



**Escuela Superior
de Ingeniería y Tecnología**
Universidad de La Laguna

Trabajo de Fin de Grado

Grado en Ingeniería Informática

Videojuegos Para la Automatización

Game4Automation

Jonathan Eliot Viera Rivas

La Laguna, 8 de septiembre de 2021

D. **Pedro Antonio Toledo Delgado**, con N.I.F. 45.725.874-B Profesor Contratado Doctor adscrito al Departamento de Ingeniería Informática y de Sistemas de la Universidad de La Laguna, como tutor

CERTIFICA

Que la presente memoria titulada:

“Game4Automation”

ha sido realizada bajo su dirección por D. **Jonathan Eliot Viera Rivas**, con N.I.F. 78.856.558-T.

Y para que así conste, en cumplimiento de la legislación vigente y a los efectos oportunos firman la presente en La Laguna a 8 de septiembre de 2021

Agradecimientos

Me gustaría agradecer a la Universidad de La Laguna por toda la formación que me han proporcionado durante todos los años, además de la ayuda que me han otorgado para facilitarme la elaboración de este trabajo final de grado, en especial a mi tutor Pedro Antonio Toledo Delgado, que ha procurado en todo momento en ayudarme y brindarme todo lo necesario para poder avanzar cada día en el desarrollo del trabajo.

También quiero agradecer a mis compañer@s de la universidad y a mis compañer@s del trabajo que me han ayudado tanto en mi formación en el ámbito profesional como en lo personal, aportando su experiencia y su granito de arena para que pudiera alcanzar nuevas metas.

Por último, quiero agradecer enormemente a mi familia y a mis amig@s que han confiado en mi en todo momento y han dedicado todo su tiempo, esfuerzo y recursos para sacar lo mejor de mi e impulsarme y guiarme en mi trayectoria de formación profesional como en la realización de este trabajo.

Licencia



© Esta obra está bajo una licencia de Creative Commons Reconocimiento-NoComercial-CompartirIgual 4.0 Internacional.

Resumen

La tecnología basada en autómatas se hace notar cada vez más en la vida cotidiana, ya sea para tareas sencillas como desplazar una cinta transportadora con alimentos, como complejas, por ejemplo, el coche autónoma de Google. Estas tecnologías buscan facilitar y mejorar la vida diaria de las personas realizando las tareas específicas para la que han sido creados con mejor eficiencia, seguridad y detalle. Cabe destacar que se ven respaldadas principalmente por el uso y avance tecnológico de las empresas que buscan beneficios a costo de la mano de obra al igual que una ventaja para el futuro consumidor del producto y apoyadas también por estos últimos por las características nombradas anteriormente.

El objetivo fundamental que se persigue con la realización de este proyecto es enseñar y crear la simulación de plantas industriales para demostrar el beneficio que tiene generar un producto previo antes de la construcción de este. Además, como medio de acercar al usuario objetivo alojaremos nuestro proyecto en una página web para que sea más intuitiva y fácil de abordar.

Es necesario destacar que gran parte del proyecto se va a realizar con la plataforma de Unity que es una herramienta popular de entorno de desarrollo de videojuegos comerciales y que incluye soporte para VR(realidad virtual) y AR(realidad aumentada), en concreto, añadiremos a esta plataforma un paquete denominado Game4Automation para extender esa funcionalidad al diseño e implementación de plantas industriales donde crearemos nuestras simulaciones de elementos automatizados con los objetos que vienen por defecto en este Add-On.

El trabajo está estructurado en tres partes: en la primera, se ha realizado la configuración de los elementos de Game4Automation para la creación de las escenas en Unity con diferentes niveles de dificultad entre escenas; una segunda parte, que en conjunto con la primera se trata de añadir el comportamiento y la funcionalidad de los autómatas generando las PLCs (Controladores Lógicos Programables) con la herramienta de Visual Studio 2017 y programado en C#; y en la tercera, se ha desarrollado una página web de autenticación de usuarios para la visualización del contenido y descarga de las simulaciones haciendo uso de HTML5,CSS y PHP comunicándose a una base de datos creada con MySQL.

Palabras clave: Autómatas, Simulaciones, Game4Automation, Unity, PLC, Pagina web.

Abstract

Automaton-based technology is increasingly noticeable in daily life, whether for simple tasks such as moving a conveyor belt with food, or complex tasks, for example, Google's automaton car. These technologies seek to facilitate and improve people's daily lives by performing the specific tasks for which they have been created with better efficiency, safety and detail. It should be noted that they are supported mainly by the use and technological progress of companies that seek benefits at the cost of labor as well as an advantage for the future consumer of the product and also supported by the latter due to the characteristics mentioned above.

The main objective pursued with the realization of this project is to teach and create the simulation of industrial plants to demonstrate the benefit of generating a previous product before its construction. In addition, as a means of bringing the target user closer, we will host our project on a web page to make it more intuitive and easy to approach.

It should be noted that a large part of the project will be carried out with the Unity platform, which is a popular tool in the commercial video game development environment and includes support for VR (virtual reality) and AR (augmented reality), specifically, we will add to this platform a package called Game4Automation to extend this functionality to the design and implementation of industrial plants where we will create our simulations of automated elements with the objects that come by default in this Add-On.

The project is structured in three parts: in the first, the configuration of the Game4Automation elements for the creation of scenes in Unity with different levels of difficulty between scenes has been carried out; a second part, which together with the first is to add the behavior and functionality of the automata generating the PLCs (Programmable Logic Controllers) with the Visual Studio 2017 tool and programmed in C #; and in the third, a user authentication web page has been developed to view the content and download the simulations using HTML5, CSS and PHP, communicating with a database created with MySQL.

Keywords: Automata, Simulations, Game4Automation, Unity, PLC, Web page.

Índice general

Capítulo 1: Introducción	13
1.1 Simulación de procesos industriales.....	14
1.2 Antecedentes y Actualidad	14
1.3 Casos de uso	17
1.4 Objetivo	18
Capítulo 2:Entorno de Desarrollo	21
2.1 Unity	21
2.1.1 UnityHub.....	22
2.1.2 Unity2020.3.X	22
2.2 Base de datos	22
2.2.1 MySQL	22
2.3 Página Web.....	23
2.3.1 HTML5.....	23
2.3.2 CSS	23
2.3.3 PHP	24
2.3.4 Bootstrap	24
2.4 Herramientas para el desarrollo.....	25
2.4.1 Visual Studio 2017 y Code.....	25
2.4.3 Github.....	25
2.4.4 Google Chrome y Drive	25
2.4.5 Wamp	26
2.4.6 Apache	26
2.4.7 Word Office.....	27
Capítulo 3:Game4Automation	28
3.1 Qué es Game4Automation	28

3.2	Funcionamiento	28
3.3	Ventajas y desventajas	29
3.4	Interfaces y PLC	29
3.4.1	Creadas Manualmente	30
3.4.2	Siemens7.....	30
3.4.3	OPC-UA.....	31
3.5	Componentes	31
3.5.1	Caja	31
3.5.2	Cinta transportadora grande	31
3.5.3	Cinta transportadora pequeña	32
3.5.4	Lata	32
3.5.5	Robot.....	33
3.5.6	Gancho.....	33
3.5.7	Lampara y caja de mandos.....	34
3.5.8	Sensor	34
3.5.9	Sumidero	34

Capítulo 4:Diseño 35

4.1	Página web.....	35
4.1.1	Login, registro y recordar contraseña	35
4.1.2	Pantalla de bienvenida	36
4.2	Unity: Plantas	38

Capítulo 5:Desarrollo 41

5.1	Página web.....	41
5.1.1	Funcionalidades.....	42
5.2	Bases de datos.....	45
5.3	Unity.....	46
5.3.1	Configuración de Unity.....	46
5.3.2	Desarrollo planta 1.....	46
5.3.3	Desarrollo planta 2.....	47
5.3.4	Desarrollo planta 3.....	49

Capítulo 6:Problematicas 52
Capítulo 7: Conclusiones y líneas futuras 53
Capítulo 8: Summary and Conclusions 55
Capítulo 9:Presupuesto 57

Índice de figuras

Figura 1.1: Fases de una simulación.....	14
Figura 2.1: Logo Unity	21
Figura 2.2: Logo UnityHub	22
Figura 2.3: Logo MySQL	23
Figura 2.4: Logo HTML5	23
Figura 2.5: Logo CSS.....	24
Figura 2.6: Logo PHP.....	24
Figura 2.7: Logo Bootstrap.....	24
Figura 2.8: Logo Visual Studio 2017 y Visual Studio Code.....	25
Figura 2.9: Logo Github	25
Figura 2.10: Logo Google Chrome y Google Drive.....	26
Figura 2.11: Logo Wampserver	26
Figura 2.12: Logo Apache	26
Figura 2.13: Logo Word Office	27
Figura 3.1: Logo Game4Automation	28
Figura 3.2: Cajas.....	31
Figura 3.3: Cinta transportadora grande.....	31
Figura 3.4: Cinta transportadora pequeña.....	32
Figura 3.5: Lata.....	32
Figura 3.6: Robot	33
Figura 3.7: Plataforma XY	33
Figura 3.8: Lampara y caja de mandos	34
Figura 3.9: Sensor.....	34

Figura 4.1: Vista login.php.....	35
Figura 4.2: Vista register.php	36
Figura 4.3: Vista forgotpw.php	36
Figura 4.4: Página de bienvenida.....	37
Figura 4.5: Vista planta 1	38
Figura 4.6: Vista planta 2	39
Figura 4.7: Vista planta 3	40
Figura 5.1: Estructura organizativa de la página web	41
Figura 5.2: Funcion para redirigir login o pantalla bienvenida	42
Figura 5.3: Funcion logout.....	42
Figura 5.4: Funcion login.....	43
Figura 5.5: Funcion para registrar al usuario	44
Figura 5.6: Verificar si existe correo y enviar correo.....	45
Figura 5.7: Diseño de la base de datos	45
Figura 5.8: Interfaces y PLCs de la planta 1.....	46
Figura 5.9: Panel de control planta 1.....	47
Figura 5.10: Interfaces y PLCs de la planta 2.....	48
Figura 5.11: Panel de control planta 2.....	49
Figura 5.12: Interfaces y PLCs de la planta 3.....	50
Figura 5.13: Panel de control planta 3.....	51

Índice de tablas

Tabla 1.1: Comparativas	15
Tabla 9.1: Presupuesto del proyecto	57

Capítulo 1

Introducción

La simulación de procesos industriales es una nueva tecnología en potencia que se basa en generar simulaciones cercanas de posibles desarrollos en plantas industriales. Además, el beneficio de esto se demuestra en la estimación aproximada de los datos del diseño para llevar a cabo la construcción a escala real, ahorrándonos recursos antes de tiempo y tomando decisiones más acertadas en la implementación.

El objetivo principal en este trabajo es dar a conocer una herramienta específica de este sector y explotarla para el ámbito de la enseñanza o por lo menos como toma de contacto para posibles usuarios objetivos que estén interesados.

Esta memoria está estructurada en nueve capítulos y la sección bibliográfica: en el primero, se explica el concepto de la simulación industrial y la relación de este tema con lo propuesto para la realización de mi proyecto, también se exponen los antecedentes y el estado actual, detallando algunos ejemplos de casos de uso. También, se expone el objetivo del trabajo y las tareas llevadas a cabo para alcanzarlo. En el segundo, se describen las diferentes herramientas y tecnologías que se han utilizado para superar y completar los hitos de proyecto. En el tercero, se explica la investigación que se hizo sobre el paquete de Game4Automation, las funcionalidades que planteaba, las ventajas y desventajas, al igual que los diferentes protocolos de conexión de interfaces a herramientas externas a Unity para la creación y uso de PLCs. Cabe añadir también la información descriptiva de los componentes por defecto de Game4Automation. En el cuarto, se muestra el diseño de las vistas de la página web en sus respectivas funciones y una visualización de las 3 plantas usando la cámara de las escenas. En el quinto, se explica el desarrollo realizado para las principales partes del proyecto: la página web, las simulaciones industriales de las tres plantas y el diseño de la base de datos. En el capítulo sexto, se explican las problemáticas que han aparecido al realizar las tareas del proyecto. En el séptimo y octavo capítulo, se redacta tanto en español como en inglés las conclusiones y las futuras implementaciones que se pueden hacer para mejorar el producto. En el noveno, se incluye el presupuesto con los costes estimados del proyecto. Y, por último, la bibliografía se compone de los enlaces y recursos de referencia empleados durante el proceso de desarrollo del proyecto.

1.1 Simulación de procesos industriales

La simulación de procesos industriales es una herramienta que nos permite reproducir virtualmente los procesos y estudiar el comportamiento, para analizar el impacto de las distintas variables que pueden intervenir en la realización de la tarea concreta, además, nos sirve para comparar alternativas de diseño sin el alto coste de los experimentos a escala real. Esto es de gran ayuda a la hora de disminuir los riesgos y optimizar la toma de decisiones.

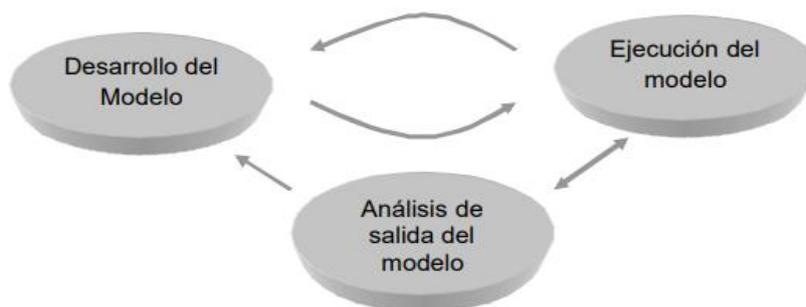


Figura 1.1: Fases de una simulación

Para complementar este esquema de simulación con un caso práctico haremos uso de una extensión de Unity, Game4Automation, que es el tema que ocupa este trabajo. Game4Automation es una extensión de Unity y está registrada por In2Sight GmbH, una empresa con sede en Sinzheim, Baden-Württemberg, Alemania y forma parte de la industria de diseño de sistemas informáticos y servicios relacionados.

En resumen, la simulación de procesos industriales nos permite abarcar un modelo de generación virtual con el objetivo de minorizar las dificultades y costes que tendríamos al implementarlo a escala real, valiéndonos de aplicaciones de desarrollo como Game4Automation de Unity, una plataforma de creación de videojuegos comerciales transformada con este add-on en una aplicación de simulación de procesos industriales. A continuación, se explicarán los antecedentes y el estado actual de esta tecnología.

1.2 Antecedentes y actualidad

La aparición de estas tecnologías basadas en simulación se generó tratando de solventar los problemas del coste de implementación real y para obtener una aproximación final de un producto o servicio, además, abarca un ámbito grande de operación desde fabricas industriales a simuladores de entrenamiento aéreo. Hay que añadir que la informática está ligada a todas estas tecnologías basadas en autómatas que realizan la tarea para la que han sido creados. Vamos a disminuir el rango de tecnologías de simulación a lo más cercano al proyecto, tomando en cuenta solo aquellas que generen un desarrollo virtual en 3D orientado a la industria y explicaré resumidamente cada una de ellas.

En primer lugar y como es tema de mi proyecto, es Game4Automation una extensión de Unity que añade a la funcionalidad la posibilidad de generar nuestros propios diseños de simulación de plantas industriales con objetos ya prefabricados o incluso generarlo con AutoCAD y exportarlos como nuevos elementos a utilizar.

Una segunda aplicación de simulación virtual es Factory I/O, que simula una fabrica en 3D como Unity y permite construir rápidamente usando la selección de piezas industriales comunes, además viene con varios ejemplos de escenas inspiradas en el sector.

Una tercera aplicación es FlexSim, que posee todos los beneficios de las anteriores aplicaciones nombradas y que genera modelos con gráficos 3D de eventos discretos, y también añade fondos inmersivos a la simulación pareciendo casi reales.

Otra plataforma para diseñar simulaciones industriales es Isaac Sim, creada por NVIDIA es una aplicación de simulación robótica y una herramienta de generación de datos sintéticos diseñada para probar, desarrollar y administrar robots en un entorno 3D. Ofrece escalabilidad y la interoperabilidad de la plataforma Omniverse, además funciona tanto en sistemas locales como basados en la nube.

Por último, aunque tal vez no se asemeja a las aplicaciones anteriores, Virtual Plant es una implementación de realidad virtual en donde tomamos el rol de un usuario para realizar acciones dentro de una planta industrial desde un entorno seguro y controlado y que sirve a su vez como modelo de enseñanza.

Finalizada la explicación de otras posibles aplicaciones a utilizar para este proyecto, vamos a generar una tabla de pros y contras en base a 3 valores al igual que un breve resumen de porqué escogimos Game4Automation sobre las demás.

	Coste	Versatilidad	Comunidad
Game4Automation	<p>Depende del paquete que se vaya a comprar, aun así, no hay un paquete gratuito.</p> <p>El único punto positivo en este aspecto es que el paquete starter es asequible, no obstante, la desventaja es que la interfaz que puedes generar es solo de Siemens7 siendo este entorno de desarrollo de pago también.</p>	<p>Depende del paquete comprado ya que uno que te cueste más tendrá añadidas más implementaciones como importar diseños de AutoCAD y el uso de nuevas interfaces como OPC-UA, TwinCat, Simulink...</p> <p>También gracias a que se trabaja en el entorno de Unity se puede utilizar las herramientas de realidad virtual y</p>	<p>Para ser una extensión que es relativamente nueva de Unity tiene una comunidad activa en el fórum de la página principal.</p>

		<p>aumentada en la simulación al igual que otras extensiones en la misma escena creada.</p> <p>Fácil de utilizar si tienes cierta experiencia en desarrollo en Unity.</p> <p>La desventaja de este paquete es que carece de muchos objetos prefabricados.</p>	
Factory I/O	<p>Depende del paquete que se vaya a comprar, aun así, hay un paquete gratuito de 30 días de uso.</p> <p>Además, todas las interfaces son para productos de Siemens7.</p>	<p>Tiene la misma versatilidad que Unity con la diferencia de que los paquetes a comprar te vienen indicados por las diferentes interfaces que quieras utilizar a no ser que compres el Ultimate que implementa todo.</p> <p>Con respecto a Game4Automation es menos completo en las interfaces a usar, pero más extenso en los objetos prefabricados que tiene.</p>	<p>Al igual que Game4Automation la comunidad es relativamente activa en los fórums no obstante en menor medida de usuarios y consultas que se desarrollan sobre la plataforma.</p>
Isaac Sim	<p>Gratuito para descargar.</p>	<p>Contiene todas las funcionalidades para desarrollo de autómatas de industria. Además, NVIDIA lo lanzo como una plataforma de código abierto</p>	<p>Fórum grande de usuarios debido a que es una herramienta de acceso gratuito, al igual que una comunidad activa en el uso de las</p>

		para usuarios que estén interesado en el desarrollo de robots, por tanto, está constantemente en feedback.	aplicaciones de NVIDIA.
FlexSim	Depende del paquete que se vaya a comprar, existe una versión gratuita.	Intuitivo y fácil de usar para plantas de baja complejidad. Dispone de las interfaces de conexión nombradas en los entornos anteriores.	Comunidad de gran tamaño, con mucho feedback.
Virtual Plant	Aplicación de pago	Para el desarrollo de metodologías enseñanza es el mayor punto que tiene este entorno, con respecto a la simulación de los autómatas carece de los efectos visuales fluidos que se dan en las anteriores aplicaciones nombradas.	No existe una comunidad de renombre.

Tabla 1.1: Comparativas

En resumen, Game4Automation, aunque tal vez es relativamente nueva y no puede ponerse en el top de aplicaciones de simulación todavía, tiene mucho margen de mejora y la potencia de hacer grandes diseños virtuales gracias a las funcionalidades y complementos que tiene Unity como motor de videojuegos.

Actualmente, existen muchas tecnologías con las que podríamos abarcar este proyecto de simulación virtual y tomando en cuenta incluso las aplicaciones de modelos 2D, existen aún más aplicaciones no nombradas que generan también estos resultados, aunque visualmente son más técnicas y cercanas a circuitos.

1.3 Casos de uso

Existen una variedad de casos de uso para aplicaciones de simulación virtual de procesos industriales, ya que es básico antes de generar un proyecto a escala real verificar el funcionamiento y comparar con posibles alternativas. A continuación, se indican algunos ejemplos de proyectos que han tenido éxito usando estas tecnologías de virtualización industrial.

- BMW Group: Lo último en tecnología de IA se está utilizando para crear la máquina de conducir definitiva. Se trata de la compañía BMW que anunció en la conferencia de GTC 2020 de NVIDIA que estaban redefiniendo la logística de la fábrica con la plataforma robótica Isaac de NVIDIA, mejorando sus procesos logísticos de fábrica con computación avanzada de IA y visualización para robots logísticos basados en la plataforma de simulación NVIDIA Isaac SDK e Isaac Sim.

BMW Group está desarrollando cinco nuevos tipos de robots en la plataforma robótica de NVIDIA Isaac, impulsados por computadores de borde como Jetson y EGX y que se ejecutan en una sola arquitectura de software Isaac SDK. Además, estos mismos robots son entrenados y validados con Isaac Sim.

- Virtual Plant: Virtual Plant es una plataforma virtual de aprendizaje para aplicar los conceptos técnicos, científicos o de ingeniería, por medio de experiencias prácticas. Es una herramienta en la que puedes interactuar con escenarios y situaciones del mundo real a través de un medio virtual.
- Two Reality: Es una empresa que desarrolla aplicaciones de realidad virtual y aumentada en la industria para dispositivos llevables como gafas inteligentes y cascos, además de dispositivos móviles.

Este entorno conecta a los operadores en el campo con la información necesaria para completar un trabajo determinado usando iconos, modelos 3D y animaciones para proporcionar instrucciones para finalizar dichas tareas. Estos flujos de trabajo se integran al backend y se asocian con la tarea en ejecución permitiendo enviar la información digital sobre la actividad en tiempo real.

1.4 Objetivo

El objetivo fundamental que se persigue con la realización de este proyecto es crear una página web integrada con las plantas realizadas en Unity como modelo de enseñanza, pudiendo acceder a esta página y sacar las plantas para visualizarla desde la máquina local. Vamos a definir los objetivos propuestos para la realización de este proyecto:

1. Crear nuestras plantas de simulación de procesos industriales con el entorno de trabajo de Unity y Game4Automation. Para demostrar la potencia de la herramienta de Game4Automation crearemos al menos 3 plantas con diferentes niveles de dificultad.

2. Desarrollar una página web que será el medio de bienvenida para los usuarios objetivos que quieran aprender sobre la simulación industrial y en concreto a usar el add-on de Unity.
3. Validar el funcionamiento de las plantas desarrolladas en la página web por los usuarios utilizando.

Para ello, estos objetivos propuestos se han dividido en dos conjuntos de Actividades principales que a su vez se divide en tareas más concretas para cada una de las partes. La jerarquía de actividades tendría una estructura como la que se ve a continuación:

- Actividades para la página Web:
 - Creación de un login y logout. Para que el usuario pueda acceder a la página web con su cuenta de usuario y contraseña.
 - Creación de una página de registro. Un acceso para que pueda inscribirse en la base de datos con su información de usuario.
 - Creación de una pantalla de bienvenida. Pantalla de bienvenida con la información de las plantas a descargar y del Add-on de Game4Automation
 - Creación de una pantalla de olvidar la contraseña. Un acceso en caso de que el usuario se olvide de la contraseña pueda obtenerla de nuevo a través de un correo electrónico.
 - Configurar y crear la base de datos. Configuración de la base de datos con las variables que se van a almacenar del registro.
- Actividades para Unity
 - Creación de las 3 plantas industriales. Creación de plantas industriales con diferentes niveles de dificultad: fácil, medio y difícil.
 - Creación de los PLCs. Desarrollo de programas que van a generar el comportamiento de los componentes de la simulación y que también tiene toda la lógica de comunicación entre ellos.

- Creación de panel de pulsadores. Desarrollar un panel para que sea interactivo con el usuario/operador de la simulación.
- Validación
 - A parte de testear como usuario/creador del trabajo, me ayude de otras 2 personas para testear la planta industrial y me diesen feedback de la misma.

La finalización de estas actividades generaría las funcionalidades del proyecto y aportaría muchos beneficios como:

- Ofrece al usuario objetivo una plataforma intuitiva y de fácil acceso para obtener estas simulaciones de ejemplo y que, además, es gratuita.
- Ayuda al usuario a entender la importancia de poder generar un boceto virtual de un posible desarrollo de escala real.
- Ofrece La posibilidad de interactuar para satisfacción del usuario con las simulaciones de Unity.

Capítulo 2

Entorno de Desarrollo

En este capítulo se explicarán las distintas herramientas y tecnologías empleadas para llevar a cabo el desarrollo del proyecto. Se agrupan en tres categorías: en la primera, se agrupan las aplicaciones para generar el proyecto en Unity; en la segunda, se explica la base de datos que se usó; en la tercera, se agrupan las aplicaciones para la creación de la página web; y en la cuarta, se agrupan todas las demás aplicaciones que fueron necesarias para la realización de tareas concretas, diseño y gestión del proyecto.

2.1 Unity

Para la realización de este proyecto la plataforma clave de trabajo es Unity. Unity es un entorno de desarrollo de videojuegos creado por la empresa Unity Technologies que fue fundada en 1988 en Copenhague, Dinamarca por David Helganson, Nicholas Francis y Joachim Ante con la idea de ser asequible para todos los usuarios que quisiesen desarrollar aplicaciones independientes. Gracias a la consistencia del motor gráfico que usa OpenGL, Direct3D, OpenGL ES e Interfaces podemos generar aplicaciones tanto de escritorio como para sistemas móviles, por ejemplo, Android o iOS. Con las nuevas versiones a partir de la 5.4.0 podemos generar proyecto en navegadores web que soporten WebGL, que suele ser actualmente la mayoría de los navegadores.

Además, el paquete que vamos a utilizar para generar este proyecto está disponible en la tienda de Unity Asset Store, que va a añadir a la funcionalidad de la plataforma objetos prefabricados de una planta industrial al igual que elementos específicos como interfaces de comunicación y scripts.



Figura 2.1: Logo de Unity5

2.1.1 UnityHub

UnityHub es el entorno en donde administrar las varias instalaciones del editor junto con los componentes asociados, así como para crear nuevos proyectos o abrir proyectos existentes. También puedes encontrar recursos, como videotutoriales que dan una idea de las posibilidades a crear en Unity.



Figura 2.2: Logo UnityHub

2.1.2 Unity2020.3.X

Para este proyecto en particular, necesitaremos la versión más nueva LTS del editor de Unity ya que el paquete de Game4Automation (Pro) que utilizaremos requiere de una versión de editor superior a la 2019.X.X, por tanto, instalaremos esta versión para evitar problemas de compatibilidad.

Quiero añadir que esta instalación ya es el editor en donde crearemos las escenas que serán alojadas después en una página web.

2.2 Base de datos

La estructura y almacenamiento de los datos de los usuarios que acceden a la página web se realizó mediante la herramienta de MySQL. A continuación, se explica que es MySQL.

2.2.1 MySQL

MySQL es un sistema de gestión de bases de datos relacional desarrollado bajo la licencia de Oracle Corporation y es una de las bases de datos más populares para entorno de desarrollo web. Elegí esta base de datos por la experiencia de uso que tenía con ella además de que se instala por defecto cuando añades la aplicación Wampserver64 al equipo. Además, existen varias interfaces de programación de aplicaciones que permiten a aplicaciones escritas en diversos lenguajes de programación, acceder a la base de datos de MySQL.



Figura 2.3: Logo MySQL

2.3 Página web

La página web será la puerta de enlace para comunicar a los usuarios objetivos con las plantas diseñadas en Unity. Además, su construcción se realiza usando tecnologías web como HTML5, CSS y PHP para lenguaje de máquina servidor. También usaremos una plantilla de Bootstrap que tomaremos como base para modificar y generar nuestro diseño de página web responsiva. A continuación, detallaremos el uso de cada una en nuestra página.

2.3.1 HTML5

Este lenguaje de marcado lo utilizaremos para generar las etiquetas en nuestro diseño web y estructurar así la información de las vistas. HTML5 presenta dos variantes de sintaxis para HTML, no obstante, en nuestro proyecto utilizaremos la <<clásica>> conocida con la etiqueta inicial (text/html).



Figura 2.4: Logo HTML5

2.3.2 CSS

CSS es un lenguaje de diseño gráfico para definir y crear presentaciones de un documento escrito en lenguaje de marcado. Se usa principalmente para embellecer el diseño visual de un documento web escrito en HTML.



Figura 2.5: Logo CSS

2.3.3 PHP

PHP es un lenguaje popular de scripting de propósito general que sirve muy bien para el desarrollo web ya que es rápido, flexible y pragmático. Utilizaremos para este proyecto la versión 7.4.9 que viene por defecto en el paquete Wampserver64, no obstante, habrá que configurarlo para añadirlo en el path de las variables de entorno de Windows debido a que por defecto se configura con otra versión desactualizada de PHP que viene en Wampserver64, la versión 5.6.

Esta parte es clave de la pagina web ya que se encargará de realizar las consultas a la base de datos, generar los nuevos registros de usuarios y enviar los correos electrónicos de petición de contraseña.



Figura 2.6: Logo PHP

2.3.4 Bootstrap

Bootstrap es una biblioteca multiplataforma de código abierto para diseño de sitios y aplicaciones web. Contiene plantillas de diseño basado en HTML y CSS, así como extensiones de JavaScript adicionales. A diferencia de muchos frameworks web, solo se ocupa del desarrollo front-end.



Figura 2.7: Logo Bootstrap

2.4 Herramientas para el desarrollo

Durante el desarrollo del proyecto se han utilizado distintas herramientas que han contribuido en el diseño y a una mejor gestión del código del proyecto.

2.4.1 Visual Studio 2017 y Code

Los editores utilizados son Visual Studio 2017 que es un IDE (entorno de desarrollo integrado) que es compatible con múltiples lenguajes de programación y Visual Studio Code que es un editor de código fuente desarrollado por Microsoft para Windows, Linux y macOS, a diferencia de Visual Studio 2017 que solamente esta para Windows y macOS, e incluye soporte para la depuración, control integrado de git, resaltado de sintaxis, finalización de código inteligente, fragmentos y refactorización de código. Además, es personalizable, gratuito y de código abierto y si necesitas usar un lenguaje de programación en específico puedes instalar las extensiones necesarias.

No es necesario tener los dos programas instalados ya que con uno puedes realizar tanto el diseño web como la programación en C#, solamente habría que configurar las extensiones instaladas.

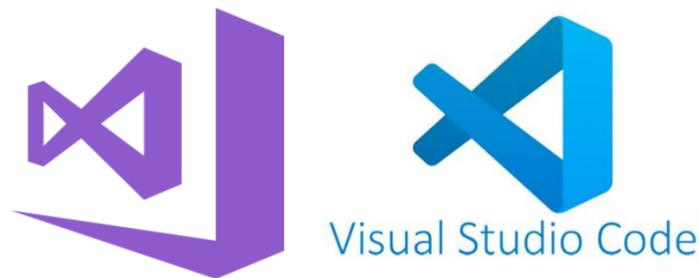


Figura 2.8: Logo Visual Studio 2017 y Logo Visual Studio Code

2.4.2 Github

Github es una plataforma de desarrollo colaborativo para alojar proyectos utilizando el sistema de control de versiones Git.

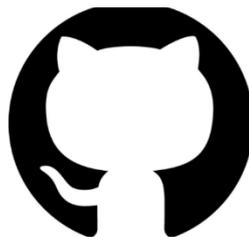


Figura 2.9: Logo Github

2.4.3 Google Chrome y Drive

Google Chrome es un navegador web de código cerrado desarrollado por Google que funciona en cualquier sistema operativo. Es el navegador más completo para la

creación de aplicaciones web y dispone de una gran variedad de funcionalidades que facilitan el desarrollo y testeado del código. La principal funcionalidad que use para depurar y testear fueron las herramientas de desarrolladores (Chrome DevTools) que está formado por un conjunto de pestañas con un amplio catálogo de recursos de depuración. También hice uso de Google Drive para alojar las plantas convertidas en aplicación de escritorio como archivos públicos y así poder añadirlos a la web para descargar.

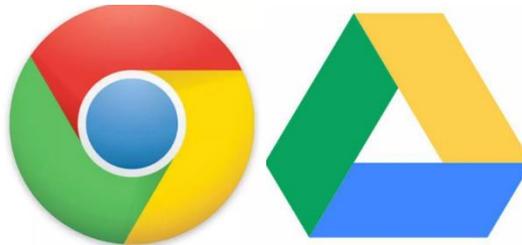


Figura 2.10: Logo Google Chrome y Logo Google Drive

2.4.4 Wamp

WampServer es una plataforma de desarrollo web basada en Windows para aplicaciones web dinámicas que utiliza el servidor Apache2, el lenguaje de scripting PHP y una base de datos MySQL. En este proyecto se utilizó para tener la instalación de las 3 herramientas que vienen en la plataforma y también para lanzar un servicio en localhost y depurar la página web.



Figura 2.11: Logo de Wampserver

2.4.5 Apache

Apache es un servidor HTTP de código abierto para plataformas Unix, Windows, Macintosh y otras que implementa un protocolo HTTP/1.1. en este proyecto se utilizó para generar una conexión local de Apache con el fin de previsualizar y probar código mientras se está desarrollando.



Figura 2.12: Logo de Apache

2.4.6 Word Office

La elaboración de la memoria del trabajo se ha hecho usando Word Office. Es un editor basado en la nube que se utilizar para escribir, editar y publicar documentos de texto.



Figura 2.13: Logo de Word Office

Capítulo 3

Game4Automation

En este capítulo nos centraremos en explicar el análisis hecho sobre la herramienta de Game4Automation, haciendo hincapié en el funcionamiento, las ventajas y desventajas que presenta, las posibles interfaces que se pueden generar y la programación lógica detrás de estas; y, por último, los componentes utilizados para realizar la simulación.

3.1 Qué es Game4Automation

Game4Automation es un paquete desarrollado por Insight2 GmbH, una empresa con sede en Alemania y que forma parte de la industria de diseño de sistemas informáticos y servicios relacionados. Esta extensión de la aplicación de Unity que se puede obtener comprándola en Unity Asset Store añade una variedad de implementaciones, convirtiendo el motor gráfico para diseño de videojuegos 3D en un elemento para diseño de simulaciones virtuales de plantas industriales. Además, trae elementos prefabricado y un catálogo de funciones que acompañan a estos componentes en la creación de las escenas.



Figura 3.1: Logo Game4Automation

3.2 Funcionamiento

El funcionamiento de Game4Automation lo podemos dividir en tres partes destacadas:

- La primera parte orientada a los elementos de fábrica que son el conjunto de

los modelos en 3D de los autómatas, texturas y materiales que utilizaremos para visualizar en las escenas.

- La segunda parte orientada al comportamiento individual de cada pieza dentro de la simulación, ya que cada elemento tiene enlazado en sus propiedades un script con el respectivo funcionamiento básico ya sea una cinta transportadora que se mueva en el eje de la cinta con cierta velocidad o un brazo robótico que pueda agarrar un elemento MU dentro de la escena.
- La tercera parte es la que se encarga de generar la sincronización de los comportamientos individuales de los elementos de la planta, creando interfaces de sensores que captarán las señales input y output de la simulación y que a su vez estos datos serán tratados por las PLCs (Controlador Lógico Programable) para completar el circuito.

Hay que tener en cuenta que detallaremos más adelante en este capítulo la importancia de la tercera parte que es la clave para poder desarrollar con éxito la simulación de una planta industrial.

3.3 Ventajas y desventajas

Con el uso de esta tecnología la mayoría de las aportaciones son beneficiosas, excepto por algunos pequeños inconvenientes que se describen a continuación.

Las principales ventajas de las simulaciones de procesos industriales son:

- La simulación puede ser aplicada en la fase de diseño de un sistema o bien cuando la producción está en marcha, con el fin plantear alternativas y del mismo modo mejorar ciertos aspectos.
- Enlazando con el punto anterior, la simulación permite evaluar tantos escenarios como se requieran, es decir, se pueden simular diferentes alternativas aplicables al sistema real hasta dar con la mejor solución.
- Permite conocer en profundidad el comportamiento del sistema con rapidez, sin necesidad de realizar una gran inversión. Además, no interfiere en la producción, ya que la experimentación se realiza en el modelo creado a partir del sistema real.
- Aporta gran facilidad para identificar áreas que suponen un problema en la capacidad de producción, ya sea cuellos de botella o por el contrario largos periodos de inactividad.
- Posee una gran flexibilidad para modelizar cualquier tipo de sistema, por muy complejo que este pueda llegar a ser. De esta manera permite que un problema complejo, pueda ser resuelto de manera matemática a través del software de simulación.
- Completando el punto anterior, los grandes avances en la informática garantizar mejores softwares de simulación, y por lo tanto la capacidad de

estos para resolver problemas cada vez más complejos y en periodos de tiempo más reducidos.

Respecto a las desventajas, podemos destacar las siguientes:

- En ocasiones se hace un uso incorrecto de la simulación, es decir, si un modelo analítico es válido, es preferible debido a que la solución será más acertada que la ofrecida por la simulación.
- La simulación no proporciona soluciones exactas, los resultados que ofrece son estimaciones acotadas dentro de un margen de confianza.
- Al tratarse en la mayoría de las ocasiones de simulaciones con cierta variabilidad en el modelo, los resultados también serán aleatorios.
- En ciertas ocasiones y dependiendo de la complejidad del sistema, no es posible tener la certeza de que el modelo de simulación sea válido

3.4 Interfaces y PLC

En esta parte explicaremos porque es importante la creación de las interfaces y que tiene que ver el diseño de las PLCs con ellas.

Las interfaces de automatización manejan la comunicación entre controladores de automatización reales o virtuales y utilizan objetos PLCInput y PLCOutput estándar (como PLCInputFloat o PLCOutputBool) para almacenar valores de señal. La propia interfaz maneja los valores en tiempo real de las señales. Algunas interfaces proporcionan la capacidad de importar todas las señales y crear automáticamente los objetos PLCInput o PLCOutput adecuados en la jerarquía de Game4Automation. A veces, las señales se pueden importar mediante comunicación con el propio controlador, o a veces se puede exportar y utilizar una lista de señales utilizada para programar el controlador de automatización. A continuación, comentaremos tres tipos de interfaces de todas las que se pueden generar con Game4Automation debido a que son las principales que tuvimos en cuenta a la hora de realizar los controladores de la simulación.

3.4.1 Creadas Manualmente

En Game4Automation se puede generar estas interfaces como objetos independientes a un controlador específico y manejar el tratamiento de estos datos con un controlador propio programado en C# por el mismo operador. En este proyecto se ha realizado de ese modo, generando las interfaces de cada pieza dentro de la simulación y añadiendo un objeto independiente asociado con un script en C# para realizar el control de las señales en tiempo real.

3.4.2 Siemens7

La interfaz TCP-IP del S7 proporciona una conexión directa a los controladores de hardware de Siemens, además se ejecuta en un subproceso separado para lograr el máximo rendimiento. La interfaz TCP-IP del S7 se basa en Snap7 de Davide Nardella y se puede comprar la aplicación del controlador TIA Portal desde la página web oficial de

Siemens. Es debido a esto que al final tuvimos que descartarla como opción a desarrollar.

3.4.3 OPC-UA

OPCUA es un estándar, mantenido por la fundación OPC, para la comunicación industrial. La mayoría de los proveedores de dispositivos de automatización, como los fabricantes de PLC y robots, están proporcionando servidores OPC-UA. A veces, el servidor OPC-UA se integra directamente en el dispositivo de automatización y, a veces, se instala un software especial en una PC. Se puede utilizar una aplicación externa para generar el controlador como TwinCat 3 y conectarlo a la interfaz de OPC-UA en Game4Automation, así se tendría separada la lógica del proyecto con la simulación.

3.5 Componentes

En este apartado, explicaremos por encima los diferentes elementos que aparecerán en las simulaciones de nuestro proyecto.

3.5.1 Caja

Es un elemento MU, es decir, una unidad móvil que nos servirá para poder introducir otros MU dentro del mismo o desplazarlo en la simulación.

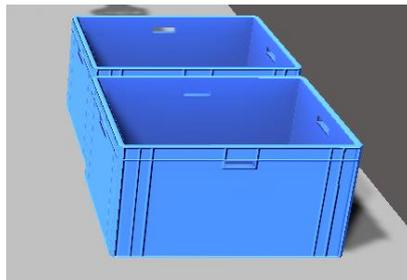


Figura 3.2: Cajas

3.5.2 Cinta transportadora grande

Es un elemento que genera un movimiento en una dirección dependiendo del sentido y que usaremos en el proyecto para desplazar las cajas.

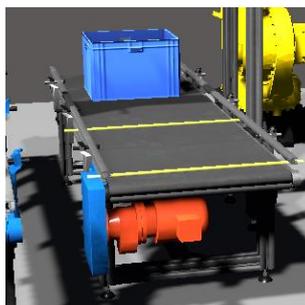


Figura 3.3: Cinta transportadora grande

3.5.3 Cinta transportadora pequeña

Es un elemento que, al igual que la cinta transportadora grande, genera un movimiento en una dirección dependiendo del sentido y que usaremos en el proyecto para desplazar las latas.



Figura 3.4: Cinta transportadora pequeña

3.5.4 Lata

Es un elemento MU que utilizaremos para introducir dentro de las cajas al igual que ser tomado y desplazado por el gancho.



Figura 3.5: Lata

3.5.5 Robot

Es un elemento automático que puede agarrar objetos MU y desplazarlos a otra posición, se genera el movimiento mediante animaciones.

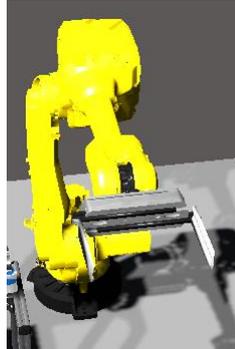


Figura 3.6: Robot

3.5.6 Plataforma XY

Es un elemento automático que al igual que el robot puede agarrar objetos MU y desplazarlos a otra posición con la diferencia que se mueve con valores decimales de distancia y en 2 ejes (X,Y).

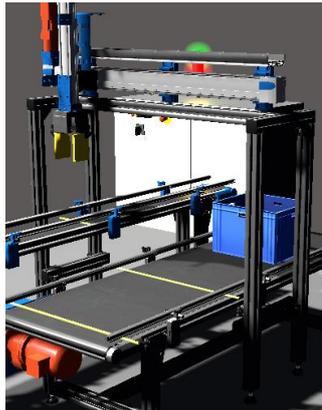


Figura 3.7: Plataforma XY

3.5.7 Lampara y caja de mando

Elemento decorativo de la simulación para asemejarse a un ambiente industrial.

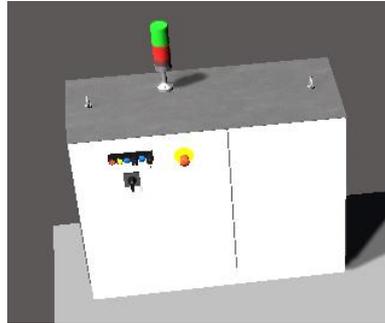


Figura 3.8: Lampara y caja de mando

3.5.8 Sensor

Elemento clave de nuestro proyecto ya que es quien se va a encargar de coordinar los movimientos de los demás componentes en la simulación.

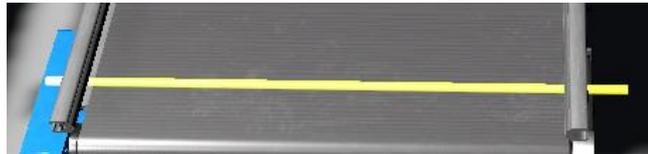


Figura 3.9: Sensor

3.5.9 Sumidero

Elemento que sirve para eliminar los MU dentro de la simulación.

Capítulo 4

Diseño

En este capítulo se presentan los diseños realizados para las diferentes vistas del proyecto separando en dos partes la página web y la simulación de las plantas. A continuación, se muestran cada una de las partes.

4.1 Página web

En este apartado veremos la vista con respecto al diseño web en donde se van a subir las plataformas de Unity.

4.1.1 Login, registro y recordar contraseña

Al acceder a la pagina web se carga la vista para iniciar sesión en la aplicación. En ella se solicita el nombre de usuario y la contraseña para entrar. En caso de no recordar la contraseña, hay una opción que te redirige a una nueva pantalla para recuperar la contraseña. Por otro lado, si el usuario no tiene cuenta registrada puedes hacerlo mediante otro enlace, en donde tienes que facilitar la información como el nombre completo del usuario, un correo electrónico, un nombre de usuario para la autenticación y la contraseña.

Game4Automation Trabajo de Fin de Grado



The image shows a login form with a light blue background. At the top, the word "Login" is centered in a bold, black font. Below it, there are two input fields: the first is labeled "Nombre De Usuario" and the second is labeled "Contraseña". Below the password field is a button labeled "Entrar". At the bottom of the form, there are two links: "¿Olvidaste la contraseña? [Pincha Aquí!](#)" and "¿No estas registrado? [Registrate Aquí!](#)".

Figura 4.1: Vista login.php

Game4Automation Trabajo de Fin de Grado



The image shows a registration form titled "Registrar" on a light purple background. It contains four input fields: "Nombre Completo", "E-mail", "Nombre De Usuario", and "Contraseña". Below the fields is a "Registrar" button and a link that says "¿Ya tienes una cuenta? ¡Login Aquí!!".

Registrar

Nombre Completo

E-mail

Nombre De Usuario

Contraseña

¿Ya tienes una cuenta? [¡Login Aquí!!](#)

Figura 4.2: Vista register.php



The image shows a password reset form titled "Reenviar contraseña" on a light purple background. It contains one input field for "E-mail" and an "Enviar" button.

Reenviar contraseña

E-mail

Figura 4.3: Vista forgotpw.php

4.1.2 Pantalla de bienvenida

Después de haber iniciado sesión, se presenta la pantalla de bienvenida. Esta consta de una breve introducción, la zona donde descargar las simulaciones de las plantas industriales y los enlaces tanto a la web oficial de Game4Automation como a el canal de YouTube oficial.

Game4Automation Trabajo de Fin de Grado

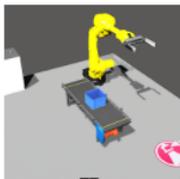
¡Bienvenido, jonathan!

[Logout](#)

Introducción a Game4Automation

Actualmente la robótica es un sector que está tomando fuerza en la industria, ya sea desde la creación de automóviles como la realización de pequeñas tareas. Esto se debe a que se ha visto la importancia que tiene generar estos automatats que pueden trabajar el doble de eficiente, o más, que un ser humano por un coste menor de mano de obra. Además evitando riesgo en algunas tareas complejas para el ser humano debido a los límites que tiene el cuerpo, por ejemplo levantar objetos pesados o mantener durante mucho tiempo la concentración en tareas minuciosas.

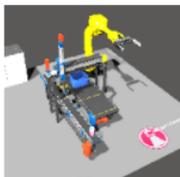
En esta página te vas a encontrar simulaciones de plantas industriales creadas en Unity3D con el addon de Game4Automation para demostrar la importancia que tiene tener un boceto del montaje y ver el funcionamiento sin haber generado todavía la construcción del mismo. A veces ver es creer, solamente a veces.



Planta 1 : Desarrollo básico

Primera planta realizada con un sencillo ejemplo de un circuito en bucle. Dos elementos dentro de la planta: una cinta que desplazará las cajas y un robot que interactuará con las cajas para colocarlas en el extremo contrario dependiendo del sentido del circuito.

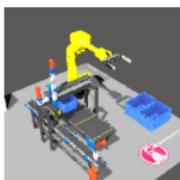
[Download](#)



Planta 2 : Desarrollo Intermedio

Segunda planta desarrollada al igual que la primera con un circuito circular con la diferencia de nuevos elementos como un gancho y el uso del objeto "sink" para borrar específicamente las latas dentro de la caja. Además en este circuito existe una dependencia entre 2 elementos claves que son el gancho y la cinta grande.

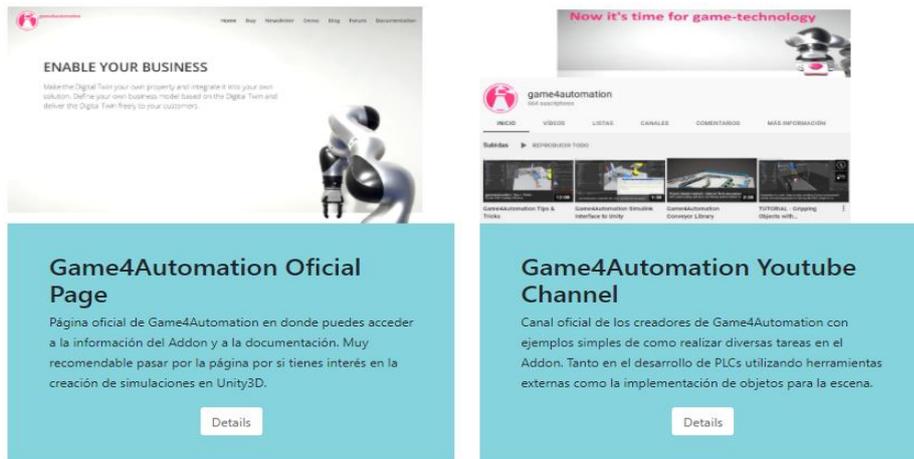
[Download](#)



Planta 3 : Desarrollo Avanzado

Tercera planta desarrollada tomando como base la segunda y añadiendo complejidad de elección de las cajas con 2 posibles finales, fallo y acierto, que dependerá de la decisión del usuario que la controla. Igualmente añadir que la coordinación ahora se toma entre 3 elementos bases que son: el gancho, la cinta grande y el brazo robótico.

[Download](#)



Copyright © 2021 - TFG - Jonathan Eliot Viera Rivas

Figura 4.4: Pagina de bienvenida

4.2 Unity: Plantas

En este apartado veremos una vista de las tres plantas creadas y se puede apreciar el nivel de dificultad por los elementos que aparecen en cada una de ellas.

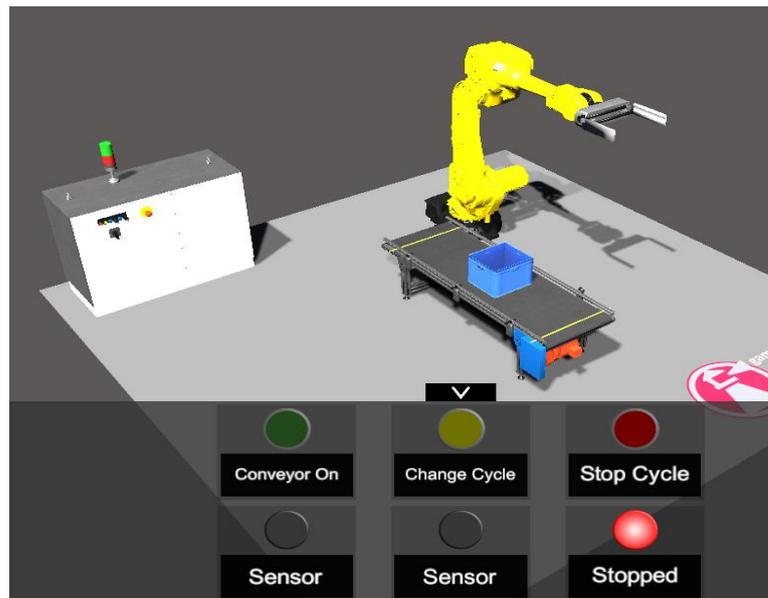


Figura 4.5: Vista planta 1

En la vista de la planta 1 apreciamos 3 elementos: el robot que se encargará de desplazar la caja de un extremo a otro, la cinta que desplazará la caja de un lado a otro

dependiendo del sentido y la caja que es el objeto movable dentro del circuito. Además del panel de control con los botones de parar la cinta, cambiar de sentido y parar el circuito.

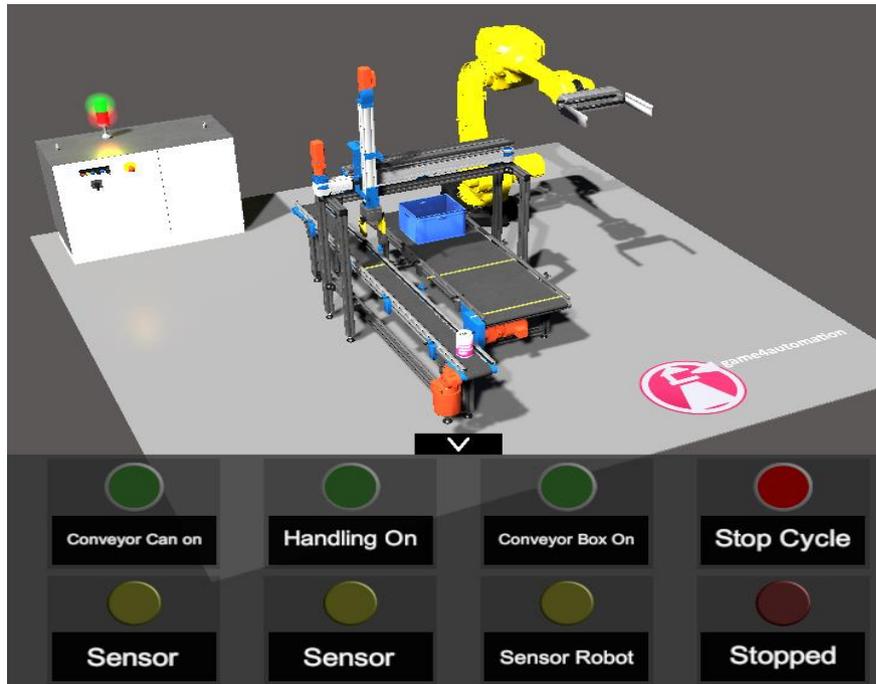


Figura 4.6: Vista planta 2

En la vista de la planta 2 tendremos un circuito donde una plataforma XY tomará las latas de la cinta transportadora pequeña para rellenar una caja de la cinta transportadora grande y el robot se encargará de coger la caja y ponerla al principio. El panel de control del circuito nos deja desactivar y activar los elementos de la simulación como parar el circuito.

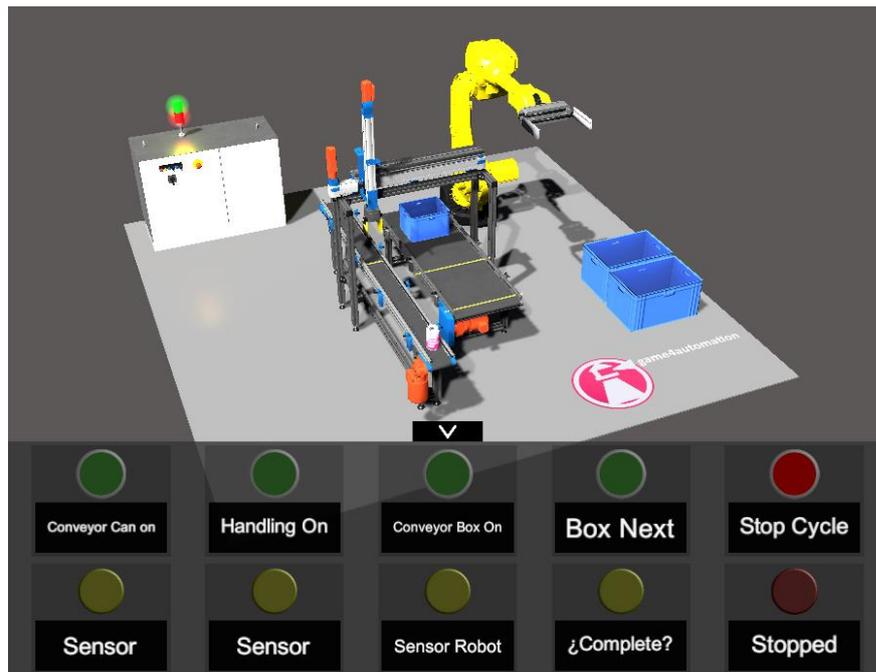


Figura 4.7: Vista planta 3

En la vista de la planta 3 tendremos un circuito complejo de coordinación en donde tomando como base el circuito de la planta 2 añadiremos la funcionalidad de decisión del operador con respecto a la caja. Si queremos la caja, entonces el comportamiento será igual que en la planta 2 dejando esta vez la caja como un acierto en una de las cajas grandes, y si queremos desechar la caja solo tendremos que indicar que sea la caja siguiente para que cambie el comportamiento del circuito y esta se deseche si ya estaba en el sensor. También quiero destacar que si se activa mientras la caja no ha llegado al sensor para rellenar entonces la cinta no va a parar ni va a actuar la plataforma XY llegando al robot para desecharla en la otra caja grande.

Capítulo 5

Desarrollo

5.1 Página web

La página web ha sido desarrollada en el entorno de trabajo de Visual Studio Code con las extensiones de PHP IntelliSense, PHP Debug, HTML CSS Support y Prettier, necesarias para poder llevar cabo la programación de la página web. La estructura del desarrollo se ha implementado de la forma más organizada posible para facilitar el desarrollo. La estructura del código se divide en cinco partes: el núcleo, que son las páginas web con las distintas funcionalidades de login.php, logout.php, register.php, forgotpw.php e index.php; la carpeta includes que contiene la cabecera y la conexión con la base de datos; la carpeta phpmailer que contiene funciones de creación de envío de mensajes electrónicos; la carpeta de img que contiene las imágenes de la página web y, por último, la carpeta css que contiene los estilos del lenguaje de marcado.

Cabe destacar el uso también de la aplicación de Wampserver para ir depurando la página en paralelo con la programación web.

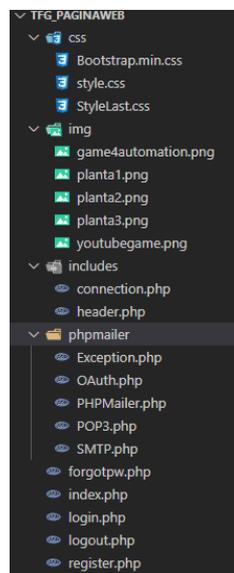


Figura 5.1: Estructura organizativa de la página web.

5.1.1 Funcionalidades

En esta parte vamos a detallar como se desarrollaron las funcionalidades del login.php, logout.php, register.php, forgotpw.php e index.php.

El archivo index.php es la página de inicio de nuestra aplicación web y funciona tanto como página de bienvenida como de verificación de autenticación. Mientras no se encuentre una sesión iniciada con un valor de usuario, nos mandará a la pagina de login.php para que nos autentiquemos. En caso de si tener un valor de usuario de sesión, actuará como una página de bienvenida.

Mientras estemos en la pagina de bienvenida podremos hacer un logout para cerrar la sesión actual, descargar las simulaciones a través de los botones de download y acceder a los enlaces externos de la pagina oficial de Game4Automation como el canal de YouTube.

```
<!DOCTYPE html>
<html>

<?php include("includes/header.php"); ?>

<?php
session_start();
if(!isset($_SESSION["session_username"])) {
    header("location:login.php");
} else {
?>

<body>
<div class="container">
    <section class="tm-section-head" id="top">
        <header id="header" class="text-center tm-text-gray">
            <h1>Game4Automation Trabajo de Fin de Grado</h1>
        </header>
    </section>
```

Figura 5.2: Función para redirigir al login o pantalla bienvenida

Con el archivo logout.php finalizaremos la sesión actual.

```
<?php
session_start();
unset($_SESSION['session_username']);
session_destroy();

header("location:login.php");
?>
```

Figura 5.3: Función para el logout

Como comentamos anteriormente cuando nos encontremos sin un valor de sesión de usuario, la página que se mostrará es la de login.php que se encargará de verificar la autenticación de credenciales, nombre de usuario y contraseña con las añadidas en la

base de datos. Esta verificación se realiza mediante una petición para recorrer las filas de la base de datos con los valores de username y password hasta encontrar semejanza entre los valores escritos en el formulario con los valores alojados en la base de datos. Pueden ocurrir 3 casos con 2 posibles consecuencias:

- Sean semejantes, lo que modificaría la sesión actual con el nombre de usuario que se autentica y se nos redirigirá a index.php para acceder a la página de bienvenida.
- No sean semejantes y los campos están rellenos, generando un error que indica que no existe ese usuario.
- No estén rellenos algunos de los campos, generando un error que indica que tienen que estar todos los campos rellenos.

```

?php

if(isset($_SESSION["session_username"])){
    header("Location: index.php");
}

if(isset($_POST["login"])){
    if(!empty($_POST['username']) && !empty($_POST['password'])) {
        $username=$_POST['username'];
        $password=$_POST['password'];

        $query = mysqli_query($conexion, "SELECT * FROM usertbl WHERE username='".$username."' AND password='".$password."'");

        $numrows = mysqli_num_rows($query);
        if($numrows!=0)
        {
            while($row=mysqli_fetch_assoc($query))
            {
                $dbusername=$row['username'];
                $dbpassword=$row['password'];
            }

            if($username == $dbusername && $password == $dbpassword)
            {
                $_SESSION['session_username']=$username;

                /* Redirigir a buscador */
                header("Location: index.php");
            }
        } else {
            $message = "Nombre de usuario o contraseña invalida";
        }
    } else {
        $message = "Todos los campos son requeridos";
    }
}
?

```

Figura 5.4: Función de login

En la pagina register.php tendremos un formulario a rellenar que generara la fila correspondiente en la base de datos, por obligación, todos los campos tienen que estar rellenos en el registro y no puede haber otra cuenta con el mismo username ya que es un valor único.

```

<?php
if(isset($_POST["register"])){
    if(!empty($_POST['full_name']) && !empty($_POST['email']) && !empty($_POST['username']) && !empty($_POST['password'])) {
        $full_name=$_POST['full_name'];
        $email=$_POST['email'];
        $username=$_POST['username'];
        $password=$_POST['password'];

        $query = mysqli_query($conexion, "SELECT * FROM usertbl WHERE username='".$username."'");
        $numrows=mysqli_num_rows($query);

        if($numrows==0)
        {
            $sql="INSERT INTO usertbl
            (full_name, email, username,password)
            VALUES('$full_name','$email', '$username', '$password')";

            $result=mysqli_query($conexion, $sql);

            if($result){
                $message = "Cuenta Correctamente Creada";
            } else {
                $message = "Error al ingresar datos de la informacion!";
            }
        } else {
            $message = "El nombre de usuario ya existe! Por favor, intenta con otro!";
        }
    }else {
        $message = "Todos los campos no deben de estar vacios!";
    }
}
?>

```

Figura 5.5: Función para registrar al usuario

Por último, la página de forgotpw.php podremos obtener a través de un mensaje de correo electrónico la contraseña de usuario que habíamos creado a la hora de registrarnos. Para esto, tenemos que colocar el correo electrónico de nuestra cuenta creada y verificará en la base de datos si existe, si es así enviará un mensaje formateado en HTML con la contraseña asociada a ese correo electrónico. Para el paso de este mensaje usamos la librería de PHPMailer, una librería creada por usuarios que se puede descargar desde Github y nos deja realizar el envío de mensajes por protocolo SMTP.

```

if($numrows!=0)
{
    while($row=mysqli_fetch_assoc($query))
    {
        $dbemail=$row['email'];
        $dbusername=$row['username'];
        $dbpassword=$row['password'];
    }

    if($email == $dbemail)
    {
        try {
            //Server settings
            $mail->SMTPDebug = 0; //Enable verbose debug output
            $mail->isSMTP(); //Send using SMTP
            $mail->Host = 'smtp.gmail.com'; //Set the SMTP server to send through in this case gmail
            $mail->SMTPAuth = true; //Enable SMTP authentication
            $mail->Username = 'Correo para autenticarse'; //SMTP username
            $mail->Password = 'clave'; //SMTP password clave del correo que envia las notificaciones
            $mail->SMTPSecure = 'tls'; //Enable implicit TLS encryption
            $mail->Port = 587; //TCP port to connect to; use 587 if you have set `SMTPSecure = PHPMailer::ENCRYPTION_STARTTLS`

            //Recipients
            $mail->setFrom('Correo autenticado que envia la contraseña', 'Game4Automation Mailer');
            $mail->addAddress($email);

            //Content
            $mail->isHTML(true); //Set email format to HTML
            $mail->Subject = 'Recordatorio de tu contraseña';
            $mail->Body = 'html'
            <html>
            <head>
            <title>Aquí esta tu contraseña</title>
            </head>
            <body>
            <h1>Hola '.$dbusername.'</h1>
            <p>Tu contraseña es '.$dbpassword.'. No la olvides. </p>
            </body>
            </html>;

            $mail->send();
            echo 'Message has been sent';
        } catch (Exception $e) {
            echo "Message could not be sent. Mailer Error: ($mail->ErrorInfo)";
        }
    }

    /* Redirigir a buscador */
    header("Location: index.php");
}

```

Figura 5.6: Verifica si existe correo y envía el correo.

5.2 Bases de datos

Para este apartado indicaremos la estructura que va a tener nuestra base de datos en donde se crearan nuestras cuentas de usuarios.

Se va a realizar una creación sencilla mediante instrucciones en MySQL de una estructura que abarca 5 campos: id, full_name, email, username y password. Las restricciones de estos campos son que no pueden ser nulos, el valor de id tiene que irse incrementando en 1 cada vez que se añade un usuario ya que no puede haber dos usuarios con la misma id y, por último, indicamos que id va a ser nuestra clave primaria y username es una clave única.

```

CREATE DATABASE IF NOT EXISTS tfqdb;
CREATE TABLE IF NOT EXISTS `usertbl` (
  `id` int(11) NOT NULL auto_increment,
  `full_name` varchar(32) collate utf8_unicode_ci NOT NULL default '',
  `email` varchar(32) collate utf8_unicode_ci NOT NULL default '',
  `username` varchar(20) collate utf8_unicode_ci NOT NULL default '',
  `password` varchar(32) collate utf8_unicode_ci NOT NULL default '',
  PRIMARY KEY (`id`),
  UNIQUE KEY `username` (`username`)
) ENGINE=MyISAM DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci;

```

Figura 5.7: Diseño de la base de datos

5.3 Unity

Para este apartado desarrollaré la metodología que seguí en la aplicación de Unity 2020.3.X para la creación y simulación de los procesos industriales. Además, definiré el uso de Visual Studio 2017 para la programación de las PLCs en el lenguaje de programación C#.

5.3.1 Configuración de Unity

Game4Automation Pro es un paquete que necesita unas versiones actualizadas del editor de Unity, por tanto, primero tuvimos que instalar la versión del editor 2020.3.X para poder trabajar con el add-on. Después de realizados estos pasos de instalación y haber lanzado el editor tenemos que agregar el paquete externo de Game4Automation accediendo a Assets e importándolo como un paquete personalizado tomando en cuenta la ruta en donde lo tenemos guardado. Después de haber cargado las dependencias del paquete, podemos hacer uso de éste para nuestro proyecto.

Con esto configurado ya podemos generar nuestra primera escena de Game4Automation accediendo a la barra de tareas de Game4Automation>Create New Game4Automation Scene.

5.3.2 Desarrollo planta 1

El objetivo de esta planta es desarrollar un circuito cíclico en donde una cinta transporta una caja hasta el sensor del extremo siguiendo el sentido actual, cuando la caja llega a tocar el sensor se para la cinta y actúa el robot para colocar esta caja al final de la cinta en el otro extremo del sentido. Es posible modificar el sentido del circuito a través de los paneles de control al igual que parar el circuito completo o elementos concretos.

La estructura de las interfaces y PLCs se muestra en la figura siguiente:

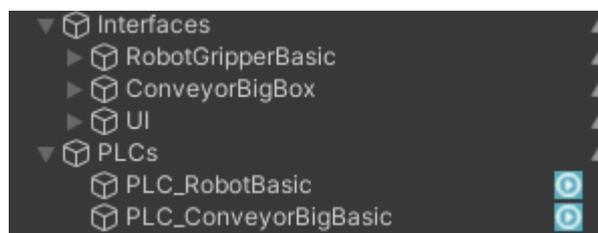


Figura 5.8: Interfaces y PLCs de la planta 1

En las interfaces tendremos los Inputs y Outputs de los elementos de la simulación y la PLC correspondiente al autómata. Definiremos brevemente que valores de nuestros sensores nos afectan para generar el comportamiento lógico de la simulación:

- Para el RobotGripperBasic (Robot), tendremos que tomar en cuenta las señales de si el final del brazo se está abriendo o cerrando, Inputs, o si ya se encuentra cerrado o abierto, Outputs.

La PLC asociada del robot organiza los valores de estados del autómatas y lanza la animación correspondiente al estado actual. Inicialmente empieza en un estado de espera y solamente cuando el sensor de alguno de los extremos esté activado cambiará de estado. Si se encuentra en un estado como closegripper cerrará el final del brazo modificando los valores de close a true, cuando se complete se pondrá closed a true. En caso de opengripper se generará una acción parecida, pero para los sensores de open y opened.

- Para ConveyorBigBox (Cinta transportadora grande), tomaremos en cuenta las señales de la posición actual en la cinta, si el movimiento es hacia delante o atrás, y los dos valores de los sensores alojados a los extremos.

La PLC asociada a la cinta transportadora se moverá principalmente en 2 sentidos dependiendo de si está cambiado el sentido o no, también cuando detecte una caja en alguno de los sensores de los extremos, parará la cinta.

- Por último, la interfaz de UI contendrá los sensores de los botones y lámparas del panel de control. Con respecto a los botones que son la primera fila: uno para activar o desactivar la cinta, otro para cambiar el ciclo y otro para parar todo el circuito. Con respecto a las lámparas que se sitúan en la segunda fila nos ayudaran a notificarnos que está ocurriendo en la simulación, en este caso que sensores de los extremos se han activado y si está parado el circuito.

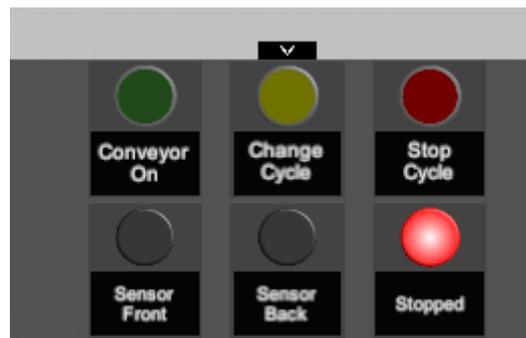


Figura 5.9: Panel de control planta 1

5.3.3 Desarrollo planta 2

Para el desarrollo de la planta 2 tomaremos como ejemplo la planta 1, no obstante, realizaremos modificaciones en el diseño ya que quitaremos los cambios de sentido y añadiremos nuevas funcionalidades.

Se añade una cinta transportadora pequeña que desplazará latas que se crean al inicio de ésta hasta un sensor. Cuando el valor del sensor de las latas cambie a true se para la cinta para ser tomadas por un nuevo elemento que es una plataforma XY. Este autómatas tiene que agarrar la lata y colocarla dentro de la caja en la cinta grande cuando la caja se encuentre en posición del sensor de caja y la cinta se pare cuando cambie el valor del sensor a true. Cuando se llene una fila en la caja se desplazará la cinta grande un rango de distancia para continuar llenando la caja. Si la caja esta completa se moverá

la cinta de nuevo hasta el sensor del robot donde se volverá a parar. Cuando el valor del sensor del robot sea true, además de parar la cinta, va a cambiar el estado del robot con la finalidad de colocar la caja de nuevo al principio de la cinta transportadora grande.

Hay un nuevo elemento al inicio de la cinta que hace de sumidero borrando las latas que se encuentran en la caja vaciándola para repetir de nuevo el circuito.

La estructura de las interfaces y PLCs se muestra en la figura siguiente:

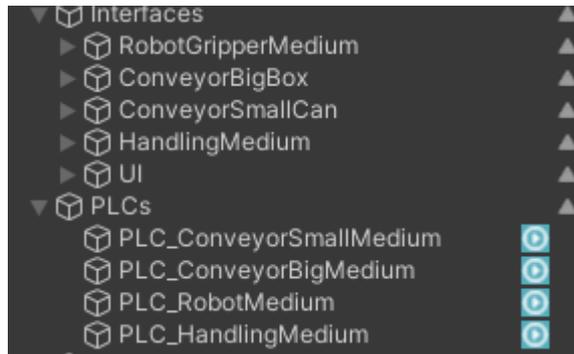


Figura 5.10: Interfaces y PLCs de la planta 2

Definiremos brevemente que valores de nuestros sensores nos afectan para generar el comportamiento lógico de la simulación:

- Para el RobotGripperMedium (Robot), tendremos que tomar en cuenta las señales de si el final del brazo se está abriendo o cerrando, Inputs, o si ya se encuentra cerrado o abierto, Outputs.

La PLC asociada del robot organiza los valores de estados del autómata y lanza la animación correspondiente al estado actual. Inicialmente empieza en un estado de espera y solamente cuando el sensor de alguno de los extremos esté activado cambiará de estado. Si se encuentra en un estado como closegripper cerrará el final del brazo modificando los valores de close a true, cuando se complete se pondrá closed a true. En caso de opengripper se generará una acción parecida, pero para los sensores de open y opened.

- Para ConveyorBigBox (Cinta transportadora grande), tomaremos en cuenta las señales de la posición actual en la cinta, el movimiento hacia delante y los dos valores de los sensores alojados a un extremo y a mitad de la cinta.

La PLC asociada a la cinta transportadora se moverá en un sentido y también cuando detecte una caja en alguno de los 2 sensores, parará la cinta.

- Para ConveyorSmallCan (Cinta transportadora pequeña), tomaremos en cuenta las señales de desplazamiento hacia delante y el sensor de las latas.

La PLC asociada a la cinta transportadora pequeña se moverá en un sentido para desplazar las latas que se encuentran encima de ella hasta el sensor de latas.

- Para HandlingMedium (Plataforma XY) se tomarán en cuenta muchos valores de

sensor y se dividen en 3 partes:

- El gancho al final del brazo tendrá los mismos sensores que el robot.
- La dirección X, donde verificamos si se está desplazando actualmente, el destino de la dirección en valor decimal, la posición actual también en valor decimal y si ya se encuentra en el destino.
- La dirección Y, donde verificamos si se está desplazando actualmente, el destino de la dirección en valor decimal, la posición actual también en valor decimal y si ya se encuentra en el destino.

La PLC asociada a la plataforma XY moverá en una dirección X la parte del gancho al destino desde un punto inicial en el eje X; y en esos puntos iniciales y de destino se moverá en el otro eje Y para recoger la lata de la cinta transportadora pequeña cuando se active el sensor de latas, o soltar la lata en la caja, cuando esté activado el sensor en medio de la cinta transportadora grande, realizando la misma acción comentada en el robot. También cuando se complete una fila en la caja se desplazará la cinta transportadora grande para rellenar la siguiente fila en la caja.

- Por último, la interfaz de UI contendrá los sensores de los botones y lámparas del panel de control. Con respecto a los botones que son la primera fila: uno para activar o desactivar la cinta pequeña, otro para la cinta grande, otro para la plataforma XY y un último para parar todo el circuito. Con respecto a las lámparas que se sitúan en la segunda fila nos ayudaran a notificarnos que está ocurriendo en la simulación, en este caso que sensores se han activado y si está parado el circuito.

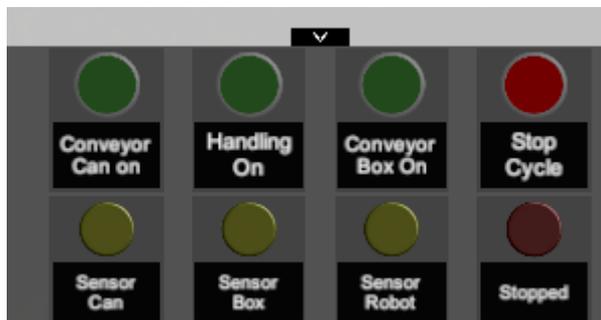


Figura 5.11: Panel de control planta 2

5.3.4 Desarrollo planta 3

Para la planta 3 con nivel de dificultad elevado, partiremos del desarrollo de la planta 2, y añadiremos la posibilidad de decidir si queremos o no queremos la caja que se encuentra en la cinta grande. En caso de querer la caja como un acierto, se dejaría que el circuito actúe como en el desarrollo de la planta 2, llevando la caja el robot a la pila de aciertos. Si no queremos la caja de la cinta, pulsando un botón en el panel de control dejará de rellenarse la caja para desplazarse hacia el sensor del robot, que la tomará y colocará en la pila de fallos. También si este botón esta activado antes de que llegue una

caja al sensor en medio de la cinta grande esa caja no se parará, aunque se desactive el botón a mitad del sensor, dejando que la caja llegue hasta el sensor del robot y se deseche.

Para esta simulación no hay un nuevo elemento si no que se modifica las PLC para generar una coordinación entre elementos más compleja.

La estructura de las interfaces y PLCs se muestra en la figura siguiente:

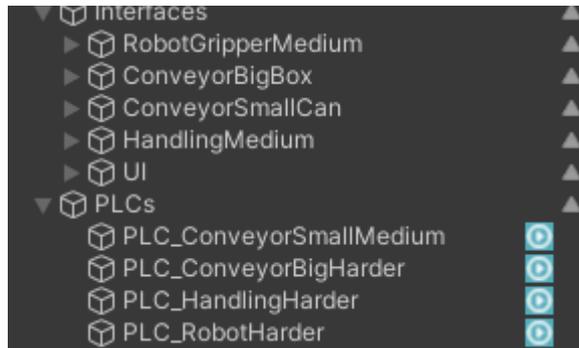


Figura 5.12: Interfaces y PLCs de la planta 3

Definiremos brevemente que valores de nuestros sensores nos afectan para generar el comportamiento lógico de la simulación:

- Para el RobotGripperMedium (Robot), tendremos que tomar en cuenta las señales de si el final del brazo se está abriendo o cerrando, Inputs, o si ya se encuentra cerrado o abierto, Outputs.

La PLC asociada del robot organiza los valores de estados del autómata y lanza la animación correspondiente al estado actual. Inicialmente empieza en un estado de espera y solamente cuando el sensor de alguno de los extremos esté activado cambiará de estado. Si se encuentra en un estado como closegripper cerrará el final del brazo modificando los valores de close a true, cuando se complete se pondrá closed a true. En caso de opengripper se generará una acción parecida, pero para los sensores de open y opened.

También será quien desactive la lámpara de complete.

- Para ConveyorBigBox (Cinta transportadora grande), tomaremos en cuenta las señales de la posición actual en la cinta, el movimiento hacia delante y los dos valores de los sensores alojados a un extremo y a mitad de la cinta.

La PLC asociada a la cinta transportadora se moverá en un sentido y también cuando detecte una caja en alguno de los 2 sensores, parará la cinta.

La acción anterior no se tendrá en cuenta si se activa el botón boxnext.

- Para ConveyorSmallCan (Cinta transportadora pequeña), tomaremos en cuenta las señales de desplazamiento hacia delante y el sensor de las latas.

La PLC asociada a la cinta transportadora pequeña se moverá en un sentido para desplazar las latas que se encuentran encima de ella hasta el sensor de latas.

- Para HandlingMedium (Plataforma XY) se tomarán en cuenta muchos valores de sensor y se dividen en 3 partes:
 - El gancho al final del brazo tendrá los mismos sensores que el robot.
 - La dirección X, donde verificamos si se está desplazando actualmente, el destino de la dirección en valor decimal, la posición actual también en valor decimal y si ya se encuentra en el destino.
 - La dirección Y, donde verificamos si se está desplazando actualmente, el destino de la dirección en valor decimal, la posición actual también en valor decimal y si ya se encuentra en el destino.

La PLC asociada a la plataforma XY moverá en una dirección X la parte del gancho al destino desde un punto inicial en el eje X; y en esos puntos iniciales y de destino se moverá en el otro eje Y para recoger la lata de la cinta transportadora pequeña cuando se active el sensor de latas, o soltar la lata en la caja, cuando esté activado el sensor en medio de la cinta transportadora grande, realizando la misma acción comentada en el robot. También cuando se complete una fila en la caja se desplazará la cinta transportadora grande para rellenar la siguiente fila en la caja.

Se reiniciará las posiciones de filas y columnas a rellenar en la caja al igual que se pone en espera si está activado el valor de boxnext. También asigna el valor de boxnext a la lámpara complete

- Por último, la interfaz de UI contendrá los sensores de los botones y lámparas del panel de control. Con respecto a los botones que son la primera fila: uno para activar o desactivar la cinta pequeña, otro para la cinta grande, otro para la plataforma XY, el botón de decisión y un último para parar todo el circuito. Con respecto a las lámparas que se sitúan en la segunda fila nos ayudaran a notificarnos que está ocurriendo en la simulación, en este caso que sensores se han activado, si el usuario ha decidido desechar la caja y si está parado el circuito.



Figura 5.13: Panel de control planta 3

Capítulo 6

Problemáticas

En este capítulo comentaremos los inconvenientes y problemas que surgieron a la hora de desarrollar este proyecto. A continuación, se indican los principales problemas que tuve que abordar, alguno ya solucionados y otros que por motivos de tiempo o por la incapacidad de la tecnología actual no pudimos llevar a cabo.

- Generar el movimiento del Robot mediante animaciones, al fin y al cabo, estas animaciones son secuencias de fotogramas cortos. Estos me generaban un problema de bucle en donde se repetía constantemente. Se resolvía modificando una propiedad en las animaciones denominada Loop Time desmarcándola. Esta propiedad lo que hace es que mientras no finalice el tiempo de la animación, aunque visualmente ya haya terminado por la velocidad de esta, se siga repitiendo.
- Tomar la caja de la cinta transportadora, me genero 2 inconvenientes:
 - Uno de ellos era que se realizaba la acción del robot sin tomar la caja, para esto tuvimos que indicar que la caja se encontraba en la capa de g4a MU y crear un elemento vacío de destino.
 - El otro es que cuando la tomaba, la colocaba en una posición del robot extraña, es decir paralela al inicio del gancho, se solucionó modificando en las características de posición (X, Y, Z) del gancho del robot para que se posicionase visualmente bien.
- Enlazar los scripts en C# en objetos vacíos en las escenas me generaba un problema de incompatibilidad, se resolvía indicando tanto el mismo nombre del objeto, como el del script y el de la clase dentro del script.
- En el uso de TwinCat 3 para el desarrollo de la interfaz de OPC-UA, me generó errores de certificados de aplicación cuando trataba de conectar mi proyecto en TwinCat con el servidor OPC-UA que había creado.
- La idea principal era poder convertir estas simulaciones en objetos web para añadirlas a mi página con WebGL de Unity, no obstante, la tecnología de Game4Automation no deja compilar un proyecto para WebGL. Tomando la decisión de crearlo como archivo ejecutable para descargar.

Capítulo 7

Conclusiones y líneas futuras

En conclusión, en este proyecto se ha desarrollado un prototipo de enseñanza de modelos virtuales de procesos industriales y que cumple con la mayoría de los objetivos propuestos que se plantearon inicialmente. A continuación, indicaremos de nuevo estos objetivos y lo que se hizo para completarlos:

- La creación de las plantas con diferentes niveles de dificultad desarrolladas en la plataforma de Unity con Game4Automation y que se pueda actuar sobre ella mediante un panel de control, haciendo un diseño interactivo para el usuario y que pueda validar el funcionamiento.
- Diseñar la pagina web de toma de contacto, realizada y demostrada en capítulos anteriores haciendo uso de HTML5, CSS y PHP, para después conectarla mediante un servicio de autenticación con la base de datos creada en MySQL.
- Creación de las PLCs que generan el controlador lógico de la simulación. Principalmente la idea era externalizarlo remotamente a una herramienta propia de programación de controladores lógicos como TwinCat 3 mediante una interfaz de OPC-UA en Unity, no obstante, se realizó al final la programación de estas PLCs en el mismo entorno de Unity haciendo uso de las interfaces de los elementos y programando en C# los scripts que iban a tomar estas señales.

La programación de los scripts se tuvo en cuenta con una estructura de módulos siendo cada autómeta en la simulación una PLC del propio comportamiento y que se tiene que coordinar con los otros módulos de la simulación.

- Validación del sistema realizando testeos de las plantas y facilitando a otros usuarios externos las aplicaciones de escritorio para probarlas y dar feedback.

Por otro lado, mi experiencia sobre el desarrollo del proyecto ha sido positiva, dado que ha mejorado mucho mis habilidades como programador y he aprendido a manejar distintas tecnologías y herramientas que antes no conocía. Además, me resultó interesante trabajar con Unity ya que siempre había querido realizar algún diseño dentro

de esta plataforma y gracias a este proyecto me he dado cuenta de que puedo explotar esta herramienta para nuevos proyectos personales. En general, estoy contento con lo trabajado ya que, aunque hayan salido varios problemas en la resolución de este trabajo he podido abordarlos en su mayoría.

Como líneas de trabajo futuro, se pueden añadir características y funcionalidades al proyecto. Algunas posibles mejoras:

- Se puede tratar de realizar una interfaz como la de OPC-UA y externalizar las PLCs a proyectos de TwinCat 3.
- Modificar el diseño web para que se pueda alojar en una página de la universidad como medio de enseñanza para asignaturas de robótica o industriales.
- Poder transformar lo realizado en Unity con Game4Automation en diseño web, para que sea más intuitivo, funcional y se pueda acceder a las plantas sin tener que descargarlas.
- Si se completase el poder operar sobre las plantas en la página web se podrían captar valores de trabajo o guardar en la base de datos los cambios en una plataforma.
- También cuando una simulación este completa estaría bien verificarla en un entorno real de trabajo.

Capítulo 8

Summary and Conclusions

In conclusion, in this project a teaching prototype of virtual models of industrial processes has been developed and that meets most of the proposed objectives that were initially raised. Here, we will state these objectives again and what was done to complete them:

- The creation of plants with different levels of difficulty developed in the Unity platform with Game4Automation and that can be acted upon through a control panel, making an interactive design for the user and that can validate the operation.
- Design the contact web page, made and demonstrated in previous chapters using HTML5, CSS and PHP, and then connect it through an authentication service with the database created in MySQL.
- Creation of the PLCs that generate the logic controller of the simulation. Mainly the idea was to remotely externalize it to a logic controller programming tool such as TwinCat 3 through an OPC-UA interface in Unity, however, in the end the programming of these PLCs was carried out in the same Unity environment making use of the interfaces of the elements and programming in C # the scripts that were to take these signals.

The programming of the scripts was taken into account with a structure of modules, each automaton in the simulation being a PLC with its own behavior and that has to be coordinated with the other modules of the simulation.

- System validation by testing the plants and providing other external users with desktop applications to test them and give feedback.

On the other hand, my experience on the development of the project has been positive, since it has greatly improved my skills as a programmer and I have learned to handle different technologies and tools that I did not know before. In addition, I found it interesting to work with Unity since I had always wanted to make some design within this platform and thanks to this project I have realized that I can exploit this tool for new personal projects. In general, I am happy with what has been done since, although there have been several problems in the resolution of this work, I have been able to address

them for the most part.

As future lines of work, features and functionalities can be added to the project. Some possible improvements:

- You can try to create an interface like OPC-UA and outsource the PLCs to TwinCat 3 projects.
- Modify the web design so that it can be hosted on a university page as a teaching tool for robotics or industrial subjects.
- To be able to transform what has been done in Unity with Game4Automation into web design, so that it is more intuitive, functional and the plants can be accessed without having to download them.
- If the power to operate on the plants were completed on the website, work values could be captured or changes on a platform could be saved in the database.
- Also when a simulation is complete it would be good to verify it in a real work environment.

Capítulo 9

Presupuesto

En este capítulo se realiza una estimación del presupuesto aproximado por el trabajo realizado para el desarrollo de este proyecto. En la siguiente tabla se refleja un desglose del presupuesto en relación con las horas y las tareas realizadas.

Tareas Realizadas	Horas	Precio/hora	Total
Estudio bibliográfico	50h	15€	750€
Análisis de requisitos y planificación	20h	20€	400€
Selección de tecnologías	6h	20€	120€
Instalación y configuración de las tecnologías seleccionadas	30h	20€	600€
Implementación de la BBDD	5h	25€	125€
Diseño de la página web	20h	10€	200€
Comunicación de la página web con la base de datos	8h	15€	120€
Desarrollo de las escenas en Unity y configuración de los componentes	60h	40€	2400€
Desarrollo de las PLCs en C#	100h	40€	4000€
Desarrollo de un panel de control	4h	5€	20€
Documentación	38h	5€	190€
Testeos	20h	15€	300€
Total	361	-	9225€

Tabla 9.1: Presupuesto del proyecto

Bibliografía

1. Simulaciones industriales. Disponible en: <https://www.vld-eng.com/blog/simulacion-procesos-industriales/#:~:text=La%20simulaci%C3%B3n%20de%20procesos%20industriales%20es%20una%20herramienta,alto%20coste%20de%20los%20experimentos%20a%20escala%20real>
2. Miriam E. Álvarez – Ricardo M. García. Simulaciones Industriales. Disponible en: <https://repositorio.utn.edu.ec/bitstream/123456789/1092/4/04%20ISC%20065%20Tesis.pdf>
3. Game4Automation. Disponible en: <https://game4automation.com/en/>
4. Game4Automation Documentación. Disponible en: <https://game4automation.com/documentation/current/index.html>
5. Unity Technologies. Disponible en: <https://learn.unity.com/tutorial/how-to-publish-for-webgl#5d925085edbc2a0c00bd4608>
6. PHP. Disponible en: <https://www.php.net/>
7. PLC. Disponible en: https://es.wikipedia.org/wiki/Controlador_l%C3%B3gico_programable
8. CSS. Disponible en: <https://www.w3schools.com/Css/>
9. Plantillas Bootstrap. Disponible en: <https://startbootstrap.com/>

10. In2sight Gmbh. Disponible en:
<https://www.in2sight.eu/>
11. Flexim. Disponible en:
<https://www.flexsim.com/es/>
12. Wampserver. Disponible en:
<https://www.wampserver.com/>
13. Beckhoff. Manual TwinCat3. Disponible en:
https://download.beckhoff.com/download/Document/Catalog/TwinCAT_3_Booklet.pdf
14. Beckhoff. TwinCat3 Documentación. Disponible en:
https://infosys.beckhoff.com/english.php?content=../content/1033/tc3_installation/179473291.html&id=
15. Stackoverflow. ¿Qué es un nodo en OPC-UA? Disponible en:
<https://stackoverflow.com/questions/55050908/what-is-a-node-in-opc-ua>
16. Satoshi. ¿Qué es OPC-UA? Disponible en:
<https://www.opiron.com/que-es-opc-ua/>
17. Wikipedia OPC-UA. Disponible en:
https://en.wikipedia.org/wiki/OPC_Unified_Architecture
18. Free Software Foundation, Inc. PHPMailer. Disponible en:
<https://github.com/PHPMailer/PHPMailer>
19. NVIDIA y BMW Group. Disponible en:
<https://www.nvidia.com/es-es/autonomous-machines/embedded-systems/car-manufacturing-robotics/>
20. NVIDIA y BMW Group. Comunicado de prensa. Disponible en:
<https://nvidianews.nvidia.com/news/bmw-group-selects-nvidia-to-redefine-factory-logistics>

21. Siemens Tia Portal. Disponible en:
<https://new.siemens.com/global/en/products/automation/industry-software/automation-software/tia-portal.html>
22. Factory I/O. Disponible en:
<https://factoryio.com/>
23. NVIDIA. Isaac Sim documentación. Disponible en:
https://docs.omniverse.nvidia.com/app_isaacsim/app_isaacsim/overview.html
24. TwoReality. Disponible en:
<https://www.tworeality.com/realidad-virtual-aumentada-industria/>
25. Virtual Plant. Disponible en:
<https://www.virtualplant.co/>
26. Game4Automation. OPC-UA configuración. Disponible en:
<https://game4automation.com/documentation/current/opcu.html>