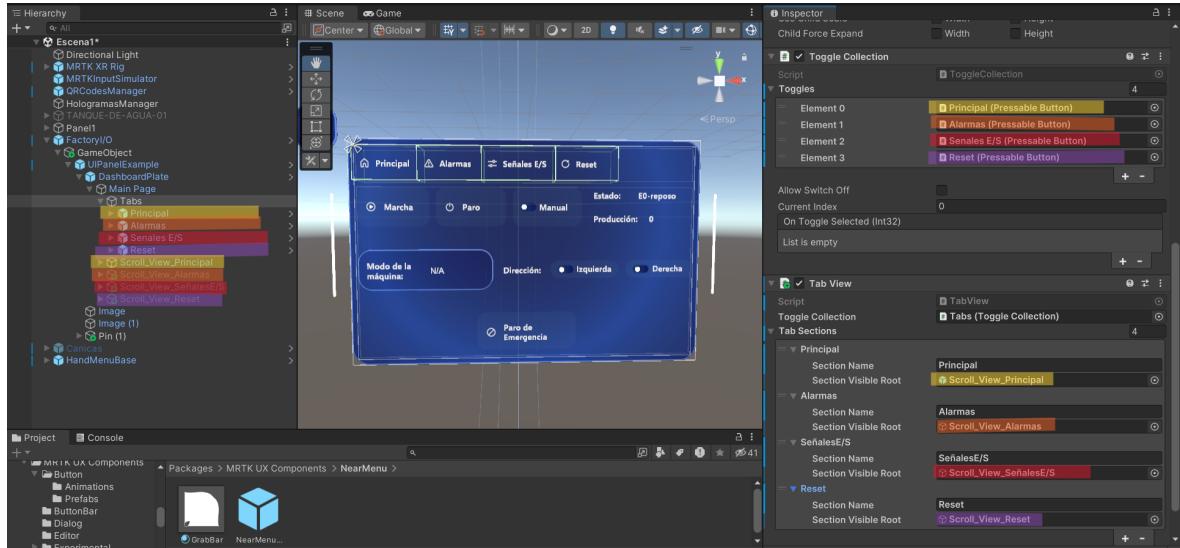


Figura 4.15: Configuracion de Tab View y Toggle Collection

4.3 Diagrama de clases

Finalmente, el diagrama de clases quedaría como se ve en la figura 4.16. En la figura se pueden diferenciar tres módulos principales:

PLC: Aquí se encuentran las clases que se encargan de la comunicación con los PLC

haciendo uso de la librería *S7NetPlus*, y la representación de los atributos. Estas clases son independientes de Unity y podrían utilizarse en otros proyectos en la plataforma .NET.

Unity: Este módulo incluye las clases que dependen de Unity, ya que heredan de la clase *MonoBehaviour*, como se ha explicado anteriormente. Este módulo es el principal de nuestra aplicación ya que define la escena y los objetos de lectura y escritura. *Escena_PLC* se comunica con un objeto de la clase *Clase_PLC* para obtener y escribir los valores y servir estos datos a los objetos de escritura y lectura, de modo que estos no dependan del módulo PLC.

HologramsManager hace de intermediario entre las escenas y el lector de QR para redirigir los eventos de los QR hacia las distintas escenas. Esta clase también mantiene un control de las escenas activas e inactivas.

QRTracking: Este módulo es el que se ha descargado del repositorio de Microsoft y se encarga de todo lo relacionado con los QR, desde su lectura y dibujado hasta de enviar los eventos hacia los objetos interesados, en nuestro caso *HologramsManager*.

Este modelo de clases cumple con los requisitos actuales del proyecto y resuelve de manera efectiva las necesidades de este.

Para desacoplar todavía más los módulos entre sí, podría hacerse uso de interfaces que luego sean implementadas por las clases que actúan de puente en los distintos módulos pero estos temas de arquitectura se dejarán de lado por el momento.

4.4 Mejoras adicionales

Además de los objetivos principales descritos anteriormente, se han añadido diversas mejoras adicionales que surgieron durante el desarrollo. Estas mejoras buscan optimizar la experiencia de usuario y aumentar la funcionalidad del sistema. Las tres mejoras realizadas son las siguientes:

4.4.1 Hologramas

En el apartado 4.2 mencionábamos como podíamos mostrar objetos en la posición de los QR que leímos. Uno de los atractivos principales del uso de la realidad aumentada era el de poder superponer hologramas en los objetos reales para aportar información adicional a los procesos.

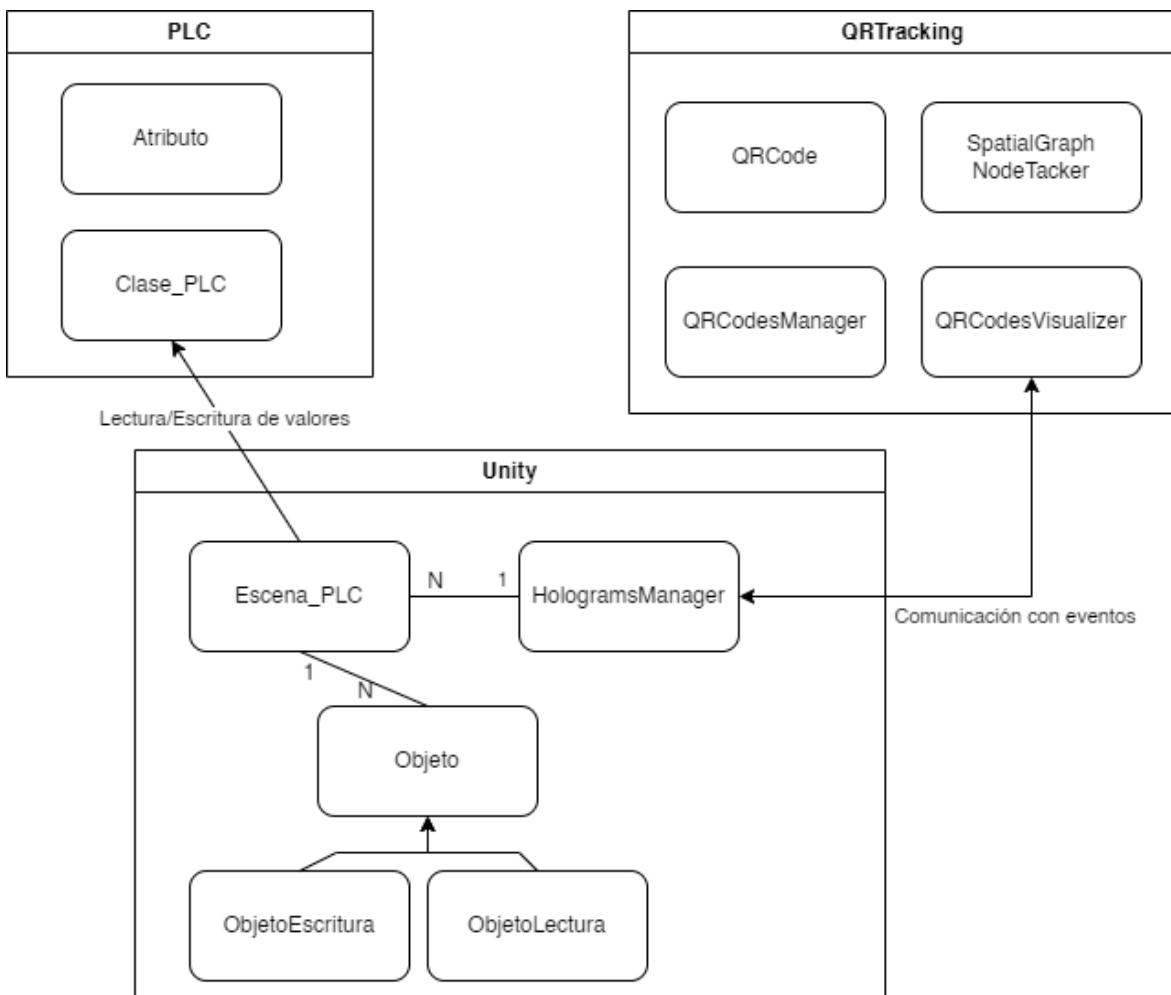


Figura 4.16: Diagrama de clases de la aplicación

Para poder ilustrar esto vamos a hacer uso de una maqueta en el que unas canicas van recorriendo un circuito transportadas por varias cintas. Esta maqueta está conectada a un PLC y podemos pararla y ponerla en marcha.

Lo primero será construir la interfaz, que podemos copiar y pegar la mostrada en la figura 4.13. Eliminamos ciertos componentes, actualizamos los valores del PLC y ya la tendríamos configurada. En la figura 4.17 podemos observar como en *Escena_PLC* hay un campo llamado "Holograma Prefab" y tiene un objeto asociado llamado "Flechas canicas".

Si recordamos lo visto en la sección 4.2 y el código 4.2, vemos cómo *HologramasManager* envía un evento a las escenas para que estas se activen o desactiven en función del valor del QR leído. Como el evento también incorpora el id del *SpatialGraphNode*

asociado al QR, desde la escena, si el QR es el que esperamos, aparte de activarnos instanciamos el "Holograma Prefab" que tenemos asociado y le asignamos el SpatialGraphNode. Esto quedaría como se ve en el código 4.8.

Código 4.8: Código para dibujar holograma

```

1 public class Escena_PLC : MonoBehaviour
2 {
3     ...
4     public bool Activar(HoloEventArgs evento)
5     {
6
7         if(evento.ValorQR == this.qrcode_name)
8         {
9             gameObject.SetActive(true);
10
11         if (instanciaHolograma == null && hologramaPrefab != null)
12         {
13             instanciaHolograma = Instantiate(hologramaPrefab, new Vector3(0, 0, 0), Quaternion.identity);
14             instanciaHolograma.GetComponent<SpatialGraphNodeTracker>().Id = evento.Tracker;
15         }
16
17         if (instanciaHolograma != null)
18             instanciaHolograma.SetActive(true);
19
20         return true;
21     } else
22     {
23         // si se desactiva el objeto con esto activo luego los botones no funcionan
24         if (TryGetComponent<ObjectManipulator>(out var obj))
25         {
26             obj.enabled = false;
27         }
28         gameObject.SetActive(false);
29
30         if (instanciaHolograma != null) instanciaHolograma.SetActive(false);
31
32         return false;
33     }
34 }
35
36 }
37 }
```

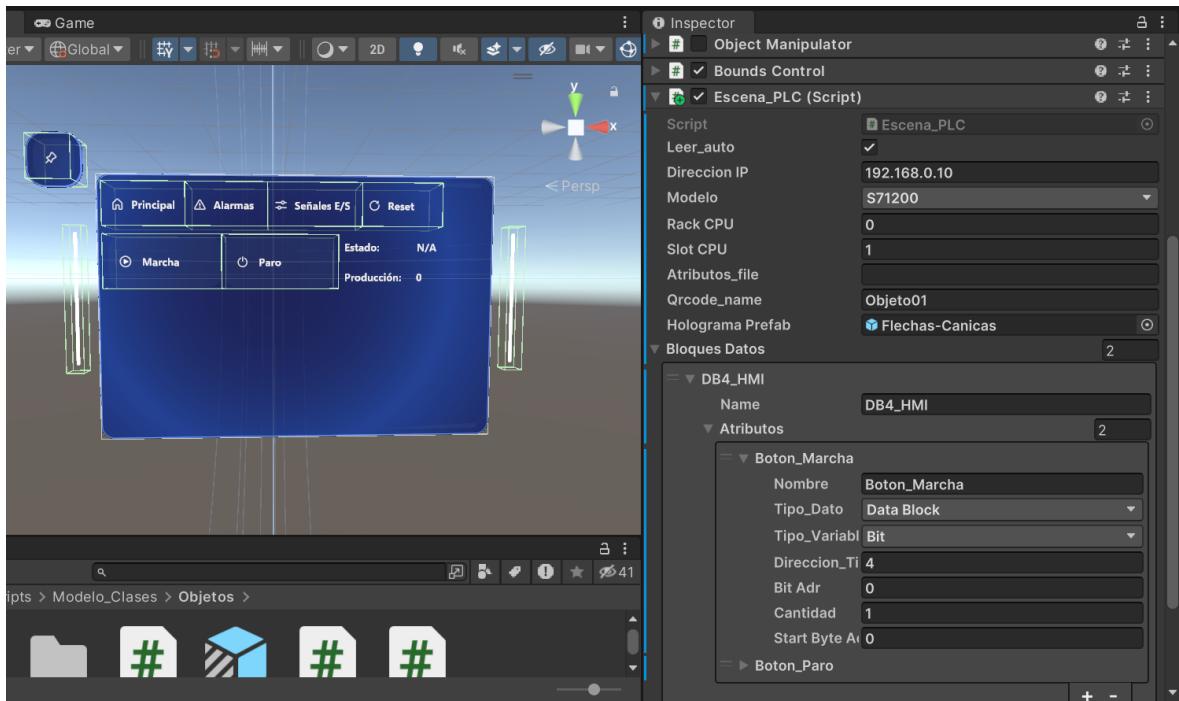


Figura 4.17: Menú para la maqueta

Ahora, si situamos el QR en un lugar concreto cerca de la maqueta, conseguiremos dibujar el objeto 3D en una posición cercana a esta y configurando manualmente la posición de cada elemento del objeto podemos conseguir colocar cada uno en la posición deseada de la maqueta.

El objeto 3D asociado a *HologramaPrefab* quedaría como se ve en la figura 4.18.

Y después de situar el QR en cierta posición relativa al QR, conseguimos el resultado de la figura 4.19

Esta solución, lejos de ser óptima, nos ayuda a visualizar el potencial de la realidad aumentada y a hacernos una idea de lo que se puede conseguir con la ayuda del uso de hologramas en tiempo real.

4.4.2 Menú Mano

Para mejorar la experiencia de usuario y crear una aplicación más completa, vamos a incorporar un menú principal que incorpore ciertas funciones de utilidad para el usuario, pero haciendo uso de la librería MRTK3.

Esta librería incorpora un componente muy conveniente y útil: *HandsConstraint*.

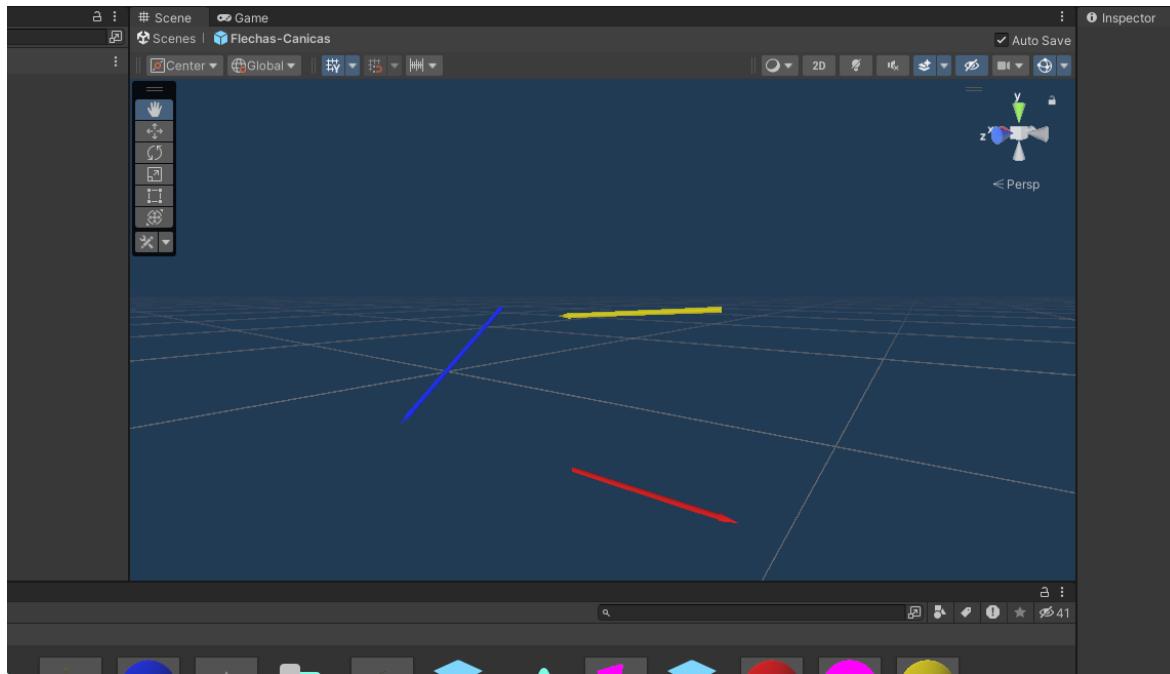


Figura 4.18: Holograma para la maqueta

Este script, al asociarlo a un objeto, es capaz de reconocer la mano del usuario y cuando este se mira la palma una de las manos, muestra el objeto asociado en una posición relativa a la mano.

Con este componente crearemos un menú (figura 4.21) que permita al usuario las siguientes funciones:

Permitir mover el panel de la escena: Aunque el comportamiento del menú es el de seguir al usuario y situarse frente a él, permitiremos interactuar con el panel para desplazarlo a la posición deseada. Esto será mejor realizarlo cuando el menú se encuentre anclado.

Anclar el panel de la escena: Aunque el panel ya incorpora un botón en la esquina superior izquierda para realizar esta acción, vamos a incorporar esta funcionalidad también en el menú para hacerlo más accesible.

Desanclar el panel de la escena: Al igual que lo anterior, esto también se realiza pulsando dicho botón cuando el panel está anclado. Sin embargo, es posible que el usuario haya perdido de vista el panel o que quiera traer de vuelta el menú sin tener que acercarse a él. Si el menú no está anclado siempre va a situarse en frente del usuario a 0.4m de distancia, por lo que si el usuario deja el menú anclado y se aleja de él, al desanclarlo este volverá inmediatamente a la posición del usuario.

Para poder implementar esta funcionalidad, el menú debe de saber que objetos *Escena_PLC* están activados para poder interactuar con ellos.

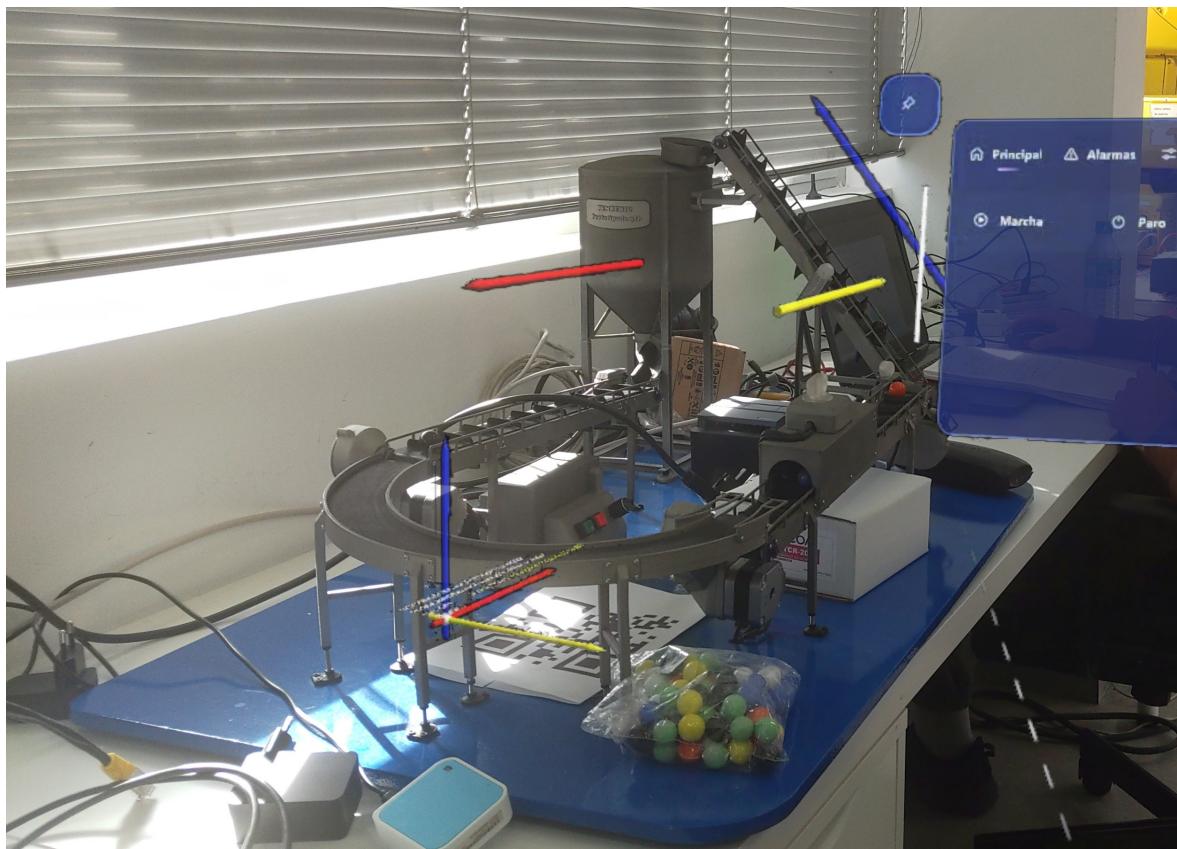


Figura 4.19: Holograma para la maqueta

El objeto *HologramsManager*, que tiene una lista con todas las escenas y redirige a estas los eventos de lectura de QR, guardará las escenas activas en cada momento y permitirá a otras clases recuperarlas. Con esto, el script *MenuMano* podrá saber qué escena está activa para anclarla, desanclarla o activar la manipulación de posición.

4.4.3 Uso de tareas

Un problema que se encontró durante el desarrollo fue el relacionado con la robustez de la aplicación ante fallos, más concretamente fallos en la conexión con el PLC. Si por algún motivo no es posible conectarse al PLC, la aplicación se ralentiza e incluso queda congelada ya que en cada *Update()* se intenta leer del PLC y esta operación, más en concreto *plc.Open()*, se queda bloqueada varios segundos intentando conectarse al PLC si este no responde inmediatamente. Esto arruina totalmente la experiencia de usuario ya que hace la aplicación inutilizable.

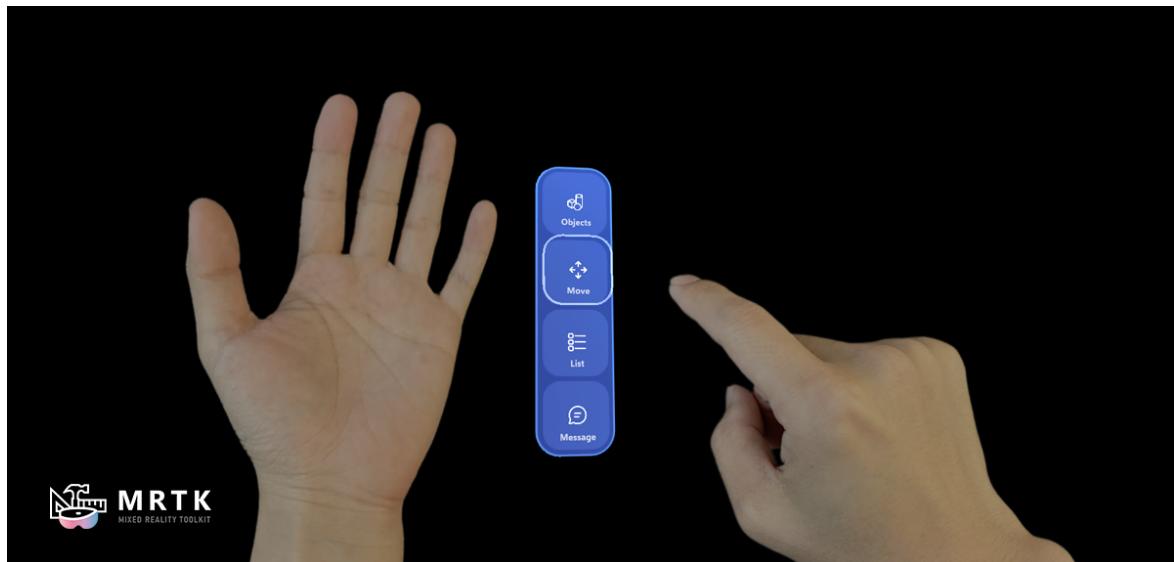


Figura 4.20: Ejemplo de menu mano

Para resolver esto hemos hecho uso de la clase *Task*¹⁴ disponible en C# para que las operaciones de lectura y escritura se ejecuten en una tarea separada del flujo principal de ejecución y así resolver este problema. Además, para la operación *LeerVariables()*, como esta se ejecuta en cada ciclo de renderizado, si se a lanzado una tarea con la función *LeerVariables()* y esta no ha terminado aun, no se vuelve a lanzar otra tarea. Con esto evitamos saturar al sistema de tareas si el PLC no está disponible. Un ejemplo de como se realiza esto se puede ver en el código 4.9

Código 4.9: Ejemplo del uso de tareas

```

1 public class ObjetoEscrituraBoolean : ObjetoEscritura
2 {
3
4     public override void EscribirValor()
5     {
6         Task.Run(() => {
7             try
8             {
9                 this.getEscena().EscribirVariable(this.nombreVariable, valor.ToString());
10                this.getEscena().EscribirVariable(this.nombreVariable, (!valor).ToString());
11                Debug.Log("Escritura task run realizada");
12            }
13            catch (Exception ex)
14            {
15                Debug.Log("Escritura task run realizada con errores: " + ex.Message);
16            }
17        });
18    }
19 }

```

¹⁴<https://learn.microsoft.com/es-es/dotnet/api/system.threading.tasks.task?view=net-8.0>

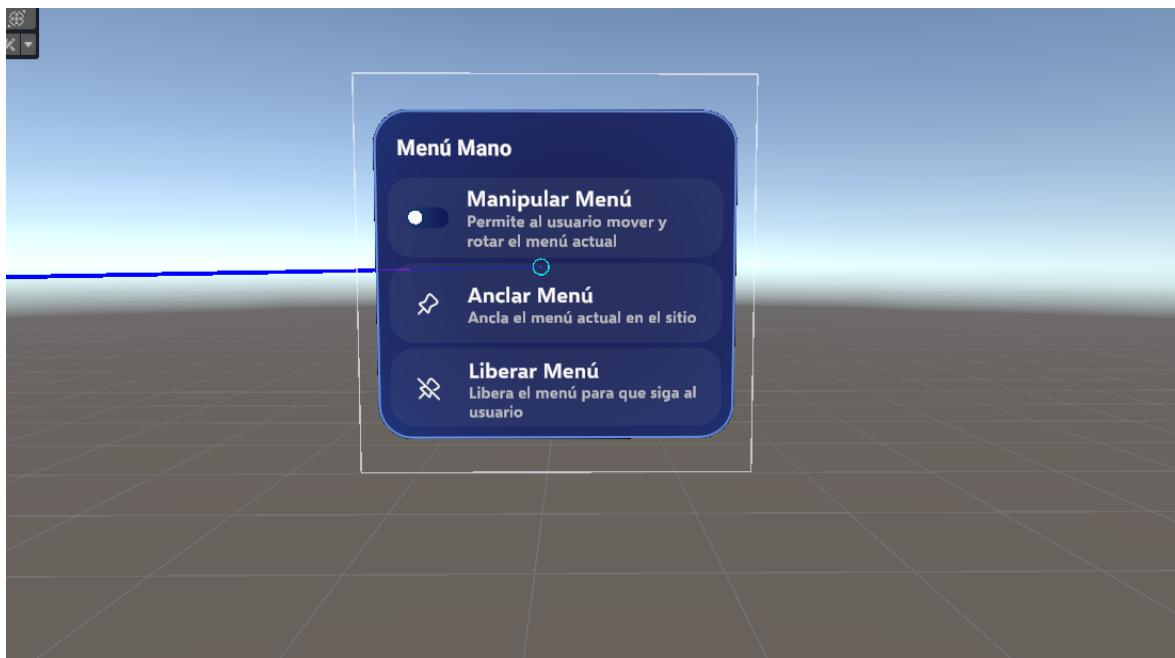


Figura 4.21: Holograma para la maqueta

21
22 }

4.5 Análisis de resultados

En este apartado vamos a realizar un análisis sobre el producto final y analizaremos los errores y virtudes que los revisores de la empresa han encontrado después de probar la aplicación.

Lo primero es ver si la aplicación cumple con lo propuesto en el apartado 3.1. Aunque la elaboración de un SCADA es algo muy complejo y que en ocasiones no tiene fin, si que hemos conseguido satisfacer el objetivo principal de este proyecto que era el de crear una aplicación en realidad aumentada que permita interactuar con distintos procesos industriales. A partir de aquí sólo quedaría añadir más información al panel y mostrar más hologramas, pero algunos de los problemas principales, como el de la comunicación y detección de objetos, ya se han resuelto.

Sobre las impresiones los revisores que lo han probado, el principal beneficio que encuentran es le poder monitorizar el proceso mientras lo tienes en frente y la comodidad de que todo sea virtual. Sin embargo, la experiencia de usuario con los QR no es perfecta y plantea algunos problemas.

El primero se refiere al de la lectura de los QR. Para escanear los códigos es necesario estar relativamente cerca y dependiendo de la luz ambiental es posible que nunca se llegue a reconocer el QR, como pasaba algunas veces cuando los rayos de sol impactaban directamente en el QR.

Otro problema con los QR es el relativo a su conservación, ya que si este se rompe es necesario volver a imprimir otro y es necesario protegerlos bien para que no se deterioren fácilmente. También, en nuestro caso, como estamos dibujando los hologramas en una posición relativa al QR, si este se desplaza puede afectar negativamente la experiencia al desplazar los hologramas.

El general, la aplicación cumple con los objetivos principales y ofrece una experiencia útil y cómoda para la monitorización de procesos industriales. Sin embargo, se han identificado áreas de mejora, especialmente en lo relacionado al uso de códigos QR, sobre lo que discutiremos en el siguiente capítulo.

5 Conclusiones y vías futuras

El desarrollo de una aplicación en realidad aumentada para la monitorización e interacción con procesos industriales ha demostrado ser una solución innovadora y efectiva para mejorar la gestión y supervisión de estos procesos.

Se ha logrado desarrollar una aplicación funcional que permite interactuar con procesos industriales a través de una interfaz en realidad aumentada. La aplicación permite la lectura y escritura de variables en un PLC, la visualización de hologramas y la interacción con elementos de la escena.

Sin embargo, como se ha expuesto en el apartado 4.5, existen bastante margen de mejora y sería necesario replantear algunas funcionalidades como lo relacionado a la detección de objetos.

Aunque el uso de QR's resuelve el problema de la detección de objetos, plantea ciertos inconvenientes relacionados sobre todo con su conservación y condiciones de iluminación. Una posible alternativa podría ser el uso de marcadores, como los utilizados por sistemas como ARToolkit o AprilTags¹. Estas marcas suelen ser más resistentes a variaciones en las condiciones de iluminación y pueden ser detectadas a mayores distancias y con ángulos de visión más amplios, como se muestra en la sección 4 de este artículo [5].

Otra alternativa, y la más eficaz, sería el reconocimiento de los objetos por visión artificial, de manera que su puedan identificar y rastrear los objetos en tiempo real, eliminando la necesidad de códigos QR o marcas físicas. El reconocimiento por visión artificial puede ofrecer mayor flexibilidad y precisión en la detección y posicionamiento de objetos. Además, esto nos permitiría colocar los hologramas en posiciones específicas del objeto, lo que eliminaría el problema que ocurre con los códigos QR y las marcas.

Siguiendo con los hologramas, para poder satisfacer los casos de uso expuestos en el apartado 1.1, tendríamos que empezar a crear tutoriales y guías de los procesos haciendo uso de hologramas y vídeos adjuntos al panel de control que faciliten las tareas de monitorización y resolución de errores, además de seguir mejorando la interfaz y funcionalidad de los paneles. Todo esto con el objetivo de mostrar a futuros interesados

¹<https://april.eecs.umich.edu/software/apriltag>

las virtudes del uso de la realidad aumentada.

Por último, otro aspecto que podríamos mejorar de la aplicación actual sería el específico a la comunicación y recolección de datos del PLC.

En este caso, estamos haciendo uso de la librería S7NetPlus para comunicarnos con PLC's de la marca Siemens. Sin embargo, existen distintas marcas y protocolos de comunicación y no sería aceptable escribir en la aplicación principal de Unity todos los protocolos para comunicarse con todos los posibles PLC's que existen en el mercado. Tendríamos que encontrar otra solución.

La solución sería hacer uso de un servidor centralizado que se encarge de la comunicación con los distintos PLC's en una planta industrial y permita a otros dispositivos (las Hololens 2) comunicarse con el servidor para leer y escribir datos. Uno de estos servidores es el llamado Kepserver², de la empresa antes mencionada PTC. Este servidor implementa casi la totalidad de protocolos de comunicación entre PLC's y permite a otros sistemas acceder a los datos comunicándose con el protocolo OPC UA³.

Con esto, desacoplaríamos de la aplicación lo relacionado con la comunicación y recolección de datos y sólo tendríamos que hacer uso de un protocolo de comunicación para acceder a los datos, ya sea OPC UA ó HTTP a través de una API REST. También, gracias al uso de un servidor centralizado, se podrían implementar otras muchas más funcionalidades como almacenamiento de un histórico de datos que podría usarse para análisis de datos.

Estos serían los tres grandes pilares en los que se centrará el desarrollo de la aplicación (kepserv, Visión artificial y hologramas), creando así una solución más completa y funcional que nos permita destacar sobre el resto.

²<https://www.kepserverexopc.com/>

³<https://www.cognex.com/es-es/blogs/machine-vision/why-opc-ua-is-essential-for-factory-automation>

Bibliografía

- [1] IBM. ¿qué es la industria 4.0 y cómo funciona? | ibm. URL <https://www.ibm.com/es-es/topics/industry-4-0>.
- [2] Heiner Lasi, Peter Fettke, Hans-Georg Kemper, Thomas Feld, and Michael Hoffmann. Industry 4.0. *Business & information systems engineering*, 6:239–242, 2014. URL <https://link.springer.com/article/10.1007/s12599-014-0334-4>.
- [3] Paula Fraga-Lamas, Tiago M Fernández-Caramés, Oscar Blanco-Novoa, and Miguel A Vilar-Montesinos. A review on industrial augmented reality systems for the industry 4.0 shipyard. *Ieee Access*, 6:13358–13375, 2018. URL <https://ieeexplore.ieee.org/abstract/document/8298525>.
- [4] Roberto Pierdicca, Emanuele Frontoni, Rama Pollini, Matteo Trani, and Lorenzo Verdini. The use of augmented reality glasses for the application in industry 4.0. In *Augmented Reality, Virtual Reality, and Computer Graphics: 4th International Conference, AVR 2017, Ugento, Italy, June 12-15, 2017, Proceedings, Part I* 4, pages 389–401. Springer, 2017. URL https://link.springer.com/chapter/10.1007/978-3-319-60922-5_30.
- [5] Oscar Blanco-Novoa, Tiago M Fernandez-Carames, Paula Fraga-Lamas, and Miguel A Vilar-Montesinos. A practical evaluation of commercial industrial augmented reality systems in an industry 4.0 shipyard. *Ieee Access*, 6:8201–8218, 2018.
- [6] SAP. ¿qué es la industria 4.0? | definición, tecnologías, beneficios | sap. URL <https://www.sap.com/spain/products/scm/industry-4-0/what-is-industry-4-0.html>.
- [7] Francesco De Pace, Federico Manuri, and Andrea Sanna. Augmented reality in industry 4.0. *Am. J. Comput. Sci. Inf. Technol*, 6(1):17, 2018. URL <https://core.ac.uk/download/pdf/234921763.pdf>.
- [8] Innoarea. Industria 4.0 a través de realidad virtual y realidad aumentada – innoarea projects. URL <https://innoarea.com/noticias/industria-4-0-a-traves-de-realidad-virtual-y-realidad-aumentada/>.

- [9] Denys Pavlov. Ar vr applications in manufacturing: Redefining productivity and quality. *Smarttek Solutions*, Abril 2023. URL <https://smarttek.solutions/blog/ar-and-vr-in-manufacturing-use-cases-and-benefits/>.
- [10] Silvia Gonzalez. Augmented reality: Seeing hmi in a new way | processing magazine. *Processing Magazine*, Junio 2023. URL <https://www.processingmagazine.com/process-control-automation/scada/hmi/article/53061206/augmented-reality-seeing-hmi-in-a-new-way>.
- [11] Light Guide. 6 uses of augmented reality for manufacturing in every industry - lightguide. Febrero 2022. URL <https://www.lightguidesys.com/resource-center/blog/6-uses-of-augmented-reality-for-manufacturing-in-every-industry/>.
- [12] Anand Nayyar and Akshi Kumar. *A Roadmap to Industry 4.0: Smart Production, Sharp Business and Sustainable Development*. Springer, Agosto 2019. URL https://www.researchgate.net/profile/Sudeep-Tanwar/publication/334836077_Additive_Manufacturing_-_Concepts_and_Technologies/links/5e834ee44585150839b1334a/Additive-Manufacturing-Concepts-and-Technologies.pdf#page=153.
- [13] Luís Fernando de Souza Cardoso, Flávia Cristina Martins Queiroz Mariano, and Ezequiel Roberto Zorjal. A survey of industrial augmented reality. *Computers & Industrial Engineering*, 139:106159, 2020.
- [14] Eleonora Bottani and Giuseppe Vignali. Augmented reality technology in the manufacturing industry: A review of the last decade. *Iise Transactions*, 51(3): 284–310, 2019.
- [15] Jeevan S Devagiri, Sidike Paheding, Quamar Niyaz, Xiaoli Yang, and Samantha Smith. Augmented reality and artificial intelligence in industry: Trends, tools, and future challenges. *Expert Systems with Applications*, 207:118002, 2022.
- [16] Chandan K Sahu, Crystal Young, and Rahul Rai. Artificial intelligence (ai) in augmented reality (ar)-assisted manufacturing applications: a review. *International Journal of Production Research*, 59(16):4903–4959, 2021.

Lista de Acrónimos y Abreviaturas

HMI	Interfaz Humano - Máquina.
PLC	Controlador Lógico Programable.
QR	Quick Response.
RA	Realidad Aumentada.
SCADA	Supervisión, Control y Adquisición de Datos.
TFG	Trabajo Final de Grado.
VR	Realidad Virtual.