



Universidad de Valladolid



**ESCUELA DE INGENIERÍAS
INDUSTRIALES**

UNIVERSIDAD DE VALLADOLID

ESCUELA DE INGENIERIAS INDUSTRIALES

Grado en Ingeniería Electrónica Industrial y Automática

**Estación de paletizado en RobotStudio
(ABB) con HMI en TIA Portal, controlable
por voz y visión artificial desde Python.**

Autor:

Robles Postigo, Adrián

Tutor:

**Herreros López, Alberto
Departamento de Ingeniería de
Sistemas y Automática**

Valladolid, septiembre de 2024.



Agradecimientos

En este momento de culminación de mi carrera universitaria, quisiera expresar mi más profundo agradecimiento a todas las personas que han hecho posible tanto mi crecimiento académico como personal durante estos cuatro años.

En primer lugar, quiero dar las gracias a mi tutor, Alberto Herreros López, cuya dirección y confianza han sido fundamentales para la realización de este trabajo. Su conocimiento y atención me han ayudado enormemente a lo largo de este proceso de aprendizaje.

Agradezco también a los profesores del Departamento de Ingeniería de Sistemas y Automática, por brindarme una educación de calidad. Sus enseñanzas han dejado una huella importante en mi formación académica.

Mis compañeros de carrera también merecen una mención especial. Gracias por todos los buenos momentos, las discusiones enriquecedoras y el apoyo mutuo durante los desafíos que hemos enfrentado juntos.

No puedo dejar de mencionar el apoyo incondicional de mi familia. Su confianza y apoyo constante han sido fundamentales para mí. Gracias por creer siempre en mí y por brindarme motivación en todo momento. Su respaldo emocional ha sido esencial para alcanzar mis metas.

Finalmente, quiero agradecer a todos aquellos que, de una manera u otra, contribuyeron a este proyecto. Cada uno de ustedes habéis sido cruciales en este viaje y estoy profundamente agradecido por cada aporte.





Resumen

Este Trabajo de Fin de Grado aborda el diseño y desarrollo de una estación robotizada para paletizado y clasificación, utilizando RobotStudio de ABB como plataforma central. El proyecto se enriquece con una Interfaz Hombre-Máquina (HMI) desarrollada en TIA Portal, diseñada para facilitar la interacción del usuario mediante comandos de voz, lo que mejora la operatividad y eficiencia del sistema.

Adicionalmente, se implementa un sistema de visión artificial, programado en Python, destinado a identificar y clasificar cajas defectuosas, elevando la precisión del proceso de clasificación.

La comunicación entre los diferentes softwares del sistema, incluyendo RobotStudio, TIA Portal y el sistema de reconocimiento de voz en Python, se maneja utilizando protocolo TCP/IP, mientras que la conexión entre el sistema de visión en Python y RobotStudio se establece mediante sockets. Esta configuración de comunicación mejora la coordinación de los componentes, proporcionando un flujo de datos constante y estable, crucial para la eficiencia operativa de la estación.

Este proyecto no solo refleja la aplicación práctica de los conocimientos técnicos adquiridos durante el grado, sino que también establece un punto de partida para futuras investigaciones y desarrollos en el campo de la Industria 4.0.

Palabras clave

RobotStudio, TIA Portal, Visión Artificial, Reconocimiento de Voz, Comunicaciones Industriales.





Abstract

This Final Degree Project addresses the design and development of a robotized station for palletizing and sorting, using ABB's RobotStudio as the central platform. The project is enhanced with a Human-Machine Interface (HMI) developed in TIA Portal, designed to facilitate user interaction through voice commands, which improves system operability and efficiency.

Additionally, an artificial vision system programmed in Python is implemented to identify and classify defective boxes, increasing the accuracy of the sorting process.

Communication between the different software components of the system, including RobotStudio, TIA Portal, and the voice recognition system in Python, is handled using the TCP/IP protocol, while the connection between the vision system in Python and RobotStudio is established via sockets. This communication setup enhances component coordination, providing a constant and stable data flow, crucial for the operational efficiency of the station.

This project not only demonstrates the practical application of the technical knowledge acquired during the degree but also lays a foundation for future research and developments in the field of Industry 4.0.

Keywords

RobotStudio, TIA Portal, Artificial Vision, Voice Recognition, Industrial Communications.





ÍNDICE DE CONTENIDO

CAPÍTULO 1. Introducción	1
Capítulo 1.1. Motivación y justificación del proyecto.....	1
Capítulo 1.2. Objetivos.....	2
Capítulo 1.3. Fundamentos previos.....	3
Capítulo 1.4. Alcance del proyecto.....	4
Capítulo 1.5. Estructura del proyecto	5
Capítulo 1.6. Convenciones.....	6
CAPÍTULO 2. Marco teórico	7
Capítulo 2.1. Robótica.....	7
Capítulo 2.1.1. Introducción a la robótica.....	7
Capítulo 2.1.2. Historia de la robótica.....	8
Capítulo 2.1.3. Robótica industrial	12
Capítulo 2.1.4. Clasificación de robots industriales.....	16
Capítulo 2.2. Automática	21
Capítulo 2.2.1. Introducción a la automática.....	21
Capítulo 2.2.2. Historia de la automática	21
Capítulo 2.2.3. Automatización industrial	25
Capítulo 2.2.4. Clasificación de autómatas industriales	27
Capítulo 2.2.5. Controlador Lógico Programable (PLC).....	29
Capítulo 2.2.6. Interfaz Hombre - Máquina (HMI)	33
Capítulo 2.3. Visión artificial.....	34
Capítulo 2.3.1. Introducción a la visión artificial	34
Capítulo 2.3.2. Historia de la visión artificial	35
Capítulo 2.3.3. Visión artificial industrial	37
Capítulo 2.3.4. Clasificación de sistemas de visión artificial industriales	38
Capítulo 2.4. Reconocimiento de voz	41
Capítulo 2.4.1. Introducción al reconocimiento de voz.....	41
Capítulo 2.4.2. Historia del reconocimiento de voz.....	42
Capítulo 2.4.3. Clasificación de sistemas de reconocimiento de voz.....	44



Capítulo 2.5. Comunicaciones Industriales	46
Capítulo 2.5.1. Introducción a las comunicaciones industriales.....	46
Capítulo 2.5.2. Redes de comunicación industrial.....	46
Capítulo 2.5.3. Pirámide CIM	47
Capítulo 2.5.4. Topologías de red industrial	49
Capítulo 2.5.5. Protocolos de comunicación industrial.....	51
CAPÍTULO 3. Estado del arte.....	55
Capítulo 3.1. Antecedentes	55
Capítulo 3.2. Crítica al estado del arte	57
Capítulo 3.3. Solución implementada.....	58
CAPÍTULO 4. Herramientas software empleadas	63
Capítulo 4.1. Software de ABB	64
Capítulo 4.1.1. RobotStudio	65
Capítulo 4.2. Siemens.....	74
Capítulo 4.2.1. TIA Portal.....	74
Capítulo 4.2.2. PLCSIM.....	85
Capítulo 4.2.3. WinCC.....	89
Capítulo 4.3. Python	90
Capítulo 4.3.1. Spyder	91
Capítulo 4.4. Comunicaciones.....	93
Capítulo 4.4.1. NetToPLCSim.....	93
CAPÍTULO 5. Diseño y desarrollo de la solución adoptada	97
Capítulo 5.1. Desarrollo en RobotStudio	97
Capítulo 5.1.1. Elementos y composición de la estación	97
Capítulo 5.1.2. Componentes inteligentes (Smart Components).....	108
Capítulo 5.1.3. RAPID	123
Capítulo 5.2. Desarrollo en TIA Portal	135
Capítulo 5.2.1. Controlador lógico programable (PLC).....	135
Capítulo 5.2.2. Interfaz Hombre – Máquina (HMI)	140
Capítulo 5.2.3. Conexión entre el PLC y la HMI	157
Capítulo 5.3. Desarrollo en Python	158
Capítulo 5.3.1. Sistema de reconocimiento de voz	158



Capítulo 5.3.2. Sistema de visión artificial.....	161
CAPÍTULO 6. Pruebas y resultados	165
CAPÍTULO 7. Estudio económico.....	183
Capítulo 7.1. Gastos del proyecto	183
Capítulo 7.1.1. Costes de desarrollo	184
Capítulo 7.1.2. Costes de implementación	186
Capítulo 7.1.3. Costes operativos.....	187
Capítulo 7.2. Beneficios del proyecto	188
Capítulo 7.3. Rentabilidad y análisis económico del proyecto.....	189
CAPÍTULO 8. Conclusiones y líneas futuras.....	191
Capítulo 8.1. Conclusiones	191
Capítulo 8.2. Propuestas para futuras investigaciones y desarrollos	193
Bibliografía	197



ÍNDICE DE FIGURAS

FIGURA 1. Reloj de agua de Ctsebio de Alejandría. (Brunete, San Segundo, & Herrero, 2024).....	8
FIGURA 2. Teatro automático de Herón de Alejandría. (Fraile Marinero, 2023).....	8
FIGURA 3. Fuente del Pavo Real de Al-Jazari. (Brunete, San Segundo, & Herrero, 2024).....	9
FIGURA 4. Joseph Engelberger con uno de los primero Unimate. (Barrientos, Peñín, Balaguer, & Aracil, 2007).....	10
FIGURA 5. Robot PUMA. (Fraile Marinero, 2023).....	10
FIGURA 6. Robot Irb6 de ASEA. (Barrientos, Peñín, Balaguer, & Aracil, 2007).....	11
FIGURA 7. TCP de un robot. (Fraile Marinero, 2023).....	13
FIGURA 8. Volumen de trabajo de un robot ABB IRB 120. (Fraile Marinero, 2023).....	13
FIGURA 9. Capacidad de carga de un robot IRB 140. (Fraile Marinero, 2023).....	14
FIGURA 10. Robot de accionamiento eléctrico. (Cortesía de Mitsubishi Electric).....	17
FIGURA 11. Robots de accionamiento hidráulico. (Barrientos, Peñín, Balaguer, & Aracil, 2007).....	17
FIGURA 12. Robot de accionamiento neumático. (Cortesía de FESTO).....	18
FIGURA 13. Estructura y campo de trabajo de robot SCARA. (Fraile Marinero, 2023).....	18
FIGURA 14. Robot angular o antropomórfico. (Barrientos, Peñín, Balaguer, & Aracil, 2007).....	18
FIGURA 15. Coordenadas, estructura y campo de trabajo de robot cartesiano. (Fraile Marinero, 2023).....	19
FIGURA 16. Coordenadas, estructura y campo de trabajo de robot cilíndrico. (Fraile Marinero, 2023).....	19
FIGURA 17. Coordenadas, estructura y campo de trabajo de robot esférico o polar. (Fraile Marinero, 2023).....	19
FIGURA 18. Molino de agua de Filón de Bizancio. (Brunete, San Segundo, & Herrero, 2024).....	22
FIGURA 19. Gallo de Estrasburgo. (Brunete, San Segundo, & Herrero, 2024).....	22
FIGURA 20. Pato de Vaucanson. (Brunete, San Segundo, & Herrero, 2024).....	23
FIGURA 21. Telar de Jacquard. (Brunete, San Segundo, & Herrero, 2024).....	23
FIGURA 22. Regulador de Watt. (Brunete, San Segundo, & Herrero, 2024).....	24
FIGURA 23. Reconstrucción de la máquina analítica de Babbage. (Brunete, San Segundo, & Herrero, 2024).....	24
FIGURA 24. Pirámide de la automatización industrial. (Brunete, San Segundo, & Herrero, 2024).....	26
FIGURA 25. Transición de tecnología cableada a programable. (Brunete, San Segundo, & Herrero, 2024).....	27
FIGURA 26. Arquitectura de un Controlador Lógico Programable. (Sanchis Llopis, Romero Pérez, & Ariño Latorre, 2010).....	30
FIGURA 27. Autómata programable modular. (Sanchis Llopis, Romero Pérez, & Ariño Latorre, 2010).....	30
FIGURA 28. Autómata programable compacto. (Sanchis Llopis, Romero Pérez, & Ariño Latorre, 2010).....	31
FIGURA 29. Ejemplo de programa de diagrama de contactos Ladder. (Sanchis Llopis, Romero Pérez, & Ariño Latorre, 2010).....	31
FIGURA 30. Ciclo de trabajo de un PLC. (Sanchis Llopis, Romero Pérez, & Ariño Latorre, 2010).....	32
FIGURA 31. Interfaz Hombre – Máquina HMI Siemens Simatic. (Cortesía de Siemens).....	33
FIGURA 32. Experimento de reconocimiento de formas con un gato. (IBM, s.f.).....	35
FIGURA 33. Alexa. (Cortesía de Amazon).....	43
FIGURA 34. Pirámide CIM. (Diseñado por el autor).....	48
FIGURA 35. Topología punto a punto. (Sicma21, 2021).....	49
FIGURA 36. Topología en bus. (Sicma21, 2021).....	49
FIGURA 37. Topología en estrella. (Sicma21, 2021).....	50
FIGURA 38. Topología en árbol. (Sicma21, 2021).....	50
FIGURA 39. Topología en anillo. (Diseñado por el autor).....	51
FIGURA 40. Estación robotizada del TFG de Álvaro Galindo de Santos. (Santos, 2016).....	55
FIGURA 41. Célula de trabajo correspondiente al TFG de Rodrigo Sancho García. (García R. S., 2021).....	56
FIGURA 42. Esquema de comunicaciones entre sistemas del proyecto.	61
FIGURA 43. Herramientas software utilizadas para cada sistema del proyecto.	63
FIGURA 44. Interfaz al iniciar RobotStudio 2023.	66
FIGURA 45. Entorno de trabajo en RobotStudio 2023.	66
FIGURA 46. Pestaña Posición inicial en RobotStudio 2023.	67
FIGURA 47. Pestaña Posición inicial en RobotStudio 2023.	67
FIGURA 48. Pestaña Simulación en RobotStudio 2023.	67
FIGURA 49. Pestaña Controlador en RobotStudio 2023.	67
FIGURA 50. Pestaña RAPID en RobotStudio 2023.	68
FIGURA 51. Pestaña Complementos en RobotStudio 2023.	68
FIGURA 52. Añadir robot a la estación en RobotStudio 2023.	69
FIGURA 53. Selección de la versión del robot en RobotStudio 2023.	69
FIGURA 54. Añadir controlador virtual desde diseño en RobotStudio 2023.	70
FIGURA 55. Selección de robots y versión de RobotWare para el controlador virtual en RobotStudio 2023.	70
FIGURA 56. Opciones del controlador en RobotStudio 2023.	71
FIGURA 57. Opciones del controlador virtual en RobotStudio 2023.	71
FIGURA 58. Interfaz de RAPID dentro de RobotStudio 2023.	72
FIGURA 59. Vista del portal en TIA Portal V16.	75



FIGURA 60. Agregar nuevo dispositivo al proyecto en TIA Portal V16.	76
FIGURA 61. Vista del proyecto en TIA Portal V16.	77
FIGURA 62. Barra de herramientas en TIA Portal V16.	77
FIGURA 63. Árbol del proyecto en TIA Portal V16.	78
FIGURA 64. Vista detallada en TIA Portal V16.	78
FIGURA 65. Área de trabajo en TIA Portal V16.	79
FIGURA 66. Herramienta de configuración en TIA Portal V16.	79
FIGURA 67. Vista de propiedades en TIA Portal V16.	80
FIGURA 68. Vista de información en TIA Portal V16.	80
FIGURA 69. Vista de diagnóstico en TIA Portal V16.	80
FIGURA 70. Bloques de programa en TIA Portal V16.	82
FIGURA 71. Iniciar simulación en TIA Portal V16.	85
FIGURA 72. Ventana de búsqueda del PLC virtual en TIA Portal V16.	86
FIGURA 73. Vista compacta de PLCSIM.	86
FIGURA 74. Vista preliminar de carga en PLCSIM y vista de compilación satisfactoria.	87
FIGURA 75. Resultado de la operación de carga en el PLC virtual.	87
FIGURA 76. Vista compacta PLCSIM con PLC virtual en modo RUN.	88
FIGURA 77. Vista del proyecto en PLCSIM.	88
FIGURA 78. Visualización pantalla HMI con WinCC.	89
FIGURA 79. Interfaz de usuario en Spyder (Python 3.11).	91
FIGURA 80. Esquema de conexión de NetToPLCSim. (InfoPLC, 2021)	93
FIGURA 81. Interfaz NetToPLCSim.	94
FIGURA 82. Datos de la estación para la comunicación con NetToPLCSim.	95
FIGURA 83. Servidor en ejecución desde NetToPLCSim.	96
FIGURA 84. Monitoreo del servidor en NetToPLCSim.	96
FIGURA 85. Robot IRB 4600-60/2.55. (Robotics, ABB, 2019).	97
FIGURA 86. Área de trabajo robot IRB 4600-60/2.55. (Robotics, ABB, 2019)	98
FIGURA 87. Herramienta "Vacuum" usada en la estación.	99
FIGURA 88. Pedestal sobre el que montar los robots de la estación.	100
FIGURA 89. Cinta transportadora de cajas usada en la estación.	100
FIGURA 90. Cinta transportadora de palés usada en la estación.	101
FIGURA 91. Valla de seguridad usada en la estación.	101
FIGURA 92. Dispensador de palés usado en la estación.	102
FIGURA 93. Dispensador de planchas usado en la estación. (Diseñado por el autor del proyecto)	102
FIGURA 94. Procedimiento para crear un dato de herramienta en RobotStudio 2023.	103
FIGURA 95. Datos de herramienta en RobotStudio 2023.	104
FIGURA 96. Procedimiento para crear un objeto de trabajo en RobotStudio 2023.	105
FIGURA 97. Parámetros para crear un objeto de trabajo en RobotStudio 2023.	105
FIGURA 98. Objetos y herramientas de trabajo de la estación en RobotStudio 2023.	106
FIGURA 99. Procedimiento para sincronizar el Controlador con RAPID en RobotStudio 2023.	107
FIGURA 100. Sincronizar Controlador con RAPID en RobotStudio 2023.	107
FIGURA 101. Acceso a lógica de estación en RobotStudio 2023.	109
FIGURA 102. Componente inteligente "SimulationEvents" en RobotStudio 2023.	110
FIGURA 103. "SC_Herramienta" en RobotStudio 2023. (Diseñado por el autor del proyecto)	111
FIGURA 104. "SC_Cinta" en RobotStudio 2023. (Diseñado por el autor del proyecto)	113
FIGURA 105. "SC_Cinta_Pale" en RobotStudio 2023. (Diseñado por el autor del proyecto)	115
FIGURA 106. "SC_Dispensador_Pale" en RobotStudio 2023. (Diseñado por el autor del proyecto)	116
FIGURA 107. "SC_Plancha" en RobotStudio 2023. (Diseñado por el autor del proyecto)	117
FIGURA 108. "SC_Camara" en RobotStudio 2023. (Diseñado por el autor del proyecto)	118
FIGURA 109. "SC_Inicio_Comunicación" en RobotStudio 2023. (Diseñado por el autor del proyecto)	120
FIGURA 110. "SC_Señales_Digitales" en RobotStudio 2023. (Diseñado por el autor del proyecto)	120
FIGURA 111. "SC_Señales_Analogicas" en RobotStudio 2023. (Diseñado por el autor del proyecto)	122
FIGURA 112. Lógica de estación del proyecto en RobotStudio 2023.	122
FIGURA 113. Símbolos normalizados para los diagramas de flujo. (Sanz, s.f.)	125
FIGURA 114. Diagrama de flujo para el proceso de paletizado.	126
FIGURA 115. Diagrama de flujo para el proceso de clasificado.	129
FIGURA 116. Organización de bloques de programa del proyecto en TIA Portal V16.	135
FIGURA 117. Organización de las variables del proyecto en TIA Portal V16.	135
FIGURA 118. Organización del bloque OB1 del proyecto en TIA Portal V16.	136
FIGURA 119. Llamada a la función FC_Recuento (FC1) desde OB1 en TIA Portal V16.	136
FIGURA 120. Organización de la función FC1 en el proyecto en TIA Portal V16.	137
FIGURA 121. Contador ascendente de palés 1 del proyecto en TIA Portal V16.	137
FIGURA 122. Organización de la FB3 del proyecto en TIA Portal V16.	138
FIGURA 123. Programación FB4 del proyecto en TIA Portal V16.	139

FIGURA 124. Programación FB5 del proyecto en TIA Portal V16.....	139
FIGURA 125. Esquema de las pantallas de la HMI del proyecto.....	140
FIGURA 126. Imagen “Ventana Principal” de la HMI implementada.....	142
FIGURA 127. Imagen “Producción” de la HMI implementada.....	143
FIGURA 128. Imagen “Estado Línea” de la HMI implementada.....	145
FIGURA 129. Imagen “Programación Manual” de la HMI implementada.....	145
FIGURA 130. Imagen “Programación clasificador manual” de la HMI implementada.....	146
FIGURA 131. Imagen “Programación paletizador manual” de la HMI implementada.....	147
FIGURA 132. Imagen “Programación por voz” de la HMI implementada.....	148
FIGURA 133. Imagen “Programación clasificador por voz” de la HMI implementada.....	150
FIGURA 134. Imagen “Programación paletizador por voz” de la HMI implementada.....	151
FIGURA 135. Programación de las propiedades de un botón en TIA Portal V16.....	152
FIGURA 136. Configuración de propiedades para cambio de gráfico en TIA Portal V16.....	153
FIGURA 137. Lista de gráficos del proyecto en TIA Portal V16.....	153
FIGURA 138. Lista de textos del proyecto en TIA Portal V16.....	154
FIGURA 139. Programación de animaciones en TIA Portal V16.....	155
FIGURA 140. Acciones para configurar eventos en TIA Portal V16.....	156
FIGURA 141. Ejemplo programación de eventos en “Botón Marcha” del proyecto en TIA Portal V16.....	156
FIGURA 142. Realización de conexión PROFINET entre el PLC y la HMI del proyecto en TIA Portal V16.....	157
FIGURA 143. Inicio de la simulación en RobotStudio 2023.....	166
FIGURA 144. Inicio de la simulación en TIA Portal V16.....	166
FIGURA 145. Servidor configurado y en funcionamiento en NetToPLCSim.....	167
FIGURA 146. Datos en el terminal al inicio de la simulación del sistema de reconocimiento por voz en Python.....	167
FIGURA 147. Datos en el terminal al inicio de la simulación del sistema de visión en Python.....	167
FIGURA 148. Datos en el terminal al establecer conexión con el servidor de RobotStudio en Python.....	168
FIGURA 149. Programación del paletizado manualmente desde la HMI en TIA Portal V16.....	168
FIGURA 150. Programación del paletizado por voz desde la HMI en TIA Portal V16.....	169
FIGURA 151. Cintas transportadoras de cajas en marcha desde la HMI en TIA Portal V16.....	169
FIGURA 152. Cintas transportadoras de cajas activas en RobotStudio 2023.....	170
FIGURA 153. Robots y cintas transportadoras de cajas activas en RobotStudio 2023.....	170
FIGURA 154. Paletizado según la configuración realizada en la figura 149 en RobotStudio 2023.....	171
FIGURA 155. Palé completado en RobotStudio 2023.....	171
FIGURA 156. Cintas transportadoras de palés activas en HMI en TIA Portal V16.....	172
FIGURA 157. Captura a cajas de vainilla desde RobotStudio 2023.....	172
FIGURA 158. Captura a cajas choco desde RobotStudio 2023.....	173
FIGURA 159. Visualización de la caja de referencia y la caja nueva perfecta de vainilla en Python.....	173
FIGURA 160. Diferencias entre la caja de referencia y la caja nueva de vainilla sin diferencias en Python.....	173
FIGURA 161. Visualización de la caja de referencia y la caja nueva perfecta de choco en Python.....	174
FIGURA 162. Diferencias entre la caja de referencia y la caja nueva perfecta de choco sin diferencias en Python.....	174
FIGURA 163. Visualización de la caja de referencia y la caja nueva defectuosa de vainilla en Python.....	174
FIGURA 164. Diferencias entre la caja de referencia y la caja nueva defectuosa de vainilla con diferencias en Python.....	174
FIGURA 165. Datos en el terminal al comparar imágenes de vainilla defectuosas en Python.....	174
FIGURA 166. Visualización de la caja de referencia y la caja nueva defectuosa de choco en Python.....	175
FIGURA 167. Diferencias entre la caja de referencia y la caja nueva defectuosa de choco con diferencias en Python.....	175
FIGURA 168. Datos en el terminal al comparar imágenes de choco defectuosas en Python.....	175
FIGURA 169. Clasificado de caja defectuosa de vainilla según el porcentaje configurado en RobotStudio 2023.....	176
FIGURA 170. Colocación en palé de caja defectuosa en RobotStudio 2023.....	176
FIGURA 171. Clasificado de caja defectuosa de choco según el porcentaje configurado en RobotStudio 2023.....	177
FIGURA 172. Cajas defectuosas paletizadas en RobotStudio 2023.....	177
FIGURA 173. Recuento de la simulación en TIA Portal V16.....	178
FIGURA 174. Seta de emergencia activada en TIA Portal V16.....	178
FIGURA 175. Datos en el terminal al pronunciar “stop” por parte del usuario en Python.....	179
FIGURA 176. Paro de emergencia en RobotStudio 2023.....	179
FIGURA 177. Reactivación movimiento después de parada de emergencia en TIA Portal V16.....	180
FIGURA 178. Reactivación movimiento después de parada de emergencia en RobotStudio 2023.....	180





ÍNDICE DE TABLAS

<i>Tabla 1. Elementos principales del lenguaje KOP.....</i>	<i>83</i>
<i>Tabla 2. Costes de software del proyecto.....</i>	<i>184</i>
<i>Tabla 3. Costes de hardware del proyecto.</i>	<i>185</i>
<i>Tabla 4. Costes de mano de obra del proyecto.....</i>	<i>185</i>
<i>Tabla 5. Costes de implementación del proyecto.</i>	<i>186</i>
<i>Tabla 6. Costes operativos anuales del proyecto.....</i>	<i>187</i>
<i>Tabla 7. Beneficios anuales del proyecto.....</i>	<i>188</i>





CAPÍTULO 1. Introducción

Capítulo 1.1. Motivación y justificación del proyecto

Mi interés por la robótica y la automatización industrial se ha ido forjando a lo largo de mi carrera en el grado de Ingeniería Electrónica Industrial y Automática. A medida que avanzaba en mis estudios, cada curso relacionado con estas disciplinas no sólo reforzaba mi pasión, sino que también ampliaba mi visión sobre su potencial transformador en el mundo actual. La decisión de enfocar mi Trabajo de Fin de Grado en esta área fue natural, impulsada por el deseo de profundizar en una tecnología que está redefiniendo los límites de la ingeniería.

El proyecto que he elegido, dedicado al diseño y desarrollo de una estación de paletizado y clasificación, encapsula este interés. Se propone integrar una Interfaz Hombre-Máquina (HMI) en TIA Portal, permitiendo que los usuarios interactúen de manera intuitiva con el sistema a través de comandos de voz. Este enfoque no sólo aumenta la accesibilidad y eficiencia del sistema, sino que también subraya la importancia de la ergonomía en el diseño industrial moderno.

Además, la implementación de un sistema de visión artificial para la detección y clasificación de defectos en las cajas usando librerías de Python ejemplifica la fusión de la robótica con la inteligencia artificial, otro pilar de mi formación académica. La comunicación entre los diferentes sistemas mediante la tarjeta de red del ordenador es un reto técnico que estoy ansioso por abordar, representando un paso crucial hacia la creación de sistemas integrados más robustos y eficientes.

Este proyecto no solo es una prueba de las habilidades y conocimientos que he adquirido durante mis estudios, sino que también pretende aportar soluciones innovadoras a problemas reales de la industria. Es una oportunidad para demostrar que los conceptos aprendidos pueden tener aplicaciones prácticas significativas, mejorando procesos y aumentando la productividad en entornos industriales.

Capítulo 1.2. Objetivos

El presente proyecto tiene como finalidad principal el diseño y desarrollo de una estación de paletizado y clasificación que utilice tecnologías avanzadas para optimizar los procesos industriales. A continuación, se detallan los objetivos específicos que se buscan alcanzar:

- **Dominio de RobotStudio de ABB:** profundizar en el uso de RobotStudio para la simulación y programación avanzada de robots mediante el lenguaje RAPID, optimizando las tareas de paletizado y clasificación en un entorno controlado.
- **Desarrollo e integración de una Interfaz Hombre-Máquina (HMI) con tecnología de voz:** diseñar una HMI en TIA Portal que incorpore funcionalidades avanzadas de control por voz, desarrolladas en Python. Este sistema permitirá a los usuarios manejar los robots de forma intuitiva y eficiente, mejorando la interactividad y accesibilidad del sistema.
- **Implementación de un sistema de visión artificial avanzado:** desarrollar un sistema de visión artificial utilizando librerías de Python para identificar y clasificar automáticamente cajas defectuosas. Este sistema estará plenamente integrado con la operativa de la estación robótica, asegurando altos estándares de calidad y eficiencia.
- **Comunicación efectiva entre sistemas:** implementar una comunicación robusta y fiable entre RobotStudio, TIA Portal y Python a través de la tarjeta de red del ordenador. Esta integración asegurará una operación coordinada y sin interrupciones entre los distintos componentes tecnológicos.
- **Simulación, validación y ajustes del sistema:** realizar una serie de simulaciones y pruebas exhaustivas para validar la funcionalidad y eficacia del sistema. Estas pruebas ayudarán a identificar y corregir posibles errores y a optimizar el rendimiento para una posible implementación en un entorno industrial real en el futuro.
- **Análisis de impacto y expansión futura:** evaluar el impacto del sistema en la productividad y la calidad del proceso industrial. Además, explorar adaptaciones y ampliaciones del sistema para diferentes aplicaciones y sectores, proponiendo mejoras continuas y nuevas funcionalidades.

Estos objetivos muestran una clara dirección hacia la integración de la robótica avanzada, la programación en Python y la interacción hombre-máquina, reflejando un compromiso con la innovación y la aplicación práctica de los conocimientos adquiridos durante la formación académica en ingeniería.

Capítulo 1.3. Fundamentos previos

El desarrollo de este proyecto de fin de grado ha requerido una base sólida en varios campos técnicos, adquiridos a través de la formación académica en Ingeniería Electrónica Industrial y Automática. Los principales pilares académicos que han respaldado este proyecto incluyen:

- **Sistemas Robotizados y Taller de Robótica:** estas asignaturas me brindaron habilidades esenciales en el manejo de RobotStudio y la programación en RAPID durante los laboratorios prácticos. A parte de las habilidades teóricas y prácticas en otros softwares que son fundamentales para la configuración y operación de los sistemas robotizados en este proyecto.
- **Mecatrónica y Electrónica Digital:** estos cursos han sido cruciales para entender y aplicar los principios de la integración de sistemas mecánicos, electrónicos y de control. La electrónica digital, en particular, ha fundamentado la lógica necesaria para la estación robótica.
- **Visión Artificial:** la capacitación en técnicas avanzadas de procesamiento de imágenes y reconocimiento de patrones me ha permitido desarrollar y optimizar un sistema de visión artificial con Python, esencial para la clasificación automatizada de cajas defectuosas.
- **Automatización Industrial, Fundamentos de la Automática y Modelado y Simulación de Sistemas:** estas asignaturas me proporcionaron un conocimiento profundo de la automatización y control de procesos, así como del modelado y simulación de operaciones complejas, aspectos clave para la integración y coordinación de la estación de trabajo.
- **Fundamentos de la Informática e Informática Industrial:** los conocimientos adquiridos en estas asignaturas han sido vitales para implementar soluciones de software.
- **Control y Comunicaciones Industriales:** mi formación en este campo ha sido indispensable para diseñar la infraestructura de comunicación necesaria para conectar de manera efectiva RobotStudio, TIA Portal y Python, asegurando una interacción fluida y coordinada entre las plataformas.

La combinación de estas disciplinas ha proporcionado una base sólida para superar los desafíos técnicos del proyecto, permitiendo la creación de una solución integrada que no solo mejora la eficiencia de los procesos de paletizado y clasificación, sino que también garantiza la adaptabilidad y escalabilidad del sistema en entornos industriales reales.



Capítulo 1.4. Alcance del proyecto

Este proyecto tiene como objetivo diseñar, desarrollar y validar una estación de paletizado y clasificación automatizada. Se comenzará con un diseño detallado utilizando RobotStudio de ABB para modelar y simular la estación, asegurando la viabilidad y eficiencia del sistema propuesto. Seguidamente, se programarán los robots con el lenguaje RAPID y se desarrollará una interfaz Hombre-Máquina en TIA Portal que incorporará control por voz mediante Python para facilitar la operación del sistema.

La integración de un sistema de visión artificial, también desarrollado en Python, permitirá la identificación y clasificación de cajas defectuosas, optimizando así el proceso de clasificación. Finalmente, el sistema será sometido a pruebas rigurosas en un entorno simulado para validar su funcionamiento y se documentará exhaustivamente cada fase del proyecto para asegurar su replicabilidad y mantenimiento.

Este enfoque integrado busca no solo aplicar conocimientos técnicos avanzados sino también mejorar los procesos de paletizado y clasificación en entornos industriales reales.

Capítulo 1.5. Estructura del proyecto

Este informe documenta exhaustivamente el proceso desde la concepción hasta la realización de una estación de paletizado y clasificación avanzada. Se detalla cada fase del proyecto con el objetivo de ofrecer una comprensión integral tanto de los fundamentos teóricos como de las aplicaciones prácticas involucradas, reflejando cómo se interconectan diversas tecnologías para crear una solución cohesiva y funcional. Para este fin, la memoria se ha estructurado en los siguientes capítulos:

- **Capítulo 1. Introducción:** se inicia con una visión general del proyecto, estableciendo la motivación y la justificación detrás del mismo. Este capítulo también define los objetivos específicos que se pretenden alcanzar y repasa los conocimientos técnicos previos que soportan el desarrollo del proyecto. Además, se delimita el alcance del proyecto y se describen las convenciones utilizadas en el documento.
- **Capítulo 2. Marco teórico:** profundiza en las bases teóricas que fundamentan el proyecto, abarcando desde la robótica y la automatización hasta la visión artificial y el reconocimiento de voz. Este capítulo también incluye un estudio sobre las comunicaciones industriales utilizadas, proporcionando un contexto técnico esencial para la comprensión de las tecnologías implementadas.
- **Capítulo 3. Estado del arte:** explora trabajos previos y desarrollos tecnológicos relacionados, ofreciendo un análisis del contexto en el que se inscribe el proyecto. Se discuten los avances recientes y cómo estos influyen o complementan la propuesta presentada.
- **Capítulo 4. Herramientas software empleadas:** detalla las herramientas de software específicas utilizadas en el proyecto, como RobotStudio, TIA Portal y Python, explicando cómo cada una contribuye al desarrollo y funcionamiento de la estación de paletizado.
- **Capítulo 5. Diseño y Desarrollo de la solución adoptada:** se describe el proceso de diseño y desarrollo de la solución, desde la programación de los robots y la configuración de la interfaz hombre-máquina hasta la integración de los sistemas de visión artificial y reconocimiento de voz, junto con la comunicación entre programas.

- **Capítulo 6. Pruebas y resultados:** este capítulo presenta los resultados obtenidos de las pruebas realizadas para validar la funcionalidad y eficacia del sistema desarrollado.
- **Capítulo 7. Estudio económico:** se realiza un análisis detallado de los costes asociados al proyecto, proporcionando una perspectiva sobre la viabilidad económica de la solución propuesta.
- **Capítulo 8. Conclusiones y líneas futuras:** finaliza con una síntesis de los hallazgos principales y el impacto del proyecto, además de sugerir direcciones futuras para la investigación y el desarrollo continuo en los diferentes campos utilizados a lo largo del proyecto.
- **Bibliografía:** el documento se completa con una sección de referencias bibliográficas que respaldan el trabajo académico.

Este enfoque estructurado no solo facilita la comprensión y la profundidad del proyecto, sino que también subraya la integración sistemática de diversas disciplinas y tecnologías para abordar un desafío de ingeniería complejo.

Capítulo 1.6. Convenciones

Para garantizar una lectura coherente y estructurada de este informe, se han establecido ciertas normativas que se aplican a lo largo de los diferentes capítulos. Estas convenciones han sido cuidadosamente seleccionadas para facilitar la comprensión y el seguimiento de los contenidos expuestos, asegurando así una experiencia uniforme y accesible para el lector.

- Se ha empleado la tipografía “Franklin Gothic Book” de tamaño 12 para el texto principal.
- Los nombres de autores se destacan en **negrita**.
- Los títulos de obras de autores se presentan en *cursiva*.
- Las palabras extranjeras no traducidas aparecen entre “comillas”.
- Los términos específicos asignados a elementos o componentes del proyecto también se enmarcan entre “comillas”.
- Justo después del índice de contenidos, se proporcionan índices separados para figuras y tablas.
- Las referencias bibliográficas se han estructurado según el estilo APA 6.



CAPÍTULO 2. Marco teórico

Capítulo 2.1. Robótica

Capítulo 2.1.1. Introducción a la robótica

La robótica es una disciplina que combina mecánica, electrónica, informática y teoría de control para el diseño y construcción de robots. Los robots son sistemas automatizados capaces de realizar tareas de manera autónoma o semiautónoma, mejorando la eficiencia, precisión y seguridad en diversas aplicaciones. Estos avances se deben a desarrollos en sensores, actuadores, inteligencia artificial y algoritmos de control. (Barrientos, Peñín, Balaguer, & Aracil, 2007)

La robótica ha ampliado significativamente su campo de aplicación, desde la manufactura industrial, donde optimiza procesos y reduce costes, hasta áreas como la medicina, donde se utilizan robots para cirugías precisas, y la exploración espacial, donde realizan tareas en entornos extremos. La combinación de diferentes tecnologías permite a los robots operar en condiciones que serían peligrosas o inviables para los seres humanos, demostrando su capacidad para transformar diversas industrias y mejorar la calidad de vida.

Capítulo 2.1.2. Historia de la robótica

Antecedentes históricos

Desde la antigüedad, el ser humano ha mostrado una gran fascinación por las máquinas que imitan las funciones y movimientos de los seres vivos. Los antiguos griegos usaban la palabra "automatos" para referirse a estas máquinas, que posteriormente evolucionó a autómata. Ejemplos tempranos incluyen la *Clépsidra* y *órgano de agua* desarrollado por **Ctesibio de Alejandría** en el siglo III a.C., y el *Teatro automático* de **Herón de Alejandría** en el siglo I a.C., que utilizaba principios de neumática e hidráulica para mover figuras y producir sonidos. (Barrientos, Peñín, Balaguer, & Aracil, 2007)

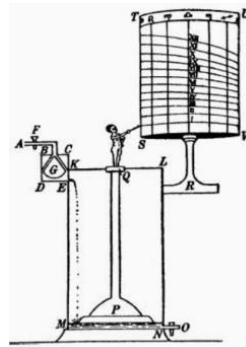


FIGURA 1. Reloj de agua de Ctesibio de Alejandría. (Brunete, San Segundo, & Herrero, 2024)

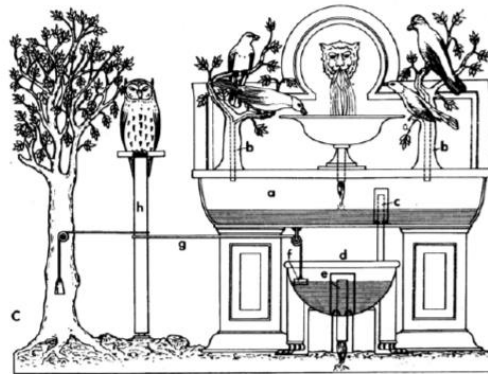


FIGURA 2. Teatro automático de Herón de Alejandría. (Fraile Marinero, 2023)

Durante la Edad Media y el Renacimiento, inventores como **Al-Jazari** y **Leonardo da Vinci** avanzaron en la creación de mecanismos automáticos. **Al-Jazari**, en el siglo XII, diseñó dispositivos como la *Fuente del Pavo Real*, un mecanismo complejo para beber o lavarse las manos. **Leonardo da Vinci**, en el siglo XV, construyó autómatas como el *León Mecánico*, que se movía usando sistemas de resortes y engranajes.

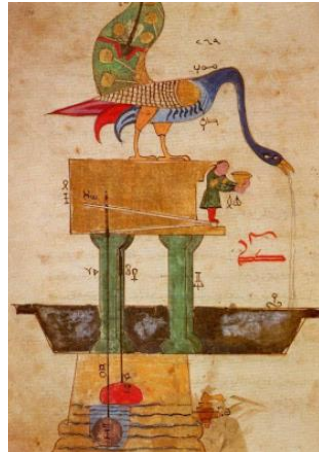


FIGURA 3. Fuente del Pavo Real de Al-Jazari. (Brunete, San Segundo, & Herrero, 2024)

Robótica moderna

La robótica moderna comenzó a tomar forma a partir del siglo XX. El término “robot” fue usado por primera vez en 1921 en la obra de teatro *Rossum’s Universal Robot (R.U.R.)* del escritor checo **Karel Capek**. Robot proviene de la palabra eslava “robota”, la cual, hacía referencia al trabajo realizado de forma forzada. (Barrientos, Peñín, Balaguer, & Aracil, 2007)

En 1948, **R.C. Goertz** del “Argonne National Laboratory” desarrolló el primer sistema de telemanipulación maestro - esclavo, diseñado para manejar materiales radiactivos sin riesgo para el operador. Este sistema fue mejorado en 1954 con la inclusión de tecnología electrónica y servocontrol, lo que permitió un control más preciso y seguro.

Mientras que el primer robot industrial fue *Unimate*, inventado por **George Devol** y **Joseph Engelberger**. En 1961, *Unimate* se instaló en una planta de General Motors para manejar piezas de metal caliente en procesos de fundición. Este evento marcó el comienzo del uso de robots en la industria, automatizando tareas peligrosas y repetitivas y mejorando la eficiencia y seguridad en las fábricas.

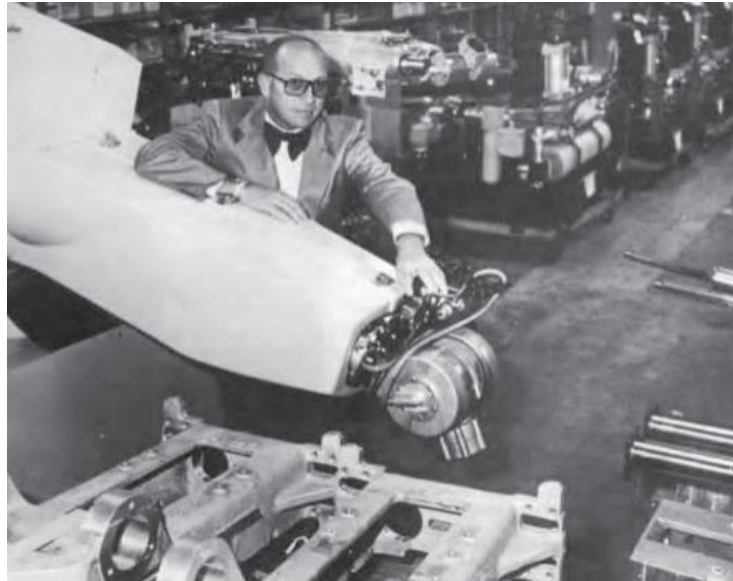


FIGURA 4. Joseph Engelberger con uno de los primeros Unimate. (Barrientos, Peñín, Balaguer, & Aracil, 2007)

Durante las décadas de 1960 y 1970, la investigación en robótica se expandió significativamente. En 1969, **Victor Scheinman** de la Universidad de Stanford desarrolló el *brazo Stanford*, sentando las bases para muchos robots industriales modernos. El *brazo de Stanford* fue el primer manipulador controlado por computadora y con la incorporación de accionamiento eléctrico. Con el tiempo, en 1978 llevó a la producción del *robot PUMA*.



FIGURA 5. Robot PUMA. (Fraile Marín, 2023)

Japón y Europa jugaron roles cruciales en el avance de la robótica industrial. En Europa, la empresa sueca ASEA (hoy ABB) construyó en 1973 el primer robot totalmente eléctrico, el *IRb6*, ofreciendo mayor precisión y control. En Japón, la empresa Kawasaki comenzó a fabricar robots tipo *Unimate*. En 1982, el profesor **Makino** de la Universidad de Yamanashi desarrolló el *robot SCARA*

(*Selective Compliance Assembly Robot Arm*), optimizado para tareas de ensamblaje con alta precisión y velocidad.



FIGURA 6. Robot Irb6 de ASEA. (Barrientos, Peñín, Balaguer, & Aracil, 2007)

Por otro lado, la robótica ha tenido aplicaciones significativas en la exploración espacial. En 1997, el satélite japonés *ETS-VII* fue pionero en la teleoperación espacial, demostrando la capacidad de realizar tareas complejas a distancia con un brazo robótico. Este avance fue crucial para la realización de misiones espaciales sin la necesidad de intervención humana directa.

Robótica en la actualidad

En el siglo XXI, la robótica ha evolucionado hacia aplicaciones más diversas, incluyendo robots de servicio y robots personales. Los avances en inteligencia artificial, visión por computadora y sistemas de control han permitido el desarrollo de robots capaces de interactuar con seres humanos y realizar tareas en entornos no estructurados. Ejemplos de ello incluyen robots asistenciales en medicina, robots de limpieza, robots de entretenimiento, etc.

La evolución de la robótica sigue transformando industrias y mejorando la calidad de vida, demostrando su potencial para revolucionar el mundo actual.

Capítulo 2.1.3. Robótica industrial

La definición de un robot industrial manipulador ha evolucionado a lo largo de los años. Según la Asociación Internacional de Estándares (ISO), un **robot manipulador industrial** se define como "un manipulador de 3 o más ejes, con control automático, reprogramable, multiaplicación, móvil o no, destinado a ser utilizado en aplicaciones de automatización industrial. Incluye al manipulador (sistema mecánico y accionadores) y al sistema de control (software y hardware de control y potencia)". (Barrientos, Peñín, Balaguer, & Aracil, 2007)

Aplicación de los robots industriales

Los robots industriales encuentran aplicaciones en diversas áreas debido a su capacidad de reprogramación y adaptación a múltiples tareas:

- **Manufactura y ensamblaje:** robots que ensamblan piezas con alta precisión y velocidad, incrementando la eficiencia y reduciendo errores humanos. Son especialmente comunes en la industria automotriz y electrónica, donde la exactitud y la rapidez son cruciales.
- **Soldadura:** robots utilizados en la industria automotriz y de fabricación para realizar soldaduras de alta calidad de manera consistente. Son capaces de realizar soldaduras por puntos y soldaduras por arco, asegurando la uniformidad y calidad de las uniones.
- **Pintura y recubrimiento:** robots que aseguran la aplicación uniforme de pintura y otros recubrimientos, mejorando la calidad del producto final y reduciendo residuos. La utilización de robots en estas tareas también minimiza la exposición de los trabajadores a sustancias químicas peligrosas.
- **Manejo de materiales:** robots que transportan y organizan materiales en almacenes y líneas de producción, mejorando la logística interna y reduciendo tiempos de espera. Estos robots son capaces de manejar cargas pesadas y realizar tareas repetitivas sin fatiga.

Estas representan solo algunas de las aplicaciones más frecuentes. Sin embargo, el espectro de su uso es significativamente más amplio y diverso.

Características generales de un robot

La selección del robot más adecuado para una aplicación específica debe considerar diversas características, entre las que destacan: (Fraile Mariner, 2023)

- **TCP (Tool Center Point):** es el punto central de la herramienta del robot. Las coordenadas de este punto serán almacenadas en el programa.

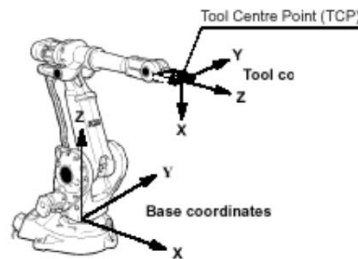


FIGURA 7. TCP de un robot. (Fraile Mariner, 2023)

- **Área de trabajo:** define el espacio en el que el extremo del robot puede operar sin tener en cuenta la herramienta. Es crucial para asegurar que el robot pueda alcanzar todos los puntos necesarios para completar su tarea.

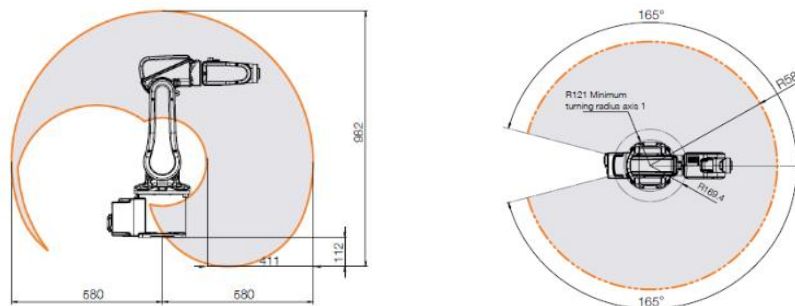


FIGURA 8. Volumen de trabajo de un robot ABB IRB 120. (Fraile Mariner, 2023)

- **Grados de libertad:** cada uno de los movimientos independientes que cada articulación puede realizar respecto a la anterior. Determina la capacidad del robot de moverse y orientarse en el espacio.

$$GDL(\text{robot}) = \Sigma GDL(\text{articulaciones})$$

- **Precisión:** distancia entre el punto alcanzado por el robot y el punto objetivo deseado.
- **Repetibilidad:** capacidad del robot para volver a la misma posición varias veces bajo condiciones idénticas.
- **Resolución:** mínima distancia de movimiento o mínimo ángulo de rotación que acepta la unidad de control del robot.

- **Velocidad y aceleración:** determinan la rapidez con la que se pueden completar las tareas, afectando directamente a la eficiencia de la producción.
- **Capacidad de carga:** peso máximo que el robot puede manejar sin comprometer su rendimiento. Debe considerarse el peso de la pinza más el de la pieza a manipular.

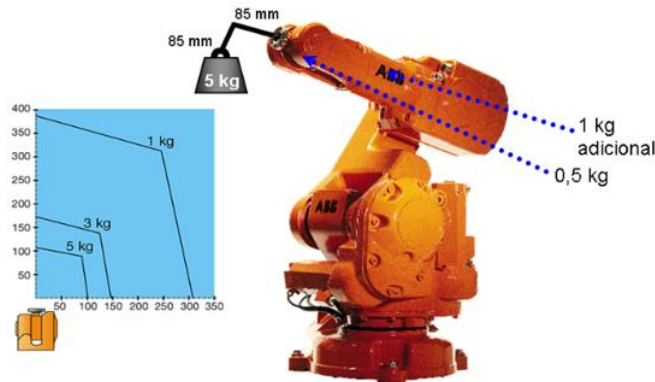


FIGURA 9. Capacidad de carga de un robot IRB 140. (Fraile Marinero, 2023)

- **Puntos singulares:** puntos del espacio de trabajo en los que es imposible para el robot realizar una trayectoria en línea recta, ya que conllevaría al movimiento de algún eje a velocidad infinita o que el valor de los ejes en esa posición sea indeterminado.
- **Sistema de control:** determina las capacidades del robot en términos de programación, comunicación y adaptación a diferentes tareas.

Seguridad en las instalaciones robotizadas

La seguridad constituye un aspecto crítico en cualquier proyecto, y esto es especialmente importante en una instalación robotizada. Algunas medidas de seguridad para tener en cuenta en una instalación robotizada son (Barrientos, Peñín, Balaguer, & Aracil, 2007):

- **Barreras de seguridad y sensores:** para evitar el acceso no autorizado a las áreas de operación del robot y detener el movimiento en caso de intrusión.
- **Sistemas de parada de emergencia:** para detener rápidamente el robot en caso de emergencia.
- **Detección de sobreesfuerzos y colisiones:** utilización de sensores que desactiven el robot si se detecta un esfuerzo excesivo o una colisión.
- **Pulsadores de seguridad:** dispositivos que impiden el movimiento accidental del robot. El más conocido es el pulsador de hombre muerto.



Justificación económica

La implementación de robots industriales representa una inversión significativa, pero sus beneficios económicos a largo plazo suelen justificar el gasto inicial. A continuación, se detallan algunos factores clave que deben considerarse al evaluar la viabilidad económica de incorporar robots en procesos industriales:

- **Reducción de costes laborales:** la utilización de robots puede reducir significativamente los costes asociados con la mano de obra, especialmente en tareas repetitivas y peligrosas.
- **Incremento de la productividad:** los robots pueden trabajar de manera continua sin descansos, aumentando la producción y reduciendo los tiempos de ciclo.
- **Mejora en la calidad:** la precisión y repetitividad de los robots pueden mejorar la calidad del producto final, reduciendo defectos y reproceso de trabajos.
- **Flexibilidad y adaptabilidad:** los robots pueden reprogramarse para diferentes tareas, proporcionando flexibilidad para adaptarse a cambios en la producción o en la demanda del mercado.

Capítulo 2.1.4. Clasificación de robots industriales

Los robots industriales pueden clasificarse según diversos criterios que reflejan su diseño, funcionalidad y aplicaciones. A continuación, se detallan las clasificaciones más relevantes:

Atendiendo a la generación

La generación de un robot se define por el contexto tecnológico en que éste aparece. El cambio de una generación a otra se produce cuando se alcanza un logro tecnológico que representa una mejora considerable en las habilidades del robot (Barrientos, Peñín, Balaguer, & Aracil, 2007). De este modo, existen las siguientes generaciones (Fraile Marinero, 2023):

- **1ª Generación:** ejecutan las tareas de manera secuencial tal como fueron programados, sin reconocer ni adaptarse a cambios en su entorno, dado que no poseen la capacidad de interactuar con el mismo.
- **2ª Generación:** pueden obtener información limitada sobre su entorno y reaccionar en función de ello. Son capaces de identificar y localizar objetos, así como de ajustar sus movimientos basándose en la detección fuerzas.
- **3ª Generación:** se programan usando lenguaje natural y pueden planificar tareas de manera autónoma. Están equipados con funciones que les confieren ciertas habilidades consideradas como “inteligentes”.

Atendiendo al área de aplicación

Dependiendo del entorno y la tarea que vayan a desempeñar se pueden dividir en dos grandes grupos (Barrientos, Peñín, Balaguer, & Aracil, 2007):

- **Automatización de la producción:** esenciales en líneas de montaje y procesos de manufactura. Realizan operaciones repetitivas con alta precisión y velocidad, como soldadura, embalaje, pintura, etc.
- **Servicios:** enfocados en tareas de inspección de productos, mantenimiento preventivo y logística. Ayudan a mejorar la eficiencia y la seguridad en ambientes de trabajo.

Atendiendo al tipo de actuadores incorporados

El tipo de actuador determina la fuerza, velocidad y precisión del robot, adaptándolos a necesidades específicas. Se clasifican en (Barrientos, Peñín, Balaguer, & Aracil, 2007):

- **Eléctricos:** precisos y fáciles de controlar, ideales para tareas de montaje y operaciones.

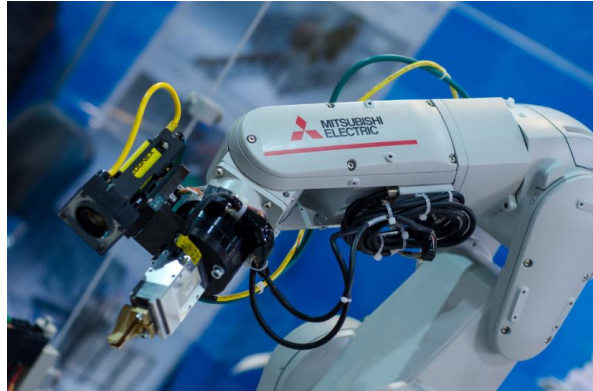


FIGURA 10. Robot de accionamiento eléctrico. (Cortesía de Mitsubishi Electric)

- **Hidráulicos:** ofrecen más potencia por lo que son preferidos en la industria pesada, donde se requieren movimientos fuertes y robustos.

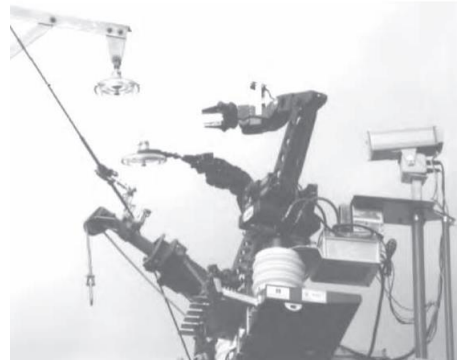


FIGURA 11. Robots de accionamiento hidráulico. (Barrientos, Peñín, Balaguer, & Aracil, 2007)

- **Neumáticos:** adecuados para tareas ligeras y rápidas, suelen ser más económicos y tienen un mantenimiento más sencillo.



FIGURA 12. Robot de accionamiento neumático. (Cortesía de FESTO).

Atendiendo a la configuración de sus ejes

El número de ejes y la configuración de ejes del robot determinan la flexibilidad y el alcance de su área de trabajo. Se distinguen (Barrientos, Peñín, Balaguer, & Aracil, 2007):

- **SCARA:** ideal para operaciones donde se requieren movimientos rápidos y precisos como tareas de “pick & place”.

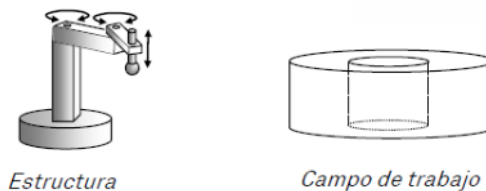


FIGURA 13. Estructura y campo de trabajo de robot SCARA. (Fraile Marín, 2023)

- **Angulares o antropomórficos:** son robots cuyos movimientos son principalmente angulares, permitiendo realizar movimientos rotacionales.

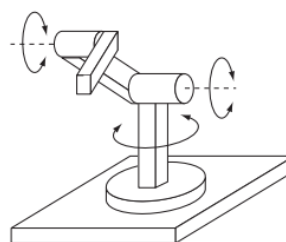


FIGURA 14. Robot angular o antropomórfico. (Barrientos, Peñín, Balaguer, & Aracil, 2007)

- **Cartesianos:** operan en ejes lineales X, Y y Z, adecuados para movimientos precisos en tres dimensiones.

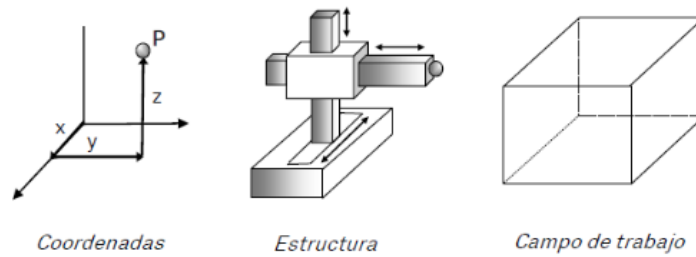


FIGURA 15. Coordenadas, estructura y campo de trabajo de robot cartesiano. (Fraile Marinero, 2023)

- **Cilíndricos:** combinan movimientos lineales y rotación alrededor de un eje vertical, útiles para tareas desarrolladas en espacios cilíndricos (de ahí el nombre).

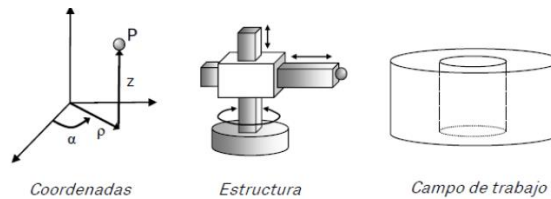


FIGURA 16. Coordenadas, estructura y campo de trabajo de robot cilíndrico. (Fraile Marinero, 2023)

- **Esféricos:** también conocidos como robots polares, sus ejes forman un sistema de coordenadas polares donde su área de trabajo abarca una esfera parcial alrededor de su base.



FIGURA 17. Coordenadas, estructura y campo de trabajo de robot esférico o polar. (Fraile Marinero, 2023)

Atendiendo al tipo de control

Los robots pueden ser clasificados en distintas categorías según el sistema de control que utilizan, basándose en las directrices de la norma ISO 8373 y las definiciones de la Federación Internacional de Robótica (IFR). A continuación, se describen cuatro tipos principales (Barrientos, Peñín, Balaguer, & Aracil, 2007):

- **Robot secuencial:** maneja sus movimientos uno a uno, controlando cada eje de forma individual y secuencial. Un movimiento comienza solo después de que el anterior ha concluido, funcionando bajo un esquema conocido como control punto a punto (PTP). Este enfoque es típico en dispositivos como los manipuladores neumáticos.
- **Robot controlado por trayectoria:** se distingue por su habilidad para ejecutar movimientos coordinados en múltiples ejes, siguiendo instrucciones detalladas que delinean una trayectoria precisa a seguir, normalmente calculada a través de métodos de interpolación. Esta capacidad permite un control más exhaustivo y dinámico sobre la ruta que debe seguir el robot.
- **Robot adaptativo:** equipado con avanzados sistemas de sensores y mecanismos de aprendizaje o adaptación, este robot puede modificar sus acciones basándose en el análisis de datos ambientales. Ejemplos incluyen el uso de visión artificial o sensores táctiles que le permiten reaccionar y adaptarse a variaciones en su entorno operativo.
- **Robot teleoperado:** operado remotamente por un humano, este robot transfiere las capacidades sensoriales y motoras del operador a lugares distantes, siendo especialmente útil en situaciones o ambientes donde la presencia humana es impracticable o peligrosa.

Capítulo 2.2. Automática

Capítulo 2.2.1. Introducción a la automática

La **automática** es la ciencia que se centra en la automatización y sus diversas aplicaciones. Su objetivo principal es desarrollar técnicas que minimicen o eliminen la necesidad de intervención humana en la producción y operación de bienes y servicios. Esto se logra sustituyendo al operador humano por dispositivos mecánicos o electrónicos que pueden realizar tareas físicas o mentales de manera programada. (Brunete, San Segundo, & Herrero, 2024)

Es una disciplina multidisciplinaria, abarcando conocimiento y tecnologías de áreas como la mecánica, la electrónica y la informática. Sus aplicaciones se extienden a múltiples campos, incluyendo la industria automotriz, la aeronáutica, la robótica, la ingeniería mecánica, la mecatrónica, etc.

La automática juega un papel crucial en la mejora de la eficiencia y la precisión de diversas industrias y servicios, impulsando la innovación y la mejora continua de los procesos mediante la integración de sistemas automáticos.

Capítulo 2.2.2. Historia de la automática

Desde los inicios de la civilización, los seres humanos han buscado formas de reemplazar el esfuerzo manual con máquinas y sistemas de control. Este esfuerzo ha llevado al desarrollo de la automática a lo largo de las diferentes épocas de la historia. (Brunete, San Segundo, & Herrero, 2024)

Edad Antigua

Los primeros indicios de la automatización se encuentran en la antigüedad. Alrededor del 2000 a.C., en Babilonia, se implementaron sistemas de regadío que utilizaban principios básicos de control automático, documentados en el *código de Hammurabi*. **Filón de Bizancio**, en el siglo I a.C., ideó el primer molino de agua, lo que le llevó a la creación de la primera bomba de agua al usar la fuerza del agua para llevarlo a un punto más alto.



FIGURA 18. Molino de agua de Filón de Bizancio. (Brunete, San Segundo, & Herrero, 2024)

Edad Media

Durante la Edad Media, la tecnología de control automático en Europa sufrió una pausa. Sin embargo, hubo algunos avances, como el *Gallo de Estrasburgo*, un autómeta construido en 1352 que funcionaba como un reloj y es el autómeta más antiguo que se conserva actualmente.



FIGURA 19. Gallo de Estrasburgo. (Brunete, San Segundo, & Herrero, 2024)

En esta época, muchos de los conocimientos sobre sistemas de control se preservaron en la cultura islámica y se redescubrieron al final del Renacimiento.

Edad Moderna

En el siglo XVIII, **Jacques de Vaucanson** creó un pato mecánico que imitaba acciones biológicas como beber y digerir alimentos, sorprendiendo a toda Europa.

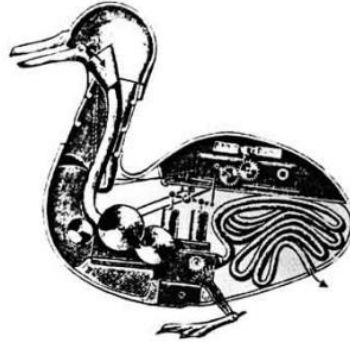


FIGURA 20. Pato de Vaucanson. (Brunete, San Segundo, & Herrero, 2024)

Uno de los hitos más importantes de esta época fue el *telar de Jacquard*, inventado en 1801. Este dispositivo usaba tarjetas perforadas para controlar el patrón del tejido, lo que sentó las bases para el desarrollo de las máquinas de control numérico.



FIGURA 21. Telar de Jacquard. (Brunete, San Segundo, & Herrero, 2024)

Edad Contemporánea

La Revolución Industrial marcó el comienzo de la era moderna en la automática. En 1788, **James Watt** inventó el *regulador centrífugo* para controlar la velocidad de las máquinas de vapor, considerado el punto de partida del control automático como disciplina científica.



FIGURA 22. Regulador de Watt. (Brunete, San Segundo, & Herrero, 2024)

A partir del siglo XVIII, la automática experimentó una evolución exponencial. **Charles Babbage** diseñó su *máquina analítica* en 1835, precursora del ordenador moderno.

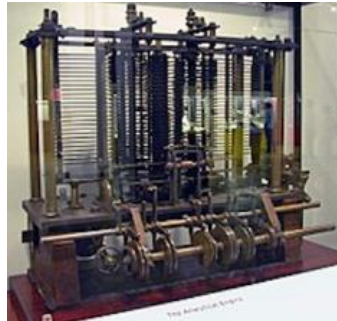


FIGURA 23. Reconstrucción de la máquina analítica de Babbage. (Brunete, San Segundo, & Herrero, 2024)

Desde la década de 1960, con la invención de los controladores lógicos programables (PLC) y los microprocesadores, la automatización se integró profundamente en la industria.

Estos avances han hecho de la automática una disciplina crucial para la mejora de los procesos industriales y el desarrollo de tecnologías que continúan transformando nuestra vida diaria.

Capítulo 2.2.3. Automatización industrial

La automatización industrial consiste en la aplicación de la automática a un proceso o dispositivo industrial. Se aplica tanto a procesos de fabricación continua como de piezas discretas, siendo fundamental para mejorar la eficiencia, calidad y flexibilidad de la producción. (Brunete, San Segundo, & Herrero, 2024)

Objetivos de la automatización industrial

- Incrementar la productividad.
- Mejorar la calidad de los productos fabricados.
- Aumentar la flexibilidad.
- Seguridad en los ambientes de trabajo.
- Reducir los costes de fabricación.

Tipos de automatización

- **Automatización rígida:** también conocida como automatización fija debido a que la secuencia de operaciones está fijada y es difícil de cambiar. Se caracteriza por la alta eficiencia en la producción de grandes volúmenes de productos idénticos.
- **Automatización flexible:** permite cambios en la secuencia de operaciones mediante reprogramación, adaptándose a diferentes tipos de productos.

La pirámide de la automatización

La pirámide de la automatización representa los distintos niveles de tecnologías y sistemas que trabajan juntos en una planta industrial para optimizar la producción. (Brunete, San Segundo, & Herrero, 2024)

- **Nivel 1 (Dispositivos de campo):** incluye sensores y actuadores que interactúan directamente con los procesos de producción.
- **Nivel 2 (Controladores de máquinas):** compuesto por PLCs, sistemas CNC y robots industriales.
- **Nivel 3 (Sistemas de control y supervisión):** los sistemas SCADA supervisan y controlan múltiples máquinas y procesos industriales,

recopilando datos en tiempo real y proporcionando interfaces gráficas para los operadores. Este nivel incluye interfaces hombre-máquina (HMI) para la interacción con los sistemas.

- **Nivel 4 (Sistemas de operación o planificación):** los Sistemas de Ejecución de Manufactura (MES) gestionan la planificación y ejecución de la producción de toda la planta, monitoreando el flujo de materiales y el estado de los procesos desde las materias primas hasta los productos terminados.
- **Nivel 5 (Sistemas de gestión):** en este nivel se integran sistemas de Planificación de Recursos Empresariales (ERP) incorporando todas las funciones de la empresa, desde la producción hasta las finanzas, proporcionando una visión completa de las operaciones y facilitando la toma de decisiones estratégicas.

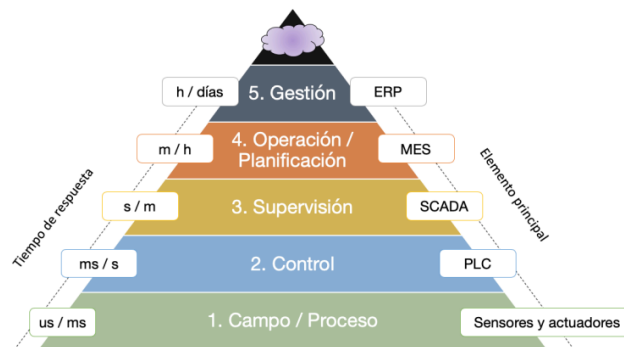


FIGURA 24. Pirámide la automatización industrial. (Brunete, San Segundo, & Herrero, 2024)

En la actualidad, se podría reconocer un nivel adicional por encima del nivel 5 denominado “la nube”. Este nivel permite que los datos de los niveles inferiores se almacenen y procesen centralmente, facilitando la transformación digital hacia la Industria 4.0.

Capítulo 2.2.4. Clasificación de autómatas industriales

La clasificación de los autómatas industriales se puede realizar según diferentes criterios. Una posible clasificación es según la tecnología utilizada: (Sanchis Llopis, Romero Pérez, & Ariño Latorre, 2010)

- **Automatismos cableados:** se implementan mediante uniones físicas (cableado) para cumplir una función lógica y son conocidos por su robustez y simplicidad. Son adecuados para sistemas sencillos donde las operaciones de control son básicas y no requieren modificaciones frecuentes. Para pasar a la tecnología programada es necesario incorporar un sistema de control programable como se muestra en la figura 25.

Ejemplos: sistemas de relés, PLDs (Programmable Logic Devices).

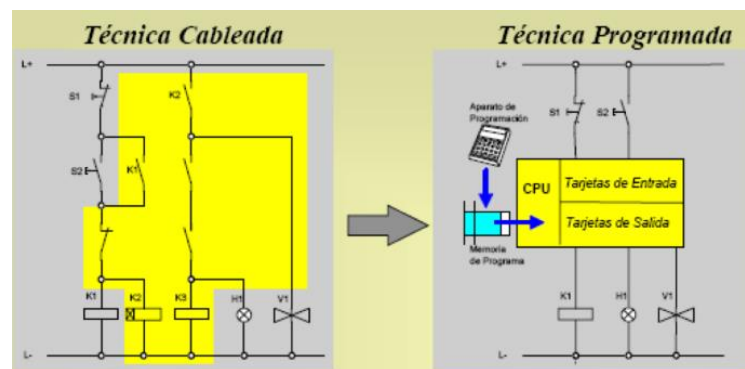


FIGURA 25. Transición de tecnología cableada a programable. (Brunete, San Segundo, & Herrero, 2024)

- **Automatismos programados:** se implementan mediante la programación de dispositivos electrónicos. Permiten una mayor flexibilidad y capacidad para realizar funciones complejas de control y comunicación. Ocupan menos espacio y son adecuados para procesos más complejos.

Dentro de los automatismos programados se distinguen:

- **Controladores Lógicos Programables (PLCs):** son computadoras especializadas diseñadas para operar en ambientes industriales, ya que son capaces de manejar múltiples señales de entrada y salida en tiempo real.
Aplicación: control de maquinaria en líneas de montaje, procesos de producción.
- **Microcontroladores:** son circuitos programables que incluyen un microprocesador, memoria y los periféricos necesarios.
Aplicación: sistemas de control de máquinas que se fabrican en grandes cantidades donde la reducción de costes es esencial.
- **Computadores industriales (PCs industriales):** son ordenadores compatibles con PCs de sobremesa en cuanto a software, pero están diseñados con un hardware más robusto para entornos industriales.
Aplicación: control de procesos industriales, supervisión y recopilación de datos.

Capítulo 2.2.5. Controlador Lógico Programable (PLC)

Un Controlador Lógico Programable (PLC), también conocido como API (Autómata Programable Industrial), es un dispositivo electrónico diseñado para controlar procesos industriales en tiempo real. Se programa en un lenguaje específico no informático para ejecutar algoritmos de control de procesos secuenciales, esenciales en la automatización industrial. Estos dispositivos son robustos y están diseñados para operar en ambientes industriales adversos, soportando condiciones de polvo, temperatura y vibraciones. (Sanchis Llopis, Romero Pérez, & Ariño Latorre, 2010)

Arquitectura del PLC

La arquitectura de un PLC se basa en una estructura modular que facilita su configuración y adaptación a diferentes necesidades industriales. Los componentes principales incluyen: (Sanchis Llopis, Romero Pérez, & Ariño Latorre, 2010)

- **Procesador (CPU):** encargada de ejecutar el programa almacenado en la memoria.
- **Memoria:** se divide en tres tipos:
 - **Memoria de programa:** almacena el código del programa que controla el proceso.
 - **Memoria de datos:** almacena datos de configuración o del funcionamiento del autómata y del proceso.
 - **Memoria interna:** almacena tanto datos de variables internas del autómata como valores de entradas y salidas.
- **Módulos de entrada/salida:** hacen de interfaz entre el PLC y los sensores y actuadores del proceso.
- **Control de periféricos:** permite la conexión del PLC con otros dispositivos, como pantallas táctiles y computadoras.
- **Fuente de alimentación:** encargada de suministrar la tensión necesaria a todos los módulos del PLC. Generalmente es de 24 V.

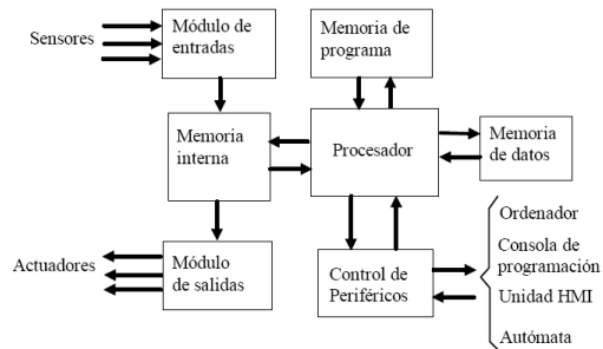


FIGURA 26. Arquitectura de un Controlador Lógico Programable. (Sanchis Llopis, Romero Pérez, & Ariño Latorre, 2010)

Tipos de PLC

Existen dos tipos principales de PLCs según su configuración física:

- **Modulares:** compuestos por módulos que se pueden conectar, añadir y configurar según las necesidades específicas del proceso. Estos incluyen módulos de CPU, fuentes de alimentación, módulos de entrada/salida y módulos de comunicación, entre otros.

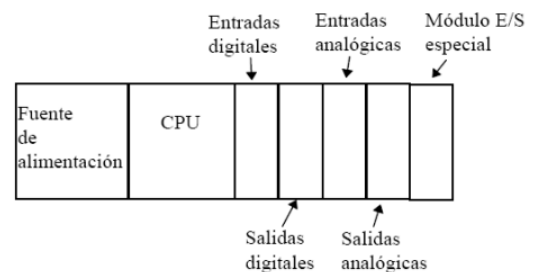


FIGURA 27. Autómata programable modular. (Sanchis Llopis, Romero Pérez, & Ariño Latorre, 2010)

- **Compactos:** integran todos los componentes esenciales en una única unidad. Son menos flexibles que los modulares, pero suelen ser más económicos y adecuados para aplicaciones más sencillas.

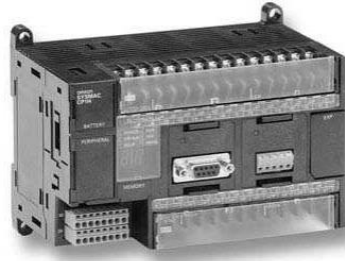


FIGURA 28. Autómata programable compacto. (Sanchis Llopis, Romero Pérez, & Ariño Latorre, 2010)

Programación del PLC

La programación de un PLC se realiza comúnmente a través de un ordenador personal utilizando un software específico del fabricante. Este software permite la edición, depuración, compilación y carga del programa en el PLC. Los lenguajes de programación más utilizados son: (Sanchis Llopis, Romero Pérez, & Ariño Latorre, 2010)

- **Diagrama de contactos (Ladder):** lenguaje intuitivo basado en esquemas de circuitos eléctricos. Este es el lenguaje de programación más utilizado en los autómatas programables y representa las ecuaciones lógicas por medio de contactos.

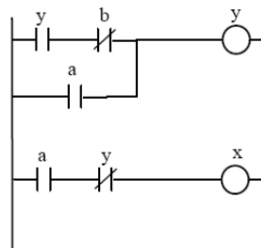


FIGURA 29. Ejemplo de programa de diagrama de contactos Ladder. (Sanchis Llopis, Romero Pérez, & Ariño Latorre, 2010)

- **Lista de instrucciones:** lenguaje de bajo nivel que se asemeja al ensamblador, utilizado para programar funciones más detalladas y específicas.
- **Diagrama de bloques funcionales:** permite la programación mediante bloques funcionales predefinidos, facilitando la creación de programas complejos de manera más visual.

Modos de funcionamiento

Se distinguen varios modos de funcionamiento esenciales para gestionar las diferentes etapas de programación y operación del PLC: (Sanchis Llopis, Romero Pérez, & Ariño Latorre, 2010)

- **Modo de programación (modo STOP):** en este modo, el PLC está preparado para recibir y almacenar el programa de control. Durante el modo de programación, el PLC no está ejecutando el control del proceso. En lugar de ello, está en una fase de configuración y carga del software necesario para su operación. Este modo permite realizar modificaciones en el programa, ajustar parámetros y asegurar que todo esté configurado correctamente antes de comenzar la ejecución de la operación real.
- **Modo de ejecución (modo RUN):** una vez que el programa ha sido cargado en el PLC y se han realizado todas las verificaciones necesarias, se cambia al modo RUN. En este modo, el PLC ejecuta continuamente el programa de control, procesando las entradas, ejecutando las instrucciones y actualizando las salidas. La ejecución del programa se lleva a cabo de forma cíclica, lo que se conoce como ciclo de scan. Durante cada ciclo de scan, el PLC lee las entradas, ejecuta el programa de usuario y actualiza las salidas en un proceso continuo e indefinido. La estructura más común de este ciclo de trabajo viene dada por la siguiente secuencia:

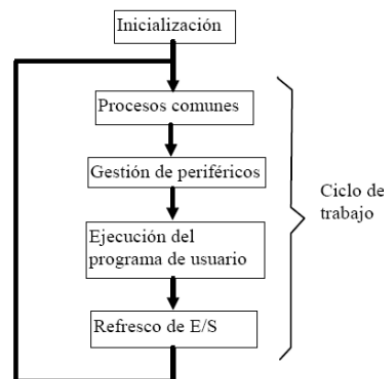


FIGURA 30. Ciclo de trabajo de un PLC. (Sanchis Llopis, Romero Pérez, & Ariño Latorre, 2010)

- **Otros modos de funcionamiento:** además de los modos principales de programación y ejecución, el PLC puede operar en otros modos, dependiendo del diseño y las capacidades del sistema. Estos pueden incluir modos de prueba o simulación, donde el PLC ejecuta el programa sin activar las salidas físicas, permitiendo la verificación del programa sin interferir con el proceso real.

Capítulo 2.2.6. Interfaz Hombre – Máquina (HMI)

Un HMI (Interfaz Hombre - Máquina) es un dispositivo periférico utilizado ampliamente en la automatización industrial para facilitar la interacción entre los operadores y los sistemas de control automatizados, como los PLCs. Permite a los operadores monitorear el estado del proceso y realizar ajustes necesarios, tales como órdenes de marcha, paro, y cambios en los parámetros de funcionamiento. (Sanchis Llopis, Romero Pérez, & Ariño Latorre, 2010)

Las unidades HMI suelen estar equipadas con un teclado y una pantalla, que puede variar en tamaño, o pueden consistir únicamente en una pantalla táctil.

Las interfaces HMI contienen un procesador y una memoria no volátil donde se almacena la aplicación de interfaz con el usuario. La programación se realiza típicamente a través de un ordenador utilizando un puerto serie o USB y el software adecuado proporcionado por el fabricante. Una vez programadas, se comunican con el autómat programable a través de puertos serie, puertos de periféricos, o incluso mediante redes de mayor nivel como Ethernet.

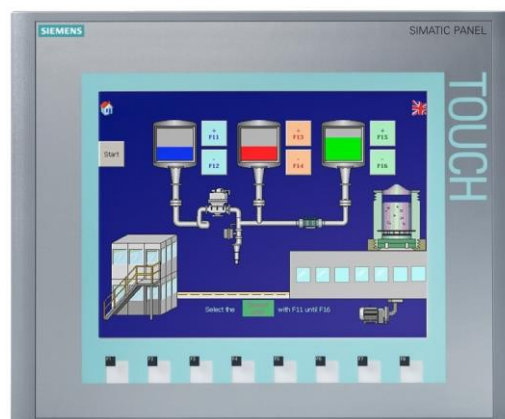


FIGURA 31. Interfaz Hombre – Máquina HMI Siemens Simatic. (Cortesía de Siemens)



Capítulo 2.3. Visión artificial

Capítulo 2.3.1. Introducción a la visión artificial

La visión artificial es una disciplina de la inteligencia artificial que capacita a los sistemas informáticos para interpretar y comprender el entorno visual mediante imágenes digitales y videos. A través del uso de cámaras y algoritmos avanzados, estos sistemas pueden procesar, analizar y tomar decisiones basadas en datos visuales. Esta tecnología emula la capacidad humana de visión, aplicándola a diversas aplicaciones industriales y comerciales, como el reconocimiento de objetos y la inspección de calidad en productos. (IBM, s.f.)

El campo de la visión artificial ha evolucionado considerablemente gracias a los avances en el aprendizaje automático y las redes neuronales convolucionales (CNN). Estos avances permiten a los sistemas descomponer y analizar imágenes a nivel de píxeles, identificando patrones y características con alta precisión.

La visión artificial no solo mejora la eficiencia y precisión en tareas industriales, sino que también abre nuevas posibilidades en áreas como la medicina, la seguridad y la automatización.

Capítulo 2.3.2. Historia de la visión artificial

La visión artificial ha recorrido un largo camino desde sus inicios a finales de la década de 1950. El desarrollo de esta tecnología comenzó en 1959, cuando neurofisiólogos mostraron a un gato una matriz de imágenes con el objetivo de vincular las respuestas en su cerebro. Este experimento reveló que el procesamiento de imágenes empieza con la identificación de formas simples, como líneas y bordes. (IBM, s.f.)

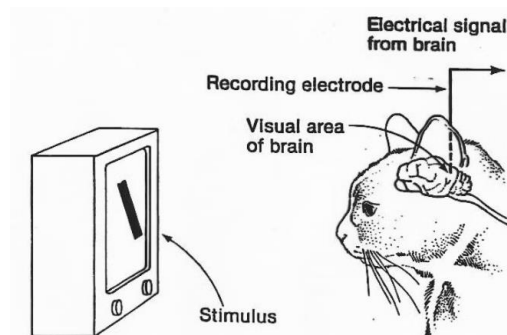


FIGURA 32. Experimento de reconocimiento de formas con un gato. (IBM, s.f.)

Simultáneamente, la primera tecnología de escaneo de imágenes fue desarrollada, permitiendo a los ordenadores digitalizar y adquirir imágenes. En 1963, los ordenadores lograron transformar imágenes bidimensionales en formas tridimensionales, marcando un evento importante en la evolución de la visión artificial.

Durante la década de 1960, la inteligencia artificial (IA) emergió como un campo académico de estudio, y la visión artificial se convirtió en una de sus principales áreas de investigación.

En 1974, se introdujo la tecnología para el reconocimiento óptico de caracteres (OCR), que permitía a los ordenadores detectar texto impreso en cualquier fuente. Al mismo tiempo, el reconocimiento inteligente de caracteres (ICR) utilizando redes neuronales comenzó a descifrar texto escrito a mano.

Un avance significativo ocurrió en 1982, cuando el neurocientífico **David Marr** propuso que la visión funciona de manera jerárquica e introdujo algoritmos para que las máquinas detectaran bordes, esquinas y curvas. De forma paralela, el científico informático **Kunihiko Fukushima** creó el *Neocognitron*, una red neuronal con capas convolucionales capaz de reconocer patrones.

Para el año 2000, la investigación en visión artificial se centró en el reconocimiento de objetos, y en 2001 surgieron las primeras aplicaciones de reconocimiento facial en tiempo real. La década de 2000 también vio la



estandarización de la anotación y etiquetado de conjuntos de datos visuales, culminando con la publicación del conjunto de datos *ImageNet* en 2010. Este conjunto de datos, que contiene millones de imágenes etiquetadas, proporcionó una base sólida para las redes neuronales convolucionales (CNN) y los modelos de aprendizaje profundo utilizados actualmente.

En 2012, un equipo de la Universidad de Toronto introdujo *AlexNet*, un modelo de CNN que redujo significativamente la tasa de error en el reconocimiento de imágenes. Este avance marcó el comienzo de una nueva era en la visión artificial, con tasas de error que se han reducido a solo unos pocos porcentajes desde entonces.

Esta evolución ha permitido que la visión artificial se integre en una amplia gama de aplicaciones industriales y comerciales, transformando sectores como la medicina, la seguridad y la industria.

Capítulo 2.3.3. Visión artificial industrial

La visión artificial industrial se refiere a la aplicación de tecnologías de visión artificial en entornos industriales con el objetivo de mejorar la eficiencia, la calidad y la seguridad de los procesos productivos. Esta tecnología utiliza cámaras, sensores y algoritmos avanzados para capturar y analizar imágenes en tiempo real, permitiendo a las máquinas realizar tareas que requieren reconocimiento visual, inspección y control de calidad.

Aplicaciones en la industria

Las aplicaciones de la visión artificial en la industria son variadas y abarcan múltiples sectores. Algunas de las más comunes incluyen:

- **Inspección de calidad:** para detectar defectos en productos durante la producción. Esto puede incluir la identificación de imperfecciones en superficies, errores de ensamblaje o variaciones en el color y tamaño.
- **Automatización de procesos:** en líneas de ensamblaje, la visión artificial permite la automatización de tareas repetitivas y la supervisión constante de la producción. Permite reducir el error humano y mejorar la calidad del producto final.
- **Gestión de inventarios:** las cámaras de visión artificial pueden monitorear y rastrear inventarios en tiempo real, facilitando la gestión de almacenes y optimizando la cadena de suministro. Aseguran que los niveles de stock se mantengan dentro de los parámetros adecuados, evitando tanto el exceso como la escasez de productos.
- **Seguridad y mantenimiento:** se aplica en supervisión de equipos y la seguridad en el lugar de trabajo. Se puede detectar condiciones peligrosas o el desgaste de maquinaria, permitiendo un mantenimiento predictivo y la prevención de accidentes.

Ventajas y beneficios

La integración de sistemas de visión artificial en entornos industriales ofrece numerosos beneficios:

- **Mejora de la calidad:** al detectar defectos en tiempo real, se minimizan los productos defectuosos y se asegura un alto estándar de calidad.
- **Reducción de costes:** al mejorar la eficiencia y reducir el desperdicio, la visión artificial contribuye a la reducción de los costes operativos.

- **Seguridad:** la supervisión constante y el mantenimiento predictivo reducen el riesgo de fallos en equipos y accidentes laborales.

La visión artificial industrial es una herramienta que está cambiando la forma en que se realizan los procesos productivos, brindando precisión, velocidad y fiabilidad. Esto impulsa la innovación y mejora la competitividad en la industria.

Capítulo 2.3.4. Clasificación de sistemas de visión artificial industriales

Los sistemas de visión artificial más usados actualmente son los siguientes: (EDS Robotics, 2020) (Elecproy, 2024)

- **Sensores de visión:** son dispositivos diseñados para resolver problemas específicos y relativamente simples. Estos sistemas son más avanzados que los sensores fotoeléctricos tradicionales, ya que pueden ofrecer resultados binarios (paso o fallo) con mayor precisión y confiabilidad. Su uso es ideal en aplicaciones donde se requiere una instalación y operación sencilla, además de ser una opción económica debido a su bajo costo y tamaño compacto.

Aplicaciones:

- Inspección básica de calidad.
- Detección de presencia o ausencia de componentes.
- Verificación de ensamblajes sencillos.

- **Cámaras inteligentes y sistemas de visión integrados:** son sistemas más avanzados que los sensores de visión y están diseñados para abordar una variedad más amplia de necesidades industriales. Combinan alta resolución con una potente capacidad de procesamiento permitiendo realizar tareas complejas como la inspección detallada y el control de calidad en tiempo real.

Aplicaciones:

- Inspección de alta precisión en líneas de montaje.
- Verificación de ensamblajes complejos.
- Integración con otros sistemas automatizados para una evaluación integral de productos.

- **Sistemas de visión avanzados:** representan la cima de la tecnología en visión artificial industrial. Equipados con hardware y software de última generación, pueden analizar grandes volúmenes de datos simultáneamente y realizar tareas altamente complejas. Son esenciales en industrias que requieren un control de calidad muy riguroso y detallado.

Aplicaciones:

- Inspección y control de calidad en la industria automotriz, aeroespacial y electrónica.
- Análisis en tiempo real de producto y procesos de alta complejidad.
- Implementación en entornos que demandan alta precisión y fiabilidad.

- **Visión estéreo:** la visión estéreo usa dos cámaras para capturar imágenes desde diferentes ángulos y crear modelos tridimensionales del entorno. Fundamental en aplicaciones donde es necesario calcular con precisión la distancia y la profundidad de los objetos.

Aplicaciones:

- Navegación y control de robots autónomos.
- Inspección detallada de piezas.

- **Visión 3D:** permite la creación de modelos tridimensionales a partir de imágenes en 2D.

Aplicaciones:

- Modelado tridimensional.
- Aplicaciones en realidad virtual.

- **Visión térmica:** a partir de cámaras infrarrojas se captura la radiación de calor emitida por los objetos. Esta tecnología es especialmente útil en situaciones donde la visibilidad es baja o nula, como en la oscuridad total o en condiciones de humo y polvo. Los sistemas de visión térmica pueden detectar diferencias de temperatura en los objetos, lo que permite identificar áreas calientes o frías, haciendo visible lo invisible al ojo humano.

Aplicaciones:

- Detección de puntos calientes para prevenir fallos.
- Identificación de intrusos o localización de personas en condiciones desfavorables.

- **Visión multispectral:** emplea cámaras que capturan imágenes en varias bandas del espectro electromagnético, más allá del rango visible para el ojo humano. Esto permite analizar los materiales y sus propiedades de manera más detallada, detectando características que no son visibles en el espectro de luz normal.

Aplicaciones:

- Evaluación del estado de los productos.
- Control de calidad de los productos.
- Detección de plagas o contaminantes.

- **Visión por computadora:** implica el uso de algoritmos y modelos matemáticos para que las máquinas interpreten y comprendan imágenes y videos, realizando tareas específicas como el reconocimiento de objetos, el seguimiento de movimientos y la identificación de patrones. Esta tecnología se utiliza ampliamente para automatizar procesos y mejorar la precisión y eficiencia en diversas aplicaciones industriales.

Aplicaciones:

- Reconocimiento facial y detección de comportamientos sospechosos.
- Análisis de imágenes para diagnóstico predictivo.



Capítulo 2.4. Reconocimiento de voz

Capítulo 2.4.1. Introducción al reconocimiento de voz

El reconocimiento de voz es una tecnología que permite a máquinas y programas identificar y comprender el habla humana. Como parte de la inteligencia artificial, esta tecnología no solo se encarga de reconocer palabras, sino también de interpretar el significado, contexto, intención y emociones del discurso. (FasterCapital, 2024)

Mediante el uso de algoritmos avanzados de procesamiento de lenguaje natural y aprendizaje automático, el reconocimiento de voz ha progresado significativamente, especialmente con el apoyo de redes neuronales profundas. Estos avances han dado lugar a aplicaciones como asistentes virtuales, sistemas de transcripción automática y herramientas para mejorar la accesibilidad.

El principal reto es la interpretación precisa de la semántica y el contexto del habla, aspectos cruciales para interactuar eficazmente con los usuarios. Esta tecnología es fundamental en el desarrollo de la inteligencia artificial y tiene un impacto significativo en nuestra vida diaria, con un gran potencial para el futuro.

Capítulo 2.4.2. Historia del reconocimiento de voz

La tecnología de reconocimiento de voz ha evolucionado considerablemente desde sus inicios, y su historia se puede dividir en cuatro fases principales: (FasterCapital, 2024)

Fase inicial (décadas de 1950-1970)

Durante las primeras décadas del desarrollo del reconocimiento de voz, los investigadores se enfocaron en sistemas simples debido a las limitaciones de hardware y software. Los primeros sistemas estaban limitados a reconocer números aislados o palabras emitidas por una sola voz. Los diseños más importantes fueron:

- **Audrey (1952):** creado por Laboratorios Bell, el primer sistema de reconocimiento de voz, que podía identificar los dígitos del 0 al 9 empleando un micrófono y un conjunto de filtros.
- **Shoebox (1962):** lanzado por IBM, Shoebox era un sistema que podía identificar 16 palabras y ejecutar operaciones básicas como la suma, para controlar una máquina de juguetes.
- **Harpy (1976):** creado por la Universidad Carnegie Mellon, podía distinguir 1011 palabras diferentes, empleando una red de secuencias de palabras denominada Modelo Oculto de Márkov (HMM).

Fase intermedia (décadas de 1980-1990)

Durante este periodo se avanzó significativamente implementado redes neuronales, técnicas de distorsión dinámica del tiempo y modelado estadístico del lenguaje. Estos avances permitieron a los sistemas reconocer el habla continua de varias personas, adaptarse al ruido ambiente y a diferentes acentos. Entre las creaciones más destacadas de esta fase se incluyen:

- **Dragon (1982):** diseñado por Dragon Systems, fue el primer sistema comercial de reconocimiento de voz, que permitía la transcripción y el dictado de discursos continuos de múltiples hablantes.
- **Sphinx (1986):** creado por la Universidad Carnegie Mellon, utilizando modelos acústicos y el modelo HMM fue el primer sistema en reconocer el habla en tiempo real.
- **ViaVoice (1997):** lanzado por IBM, fue el primer sistema que podía reconocer el habla continua en varios idiomas, usando redes neuronales y modelos de lenguaje.

Fase moderna (década de 2000 - presente)

Esta etapa se caracteriza por la incorporación del reconocimiento de voz en una variedad de dispositivos y aplicaciones de uso diario:

- **Siri (2011):** Apple presentó este asistente de voz que facilita tareas como buscar en la web, enviar mensajes y configurar recordatorios mediante comandos de voz en lenguaje natural.
- **Google Assistant (2016):** Google lanzó un asistente de voz capaz de realizar numerosas tareas, desde reproducir música hasta controlar dispositivos inteligentes, destacándose por su habilidad para gestionar conversaciones contextuales.
- **Alexa (2014):** Amazon introdujo un asistente de voz que alimenta los dispositivos Echo, permitiendo a los usuarios hacer compras, escuchar noticias y controlar dispositivos domésticos a través de comandos de VOZ.



FIGURA 33. Alexa. (Cortesía de Amazon)

Fase futura (década de 2020 en adelante)

El futuro del reconocimiento de voz presenta nuevos desafíos y oportunidades, destacando:

- **Traductor universal:** se espera el desarrollo de dispositivos capaces de traducir cualquier idioma en tiempo real con alta precisión, facilitando la comunicación global.
- **Agente emocional y social:** sistemas que reconocerán y responderán a señales emocionales y sociales del habla, mejorando la naturalidad y la profundidad de las interacciones entre humanos y máquinas.
- **Asesor ético y social:** sistemas que analizarán y aconsejarán sobre las implicaciones éticas y sociales del discurso humano, promoviendo una comunicación más responsable.

Capítulo 2.4.3. Clasificación de sistemas de reconocimiento de voz

El reconocimiento de voz ha cambiado drásticamente la manera en que interactuamos con la tecnología, haciendo posible una comunicación más natural y accesible. A lo largo de los años, estos sistemas han mejorado considerablemente en su capacidad para comprender e interpretar el habla humana. A continuación, se describen los diferentes tipos de sistemas de reconocimiento de voz, junto con sus características, ventajas y desventajas: (FasterCapital, 2024)

Sistemas dependientes del hablante

Diseñados para reconocer y comprender la voz de usuarios individuales. Requieren un entrenamiento extenso, donde el usuario proporciona numerosos datos de voz para que el sistema aprenda sus patrones únicos. Aunque son muy precisos para el usuario que los ha entrenado, presentan dificultades para adaptarse a nuevas voces o variaciones en el patrón de habla.

Ejemplos: asistentes de voz como Siri y Google Assistant.

Sistemas independientes del hablante

Estos sistemas no necesitan un entrenamiento previo con voces específicas. Son entrenados con una amplia variedad de muestras de voz de diferentes personas, permitiéndoles reconocer un amplio rango de voces. Aunque no alcanzan la precisión de los sistemas específicos del usuario, ofrecen mayor flexibilidad y adaptabilidad.

Sistemas de reconocimiento de palabras aisladas

Diseñados para reconocer palabras aisladas pronunciadas por el usuario. Son útiles en aplicaciones que requieren la identificación de comandos específicos o palabras clave. Funcionan bien cuando las palabras se pronuncian claramente y con pausas entre ellas, pero enfrentan dificultades con el habla continua o el procesamiento del lenguaje natural.

Ejemplos: sistemas automatizados de atención telefónica.

Sistemas de reconocimiento continuo de voz

Capaces de transcribir y comprender el habla natural y continua. Procesan un flujo constante de audio, transformándolo en texto o ejecutando comandos basados en el habla. Su precisión ha mejorado significativamente gracias a los avances en aprendizaje automático y redes neuronales.

Ejemplos: asistentes virtuales como Alexa de Amazon y Cortana de Microsoft.

Sistemas de comando y control

Son sistemas concebidos para comprender comandos específicos del usuario. Identifican frases o patrones establecidos y realizan las acciones correspondientes, siendo frecuentemente utilizados en la automatización del hogar mediante control por voz.

Ejemplos: sistema inteligente para el hogar que enciende o apaga la calefacción, la luz, etc.

Sistemas de procesamiento del lenguaje natural (PLN)

Estos sistemas no solo convierten el habla en texto, sino que también entienden su significado y contexto. Emplean algoritmos avanzados y modelos lingüísticos para interpretar el lenguaje humano, lo que facilita interacciones más complejas entre personas y máquinas. Son capaces de captar intenciones, extraer información relevante y generar respuestas precisas.

Ejemplos: chatbots y asistentes virtuales como Siri o Google Assistant incorporan sistemas de PLN.

Capítulo 2.5. Comunicaciones Industriales

Capítulo 2.5.1. Introducción a las comunicaciones industriales

Las comunicaciones industriales son fundamentales para la automatización y el control de procesos en el ámbito industrial. Permiten que una amplia variedad de dispositivos, como sensores, actuadores, controladores programables y sistemas de gestión, se conecten y transmitan datos entre sí, asegurando un correcto funcionamiento y supervisión de los sistemas industriales.

Capítulo 2.5.2. Redes de comunicación industrial

Las redes de comunicación industrial facilitan el intercambio de información y el control en tiempo real entre dispositivos utilizando enlaces como cobre, fibra óptica o inalámbricos. Son esenciales para manejar grandes volúmenes de datos en entornos industriales usando un ancho de banda limitado. (Sicma21, 2021)

Estas redes permiten el intercambio de datos entre dispositivos de campo, controladores lógicos programables (PLC), sistemas de control de supervisión y adquisición de datos (SCADA), y sistemas de control distribuido (DCS). La comunicación se realiza mediante protocolos específicos que aseguran la transferencia de datos de manera eficiente. Los datos se transmiten en serie, organizados en paquetes, a una velocidad determinada denominada velocidad de transmisión, que se mide en bits por segundo (bps), expresada comúnmente en Kbps o Mbps.

Un desafío histórico ha sido la conexión de dispositivos de distintos fabricantes debido a los diferentes protocolos utilizados. Esto ha complicado la planificación y ha encarecido la unificación de la información. Para resolver este problema, se han desarrollado estándares abiertos que buscan simplificar y unificar las comunicaciones industriales como el modelo OSI. No obstante, alcanzar un consenso completo en la industria sigue siendo un desafío en desarrollo.

Capítulo 2.5.3. Pirámide CIM

Como ya se ha explicado anteriormente en el capítulo 2.2.3., la pirámide CIM (Computer Integrated Manufacturing) es un modelo utilizado para describir la estructura de automatización en una planta industrial.

Enfocando la atención en la comunicación llevada a cabo en cada nivel: (ningenia, 2015)

- **Nivel de proceso:** este es el nivel más bajo y está compuesto por sensores y actuadores que interactúan directamente con el proceso físico. Los sensores recopilan datos del proceso y los actuadores ejecutan las órdenes de los controladores. La comunicación es estrictamente en tiempo real, utilizando protocolos como AS-i, DeviceNet y Profibus-DP.
- **Nivel de campo:** en este nivel se encuentran los PLCs, robots y sistemas de control numérico que gestionan los sensores y actuadores del nivel inferior. La comunicación en este nivel se realiza mediante redes de control como EtherNet/IP, Profinet y Profibus, permitiendo la realización de trabajos productivos de manera autónoma.
- **Nivel de célula:** recibe las órdenes de los niveles superiores y las descompone en operaciones más simples. Aquí es fundamental contar con una comunicación robusta, determinista y fiable, adecuada para ambientes industriales hostiles. Se utilizan redes de Ethernet Industrial, como Modbus TCP, EtherNet/IP, Profinet y EtherCAT, que están diseñadas para operar en condiciones adversas y asegurar la transmisión en tiempo real.
- **Nivel de planta:** en este nivel se ubican los sistemas de control y supervisión, como HMI y SCADA. Estos sistemas permiten la monitorización en tiempo real de los procesos productivos. La comunicación se realiza mediante redes Ethernet TCP/IP, facilitando la interacción con los elementos de supervisión.
- **Nivel de fábrica:** es nivel más alto y se centra en la gestión global de la empresa. Aquí se utilizan ordenadores y servidores que manejan aplicaciones como ERP (Enterprise Resource Planning), MES (Manufacturing Execution Systems), y CAD/CAM/CAE (Computer-Aided Design/Manufacturing/Engineering). La comunicación se realiza a través de redes Ethernet estándar (Ethernet TCP/IP), permitiendo la transmisión de grandes volúmenes de datos entre los distintos niveles.

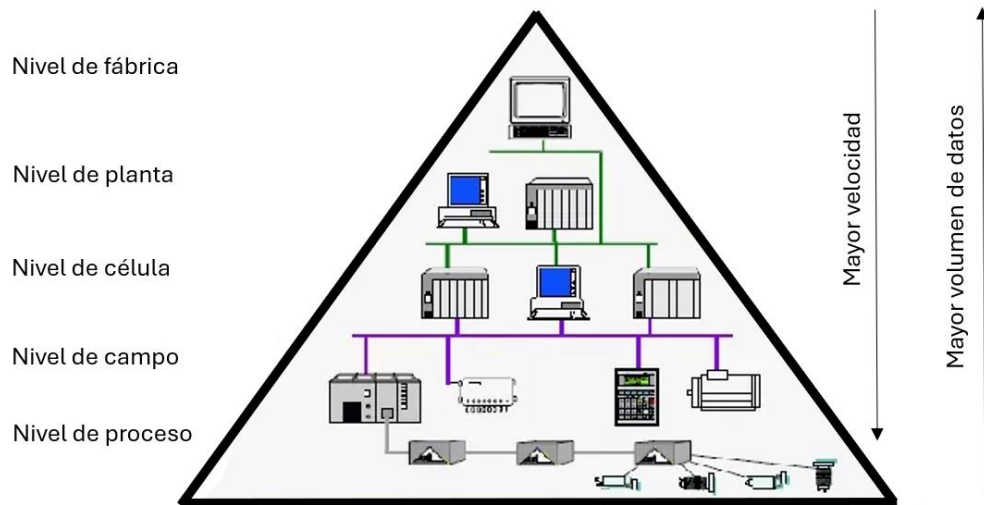


FIGURA 34. Pirámide CIM. (Diseñado por el autor)

A medida que se asciende en la pirámide CIM, la velocidad de transmisión de datos disminuye, mientras que el volumen de datos aumenta, aspecto para tener en cuenta en el diseño de la planta.

Capítulo 2.5.4. Topologías de red industrial

La topología de red determina cómo se conectan los diferentes dispositivos dentro de una red industrial, afectando directamente la comunicación y el control en una planta de producción. (Sicma21, 2021)

La topología de red se divide en:

- **Topología física:** determina la colocación, ubicación e instalación de los componentes en la red.
- **Topología lógica:** determina la circulación de los datos en la red independientemente del diseño físico.

Ambas topologías pueden ser iguales o diferentes dentro de una misma red.

Las principales topologías usadas en la industria son:

- **Topología punto a punto:** es la forma más básica de conexión, donde dos dispositivos se comunican directamente. Es adecuada para configuraciones simples y pequeñas. No obstante, su principal desventaja es la escalabilidad, ya que agregar más dispositivos requiere nuevas conexiones individuales.



FIGURA 35. Topología punto a punto. (Sicma21, 2021)

- **Topología de bus:** en esta configuración, todos los dispositivos se conectan a un único cable principal, formando una línea de comunicación compartida. Es común en sistemas de buses de campo como Profibus. Una desventaja significativa es que si el cable principal falla, todos los dispositivos conectados también fallarán.

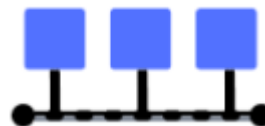


FIGURA 36. Topología en bus. (Sicma21, 2021)

- **Topología en estrella:** esta topología utiliza un punto central de conexión, como un hub o switch, al que todos los dispositivos se conectan individualmente. Ofrece la ventaja de aislar fallos, ya que un problema en un cable solo afecta al dispositivo conectado a ese cable. Sin embargo, requiere más cableado en comparación con una topología de bus.

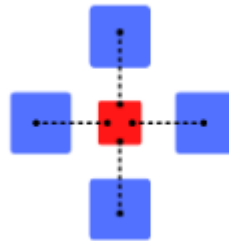


FIGURA 37. Topología en estrella. (Sicma21, 2021)

- **Topología en árbol:** similar a la topología en estrella, pero con varios niveles de conexión. Se utiliza un punto de distribución central conectado a varios dispositivos, que a su vez pueden actuar como puntos de distribución para otros dispositivos. Es común en redes Ethernet de oficinas y permite una expansión modular de la red.

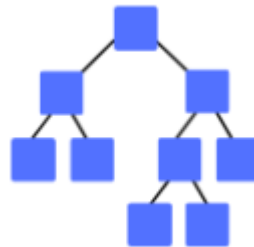


FIGURA 38. Topología en árbol. (Sicma21, 2021)

- **Topología en anillo:** en esta configuración, cada dispositivo está conectado a otros dos dispositivos, formando un círculo cerrado. Los datos se transmiten en una dirección (o en ambas, en un anillo redundante) alrededor del anillo hasta alcanzar su destino. Una ventaja de esta topología es su capacidad de recuperación en caso de fallo, ya que, si se interrumpe una conexión, los datos pueden seguir transmitiéndose en la dirección opuesta. Sin embargo, puede ser más compleja de configurar y mantener

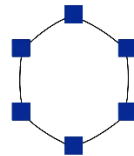


FIGURA 39. Topología en anillo. (Diseñado por el autor)

Capítulo 2.5.5. Protocolos de comunicación industrial

Un protocolo es un conjunto de reglas que facilitan la comunicación entre dispositivos en una red. También aseguran la interoperabilidad, la fiabilidad y la eficiencia en la transmisión de datos entre diferentes componentes de una red industrial. Algunos de los protocolos más utilizados en la industria incluyen: (Logicbus, 2019)

- **Modbus:** es uno de los protocolos más antiguos y sigue siendo muy utilizado debido a su simplicidad y robustez. Usado para establecer una comunicación cliente-servidor entre los dispositivos. Funciona tanto en redes serie (Modbus RTU) como en Ethernet (Modbus TCP/IP). Es ideal para aplicaciones donde la simplicidad y la interoperabilidad son cruciales
- **Ethernet/IP (Protocolo industrial Ethernet):** utiliza la tecnología Ethernet estándar. Uno de sus principales objetivos es usar los protocolos de transporte (TCP), Internet (IP) y las tecnologías de acceso y señalización de medios albergados en las tarjetas de interfaz de Ethernet. Conocido por soportar tanto la mensajería explícita como implícita, lo que lo hace versátil para diversas necesidades de automatización.
- **PROFINET:** basado en Ethernet y es ampliamente utilizado en la automatización industrial. Permite la comunicación en tiempo real y es conocido por su flexibilidad y alta precisión en la transmisión de datos. Este protocolo soporta varias configuraciones como por ejemplo

PROFINET/IO y PROFINET/IRT para aplicaciones de tiempo real isócrono.

- **PROFIBUS:** es un estándar abierto que permite altas velocidades de transmisión y es duradero en ambientes industriales hostiles. Tiene varias versiones como PROFIBUS-DP para comunicación rápida entre controladores y dispositivos de campo, PROFIBUS-PA para entornos con peligro de explosión y PROFIBUS-FMS para grandes volúmenes de información a nivel de célula.
- **ETHERCAT:** es un protocolo de alto rendimiento basado en Ethernet, utilizado para aplicaciones que requieren velocidades de comunicación extremadamente rápidas.
- **DeviceNet:** utiliza la tecnología CAN (Controller Area Network) para facilitar la comunicación en tiempo real entre dispositivos industriales como sensores y actuadores. Su arquitectura maestro-esclavo permite que un controlador maestro gestione varios dispositivos esclavos, proporcionando una red fiable y robusta para entornos industriales exigentes. Ampliamente utilizado debido a su facilidad de implementación y capacidad para integrar dispositivos de diferentes fabricantes.
- **CANopen:** es un protocolo de comunicación también basado en la tecnología CAN, diseñado para aplicaciones de automatización industrial. Se distingue por su eficiencia en la transmisión de datos en tiempo real y su capacidad para configuraciones de control distribuido. Facilita la interoperabilidad entre dispositivos de distintos fabricantes, promoviéndolo para sistemas complejos y flexibles que requieren una alta adaptabilidad y gestión distribuida de dispositivos en la red.
- **MQTT:** es un protocolo de red abierto y ligero de mensajería diseñado para aplicaciones donde el ancho de banda es limitado. Es ampliamente utilizado en entornos de Internet de las Cosas (IoT) industrial.
- **OPC-UA:** es una versión avanzada del protocolo estándar OPC, diseñada para ser independiente del sistema operativo y permitir el intercambio seguro y confiable de datos entre sistemas operativos diversos. OPC UA es ampliamente utilizado en la automatización industrial para facilitar la interoperabilidad entre diversos sistemas y dispositivos, proporcionando una plataforma unificada para el intercambio de datos.
- **S7 Communication:** es un protocolo de comunicación utilizado para conectar controladores programables (PLCs) en sistemas de automatización industrial. Desarrollado específicamente para las familias de PLCs de Siemens, como las series SIMATIC S7, facilita la transferencia de datos entre dispositivos, permitiendo controlar y monitorear procesos industriales de manera eficiente y segura. Funciona principalmente a través de Ethernet y se adapta bien a



diferentes tamaños de sistemas. Además, el protocolo S7 soporta mejoras en seguridad para proteger las comunicaciones en entornos industriales.

Los protocolos son fundamentales para garantizar la interoperabilidad y eficacia, abordando desafíos como la compatibilidad, la latencia, la seguridad y la escalabilidad.



CAPÍTULO 3. Estado del arte

Capítulo 3.1. Antecedentes

El presente Trabajo de Fin de Grado se centra en el diseño y desarrollo de una estación de paletizado y clasificación, integrando diversas tecnologías emergentes en un solo sistema. La Universidad de Valladolid cuenta con un sólido historial de investigación y aplicación en este campo, habiéndose realizado trabajos previos que combinan algunas de las tecnologías utilizadas en este proyecto. Ejemplos de trabajos anteriores realizados por estudiantes de la universidad son fundamentales para un correcto enfoque, tales como:

- En 2016, **Rocío Pérez Fernández** llevó a cabo la simulación de un proceso industrial usando RobotStudio de ABB, donde integró la programación de un PLC y una HMI utilizando Step 7 y TIA Portal. Esta integración proporciona un precedente valioso para este proyecto, que busca desarrollar una HMI que facilite el manejo del sistema a través de comandos de voz. (Fernández R. P., 2016)
- Durante ese mismo año, **Álvaro Galindo de Santos** destacó por desarrollar una comunicación remota entre una aplicación Android y un robot industrial para jugar al “tres en raya”, empleando sockets para facilitar el control y la interacción lúdica, lo que destaca la importancia de la interactividad y la conectividad en los sistemas automatizados. (Santos, 2016)

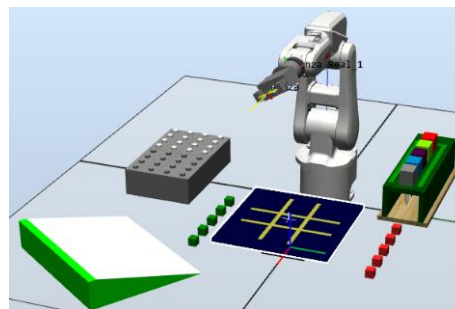


FIGURA 40. Estación robotizada del TFG de Álvaro Galindo de Santos. (Santos, 2016)

- **Daniel García Fernández**, en 2017 realizó su trabajo final de máster sobre modelado y simulación de una planta real de fabricación y logística, utilizando tecnología OPC para la comunicación entre dispositivos. Este trabajo resalta la importancia de la replicación de entornos de trabajo reales para optimizar los procesos y las comunicaciones entre equipos automatizados. (Fernández D. G., 2017)
- **José Delgado García**, en su proyecto de 2020, estableció una célula robotizada que integraba tecnología de Fanuc y Siemens, diseñada tanto para la educación como para la implementación práctica que llevo a cabo. Este proyecto refleja la flexibilidad de los sistemas automatizados para adaptarse a diferentes entornos y propósitos, siendo un ejemplo valioso para considerar en el diseño de interfaces y en la programación de automatismos. (García J. D., 2020)
- En 2021, **Rodrigo Sancho García** utilizó Autodesk Inventor para el diseño de una estación de trabajo robótica, integrando un sistema de visión artificial para la realización de tareas multitarea. La coordinación de la célula se realizó a través del protocolo OPC UA, optimizando la interoperabilidad entre RobotStudio y MATLAB. Este proyecto subraya la importancia de combinar diseño mecánico avanzado, visión artificial y comunicaciones robustas para desarrollar sistemas robotizados complejos y eficientes, aspectos clave estrechamente relacionados con este proyecto. (García R. S., 2021)

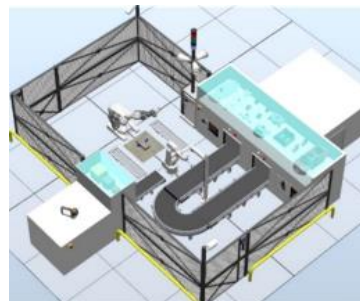


FIGURA 41. Célula de trabajo correspondiente al TFG de Rodrigo Sancho García. (García R. S., 2021)

Estos proyectos previos no solo muestran la evolución de las técnicas de simulación, control y operación en entornos robotizados, sino que también destacan la importancia de una interfaz de usuario intuitiva y comunicaciones efectivas.



Capítulo 3.2. Crítica al estado del arte

En la exploración de proyectos previos en la Universidad de Valladolid, el uso de tecnologías de robótica y automatización como RobotStudio de ABB, TIA Portal de Siemens y Matlab es notable. A pesar de su eficacia en operaciones generales, enfrentan limitaciones considerables en áreas específicas, incluyendo la integración de sistemas de visión artificial y la funcionalidad de reconocimiento de voz, esenciales para modernizar las estaciones robóticas y automáticas.

La adaptabilidad de las Interfaces Hombre-Máquina (HMI) actualmente utilizadas es insuficiente, particularmente en entornos que podrían beneficiarse significativamente del control por voz. Esta tecnología promete simplificar las operaciones permitiendo a los operarios ejecutar comandos de manera intuitiva y eficiente, reduciendo la curva de aprendizaje y los errores asociados con interfaces más complejas.

Además, la falta de interoperabilidad entre los sistemas de control y las plataformas de programación limita la capacidad para realizar ajustes en tiempo real basados en la visión artificial, crucial para la identificación precisa de objetos defectuosos o mal clasificados.

Estas deficiencias no solo complican la gestión diaria, sino que también obstaculizan la escalabilidad de las soluciones y limitan la capacidad de adaptarse a nuevos desafíos del mercado. Aumentar la eficacia de la integración de estas tecnologías y expandir las funcionalidades de las HMI para incluir control por voz son pasos cruciales hacia la optimización de la productividad y la eficiencia operativa en la industria actual.

Capítulo 3.3. Solución implementada

En respuesta a las limitaciones identificadas en el análisis del problema, se ha diseñado e implementado una estación de paletizado y clasificación que integra una gran variedad de tecnologías avanzadas. Esta solución, desarrollada con un enfoque multidisciplinario y utilizando software de última generación, no solo busca mejorar la eficiencia, precisión y facilidad de uso, sino también demostrar la viabilidad de distintas alternativas tecnológicas, rompiendo con la perspectiva tradicional. Este enfoque ayuda a proporcionar un marco referencial amplio para futuros desarrollos, fomentando un entendimiento más profundo de cómo implementar de manera efectiva estas tecnologías y adaptarlas a diversas necesidades operativas.

Como se ha venido haciendo a lo largo del informe se dividirá el problema en distintos apartados:

Estación robótica

RobotStudio de ABB ha sido seleccionado como la plataforma principal para el diseño y desarrollo de la estación robótica. Este software es altamente valorado por su capacidad para modelar y probar de manera exhaustiva el sistema en un entorno virtual antes de su despliegue en un entorno real. La experiencia previa con RobotStudio ha facilitado la elección de este software.

En cuanto al diseño de la estación, se incorporarán diversos elementos proporcionados por RobotStudio, como cintas transportadoras, dispensadores de palés y barreras de seguridad. Además, se realizará un diseño específico en Autodesk Inventor para un dispensador de planchas, obra del autor del proyecto.

Los dos robots elegidos serán del modelo IRB 4600-60/2.55 de ABB, seleccionados por su adecuación a estas aplicaciones de paletizado.

Para el desarrollo del sistema de visión artificial dentro de la simulación de RobotStudio, se realizará un componente inteligente encargado de tomar capturas cuando las cámaras hayan sido activadas.

Interfaz HMI y autómatas

Para la Interfaz Hombre-Máquina (HMI) de este proyecto, se ha elegido TIA Portal, reconocido en la industria de la automatización por su capacidad para desarrollar interfaces intuitivas y adaptativas. La HMI implementada será mejorada con una innovadora función de control por voz, desarrollada a través de un programa creado en Python. Esta capacidad amplía notablemente la accesibilidad y simplifica la operatividad de la estación, permitiendo realizar ajustes rápidos y efectivos que mejoran tanto la agilidad de las operaciones como la respuesta del sistema en tiempo real.

Para este proyecto se ha elegido el autómatas 1214C DC/DC/DC, debido a su costo asequible y tamaño compacto. Además, la familiaridad del autor con este modelo facilita su programación y su implementación, lo que es crucial para la viabilidad del proyecto en caso de querer llevarlo a cabo en un entorno real en el futuro.

En cuanto a la interfaz hombre-máquina, se ha optado por la HMI KTP1 1200 Basic PN, elegido por su compatibilidad con el autómatas seleccionado y por sus grandes dimensiones. Esta característica es fundamental para asegurar que la interfaz sea fácil de usar, con controles intuitivos y una presentación clara que los operarios puedan manejar sin dificultad, mejorando la eficiencia y la accesibilidad en el uso diario de la estación.

En la HMI, el reconocimiento de voz se ha implementará utilizando la biblioteca “speech_recognition” de Python para capturar comandos verbales. Adicionalmente, se emplea la biblioteca “pyttsx3” para convertir texto a voz, facilitando así la interacción directa del sistema con el usuario pudiendo mantener una pequeña conversación.

Sistema de visión artificial

El programa de visión artificial se ha desarrollado utilizando Python, debido a su destacada capacidad y versatilidad en este campo. Python es ampliamente reconocido por su amplia biblioteca de herramientas especializadas en procesamiento de imágenes y aprendizaje automático, lo que lo convierte en la elección ideal para aplicaciones de visión artificial.

Las imágenes capturadas por el componente inteligente en RobotStudio se almacenan automáticamente en una carpeta específica del ordenador. Este directorio es monitoreado de forma continua por el programa de visión artificial desarrollado en Python, utilizando las bibliotecas “watchdog.observers” y “watchdog.events”. Este monitoreo constante permite que cualquier nueva imagen guardada sea rápidamente detectada para posteriormente procesarla utilizando la librería “OpenCV”, optimizando así el flujo de trabajo para el análisis y tratamiento de imágenes en tiempo real.

Comunicación entre programas

La comunicación dentro del sistema está cuidadosamente estructurada para garantizar eficiencia y fiabilidad. La interfaz hombre-máquina (HMI) y el autómata interactúan utilizando el protocolo Profinet, destacado por su rapidez y fiabilidad en entornos industriales. Por su parte, la conexión entre el autómata con el programa de reconocimiento de voz y con RobotStudio, se establece mediante la tarjeta de red del ordenador, utilizando protocolo TCP/IP gracias al software NetToPLCSim, lo que asegura una transferencia de datos segura y eficiente.

Además, el sistema de visión desarrollado en Python se comunica con RobotStudio a través de sockets para saber cuál de las cámaras ha realizado la foto.

Esta metodología de conexión no solo facilita la integración efectiva de diversas tecnologías y protocolos, sino que también proporciona una base sólida de información valiosa para futuros proyectos. Además, este enfoque promueve la familiarización con una variedad de protocolos y técnicas de comunicación existentes, ampliando la competencia técnica necesaria para gestionar eficazmente diversas infraestructuras de comunicación en entornos industriales.

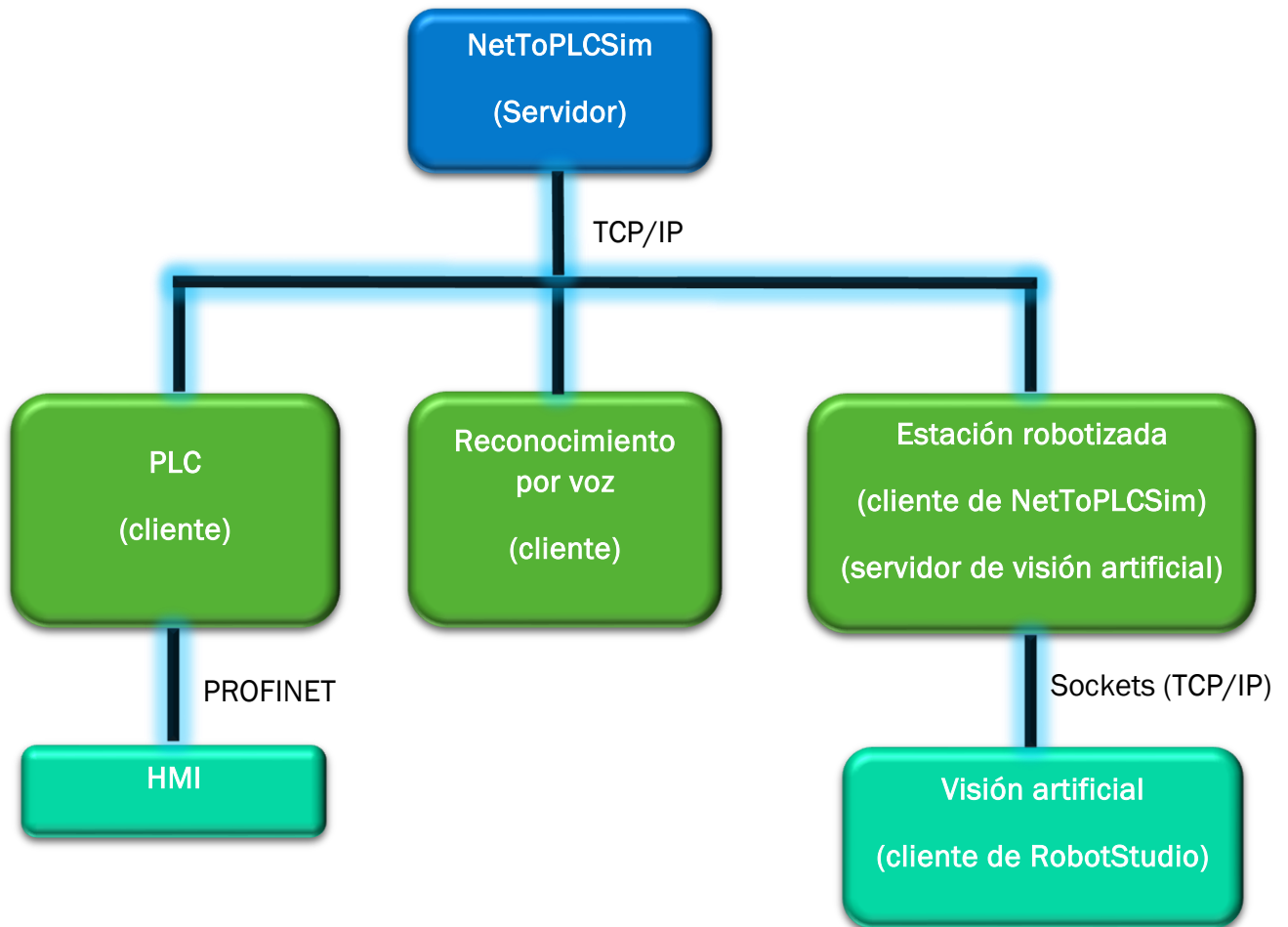


FIGURA 42. Esquema de comunicaciones entre sistemas del proyecto.



CAPÍTULO 4. Herramientas software empleadas

Los programas y herramientas utilizadas para la realización del proyecto, junto con sus respectivas finalidades, se presentan de forma esquemática:

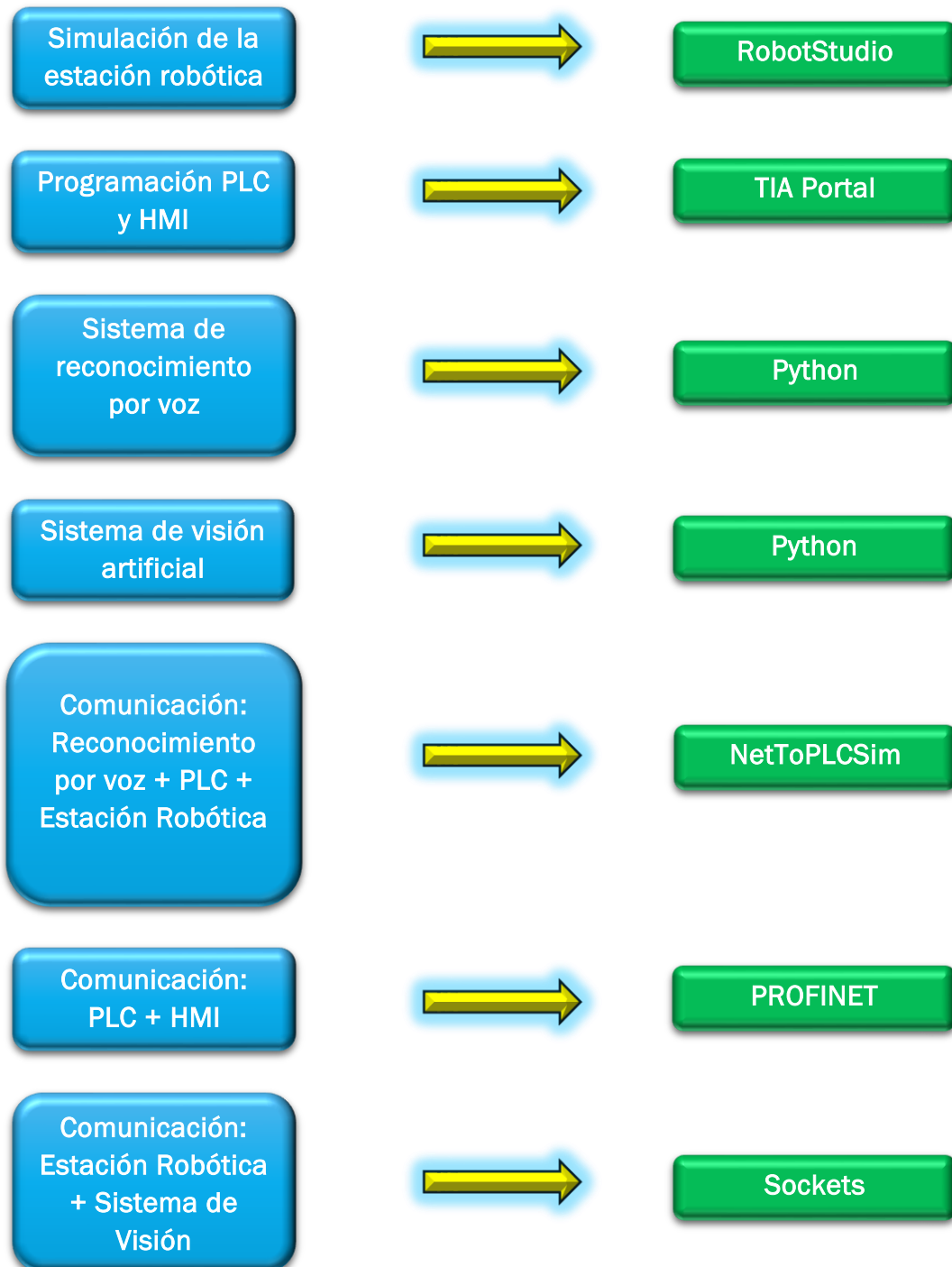


FIGURA 43. Herramientas software utilizadas para cada sistema del proyecto.



Capítulo 4.1. Software de ABB

El desarrollo de este Trabajo de Fin de Grado implica la realización de tareas específicas mediante el uso de robots industriales. Para lograr este objetivo, se requiere un software que permita la simulación y programación avanzada de los movimientos y operaciones de dichos robots.

ABB Robotics se destaca en el sector de la robótica industrial, no sólo por la calidad y versatilidad de sus robots sino también por su robusto portafolio de aplicaciones diseñadas para la simulación, programación y gestión de operaciones robóticas. Este portafolio incluye diversas soluciones como RobotWare, usado por los controladores de los robots para gestionar y ejecutar los movimientos, y RobotStudio, para la simulación general de la estación. Ambas herramientas son esenciales para el diseño y ejecución de proyectos en un entorno simulado.

En este proyecto se utilizará la versión 2023 de RobotStudio y la versión 6.15.03 de RobotWare.

El reconocimiento global de ABB Robotics y su posicionamiento como uno de los líderes en el mercado de la robótica industrial, justifica la elección de esta empresa como proveedora de software para el desarrollo de las aplicaciones robóticas en este proyecto.

Capítulo 4.1.1. RobotStudio

RobotStudio es una aplicación de PC diseñada para el modelado, programación fuera de línea y simulación de células robotizadas. Este software facilita la interacción tanto con un controlador IRC5 virtual, que funciona en la propia PC del usuario, como con un controlador IRC5 físico. El controlador virtual, conocido también como VC (controlador virtual), permite ejecutar simulaciones en un entorno virtual sin la necesidad de equipamiento físico. En contraste, la conexión con un controlador real activa el modo online de RobotStudio. (Robotics, Manual del operador RobotStudio, 2012)

El modo online de RobotStudio se utiliza combinándolo con controladores reales, ofreciendo integración con procesos operativos activos. Por otro lado, el modo offline se emplea cuando no hay conexión a un controlador físico o cuando se utiliza un controlador virtual, brindando un espacio seguro para probar y desarrollar sin riesgos, como se hará en este proyecto.

RobotStudio está disponible en diferentes modalidades de instalación para adecuarse a las necesidades de cada usuario:

- **Completa:** abarca todas las características disponibles.
- **Personalizada:** permite a los usuarios elegir los componentes específicos y establecer configuraciones de rutas a medida.
- **Mínima:** orientada exclusivamente para uso en modo online, perfecta para quienes necesitan interacción constante con controladores físicos.

Al iniciar RobotStudio, la interfaz de usuario se presenta de la siguiente manera:

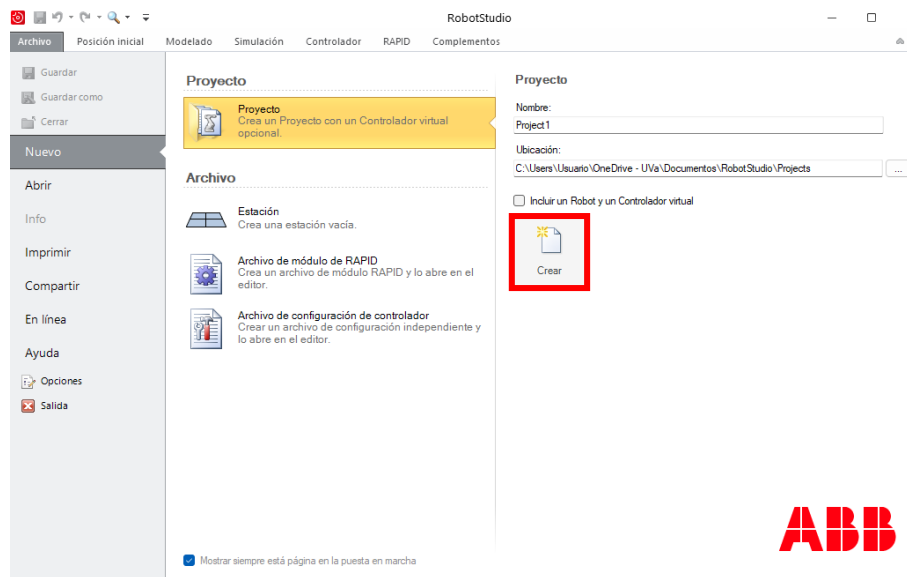


FIGURA 44. Interfaz al iniciar RobotStudio 2023.

Sin embargo, entorno de trabajo de RobotStudio se muestra en la figura 45.

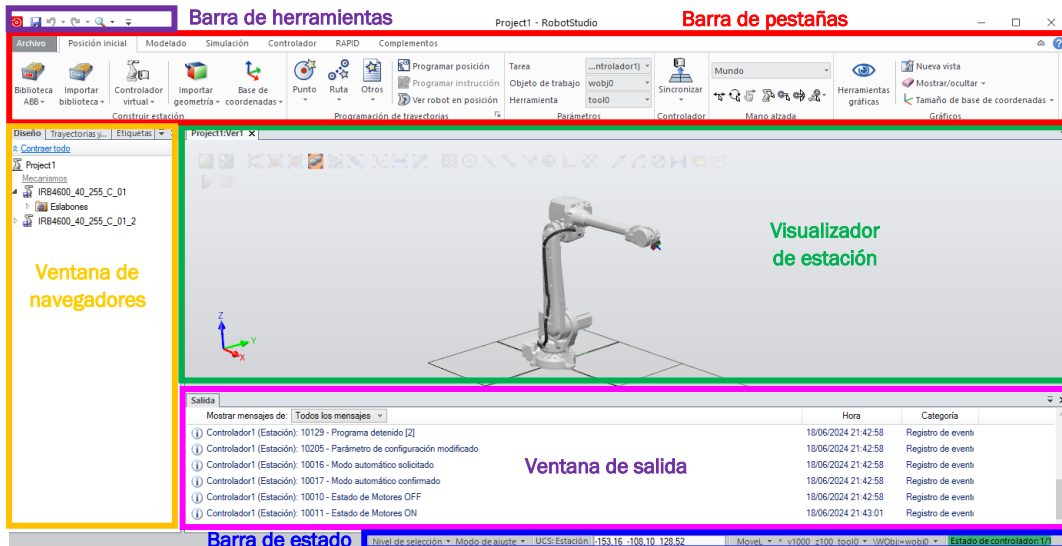


FIGURA 45. Entorno de trabajo en RobotStudio 2023.

Donde se pueden distinguir distintas áreas:

- **Barra de herramientas (púrpura):** ofrece acceso rápido a un conjunto de funciones y herramientas fundamentales, como abrir, guardar, y cerrar proyectos, realizar búsquedas rápidas de acciones en RobotStudio, configurar el diseño de la interfaz, entre otras opciones.
- **Barra de pestañas (rojo):** sirve para organizar las distintas acciones proporcionadas por RobotStudio. Para cada pestaña las acciones disponibles son distintas:
 - **Archivo:** incluye funciones para configurar una nueva estación, crear un sistema robótico, conectar con un controlador, guardar la estación en diversos formatos, etc.
 - **Posición inicial:** incorpora controles imprescindibles para diseñar estaciones, desarrollar sistemas, programar trayectorias y posicionar componentes.

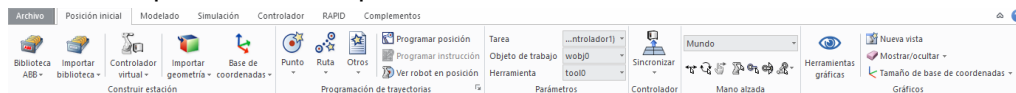


FIGURA 46. Pestaña Posición inicial en RobotStudio 2023.

- **Modelado:** incluye funciones necesarias para crear y organizar componentes, crear cuerpos, realizar mediciones, ejecutar operaciones de CAD, etc.

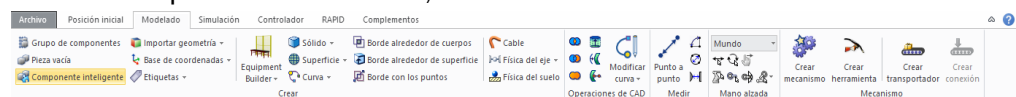


FIGURA 47. Pestaña Posición inicial en RobotStudio 2023.

- **Simulación:** dispone de los controles para crear, configurar, controlar, supervisar, registrar, etc. simulaciones.

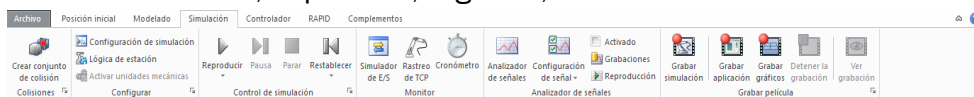


FIGURA 48. Pestaña Simulación en RobotStudio 2023.

- **Controlador:** incorpora acciones para la sincronización, configuración y asignación de tareas al controlador virtual, así como para manejar controladores reales.

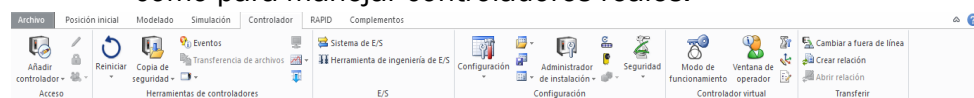


FIGURA 49. Pestaña Controlador en RobotStudio 2023.

- **RAPID:** incluye las funciones relacionadas con el editor RAPID.

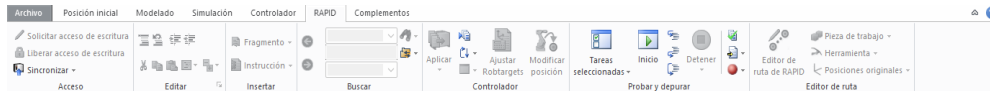


FIGURA 50. Pestaña RAPID en RobotStudio 2023.

- **Complementos:** dispone de módulos y extensiones adicionales que pueden ser instaladas para ampliar las funcionalidades de RobotStudio.

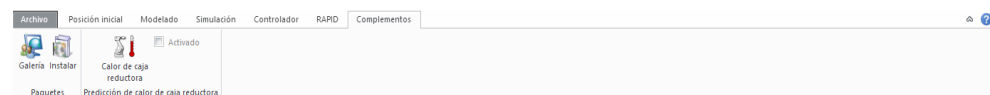


FIGURA 51. Pestaña Complementos en RobotStudio 2023.

- **Ventana de navegadores (amarillo):** se trata de una visualización jerárquica que agrupa elementos como componentes físicos, trayectorias, objetivos, entre otros. Dependiendo de la pestaña activa en la barra de herramientas, se mostrarán distintos navegadores específicos asociados a dicha pestaña.
- **Visualizador de la estación (verde):** es el espacio visual donde se observa el diseño de la estación y se sigue el progreso de los movimientos de los robots a lo largo de la simulación.
- **Ventana de salida (rosa):** muestra los eventos que ocurren dentro de la estación, tales como el inicio o la detención de una simulación. Esta información es valiosa para diagnosticar y resolver problemas en las estaciones. Los eventos se clasifican en tres categorías según su gravedad:
 - Eventos de información.
 - Eventos de aviso.
 - Eventos de error.
- **Barra de estado (azul):** proporciona información en tiempo real sobre diversos aspectos de la operación actual, como el nivel de selección, el modo de ajuste, los valores del sistema de coordenadas, así como el estado del controlador virtual y otros parámetros relevantes.

Para crear un nuevo proyecto, se debe presionar el botón “crear” destacado en rojo en la figura 44. Tras la creación del proyecto, se visualizará una estación vacía, momento en el cual será necesario añadir el robot y el controlador virtual.

El robot se debe seleccionar de uno de los disponibles desde el icono “Biblioteca ABB”, tal y como se ilustra en la figura 52.

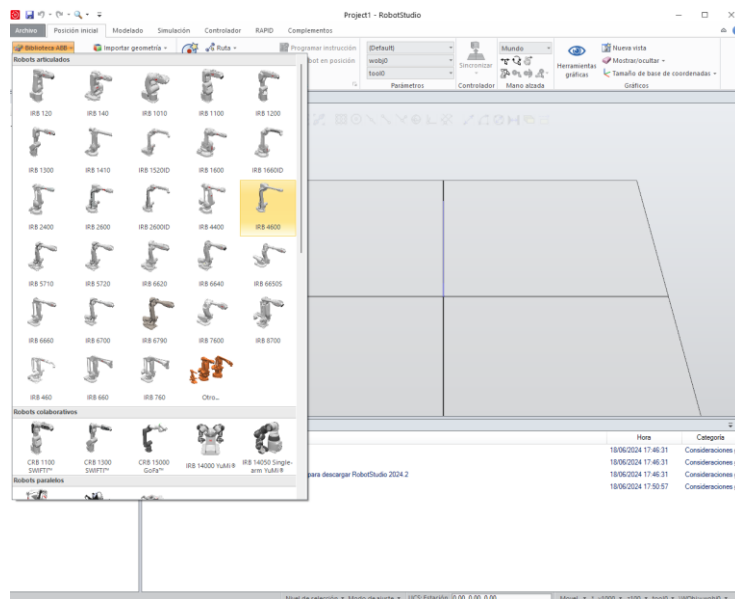


FIGURA 52. Añadir robot a la estación en RobotStudio 2023.

El paso final para incorporar el robot a la estación consiste en seleccionar la versión concreta del robot.

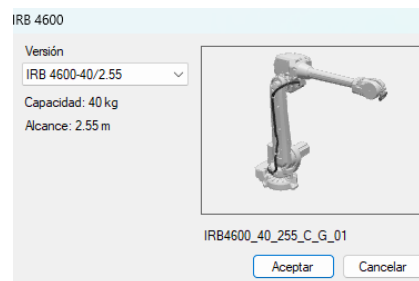


FIGURA 53. Selección de la versión del robot en RobotStudio 2023.

Por otra parte, después de insertar el robot a la estación se debe añadir el controlador virtual siguiendo los siguientes pasos:

1. Hacer clic en “Controlador virtual”.

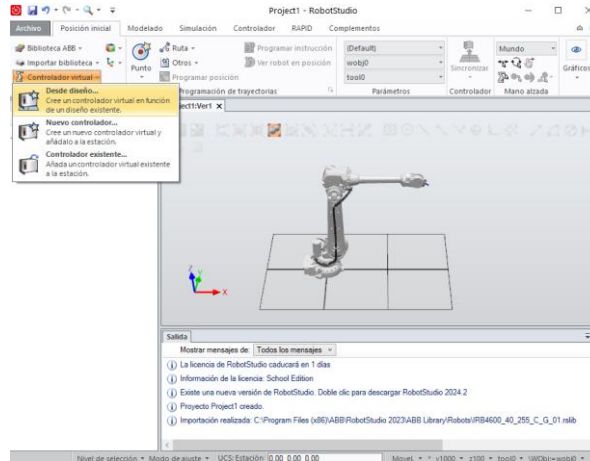


FIGURA 54. Añadir controlador virtual desde diseño en RobotStudio 2023.

2. Asignar un nombre al controlador y seleccionar la ubicación en el ordenador donde se guardará.
3. Elegir los robots que se van a asociar al controlador y elegir la versión de RobotWare deseada, como se indica en la figura 55.

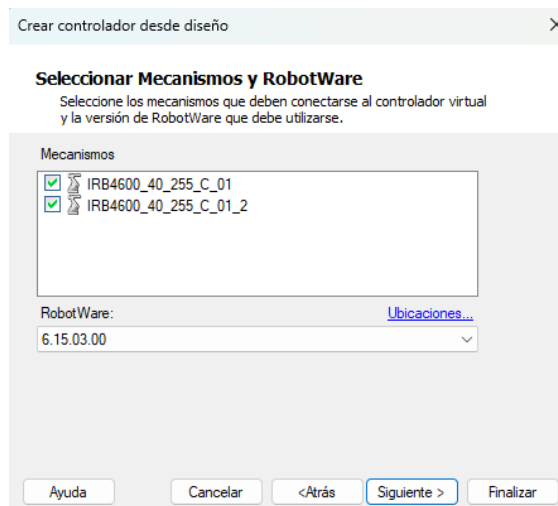


FIGURA 55. Selección de robots y versión de RobotWare para el controlador virtual en RobotStudio 2023.

4. Hacer clic en “Siguiete” mostrará una pantalla donde se pueden configurar diversas opciones del controlador.

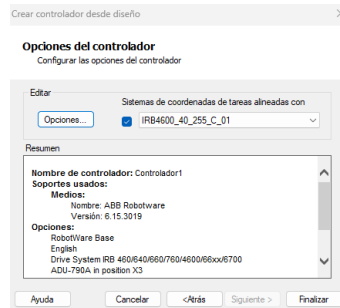


FIGURA 56. Opciones del controlador en RobotStudio 2023.

5. En la sección de opciones, se abrirá una pantalla que permite agregar configuraciones específicas dentro de cada categoría. La configuración del controlador puede variar de un proyecto a otro. En este proyecto, las opciones que se han añadido al controlador son:

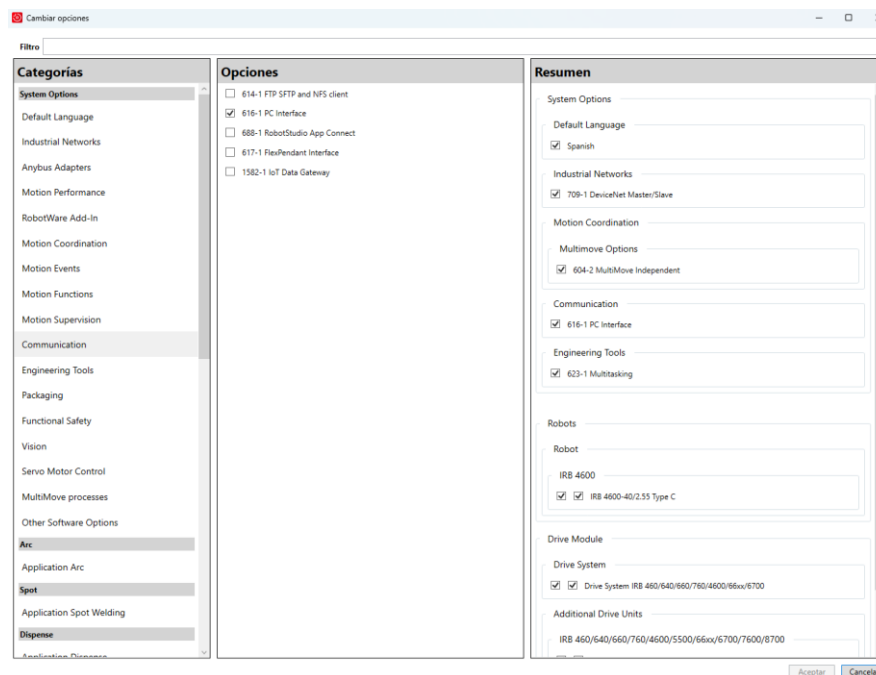


FIGURA 57. Opciones del controlador virtual en RobotStudio 2023.

Una configuración adecuada de las opciones del controlador es esencial para facilitar la programación multitarea, permitir el movimiento independiente de los robots y establecer comunicaciones efectivas con otros programas implicados en el proyecto.

6. Para concluir, se debe hacer clic en el botón "Aceptar" mostrado en la figura 57, lo que cerrará el menú de opciones. Posteriormente, se debe presionar el botón "Finalizar" que aparece en la figura 54 para completar el proceso.

RAPID

RAPID es un lenguaje de programación de alto nivel desarrollado por ABB para la programación de sus robots industriales. Este lenguaje facilita la escritura de programas usando palabras clave en inglés, como IF y FOR, haciéndolo más accesible y fácil de entender para los usuarios en comparación con el lenguaje binario compuesto exclusivamente de ceros y unos.

Con RAPID, se puede organizar el código en tareas y módulos, controlar movimientos del robot, y manejar las entradas y salidas. Además, permite la integración de sensores y la gestión de errores y protocolos de seguridad, haciendo posible la programación de operaciones complejas y la mejora en la interacción del robot con su entorno operativo.

Para iniciar la programación con RAPID, primero se debe acceder a la pestaña "RAPID" en la barra de pestañas. A continuación, se selecciona el módulo de programa correspondiente a la tarea que se desea programar, tal como se ilustra en la figura 58.

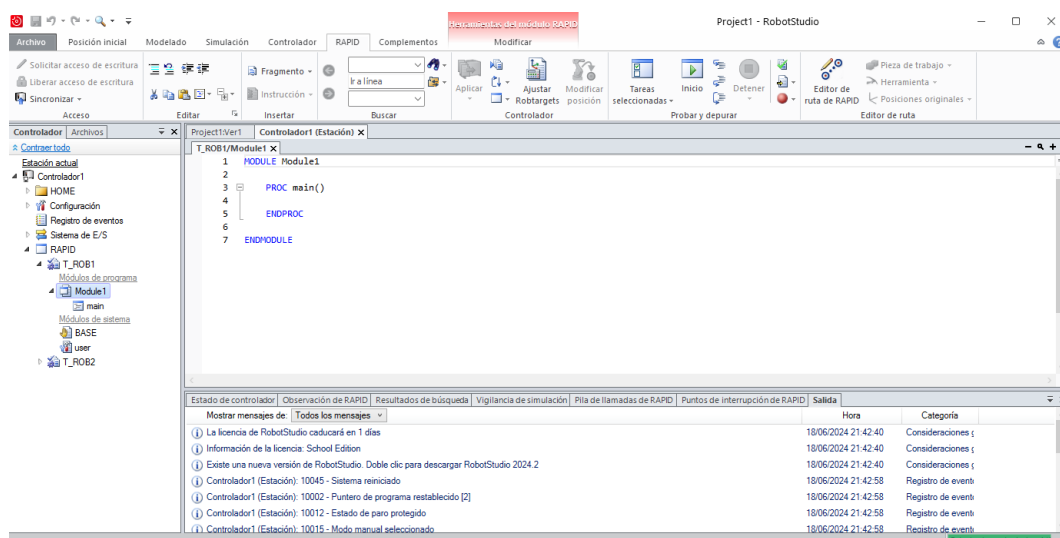


FIGURA 58. Interfaz de RAPID dentro de RobotStudio 2023.

Existen varias instrucciones básicas para controlar el movimiento de los robots, destacando principalmente:

- **MoveJ (Movimiento Articular):** esta instrucción es utilizada para posicionar el robot mediante la manipulación de sus ejes articulares. Se prioriza la velocidad más que la trayectoria. Ideal para alcanzar posiciones que requieren ajustes complejos en la orientación del robot.
- **MoveL (Movimiento Lineal):** permite que el robot se mueva en línea recta hacia la posición deseada.
- **MoveC (Movimiento Circular):** utilizada para movimientos que siguen una trayectoria curva. Está instrucción requiere dos puntos de referencia, un punto intermedio y el punto destino.

Capítulo 4.2. Siemens

En este proyecto, se ha elegido implementar tanto el autómata programable como la Interfaz Hombre-Máquina (HMI) de Siemens. Aunque Siemens opera en diversos sectores tecnológicos, su especialización en automatización industrial la convierte en un proveedor líder para aplicaciones que requieren precisión y eficacia. La integración de su autómata y HMI en la estación de paletizado y clasificación busca aprovechar las capacidades que brinda para optimizar el control y la interacción del sistema.

Capítulo 4.2.1. TIA Portal

TIA Portal (Totally Integrated Automation Portal) de Siemens es una plataforma de ingeniería robusta utilizada ampliamente en la automatización de procesos industriales. Este software permite la programación, gestión y diagnóstico de PLCs, Interfaces Hombre-Máquina (HMI) y otros dispositivos de control. Su diseño integrado soporta la configuración y operación eficiente de múltiples dispositivos en un único entorno de desarrollo, facilitando así la interconexión y sincronización de sistemas complejos.

La versión utilizada para la realización de este proyecto es TIA Portal V16.

Al abrir TIA Portal, se inicia en la vista del portal, la cual proporciona una visión de las herramientas orientada a las tareas. Su objetivo es simplificar la navegación por las tareas y datos del proyecto. Es la pantalla que aparece cuando se inicia el programa, en donde se podrá crear un proyecto desde cero o abrir un proyecto creado.

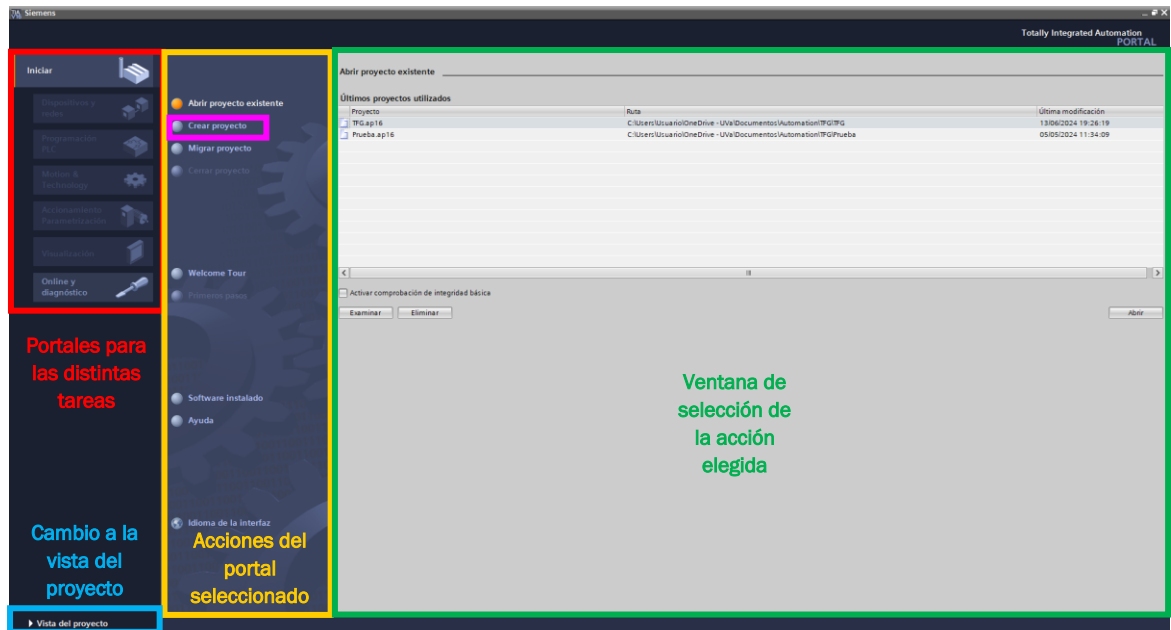


FIGURA 59. Vista del portal en TIA Portal V16.

La vista del portal está dividida en diferentes secciones como se muestra en la figura 59 y a continuación se describen:

- **Portales para las distintas tareas (rojo):** proveen funciones básicas para las diferentes tareas. Los portales disponibles en la vista del portal dependen de los productos instalados.
- **Acciones del portal seleccionado (amarillo):** aparecen opciones que se pueden ejecutar en el portal elegido, las cuales pueden variar dependiendo del portal, sin embargo, el acceso a la Ayuda siempre estará presente.
- **Ventana de selección de la acción elegida (verde):** disponible en todos los portales adaptándose a la selección actual.
- **Cambio a la vista del proyecto (azul):** permite cambiar a la otra vista disponible denominada “Vista del proyecto”.

Para crear un nuevo proyecto se selecciona la acción del portal denominada “crear proyecto” (rosa) en la figura 59.

Una vez asignado el nombre del proyecto, la ruta donde almacenarlo y demás parámetros, se agregará un nuevo dispositivo siguiendo los siguientes pasos:

1. Desde el portal selecciona “Dispositivos y redes”.
2. Selecciona la acción “Agregar dispositivo”.
3. Elige el dispositivo y la versión deseada.
4. Haga clic en el botón “Agregar”.

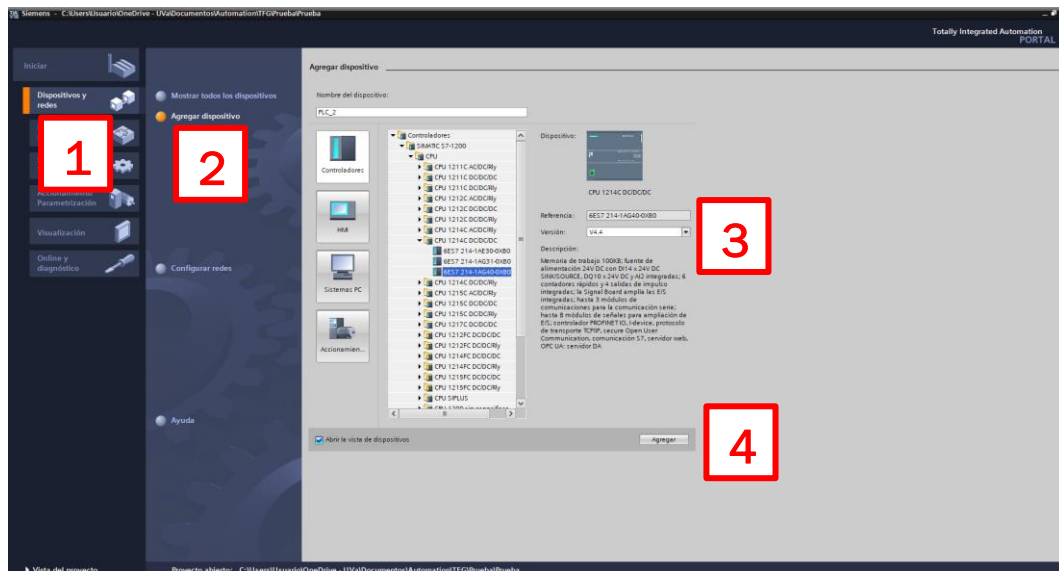


FIGURA 60. Agregar nuevo dispositivo al proyecto en TIA Portal V16.

Este es el procedimiento que hay que seguir para añadir tanto el PLC como la Interfaz Hombre-Máquina (HMI).

Una vez agregado el dispositivo con el que se trabajará, se procederá a seleccionar la otra vista disponible (remarcada en azul en la figura 59), denominada “Vista del proyecto”, desplegándose la siguiente pantalla:

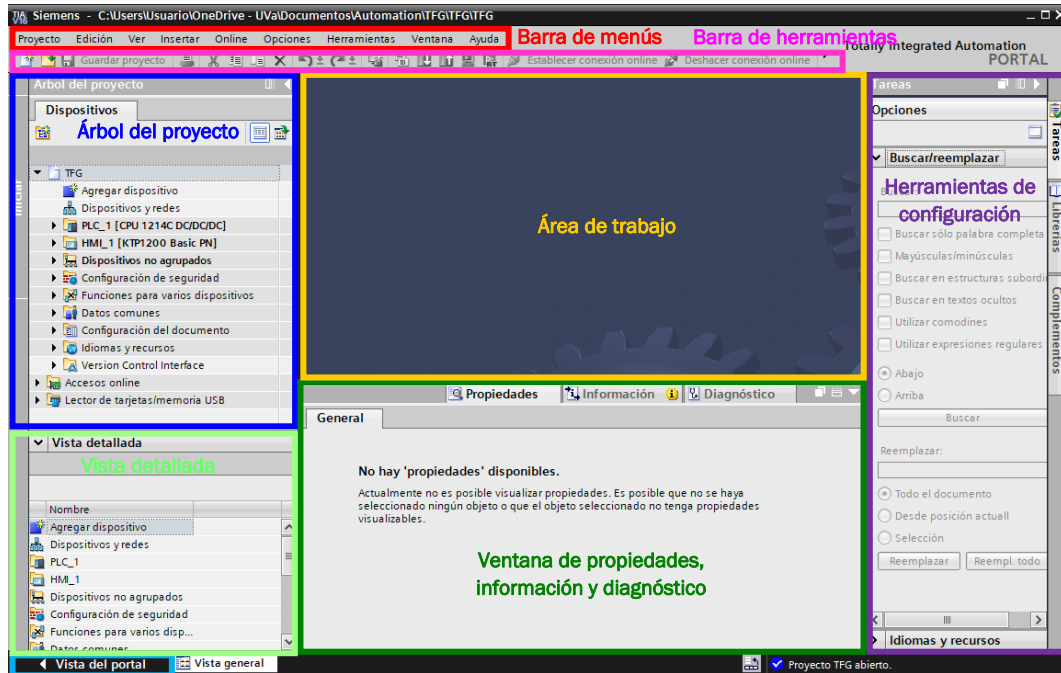


FIGURA 61. Vista del proyecto en TIA Portal V16.

Nuevamente se puede desglosar la vista del proyecto en diferentes secciones:

- **Barra de menús (rojo):** se encuentran los comandos para trabajar con el software.
- **Barra de herramientas (rosa):** alberga los botones de acceso directo a los comandos más utilizados. A su vez, se pueden distinguir principalmente cuatro tipos como se observa en la división realizada en la figura 62:
 1. Funciones relacionadas con el proyecto: crear, abrir, guardar, eliminar, copiar, etc.
 2. Funciones relacionadas con la simulación y el funcionamiento: compilar, cargar en dispositivo, cargar de dispositivo, iniciar simulación, establecer o deshacer conexión online.
 3. Funciones relacionadas con la puesta en marcha o pausa del dispositivo, además de ver los dispositivos accesibles.
 4. Funciones para configurar el área de trabajo.



FIGURA 62. Barra de herramientas en TIA Portal V16.

- **Árbol del proyecto (azul oscuro):** da acceso a todos los componentes y datos del proyecto de forma jerárquica. A través del árbol del proyecto se pueden agregar componentes, editarlos, consultar y modificar las propiedades de los componentes existentes, etc.

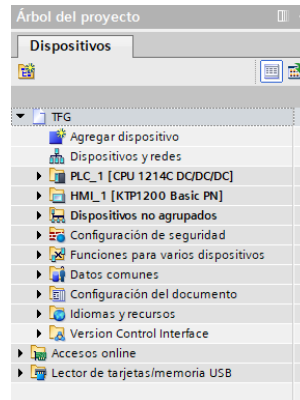


FIGURA 63. Árbol del proyecto en TIA Portal V16.

- **Vista detallada (verde claro):** muestra determinados contenidos del objeto seleccionado. Entre los contenidos posibles están por ejemplo las listas de textos o variables.

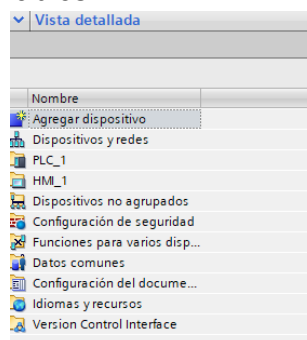


FIGURA 64. Vista detallada en TIA Portal V16.

- **Área de trabajo (amarillo):** es el espacio de trabajo principal, siendo la zona donde se visualizan los objetos y ventanas que se abren para editar los mismos.

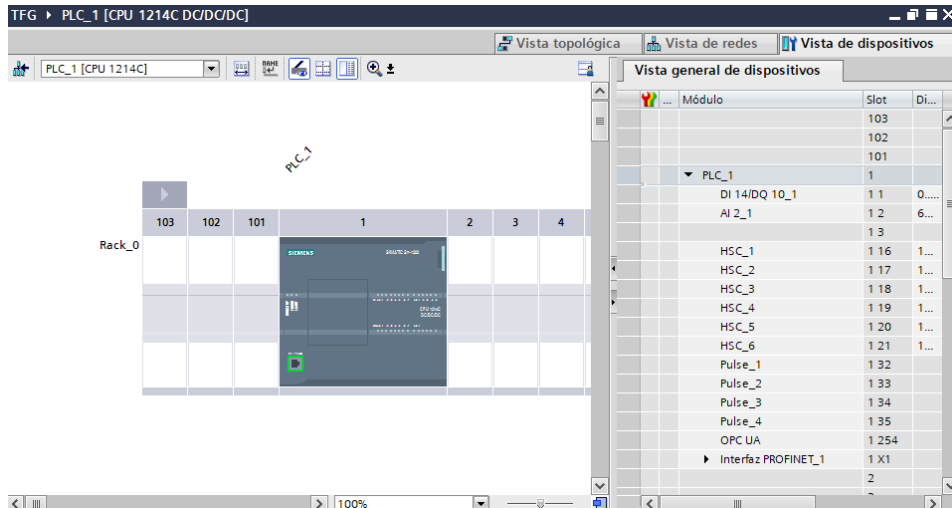


FIGURA 65. Área de trabajo en TIA Portal V16.

- **Herramientas de configuración (púrpura):** es la zona donde se encuentran distintas funciones según el objeto editado o seleccionado. Por ejemplo, en caso de que se esté trabajando con un PLC aparecerán distintas instrucciones, tareas, etc. Por otro lado, si se está trabajando con un HMI aparecerán opciones para, por ejemplo, añadir elementos a la pantalla que se esté modificando.

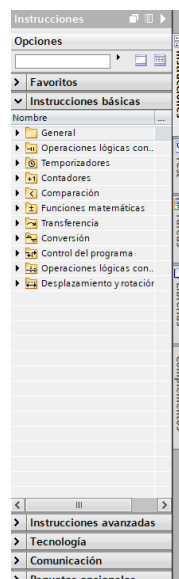


FIGURA 66. Herramienta de configuración en TIA Portal V16.

- **Vista de propiedades, configuración y diagnóstico (verde oscuro):** las tres vistas forman la denominada vista de inspección, en donde se puede visualizar información adicional sobre el objeto seleccionado o sobre las acciones realizadas. Analizándolas por separado:
 - **Propiedades:** facilita la configuración del autómata, del HMI y del Ordenador Industrial.

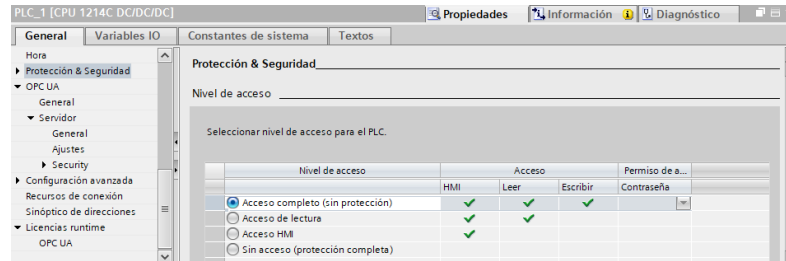


FIGURA 67. Vista de propiedades en TIA Portal V16.

- **Información:** visualiza las alertas y errores que surjan durante la compilación y ejecución del proyecto, datos sobre referencias cruzadas, notificaciones emitidas por TIA Portal, etc.

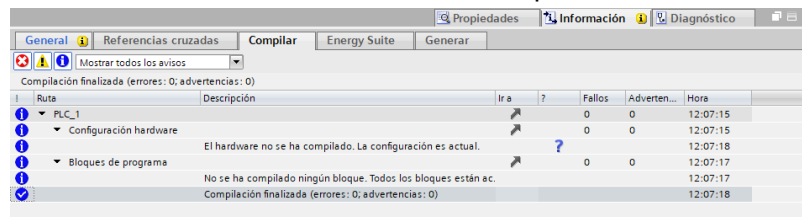


FIGURA 68. Vista de información en TIA Portal V16.

- **Diagnóstico:** exhibe el estado de los dispositivos y de la conexión, además de mostrar distintos avisos.

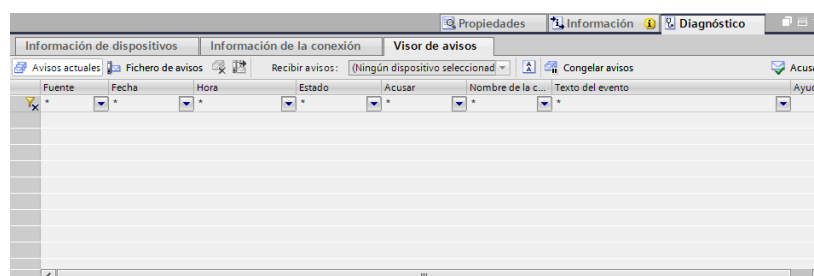


FIGURA 69. Vista de diagnóstico en TIA Portal V16.

- **Cambiar a la vista del portal (azul claro):** permite cambiar a la otra vista disponible denominada "Vista del portal".

Programación

La programación en TIA Portal se realiza principalmente a través de los denominados “Bloques de programa”. Este grupo incluye diversas estructuras disponibles para la programación de usuario, diseñadas para administrar el sistema y organizar las distintas funciones que se desean implementar en el dispositivo. Estos bloques de programa se categorizan en cuatro tipos:

- **Bloque de organización (OB):** sirve como nexo entre el sistema operativo del controlador y el programa de usuario, responsable de iniciar el sistema de automatización, ejecutar el programa de manera cíclica o por alarmas y gestionar errores. Dentro de estos, el OB1 se destaca como el bloque central que, funcionando de manera cíclica, coordina y dirige las llamadas a otros bloques y funciones. Además, se implementan bloques especializados como el OB100 para operaciones de arranque y OB30 para interrupciones cíclicas, que garantizan la ejecución programada y sistemática de tareas dentro del entorno de automatización.
- **Bloque de función (FB):** componente programable que facilita la gestión de funciones complejas que se repiten con frecuencia, como la regulación o la selección de modos. Estos bloques poseen memoria, permitiéndoles almacenar de forma permanente los valores de los parámetros en un bloque de datos de instancia (DB), accesibles en cualquier momento y desde cualquier punto del programa de usuario. Esto hace que los FB sean ideales para aplicaciones que requieren consistencia y persistencia de datos.
- **Función (FC):** bloque programable que opera sin memoria permanente. Utiliza variables temporales, las cuales se almacenan en una pila de datos local que se reinicia una vez que finaliza la ejecución del bloque. Si se necesita conservar datos de manera permanente más allá de una única ejecución, es necesario utilizar bloques de datos globales (DB). Esto hace que los FC sean adecuados para operaciones que no requieren retención de estado entre ejecuciones.
- **Bloque de datos (DB):** componente del programa de usuario diseñado para almacenar valores o cadenas de caracteres. Estos bloques pueden ser globales, accesibles desde todos los bloques lógicos del programa, o de instancia, asignados específicamente a una llamada de FB (bloque de función). Los DBs son cruciales para retener las variables declaradas dentro de los bloques de función, permitiendo su utilización a lo largo de diversas partes del programa y entre ejecuciones distintas.

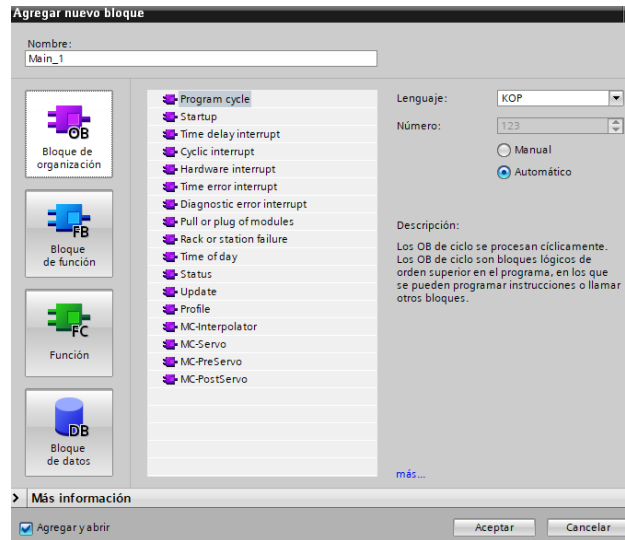


FIGURA 70. Bloques de programa en TIA Portal V16.

Lenguajes de programación

TIA Portal admite varios lenguajes de programación, permitiendo seleccionar el lenguaje adecuado en cada bloque de programa, adaptándose a diferentes necesidades. Entre los más utilizados se encuentran:

- **FUP (Function Block Diagram):** emplea lógica booleana con funciones como OR, AND y NOT, útil para visualizar lógicas agrupadas en bloques.
- **KOP (Ladder Diagram):** conocido como diagrama de contactos o de escalera, utiliza contactos eléctricos en serie y paralelo, facilitando su comprensión para aquellos familiarizados con diagramas eléctricos.
- **AWL (Statement List):** es una lista de instrucciones de bajo nivel que optimiza el tiempo de traducción del controlador.
- **S7-SCL (Structured Control Language):** proporciona sentencias de bucles y condiciones, ideal para cálculos y análisis de información compleja.
- **S7-GRAPH:** es un lenguaje híbrido entre KOP y AWL, ya que crea cadenas de etapas para representar procesos complejos de manera clara.

El lenguaje más utilizado para la programación de autómatas y el que se usará en este proyecto es el lenguaje KOP o Ladder.

A continuación, se presentarán los elementos más comunes en este lenguaje:

NOMBRE	DESCRIPCIÓN	SÍMBOLO
Contacto normalmente abierto	Un contacto normalmente abierto se cierra y conduce corriente cuando la variable asociada es "1". Si la variable es "0", el contacto permanece abierto y bloquea el flujo de corriente.	
Contacto normalmente cerrado	Un contacto normalmente cerrado se abre e interrumpe el flujo de corriente cuando la variable asociada es "1". Si la variable es "0", el contacto permanece cerrado y permite el flujo de corriente.	
Detectar flanco positivo	Un contacto para detectar flanco positivo se activa solo cuando la variable asociada cambia de "0" a "1", generando un pulso de señal en ese momento. Si la variable se mantiene constante, el contacto no reacciona.	
Detectar flanco negativo	Un contacto para detectar flanco negativo se activa solo cuando la variable asociada cambia de "1" a "0", generando un pulso de señal en ese momento. Si la variable se mantiene constante, el contacto no reacciona.	
Invertir resultado lógico	Un contacto para invertir el resultado lógico cambia el estado de la señal entrante: si la variable asociada es "1", el contacto emite "0", y si es "0", emite "1".	
Bobina de asignación	Una bobina de asignación toma el estado lógico de su entrada y lo asigna directamente a la variable de salida. Si la entrada es "1", la salida se establece en "1"; si la entrada es "0", la salida también es "0".	
Bobina para negar asignación	Una bobina para negar asignación invierte el estado lógico de su entrada y asigna ese valor invertido a la variable de salida. Si la entrada es "1", la salida se establece en "0"; si la entrada es "0", la salida se establece en "1".	
Bobina activación tipo set	Una bobina de establecimiento (set) se activa cuando su entrada se vuelve "1", configurando la salida en "1" y manteniendo este estado hasta que se reciba una señal de reseteo. Si la entrada es "0", no afecta el estado actual de la salida.	
Bobina de reseteo tipo reset	Una bobina de reseteo (reset) se activa cuando su entrada se vuelve "1", configurando la salida en "0" y manteniendo este estado hasta que se reciba una señal para cambiarla. Si la entrada es "0", no afecta el estado actual de la salida.	

Tabla 1. Elementos principales del lenguaje KOP.

Tipos de variables

TIA Portal cuenta con diferentes tipos de variables según su identificador, destacando:

- **Entradas (% I):** representan las entradas físicas al PLC. Se utilizan para leer el estado de los dispositivos de entrada como sensores, interruptores, etc. Las variables de entrada son utilizadas por el PLC para recibir datos del entorno externo y generalmente son de solo lectura.
- **Salidas (% Q):** representan las salidas físicas del PLC. Estas variables se utilizan para controlar actuadores como motores, válvulas, luces, entre otros. Las salidas envían señales del PLC a los dispositivos de salida para realizar acciones en el sistema físico.
- **Marcas (% M):** son bits o palabras de memoria interna del PLC que se utilizan para almacenar datos temporales o estados intermedios dentro de un programa. Estas variables no están directamente vinculadas a las entradas o salidas físicas, sino que sirven como almacenamiento interno para la lógica del programa.

Capítulo 4.2.2. PLCSIM

PLCSIM es un simulador de PLC que permite a los usuarios probar y validar programas de PLC sin necesidad de hardware físico. Aunque debe instalarse por separado, se integra con TIA Portal, ofreciendo un entorno virtual para emular las funcionalidades de un PLC. Permite a los desarrolladores simular entradas, salidas y procesamiento lógico, asegurando que los programas funcionen correctamente antes de su implementación en dispositivos reales. Esta herramienta es esencial para identificar y solucionar problemas de programación de manera anticipada.

Una vez creado el proyecto y desarrollado el programa, se procederá a la simulación utilizando PLCSIM para probar el programa. Para ello, se debe seleccionar el icono llamado "Iniciar simulación" en la barra de herramientas, como se muestra en la figura 71.



FIGURA 71. Iniciar simulación en TIA Portal V16.

A continuación, se desplegarán dos pantallas: una para buscar el PLC virtual donde se cargará el programa y la configuración del PLC, como se muestra en la figura 72, y otra que presenta la vista compacta de PLCSIM, como se ilustra en la figura 73.

Antes de realizar la carga se compilará el proyecto y se abrirá la vista preliminar de carga.

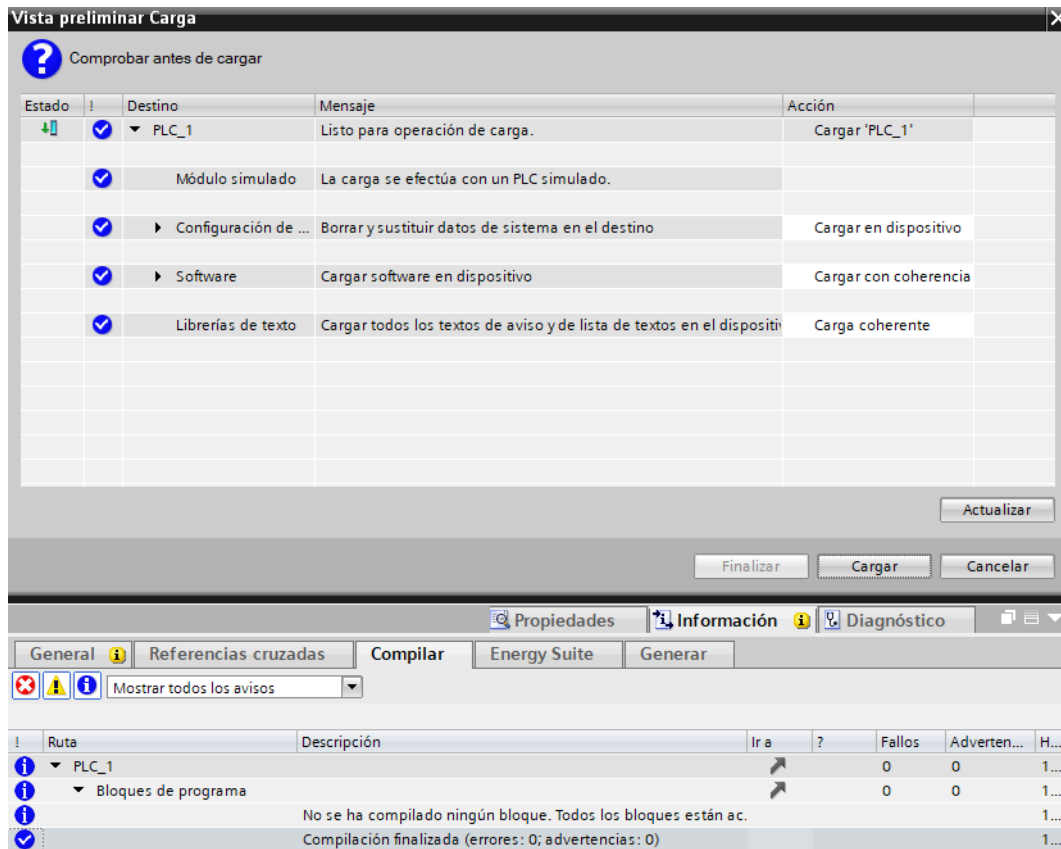


FIGURA 74. Vista preliminar de carga en PLCSIM y vista de compilación satisfactoria.

Finalmente, si la compilación ha sido exitosa, aparecerá una ventana con el resultado de la operación de la carga, desde donde se arrancará el PLC virtual.

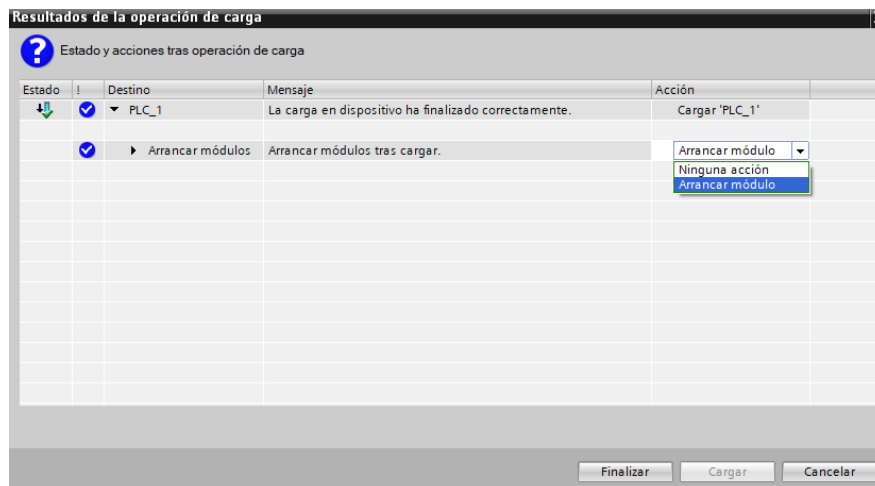


FIGURA 75. Resultado de la operación de carga en el PLC virtual.

En consecuencia, la vista compacta del PCSIM tendrá el siguiente aspecto:



FIGURA 76. Vista compacta PLCSIM con PLC virtual en modo RUN.

Para utilizar todas las funciones que ofrece PLCSIM se debe trabajar con la vista del proyecto, haciendo clic en el botón para cambiar vistas

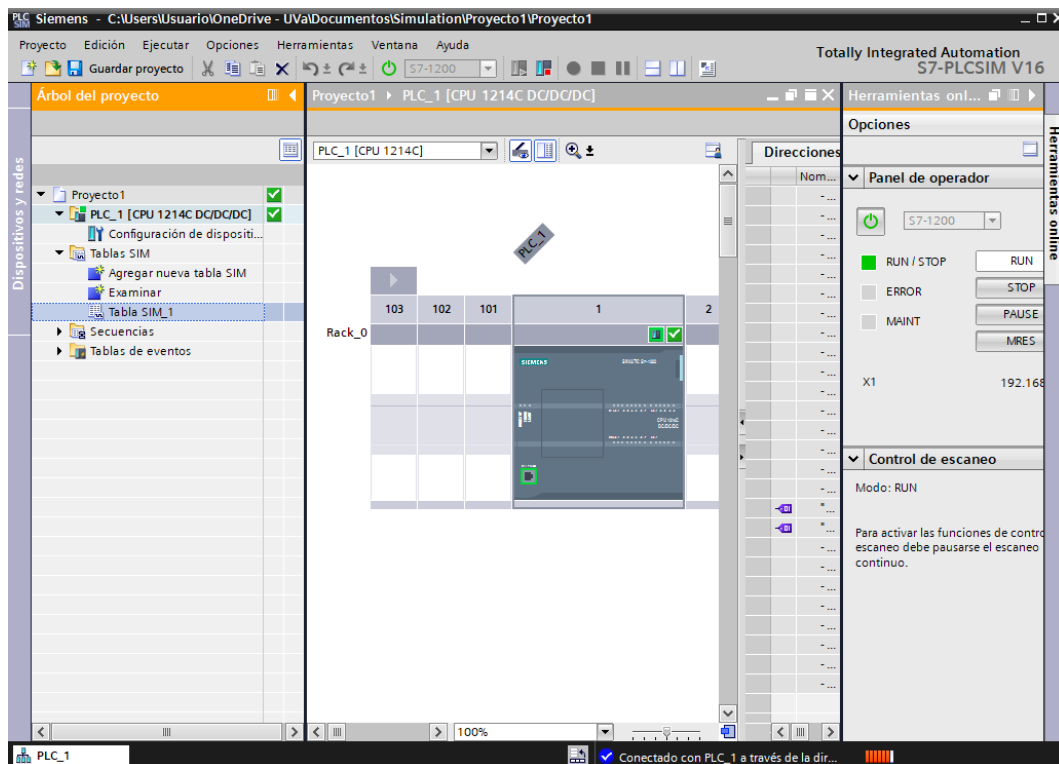


FIGURA 77. Vista del proyecto en PLCSIM.

Desde esta vista extendida, se podrá cambiar el valor de las variables y visualizar su efecto en tiempo real utilizando el simulador virtual PLCSIM, obteniendo los mismos resultados que se obtendrían con un PLC real.

Capítulo 4.2.3. WinCC

WinCC (Windows Control Center) es un software de Siemens utilizado para la supervisión y control de procesos industriales a través de interfaces hombre-máquina (HMI). De la misma manera que PLCSIM es un simulador virtual de un PLC, WinCC es un simulador virtual de un HMI.

WinCC permite el diseño, visualización y operación de sistemas automatizados, ofreciendo herramientas para crear pantallas de control, monitorear datos en tiempo real y gestionar alarmas.

Para utilizar WinCC, primero se debe crear un proyecto y agregar una HMI al mismo. Luego, en el árbol del proyecto, se selecciona la HMI agregada y, al igual que con PLCSIM, se hace clic en el ícono "Iniciar simulación" en la barra de herramientas.

TIA Portal compilará el proyecto y, si no hay errores, iniciará WinCC para simular la HMI programada.

Para que las instrucciones enviadas desde la HMI en WinCC tengan efecto, tanto WinCC como PLCSIM deben estar simulándose simultáneamente y conectados. Esto asegura que las modificaciones en los valores de las variables de la HMI estén sincronizadas con las variables del PLC.

La visualización de la HMI a través de WinCC se muestra a continuación:



FIGURA 78. Visualización pantalla HMI con WinCC.

Capítulo 4.3. Python

Python es un lenguaje de programación interpretado de alto nivel y de propósito general. Su sintaxis sencilla y legible lo hace accesible para muchos usuarios, lo que ha llevado a su uso extendido en áreas como el desarrollo web, la ciencia de datos, la inteligencia artificial y la automatización industrial.

En este proyecto, Python se emplea para desarrollar tanto el sistema de visión artificial como la gestión de comandos de voz. Python dispone de numerosas librerías que facilitan la implementación de estas funciones. A continuación, se destacan algunas de las librerías clave utilizadas:

- **OpenCV:** es una biblioteca de código abierto enfocada en la visión por computadora. Permite realizar detección de objetos, procesamiento de imágenes y análisis de vídeos. OpenCV es esencial para el sistema de visión artificial, encargado de identificar y clasificar cajas defectuosas.
- **SpeechRecognition:** esta librería facilita el reconocimiento de voz en Python, permitiendo la captura y el procesamiento de comandos de voz del usuario. Es crucial para la Interfaz Hombre-Máquina (HMI) del proyecto, permitiendo la interacción del usuario con el sistema mediante voz.
- **Snap7:** es una librería que posibilita la comunicación con PLCs de Siemens usando un formato de protocolo S7. Su principal función es facilitar la transferencia de datos entre el hardware de automatización y los sistemas de control, asegurando una conexión eficiente y fiable.

Python ha sido elegido para este proyecto debido a su versatilidad y a la amplia gama de librerías disponibles que soportan el desarrollo eficaz y robusto de aplicaciones de visión artificial, reconocimiento de voz y comunicación con hardware industrial.

Capítulo 4.3.1. Spyder

Spyder (Scientific Python Development EnviRonment) es un entorno de desarrollo integrado (IDE) de código abierto, diseñado específicamente para científicos y analistas de datos que trabajan con Python. Este IDE ofrece una amplia gama de herramientas avanzadas para la edición, depuración y análisis de datos, todo en una interfaz de usuario unificada, lo que facilita el manejo de grandes conjuntos de datos y aplicaciones científicas complejas.

La interfaz de usuario se organiza en distintas áreas funcionales, como se ilustra en la figura 79. Esta disposición sigue el mismo enfoque modular utilizado en otros softwares, facilitando la navegación y la interacción.

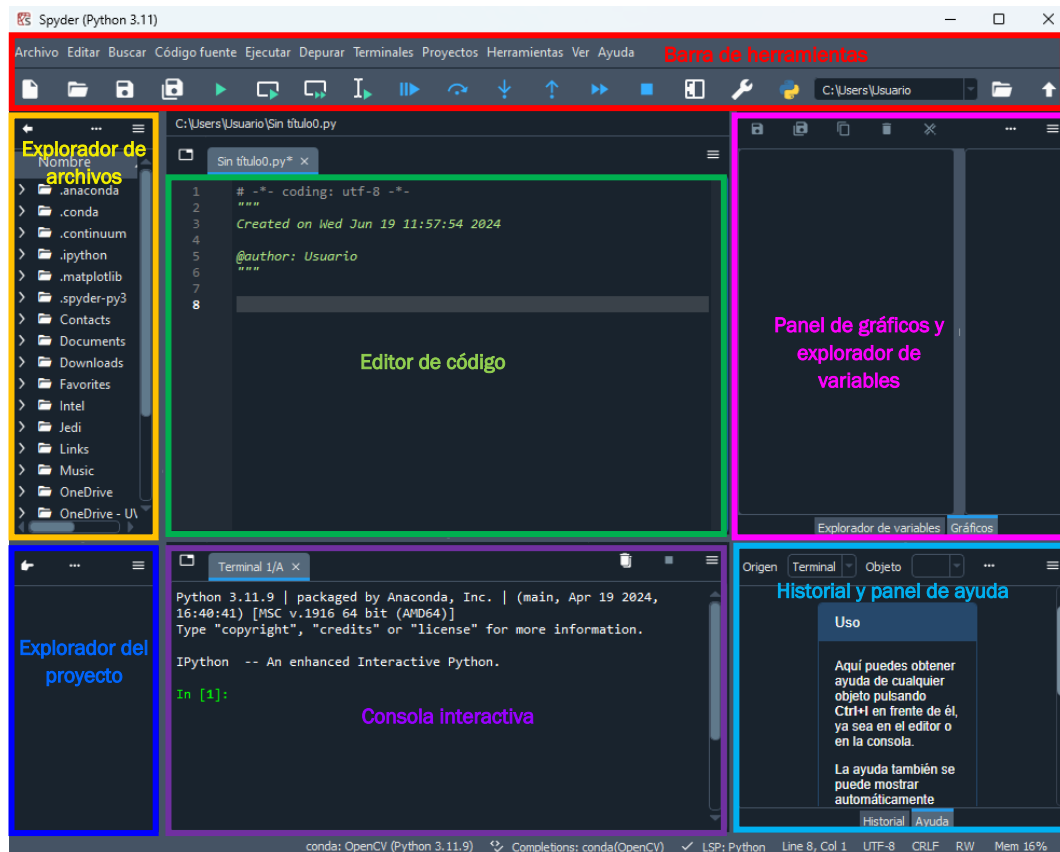


FIGURA 79. Interfaz de usuario en Spyder (Python 3.11).

- **Barra de herramientas (rojo):** ofrece acceso inmediato a diversas funciones clave, incluyendo abrir, guardar y ejecutar scripts, junto con herramientas para la depuración y la ejecución de comandos, entre otras. Esta barra optimiza la navegación y aumenta la eficiencia del usuario al desarrollar sus proyectos.

- **Explorador de archivos (amarillo):** esta sección muestra una vista de árbol de los archivos y directorios disponibles. Permite a los usuarios explorar y gestionar los diferentes scripts, documentos y recursos del proyecto. Los usuarios pueden abrir, renombrar, mover y organizar archivos directamente desde este panel.
- **Editor de código (verde):** es el área principal de trabajo donde se escribe y edita el código Python. Ofrece funcionalidades avanzadas como resaltado de sintaxis, autocompletado y análisis de código en tiempo real. Los comentarios, errores y advertencias se muestran en esta área para ayudar a los desarrolladores a escribir y depurar código de manera más eficiente.
- **Panel de gráficos y explorador de variables (rosa):** este panel permite a los usuarios ver y gestionar las variables del entorno de trabajo, mostrando información detallada sobre cada variable, incluyendo su nombre, tipo y valor actual. Además, esta área también permite la visualización de gráficos generados durante la ejecución de scripts.
- **Explorador del proyecto (azul oscuro):** muestra una lista jerárquica de todas las funciones y métodos definidos en el archivo de código actual. Permite a los usuarios navegar rápidamente entre las diferentes partes del código. Al hacer clic en una función o método, el usuario puede saltar directamente a su ubicación en el editor.
- **Consola interactiva (púrpura):** zona donde los usuarios pueden ejecutar comandos y scripts de Python en tiempo real. Permite ver los resultados de las ejecuciones de código inmediatamente, lo que facilita la experimentación y la resolución de problemas. Además, cuando se compilan y ejecutan scripts, la consola muestra los resultados, errores y mensajes de depuración, proporcionando feedback inmediato sobre el funcionamiento del código.
- **Historial y panel de ayuda (azul claro):** proporciona acceso a la documentación y ayuda sobre Python y sus librerías, permitiendo buscar información sobre funciones, clases y módulos, y ver ejemplos de uso directamente en el panel. También incluye el historial de comandos y ejecuciones, que permite a los usuarios revisar y reutilizar comandos anteriores.

Capítulo 4.4. Comunicaciones

Capítulo 4.4.1. NetToPLCSim

NetToPLCSim es una extensión que amplía las capacidades del simulador PLCSIM. (InfoPLC, 2021)

Esta herramienta permite acceder al simulador PLCSIM a través del protocolo de comunicación TCP/IP, utilizando la interfaz de red del ordenador donde se está ejecutando la simulación. Específicamente, utiliza Iso-On-TCP, que es una adaptación del protocolo ISO para operar sobre TCP/IP, permitiendo la comunicación eficaz con el simulador S7-PLCSIM. Aunque los datos intercambiados siguen el formato del protocolo S7, característico de los PLCs Siemens, la transferencia de estos datos se realiza mediante Iso-On-TCP, facilitando así la integración con sistemas de red basados en TCP/IP.

De esta manera, NetToPLCSim facilita la interacción entre el simulador del PLC y otros dispositivos conectados a la misma red, permitiendo realizar pruebas y simulaciones de forma más completa y realista.

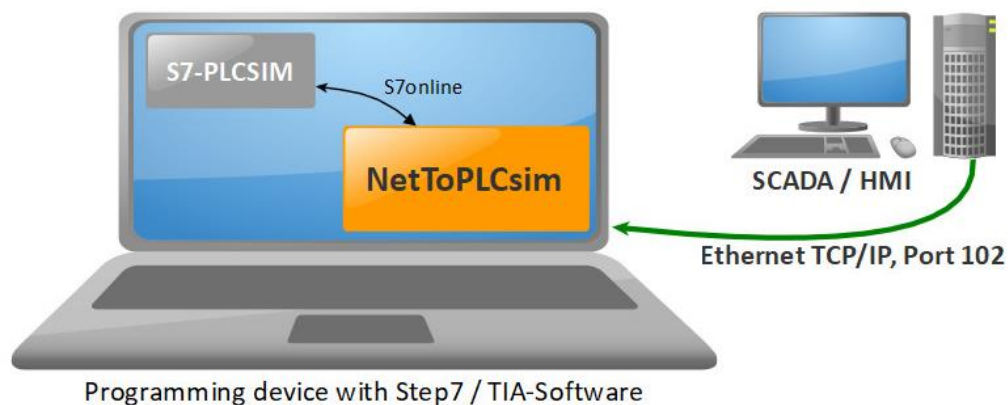


FIGURA 80. Esquema de conexión de NetToPLCSim. (InfoPLC, 2021)

Para utilizar NetToPLCSim, es necesario ejecutarlo con privilegios de administrador, debido a que requiere acceso al puerto 102 para realizar la comunicación. Al iniciar el software, la interfaz se presenta de la siguiente manera:



FIGURA 81. Interfaz NetToPLCSim.

En donde se pueden distinguir varias áreas de trabajo:

- **Barra de herramientas (rojo):** alberga herramientas básicas como abrir, guardar, etc., proyectos, así como herramientas para obtener acceso al puerto 102 o manuales de ayuda.
- **Área de conexiones (amarillo):** en esta zona se encuentran las estaciones creadas, acompañadas de sus parámetros característicos como son:
 - Nombre asignado.
 - Dirección IP del localhost.
 - Dirección IP asignada al PLCSIM.
 - Rack y Slot.
 - Estado.
- **Configuración de la estación (rosa):** es el área donde se sitúan los botones para añadir, modificar o borrar las estaciones añadidas al área de trabajo.
- **Iniciar/ Detener servidor (azul):** son botones para comenzar o finalizar la comunicación.
- **Estado del puerto 102 (verde):** indicador del estado del puerto 102. Para que la comunicación se lleve a cabo con éxito, NetToPLCSim debe tener acceso a este puerto.

El proceso a seguir para crear una estación e iniciar la comunicación es el siguiente:

1. Pulsar el botón “Add” para añadir una estación.
2. Configurar los datos de la estación:

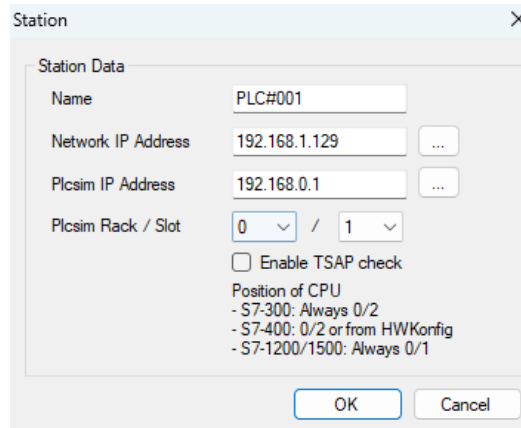


FIGURA 82. Datos de la estación para la comunicación con NetToPLCSim.

- **Nombre de la estación.**
 - **Dirección IP del localhost:** se realizará la comunicación a través de la tarjeta de red del ordenador, por lo que se seleccionará dicha dirección IP.
 - **Dirección IP de PLCSIM:** se seleccionará la dirección IP asignada al PLC que se debe estar simulando.
 - **Rack y Slot de PLCSIM:** se debe conocer en que rack y slot se encuentra el PLC.
3. Hacer clic en el botón “OK” de la figura 64 para añadir la estación al área de conexiones.
 4. Seleccionar la estación deseada en el área de conexiones y hacer clic en el botón "Start Server" para que el estado de la estación cambie a "RUNNING" y se pueda establecer comunicación.

La interfaz de NetToPLCSim con la estación en ejecución será:

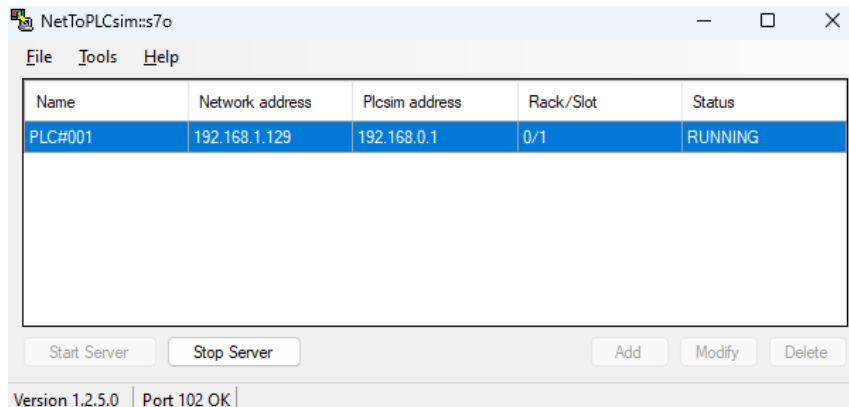


FIGURA 83. Servidor en ejecución desde NetToPLCSim.

NetToPLCSim ofrece una herramienta para monitorear la comunicación. Para utilizarla, es necesario seleccionar la estación que se desea monitorear, hacer clic derecho sobre ella y elegir la opción “Start monitoring”. Al hacerlo, se obtiene información detallada sobre la transmisión de datos.

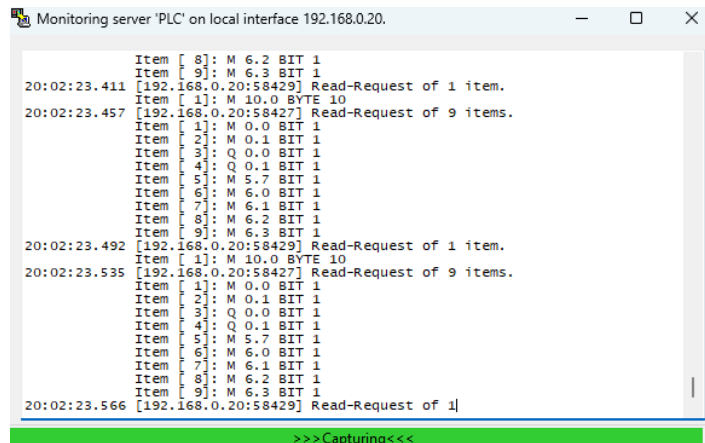


FIGURA 84. Monitoreo del servidor en NetToPLCSim.

CAPÍTULO 5. Diseño y desarrollo de la solución adoptada

Capítulo 5.1. Desarrollo en RobotStudio

En esta sección, se ha decidido estructurar la exposición en dos partes fundamentales.

La primera se centra en la configuración de la estación, abordando aspectos como la selección y disposición de los robots, así como los componentes adicionales tales como cintas transportadoras, dispensadores de palés y herramientas de trabajo. También se discutirá cómo se han diseñado los componentes inteligentes.

La segunda parte profundizará en la programación realizada utilizando RAPID, el lenguaje específico de RobotStudio.

Capítulo 5.1.1. Elementos y composición de la estación

Capítulo 5.1.1.1. Robots utilizados

Para iniciar la creación de una estación robótica es imprescindible contar con al menos un robot. En este proyecto, se ha seleccionado el modelo IRB 4600-60/2.55 de ABB como ya se ha comentado.



FIGURA 85. Robot IRB 4600-60/2.55. (Robotics, ABB, 2019)

A pesar de que este proyecto se limita a una simulación y no ha sido implementado en un entorno real, resulta crucial comprender los parámetros de operación del robot elegido. A continuación, se describen las características más relevantes del modelo IRB 4600-60/2.55, según la información proporcionada por el fabricante. (Robotics, ABB, 2019)

- **Peso del robot:** 435 Kg.
- **Capacidad de manejo:** 40 Kg.
- **Temperatura ambiente operativa:** +5 °C a +45° C.
- **Estándar de protección:** IP67 garantizando protección contra el polvo y el agua.
- **Velocidad máxima de cada eje:**
 - **Eje 1:** 175 °/s.
 - **Eje 2:** 175 °/s.
 - **Eje 3:** 175 °/s.
 - **Eje 4:** 250 °/s.
 - **Eje 5:** 250 °/s.
 - **Eje 6:** 360 °/s.
- **Área de trabajo:**
 - **Pos. A:** 2872 mm.
 - **Pos. B:** 1735 mm.
 - **Pos. C:** 1393 mm.
 - **Pos. D:** 680 mm.
 - **Pos. E:** 2202 mm.
 - **Pos. F:** 2552 mm.

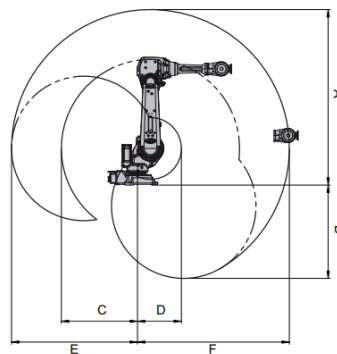


FIGURA 86. Área de trabajo robot IRB 4600-60/2.55. (Robotics, ABB, 2019)

Si bien las características mencionadas son importantes, dependiendo de la aplicación específica, podría ser necesario considerar otros parámetros adicionales. No obstante, los descritos anteriormente constituyen algunos de los más relevantes para este proyecto.

Capítulo 5.1.1.2. Elementos físicos

A pesar de que los robots constituyen el núcleo de la estación, son necesarios otros componentes adicionales. La mayoría de estos objetos físicos provienen de RobotStudio, con la excepción del dispensador de planchas, que ha sido diseñado por el autor del proyecto utilizando Autodesk Inventor.

A continuación, se enumeran los elementos que integran la estación:

Herramienta para el robot

En este proyecto, la herramienta seleccionada para manipular cajas es la denominada "Vacuum". Se ha asignado una de estas herramientas a cada robot para facilitar las operaciones de carga y descarga.

Los parámetros característicos son:

- **Largo:** 800 mm.
- **Ancho:** 280 mm.
- **Número de vigas:** 2.
- **Número de succionadores por viga:** 3.



FIGURA 87. Herramienta "Vacuum" usada en la estación.

Pedestal

Dado que la altura de los palés con cajas supera la capacidad del área de trabajo de los robots, se ha tomado la decisión de elevar cada robot sobre un pedestal ya que el modelo IRB 4600-60/2.55 admite la posibilidad de subirlo a un pedestal. Esta solución permite ganar 225 mm adicionales de altura, asegurando que los robots puedan manejar cajas en la última altura sin salirse de su zona de operación asignada.

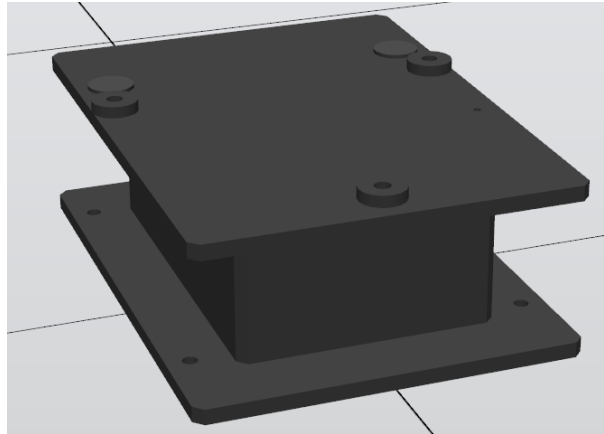


FIGURA 88. Pedestal sobre el que montar los robots de la estación.

Cinta transportadora de cajas

Las cajas destinadas a ser paletizadas o clasificadas por los robots llegarán a través de cintas transportadoras, como se muestra en la figura 86. Cada tipo de caja se trasladará por una cinta diferente.

Los parámetros característicos de estas cintas son:

- **Longitud:** 8000 mm.
- **Ancho:** 600 mm.
- **Alto:** 800 mm.

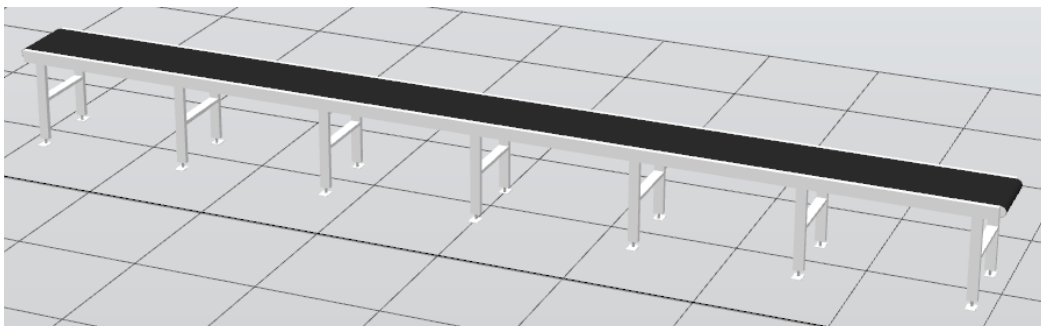


FIGURA 89. Cinta transportadora de cajas usada en la estación.

Cinta transportadora de palés

Para el proceso de paletizado, se sitúa un palé al inicio de cada cinta transportadora. Al completarse el apilado en un palé, se emitirá una orden para que este avance hacia el extremo opuesto de la cinta, donde una carretilla elevadora podrá recogerlo.

Las dimensiones de las cintas transportadoras utilizadas en este proceso son:

- **Largo:** 4000 mm.
- **Ancho:** 1300 mm.
- **Alto:** 200 mm.

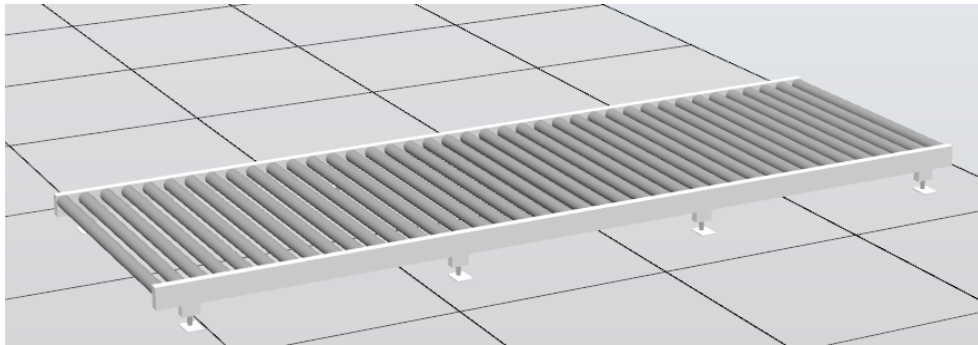


FIGURA 90. Cinta transportadora de palés usada en la estación.

Vallas metálicas de seguridad

La seguridad es prioritaria en cualquier estación robótica. Para garantizarla, se han instalado barreras de seguridad que ningún operario debe cruzar mientras los robots estén en funcionamiento. Las vallas empleadas en este proyecto varían en longitud y algunas incluyen aperturas específicas que permiten actividades como la reposición de palés o la recogida de los ya completados. La apariencia de estas verjas de seguridad es la siguiente:

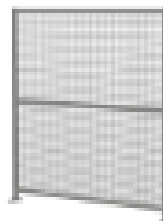


FIGURA 91. Valla de seguridad usada en la estación.

Dispensador de palés

Con el fin de asegurar un suministro continuo de palés, se ha implementado un sistema específico que permite reponerlos durante la simulación.

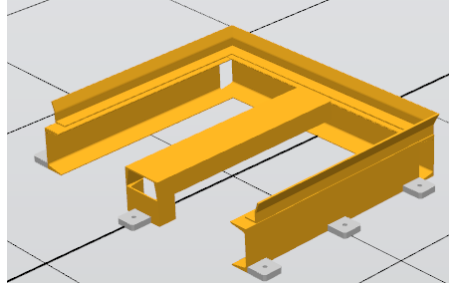


FIGURA 92. Dispensador de palés usado en la estación.

Dispensador de planchas de cartón

Como se mencionó al comienzo de este capítulo, el dispensador de planchas fue diseñado por el autor del proyecto utilizando Autodesk Inventor. Posteriormente, este componente fue importado a RobotStudio para integrarse en la simulación de la estación robótica.

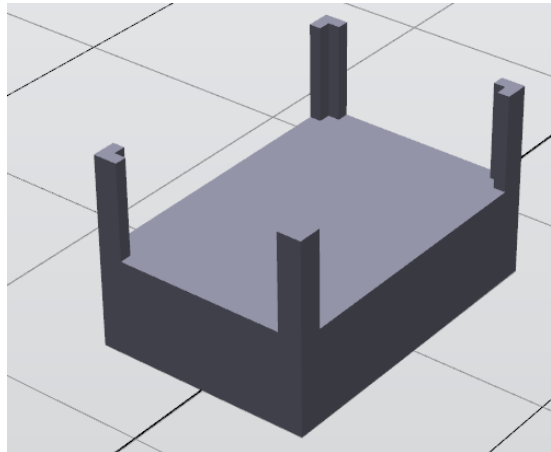


FIGURA 93. Dispensador de planchas usado en la estación. (Diseñado por el autor del proyecto)

Por ahora, todos estos componentes son meramente decorativos. En el capítulo 5.1.2 se detalla el procedimiento para configurarlos de modo que exhiban el comportamiento deseado dentro de la simulación.

Capítulo 5.1.1.3. Herramientas y objetos de trabajo

Herramientas de trabajo

Los datos de herramientas son parámetros que describen las características físicas y funcionales de las herramientas que un robot puede emplear. Estos parámetros abarcan aspectos como las dimensiones, forma, orientación y el Punto de Control de Herramientas (TCP).

En RobotStudio, se puede generar estos datos de dos maneras: automáticamente, al importar herramientas predefinidas, o de forma manual para aquellas herramientas personalizadas que no están predefinidas. En este último caso, es necesario introducir manualmente los datos requeridos por el programa, tal como se ilustra en la figura 94.

Para configurar los datos de una herramienta dentro de una estación en RobotStudio, se deben seguir los pasos que se detallarán a continuación:

1. Dirigirse a la opción “Trayectoria y puntos” dentro de la ventana de navegadores.
2. Dentro del controlador y robot específicos, hacer clic derecho sobre la carpeta “Datos de herramienta” y elegir la opción “Crear dato de herramienta”.

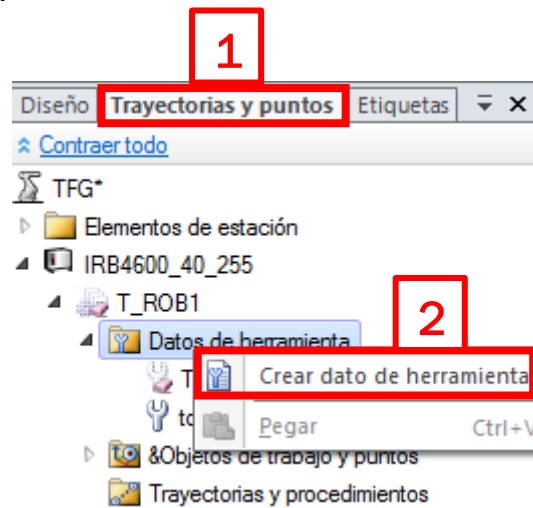


FIGURA 94. Procedimiento para crear un dato de herramienta en RobotStudio 2023.

3. El siguiente paso es configurar los datos desplegados según conveniencia y finalmente hacer clic en el botón “crear”.

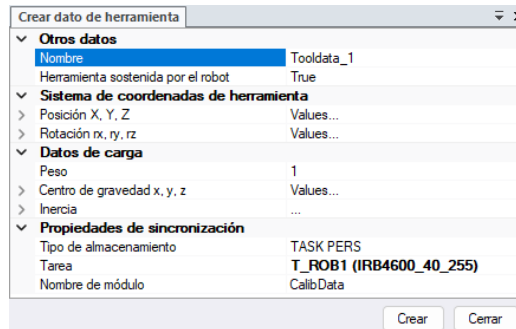


FIGURA 95. Datos de herramienta en RobotStudio 2023.

Objetos de trabajo

Un objeto de trabajo es básicamente un sistema de coordenadas que se utiliza para describir la posición de una pieza de trabajo.

Este sistema se compone de dos conjuntos de coordenadas:

- **Base de coordenadas del usuario:** sirve como marco de referencia para el objeto de trabajo. Situada en un lugar específico relacionado con el entorno de trabajo, como puede ser un tablero o una cinta.
- **Base de coordenadas del objeto:** es descendiente de la base de coordenadas del usuario. Este sistema se coloca directamente en la pieza de trabajo y se utiliza para todas las operaciones de programación.

En caso de no especificar ningún objeto de trabajo, los objetivos dependen del objeto predeterminado “Wobj0”, que será siempre coincidente con la base de coordenadas del robot.

El procedimiento para crear un objeto de trabajo es similar al procedimiento para crear datos de herramienta:

1. Dirigirse a la opción “Trayectoria y puntos” dentro de la ventana de navegadores.
2. Dentro del controlador y robot específicos, hacer clic derecho sobre la carpeta “Objetos de trabajo y puntos” y elegir la opción “Crear objeto de trabajo”.

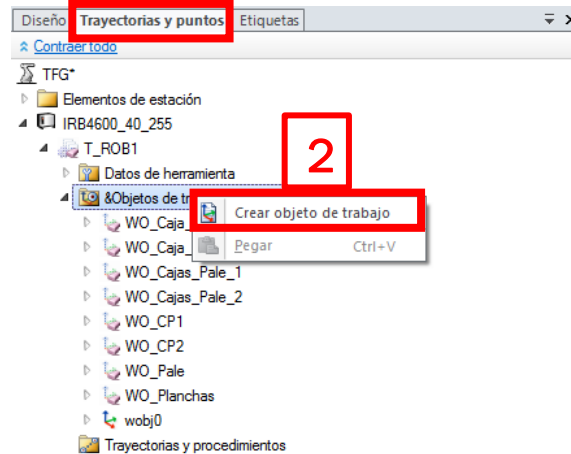


FIGURA 96. Procedimiento para crear un objeto de trabajo en RobotStudio 2023.

3. El último paso es configurar los datos desplegados según conveniencia y finalmente hacer clic en el botón “crear”.

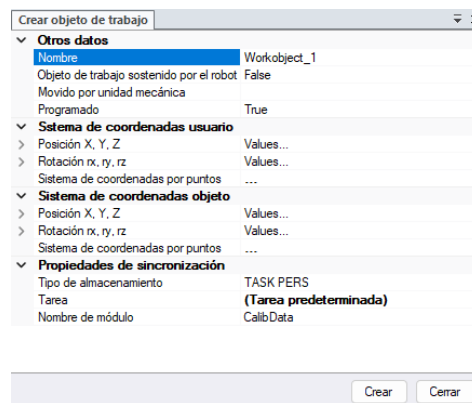


FIGURA 97. Parámetros para crear un objeto de trabajo en RobotStudio 2023.

Los **objetos de trabajo (rojo)** y las **herramientas (azul)** creados en la estación robótica del proyecto se presentan en la siguiente figura:

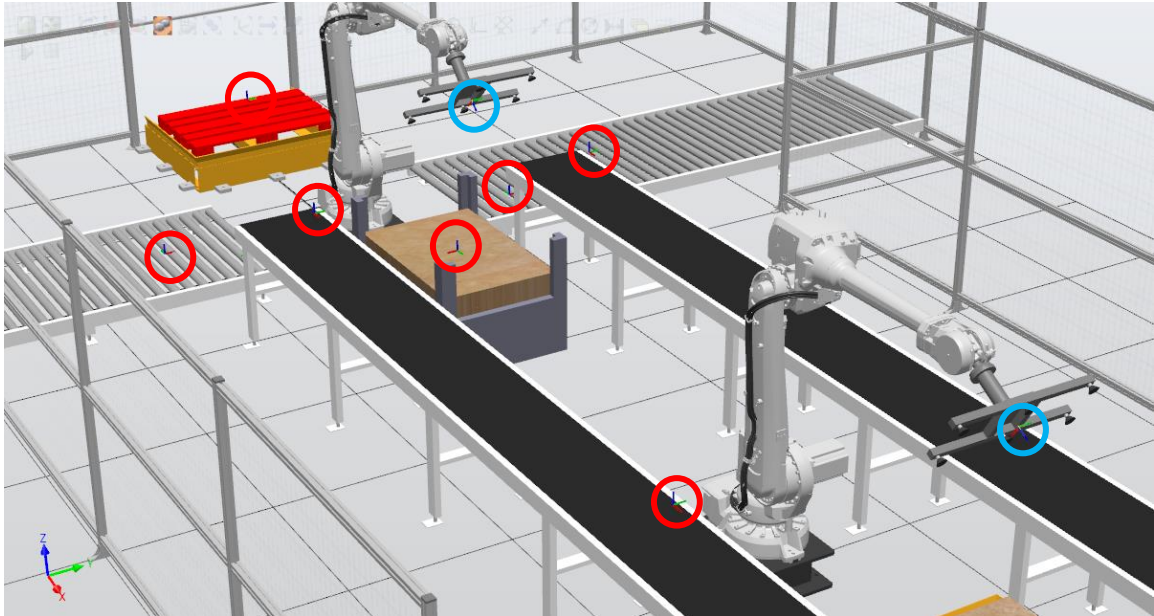


FIGURA 98. Objetos y herramientas de trabajo de la estación en RobotStudio 2023.

Finalmente, para poder usar en RAPID estos objetos y herramientas de trabajo, hay que sincronizar estos datos de la estación con RAPID.

Los pasos necesarios para completar esta sincronización se detallan a continuación:

1. Dirigirse a la pestaña “Posición Inicial” dentro de la barra de pestañas.
2. Abrir el desplegable “Sincronizar” y seleccionar la opción “Sincronizar con RAPID”.

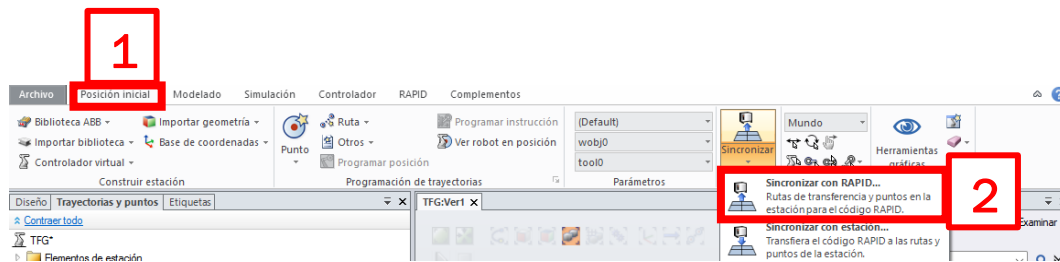


FIGURA 99. Procedimiento para sincronizar el Controlador con RAPID en RobotStudio 2023.

3. Elegir los elementos a sincronizar, el módulo donde se sincronizarán estos y el tipo de variable.

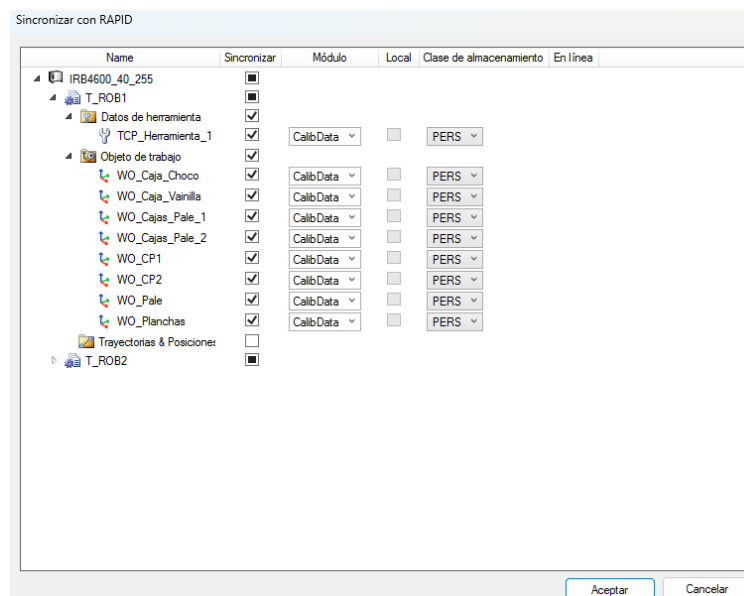


FIGURA 100. Sincronizar Controlador con RAPID en RobotStudio 2023.

Una vez hecho esto, se podrán utilizar estas herramientas y objetos de trabajo creados para programar el movimiento de los robots desde RAPID.

Capítulo 5.1.2. Componentes inteligentes (Smart Components)

Los componentes inteligentes (Smart Components) son fundamentales para simular el comportamiento de los distintos elementos dentro de la estación de trabajo. Estos componentes son imprescindibles para programar las interacciones entre los robots y su entorno de manera que se puedan recrear condiciones reales de operación durante la simulación.

Antes de programar los comportamientos específicos, es necesario establecer y categorizar cada componente dentro de la estación. Los componentes inteligentes en RobotStudio se organizan en diversas categorías, cada una con funciones específicas que permiten una amplia gama de operaciones:

- **Señales y propiedades:** incluyen bloques para lógica básica como puertas AND, OR, XNOR, además de temporizadores y bloques para expresiones matemáticas.
- **Primitivos paramétricos:** permiten la creación dinámica de sólidos o líneas durante la simulación.
- **Sensores:** esta categoría abarca una variedad de sensores que pueden detectar la presencia o contacto de objetos, activando señales específicas cuando un objeto atraviesa su campo de detección.
- **Acciones:** aquí se encuentran bloques que permiten realizar acciones como unir dos objetos, crear o eliminar objetos y alternar la visibilidad de los mismos dentro de la simulación.
- **Manipuladores:** contienen bloques para mover objetos o mecanismos siguiendo trayectorias lineales o circulares y mover objetos a unas coordenadas determinadas.
- **Controlador:** incluye bloques que permiten la lectura y escritura de variables utilizadas en RAPID.
- **Física:** estos bloques son utilizados para asignar propiedades físicas a objetos o mecanismos, mejorando el realismo de las simulaciones mediante la aplicación de dinámicas y restricciones físicas reales.
- **PLC:** proporcionan la capacidad de establecer comunicación con PLCs de Siemens a través de conexiones OPC o SIMIT.
- **Realidad virtual:** esta categoría incluye componentes como el VrHandController, que se mueve con un controlador manual en realidad virtual, VrSession, que añade botones a un panel de menú de RV y señala cuando el usuario sale e ingresa, y VrTeleporter, que teletransporta al usuario de RV a la ubicación del componente.

- **Otros:** categoría que incluye una variedad de funciones adicionales como la generación de números aleatorios, la detención de simulaciones, la reproducción de sonidos, o cambios en las propiedades de los objetos, como el color.

Una vez definidos los bloques, se procede a integrarlos en la lógica de la estación.

Este proceso implica conectar cada componente con la señal que lo controle, asegurando que todas las interacciones y movimientos sean gestionados correctamente.

La programación se lleva a cabo desde la opción “Lógica de estación” dentro de la pestaña “Simulación” como se muestra en la figura 98, donde se pueden visualizar y conectar los bloques mediante enlaces visuales o no visuales, dependiendo de las preferencias del usuario y la complejidad de la estación.

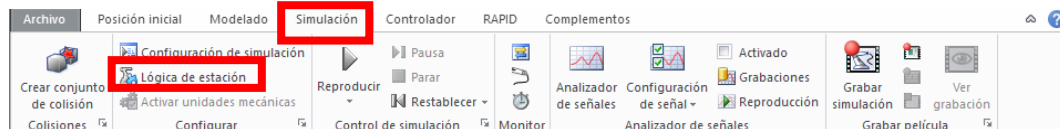


FIGURA 101. Acceso a lógica de estación en RobotStudio 2023.

A continuación, se explicará cada componente inteligente creado por separado:

SimulationEvents

El componente inteligente “SimulationEvents” permite interactuar con los eventos de la simulación. Se manejan dos eventos principalmente:

- **“SimulationStarted”**: este evento se activa cuando comienza la simulación. En el proyecto se usa para iniciar la comunicación, inicializar variables, borrar elementos que no hayan sido eliminados en anteriores simulaciones, poner en marcha procesos, etc.
- **“SimulationStopped”**: este evento se activa cuando la simulación se detiene. En la estación es usado para finalizar la comunicación, borrar elementos, detener procesos, etc.

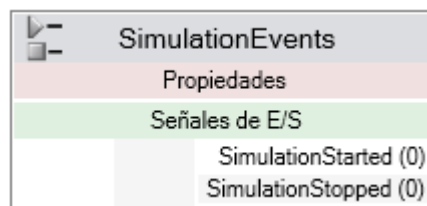


FIGURA 102. Componente inteligente “SimulationEvents” en RobotStudio 2023.

Este bloque se ha implementado en cada uno de los elementos siguientes, dado que se han observado mejoras significativas en la reducción de retrasos y fallos al usar un bloque distinto para cada componente, en lugar de un único bloque para todos ellos.

SC_Herramienta

El componente inteligente “SC_Herramienta” está diseñado para simular operaciones características de una herramienta como coger y dejar cuerpos. Se estructura mediante varios bloques como se muestra en la figura 103.

- **Señal de entrada “Vacio_Herr”:** el controlador del robot será el encargado de activarla o desactivarla para coger o soltar objetos.
- **Sensor lineal:** detecta la presencia de objetos en una zona muy próxima a las ventosas de la herramienta. Su activación desencadena operaciones siguientes.
- **“Attacher” y “Detacher”:** simulan el acoplamiento y desacoplamiento de los objetos que el sensor lineal detecta con la herramienta del robot. El "Attacher" se encarga de unir el objeto a la herramienta, mientras que el "Detacher" de liberarlo.
- **Puerta NOT:** asegura que, cuando la señal "Vacio_Herr" está desactivada, el bloque "Detacher" se active para liberar el objeto previamente sujeto.
- **Biestable RS:** mantiene un registro de si la herramienta está o no sujetando un objeto.
- **Señal de salida “Cogido_Herr”:** a través de esta señal se comunica a otros componentes el estado actual de la herramienta, específicamente si está o no sujetando un objeto.

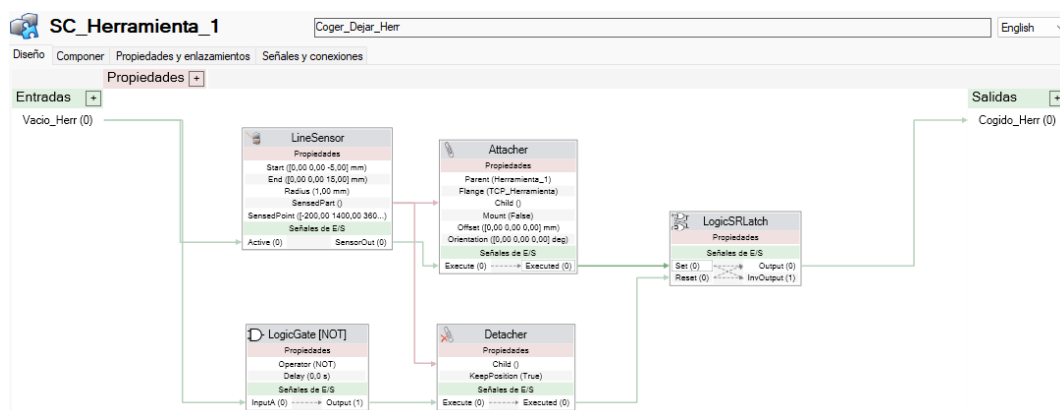


FIGURA 103. “SC_Herramienta” en RobotStudio 2023. (Diseñado por el autor del proyecto)

Aunque la imagen muestra "SC_Herramienta_1" relacionado con la herramienta del robot paletizador, la estructura del componente inteligente correspondiente a la herramienta 2 relacionada con la herramienta del robot clasificador es idéntica.

SC_Cinta

Este componente ha sido diseñado específicamente para controlar cada una de las dos cintas transportadoras, creando un componente inteligente independiente para cada una mediante la modificación de coordenadas y ajustes menores en sus parámetros distintivos. La función principal de este componente es la generación y movimiento de cajas.

Para facilitar la comprensión de este componente, los bloques más representativos se han organizado y clasificado según su propósito y las señales de entrada y salida. Esta estructuración se ilustra en la figura 104, donde cada grupo de bloques está claramente diferenciado para mostrar cómo contribuyen al funcionamiento global de la cinta transportadora.

- **Señales de entrada:**
 - **“Start_C1”:** esta señal activa el movimiento de la cinta. La cinta solo operará si esta señal está en estado alto (1) y es controlada desde la Interfaz Hombre-Máquina (HMI).
 - **“Clasificar”:** esta señal pausa la cinta en la zona específica donde el robot clasificador recoge las cajas defectuosas.
- **Sensores de recogida:** detectan la presencia de cajas en las zonas asignadas para su recogida, ya sea para el paletizado o para la clasificación.
- **Generación de cajas:** compuesto por:
 - **Sensor lineal:** activo constantemente, inicia la creación de una caja cuando no detecta presencia.
 - **Bloques “Source”:** encargados de generar cajas, uno para cajas defectuosas y otro para cajas válidas. Se utiliza un sistema donde 1 de cada 10 cajas es defectuosa, gestionado por un bloque de comparación y un bloque "Random" que genera un número aleatorio cada vez que se crea una caja. Si el número generado es mayor que 1, la caja es válida; si es menor, es defectuosa.
- **Gestión de colas:** maneja el flujo de cajas sobre la cinta, eliminando de la cola las cajas que son recogidas por cualquier robot.
- **Movimiento:** por medio del bloque “LinearMover” se trasladan las cajas a lo largo de la cinta a velocidades programadas. Para la línea 1, la velocidad es de 350 mm/s y para la línea 2, de 300 mm/s.
- **Resto de bloques no agrupados:** coordinan la lógica operativa entre bloques, asegurando el correcto funcionamiento de la cinta.

- Señales de salida:

- “Sensor_C1”: cuando esta señal esta activa, esta señal notifica al robot paletizador que hay una caja disponible para recoger al final de la cinta.
- “Coger_Caja”: similarmente, esta señal alerta al robot clasificador de la presencia de una caja defectuosa en la zona de recogida específica.

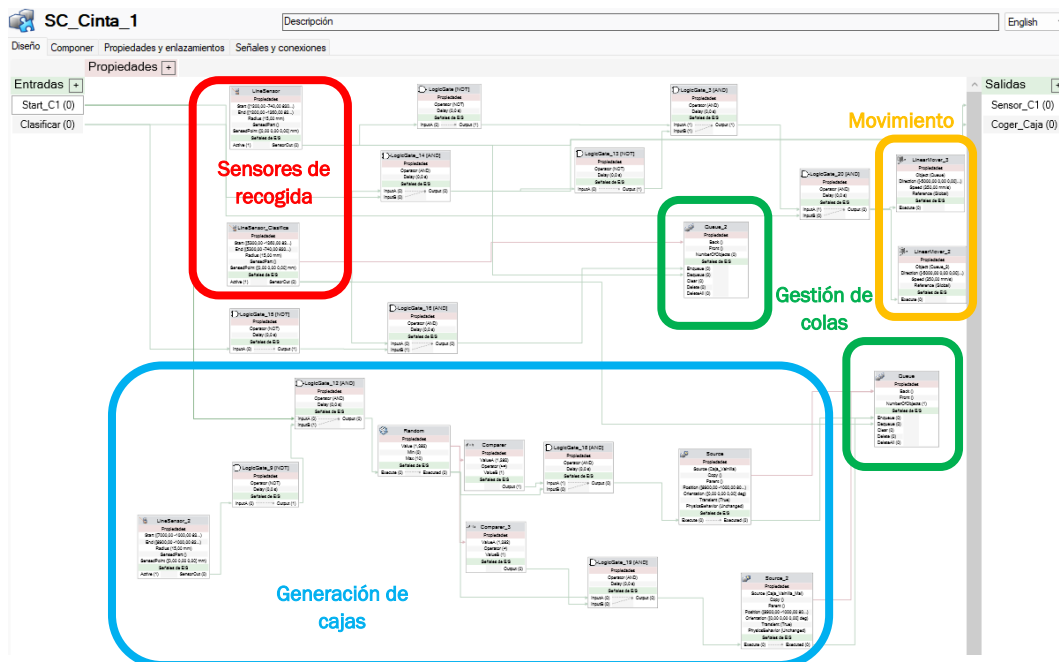


FIGURA 104. “SC_Cinta” en RobotStudio 2023. (Diseñado por el autor del proyecto)

SC_Cinta_Pale

Gracias al componente “SC_Cinta_Pale”, se ha logrado programar el movimiento de las cintas transportadoras de palés, así como la eliminación de los elementos que conforman los palés una vez completados y situados durante unos segundos en la posición de recogida. Este procedimiento se ha estandarizado para ambas cintas transportadoras de palés.

La organización y función de los bloques de este componente se ilustran en la figura 105 y se detallan a continuación:

- **Señales de entrada:**
 - **“Pale_Completado”:** activa el movimiento de la cinta para trasladar el palé hasta el extremo de la misma, donde será recogido.
 - **“Pale_Recogido”:** indica que todos los elementos del palé a ser eliminado han sido completamente borrados.
- **Sensores para crear palé:** compuesto por tres sensores de tipo planar que detectan la adición de nuevos elementos al palé, permitiendo gestionarlos como un conjunto unitario dentro de la cola.
- **Borrado de elementos:** activado por la señal "Pale_Recogido", este proceso involucra el bloque "Sink", que elimina del sistema cualquier objeto detectado por el sensor volumétrico ubicado al final de la cinta.
- **Gestión de movimiento y de cola:** estos bloques controlan tanto el movimiento de la cinta transportadora de palés como la gestión de la cola de elementos que componen el palé, hasta que son completamente eliminados de la estación.
- **Señales de salida:**
 - **“Pale_A_Recoger”:** señala que el palé está en la zona final de la cinta, listo para ser borrado (en una aplicación real, esto indicaría que está listo para ser recogido por una carretilla transportadora).
 - **“Hay_Cola”:** informa que aún quedan elementos del palé que deben ser eliminados.
 - **“Estado_CP”:** muestra el estado actual de la cinta, ya sea en movimiento o detenida, y esta información se envía a la interfaz HMI para mantener al operador informado sobre el estado operativo de la cinta.

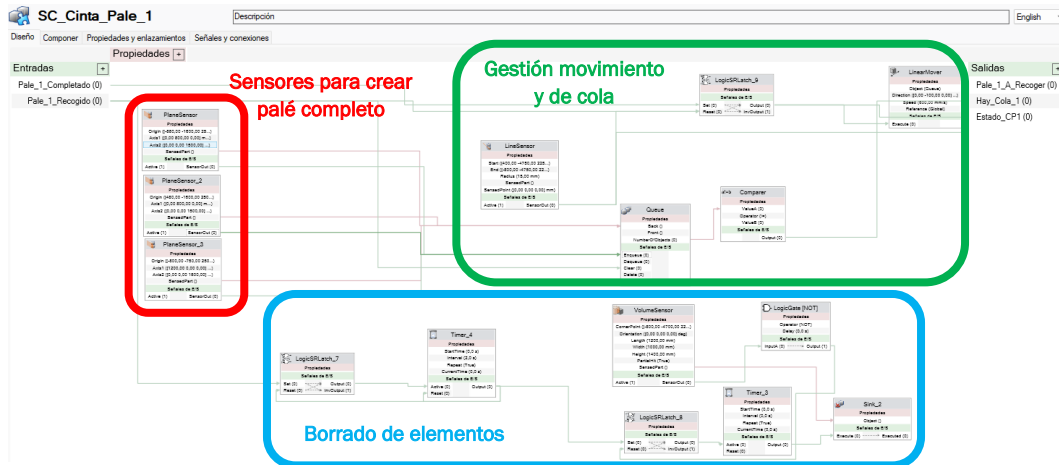


FIGURA 105. "SC_Cinta_Pale" en RobotStudio 2023. (Diseñado por el autor del proyecto)

SC_Dispensador_Pale

El componente "SC_Dispensador_Pale" ha sido diseñado para automatizar completamente la dispensación de palés, asegurando que el robot tenga acceso continuo a ellos. El funcionamiento de este sistema se detalla a continuación, mediante la descripción de los bloques empleados en su configuración:

- **"SimulationEvents"**: este bloque es relevante al inicio y al final de la simulación. Al comenzar, asegura que haya disponible un palé, y al concluir, elimina todos los palés que han sido generados y almacenados en la cola durante la simulación.
- **Sensor lineal**: permanece constantemente activo para monitorear la presencia de palés en el dispensador. Su objetivo es detectar si hay o no palés disponibles.
- **Puerta NOT**: actúa negando la salida del sensor lineal para lograr la siguiente lógica:
 - **Si hay palé**: la salida del sensor lineal es 1, y después de la puerta NOT, se convierte en 0, indicando que no es necesario generar otro palé.
 - **Si no hay palé**: la salida del sensor lineal es 0, y tras pasar por la puerta NOT, se convierte en 1, señalando la necesidad de crear un nuevo palé.
- **"Source"**: este bloque se utiliza para crear objetos cuando se recibe la señal en su entrada "Execute". En este caso, está configurado para generar copias del objeto "Palé". Una vez que el palé es creado, la salida "Executed" del bloque se activa a 1 y se envía esta señal al bloque "Queue".

- **“Queue”**: administra una cola de objetos, que en este caso son palés. Está configurado para apilar los nuevos objetos al final de la cola a medida que son creados, lo cual es notificado por la salida "Executed" del bloque "Source".

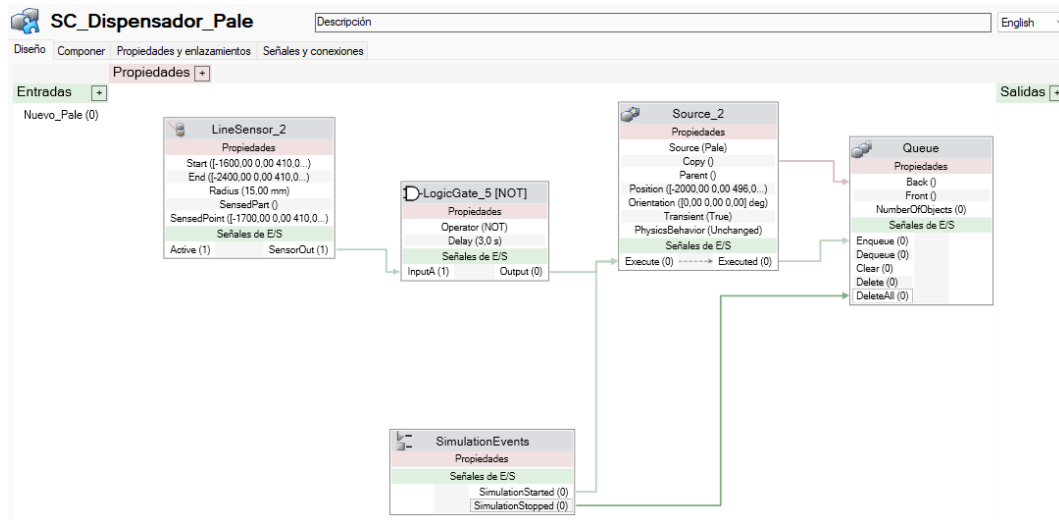


FIGURA 106. “SC_Dispensador_Pale” en RobotStudio 2023. (Diseñado por el autor del proyecto)

SC_Plancha

El dispensador de planchas de cartón ha sido configurado de manera similar al dispensador de palés para asegurar una provisión continua de planchas a medida que el robot las utiliza. Cada vez que una plancha es retirada, inmediatamente se crea una nueva.

Los componentes y su funcionamiento en este sistema se describen a continuación:

- **“SimulationEvents”**: utilizado al iniciar y finalizar la simulación. Asegura que haya una plancha disponible al empezar y elimina todas las planchas generadas y almacenadas en la cola al concluir.
- **Biastable RS**: aunque no sería estrictamente necesario, se ha incorporado para resolver problemas ocasionales en los que, al iniciar o detener la simulación, no se generaban nuevas planchas o no se eliminaban las existentes, debido a bloqueos en el sensor lineal.
- **Sensor lineal**: activo desde el inicio de la simulación, este sensor detecta la presencia o ausencia de una plancha. Informa al bloque “Source” para generar una nueva plancha en caso de que no haya.

- **Puerta NOT:** niega la salida del sensor lineal para obtener el siguiente funcionamiento:
 - **Si hay plancha:** la salida del sensor es 1, que tras pasar por la puerta NOT, se convierte en 0, indicando que no es necesario crear otra plancha.
 - **Si no hay pale:** la salida del sensor es 0, que se convierte en 1 después de la puerta NOT, señalando la necesidad de generar una nueva plancha.
- **“Source”:** este bloque crea objetos cuando recibe la señal "Execute". Está configurado para producir copias del objeto "Plancha". Tras la creación de una nueva plancha, la salida "Executed" se activa a 1 y se envía esta señal al bloque “Queue”.
- **“Queue”:** administra una cola de objetos, en este caso, planchas de cartón. Está configurado para que los objetos recién creados se apilen al final de la cola, lo cual es notificado por la salida “Executed” del bloque “Source”.

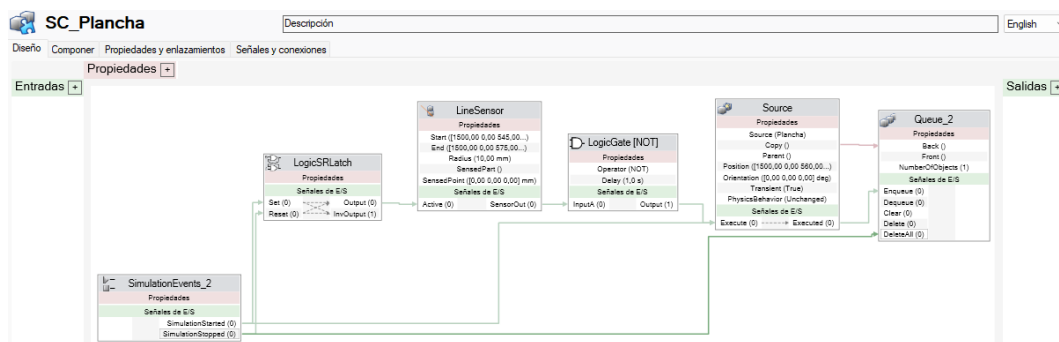


FIGURA 107. “SC_Plancha” en RobotStudio 2023. (Diseñado por el autor del proyecto)

SC_Camara

Gracias al desarrollo de este componente inteligente, se ha logrado integrar una cámara de visión artificial en RobotStudio, permitiendo la captura de imágenes. Esta innovación representa un avance significativo, ya que es la primera vez que se consigue incorporar una cámara de visión en una simulación de RobotStudio.

Los bloques que conforman la estructura de la cámara se organizan y funcionan de acuerdo con lo que se ilustra en la figura 108, proporcionando una capacidad sin precedentes para la visualización y análisis en tiempo real dentro del entorno simulado de RobotStudio.

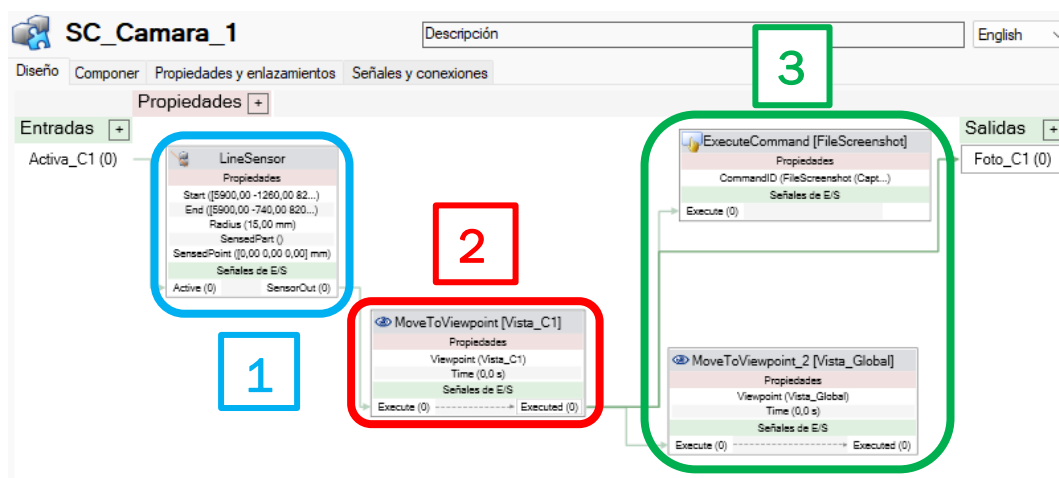


FIGURA 108. "SC_Camara" en RobotStudio 2023. (Diseñado por el autor del proyecto)

El funcionamiento global del componente se detalla a continuación:

- Si la **señal de entrada "Activa_C1"** está **desactivada**, no se realizará ninguna captura de imagen. Esto asegura que la cámara sólo opere cuando sea necesario.
- Si la **señal de entrada "Activa_C1"** está **activada**, se sigue la siguiente secuencia:
 1. **Espera de activación:** la cámara espera hasta que el sensor lineal, colocado en la cinta transportadora a la altura del robot clasificador, detecte una caja y se active.
 2. **Movimiento de vista:** una vez activado el sensor, la vista en la simulación se cambia a "Vista_C1", que está enfocada específicamente donde la caja a analizar se encuentra en ese momento sobre la cinta.

- 3. Captura y retorno de vista:** inmediatamente después de cambiar la vista, se ejecuta el comando para realizar la captura de pantalla. Posteriormente, la vista se retorna a “Vista_Global”. Además, se envía una señal de salida indicando desde que cámara ha sido tomada la foto.

Para la cámara destinada a la cinta transportadora de cajas de chocolate (cinta 2), el procedimiento es idéntico, cambiando únicamente los campos específicos que diferencian cada cámara, tales como la posición de la cámara o las coordenadas del sensor.

SC_Inicio_Comunicacion

Este componente inteligente ha sido diseñado para gestionar de manera efectiva la comunicación con el PLC durante las simulaciones.

El funcionamiento detallado es el siguiente:

- **Señales de entrada “Inicio” y “Fin” de comunicación:** al iniciar la simulación, se manda automáticamente la orden de iniciar la comunicación con el PLC, empleando para ello un biestable RS. De manera similar, cuando se detiene la simulación, se envía la orden de finalizar la comunicación. Este proceso asegura que la comunicación con el PLC solo esté activa durante el período de simulación.
- **Señales de salida:** estas señales de salida estarán conectadas al componente específico que realiza la comunicación con el PLC, el cual será detallado en las descripciones de los siguientes componentes inteligentes. Las señales de salida serán:
 - **“Conectar”:** indica que hay que iniciar la comunicación con el PLC.
 - **“Leer”:** aunque en la configuración actual esta señal no está conectada a ningún otro componente del sistema, se conserva activa dentro del diseño. La razón de mantener esta señal es su potencial utilidad en futuros proyectos o modificaciones del sistema actual. En situaciones donde la lectura continua no sea gestionada por el componente específico de conexión, esta señal podría ser utilizada para activar procesos de lectura de datos del PLC bajo demanda o en intervalos específicos.

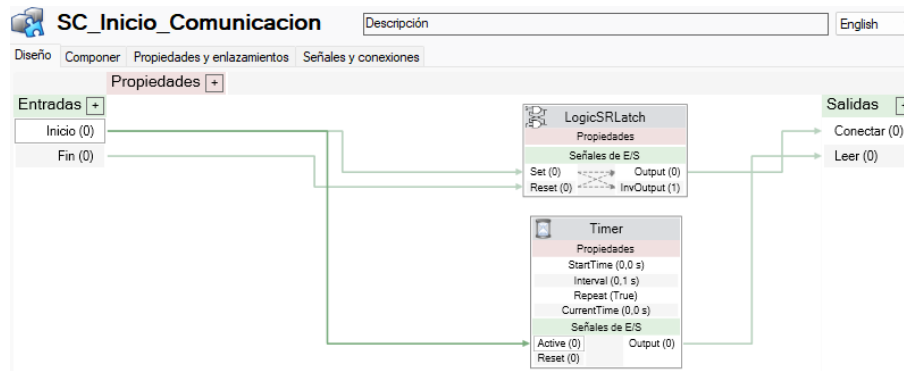


FIGURA 109. “SC_Inicio_Comunicación” en RobotStudio 2023. (Diseñado por el autor del proyecto)

SC_Señales_Digitales

Este componente inteligente se encarga de la comunicación de señales digitales entre el PLC y RobotStudio. Para ello hay que configurar:

- **Dirección IP** en la que se realizará la comunicación.
- **Rack y Slot** en los que se encuentra el PLC en TIA Portal.
- **Número de entradas digitales.**
- **Número de salidas digitales.**
- **Tipo de señal, byte y bit** en TIA Portal.
- **Señal “Connect”**: esta señal debe ser activada para que exista comunicación entre este componente inteligente y el PLC.

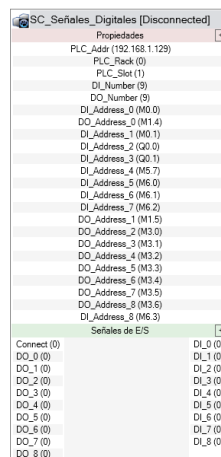


FIGURA 110. “SC_Señales_Digitales” en RobotStudio 2023. (Diseñado por el autor del proyecto)

Este componente inteligente ha sido descargado de internet, ya que no está incluido con las bibliotecas proporcionadas por RobotStudio. Es de uso gratuito. (DenisFR, 2018)

SC_Señales_Analogicas

“SC_Señales_Analogicas” es un componente inteligente creado para transferir el valor actual de las variables numéricas de TIA Portal a RobotStudio.

El funcionamiento del componente se detalla a continuación:

- **Señales de entrada:**
 - **“Actualizar_Valores”:** cuando esta señal se activa, se pasan a RAPID los valores de las variables asociadas.
 - **“Conectar”:** activa la señal “Connect” del componente “RSConnectGIOToSnap7” para conectar el PLC con RobotStudio y permitir la comunicación.
- **Bloque “RSConnectGIOToSnap7”:** este bloque, similar al componente “SC_Señales_Digitales”, ha sido descargado de Internet y configurado en RobotStudio (DenisFR, 2018). La configuración necesaria incluye:
 - **Dirección IP** en la que se realizará la comunicación.
 - **Rack y Slot** en los que se encuentra el PLC en TIA Portal.
 - **Número de entradas o salidas:** define la cantidad de entradas y salidas a utilizar. Cada señal será de 1 byte, y todas las entradas y salidas serán continuas, lo cual es importante tener en cuenta al programar en TIA Portal.
 - **Primer byte** donde se encuentran las señales analógicas de entrada y de salida.
- **Bloques “RapidVariable”:** cada uno de estos bloques está vinculado a su variable asociada en RAPID. Cuando se recibe la orden de transferir el valor a RAPID a través de la señal “Actualizar_Valores”, el bloque copia el valor del byte configurado en la variable de RAPID.

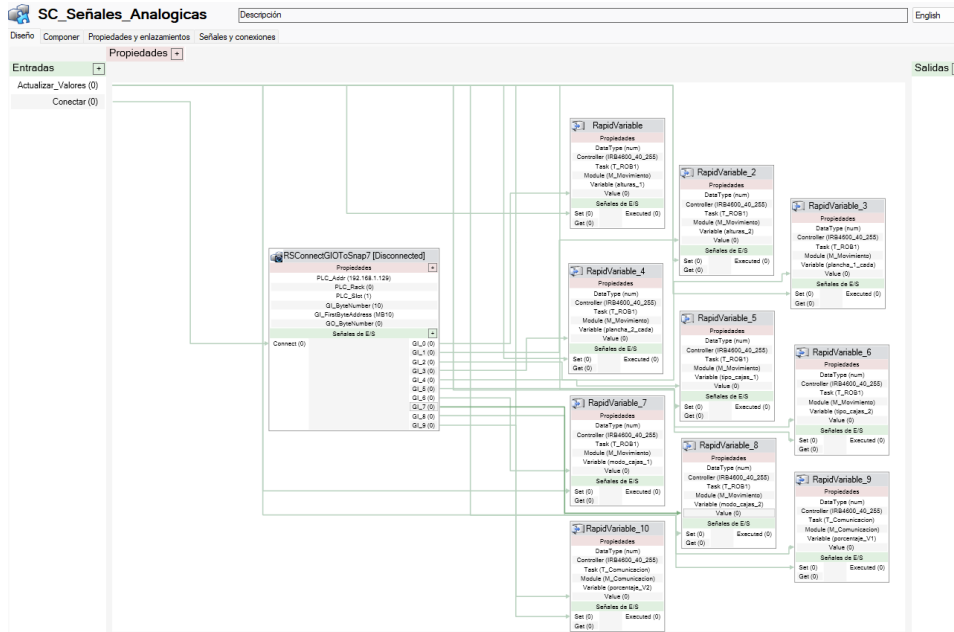


FIGURA 11.1. “SC_Señales_Analogicas” en RobotStudio 2023. (Diseñado por el autor del proyecto)

Lógica de la estación

La composición de todos los componentes anteriores junto con el controlador para el correcto funcionamiento de la estación tiene el siguiente aspecto:

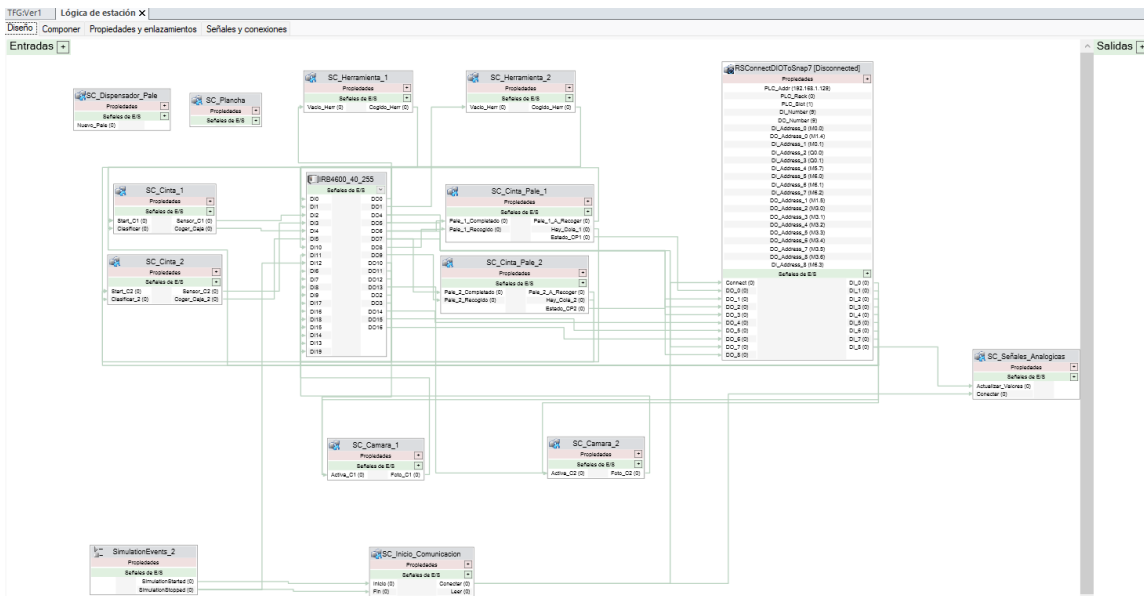


FIGURA 11.2. Lógica de estación del proyecto en RobotStudio 2023.

Gracias a esta imagen, se pueden ver todas las conexiones existentes entre los diversos componentes y señales de la estación, proporcionando una visión clara de cómo interactúan entre sí.

Capítulo 5.1.3. RAPID

Como se ha mencionado previamente, la estación robótica está equipada con dos robots: uno se ocupa de la elaboración de palés y el otro de identificar cajas defectuosas. El código se ha organizado en tres tareas diferenciadas, cada una con sus módulos, procesos y funciones particulares.

Estas tareas se describen a continuación:

- **"T_ROB1"**: contiene la programación del robot paletizador, asignando a "ROB_1" como su unidad mecánica principal.
- **"T_ROB2"**: incluye la programación del robot clasificador, utilizando a "ROB_2" como su unidad mecánica.
- **"T_Comunicacion"**: esta tarea es responsable de facilitar la comunicación entre el sistema de visión artificial desarrollado en Python y el robot clasificador "ROB_2". Sin embargo, no se asocia con ninguna unidad mecánica directamente.

La distribución de la programación en el lenguaje RAPID se detalla dentro de las tareas especificadas.

Capítulo 5.1.3.1. T_ROB1

Esta tarea está diseñada para gestionar los movimientos del robot paletizador y utilizar los parámetros proporcionados por el operario a través de la interfaz hombre-máquina (HMI) para la formación de palés.

Para optimizar la estructura del programa, se ha decidido dividirlo en los siguientes tres módulos:

Capítulo 5.1.3.1.1. CalibData

El módulo "CalibData" es esencial para definir los parámetros de las herramientas y los objetos de trabajo que interactúan con el robot. Dentro de este módulo la información es distribuida de la siguiente manera:

- **Herramientas de trabajo ("TCP_Herramienta_1"):** define la configuración de la herramienta usada por el robot.
- **Objetos de trabajo ("WO_..."):** configura espacios de trabajo como "WO_Pale", "WO_CP1", etc. Cada objeto de trabajo contiene unas coordenadas específicas y orientaciones que dictan dónde y cómo el robot debe interactuar con estos objetos en el espacio de trabajo.
- **Procedimientos globales ("Succiona" y "Suelta"):** controlan operaciones básicas de la herramienta del robot, como la activación del dispositivo para sujetar o soltar objetos. Estos procedimientos serán invocados desde otros módulos.

Capítulo 5.1.3.1.2. M_Main

Este módulo está dedicado exclusivamente a un procedimiento global, cuya función es invocar al procedimiento principal del módulo "M_Movimiento".

Esta metodología se ha adoptado con el objetivo de mejorar la organización y estructura del programa. Al centralizar las llamadas a procedimientos principales en un módulo específico, se facilita la comprensión del programa y se simplifica la implementación de futuras modificaciones o mejoras.

Capítulo 5.1.3.1.3. M_Movimiento

Este módulo centraliza toda la programación relacionada con los movimientos del robot paletizador. Además, gestiona la activación y desactivación de la mayoría de las señales digitales de la estación según sea necesario.

Para facilitar la comprensión de su funcionamiento, la explicación se apoya en un esquema que ilustra de manera general la ejecución del módulo. Este enfoque visual ayuda a clarificar el flujo de operaciones y las interacciones entre las diferentes partes del sistema, haciendo mucho más accesible la comprensión de los procesos y la secuencia de tareas que el robot lleva a cabo.

Antes se deben conocer los símbolos normalizados que se van a usar, los cuales están recogidos en la siguiente imagen: (Sanz, s.f.)

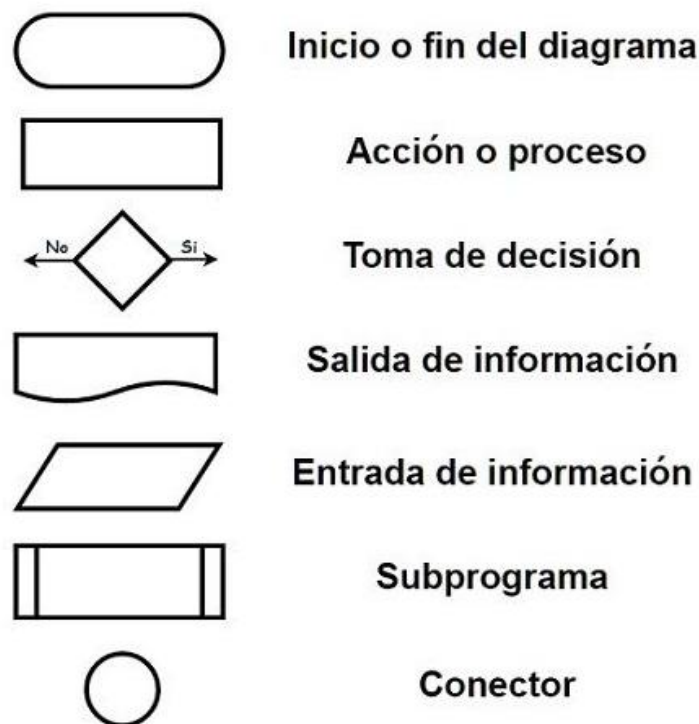


FIGURA 113. Símbolos normalizados para los diagramas de flujo. (Sanz, s.f.)

Nota: en los diagramas del proyecto utilizan símbolos ligeramente redondeados para representar acciones o procesos, debido a que la aplicación empleada no permite la creación de rectángulos perfectos. Para diferenciar estos símbolos de los bloques de inicio o fin, se ha optado por colorear los bloques de inicio o fin en verde y los bloques de acción o proceso en azul oscuro.

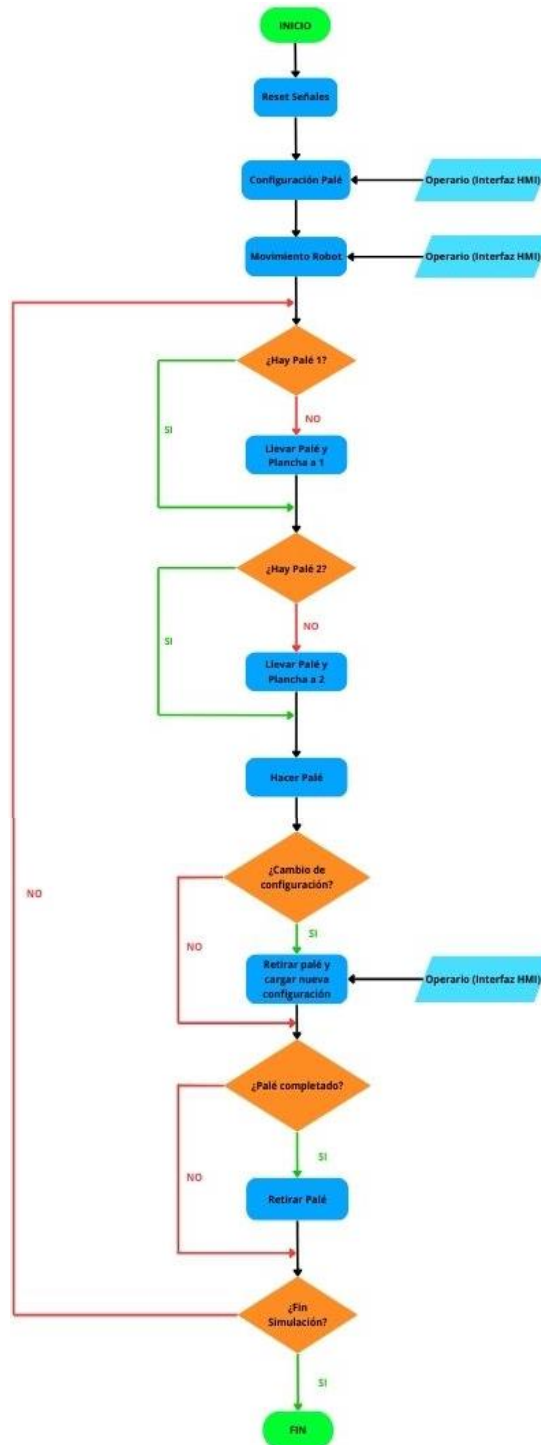


FIGURA 114. Diagrama de flujo para el proceso de paletizado.

Inicialización y configuración

El módulo comienza con la definición y configuración de objetos de trabajo, jointtargets, robtargets, velocidades y variables. Todos ellos constituyen parámetros básicos para el movimiento del robot y la ejecución del programa.

Procedimiento principal (“main_M_Movimiento”)

Este procedimiento encapsula la lógica de arranque y el bucle de operación continuo del robot:

- **Procedimiento “Inicio”:** se ejecuta al inicio de la simulación y restablece todas las señales y contadores. Prepara el robot configurando variables de control como el tipo de cajas, el modo de colocación y la alternancia entre diferentes tipos de cajas para cada palé.
- **Procedimiento “Programa_Principal”:** es un bucle infinito para asegurar que el robot este continuamente activo.

Flujo de trabajo principal

Dentro del procedimiento “Programa_Principal”, se decide entre:

- **Control de palés:** se determina si es necesario colocar un nuevo palé. Si al comenzar la simulación no hay palés disponibles, o si los palés en las cintas han sido retirados tras cambios desde la HMI, se activa el procedimiento "Coge_Palé".
- **Procedimiento “Funcionamiento_Normal”:** en este procedimiento se decide dinámicamente cómo manejar las cajas basándose en el estado de los sensores de las cintas y la configuración realizada por el operario desde la HMI.

Procedimientos específicos para manejar cajas

En el “Funcionamiento_Normal” el sistema decide como manejar las cajas entre:

- **Alternancia:** si el sistema está configurado para alternar el tipo de cajas en un palé específico, cambia entre manejar cajas de vainilla y de chocolate según sea necesario.
- **Procedimiento “Manejar_Caja_Sin_Alterar”:** cuando cada palé se realiza con un solo tipo de cajas, el robot dirige las cajas directamente al palé correspondiente, según la configuración realizada y disponibilidad.

Funciones y procedimientos de apoyo

En los siguientes procedimientos es donde se organizan los movimientos del robot:

- **“Coge_Caja_Vainilla” y “Coge_Caja_Choco”:** para recoger cajas de las cintas transportadoras utilizando el sistema de succión creado para la herramienta del robot.
- **“Llevar_Caja_A_Destino”:** transporta y coloca la caja en el palé designado.
- **“Coge_Plancha”:** se encarga de recoger y colocar planchas de cartón en los palés para conseguir una mayor estabilidad en el transporte.
- **“Posición_Caja”:** calcula la nueva posición en la que el robot debe colocar la caja en el palé.
- **“Pale_Completado”:** verifica si un palé está completado para seguir apilando cajas en él o, por el contrario, darle salida y colocar un nuevo palé.

Manejo de interrupciones

Las denominadas **TRAP** son esenciales para la gestión de interrupciones, siendo eventos inesperados o condiciones específicas que requieren ser atendidas inmediatamente, como paradas de emergencia o nuevas configuraciones por parte del operario.

Es importante destacar que, aunque la instrucción de paro de emergencia se emite desde la HMI en TIA Portal, se ha optado por configurar el cese del movimiento desde aquí por su sencillez y eficiencia.

Capítulo 5.1.3.2. T_ROB2

En esta sección, se detalla la programación de todos los movimientos del robot clasificador, tomando en cuenta diversos parámetros que se describirán más adelante.

Para mejorar la comprensión de esta explicación, se ha creado un diagrama que acompaña el texto. Este recurso visual es especialmente valioso para visualizar la interacción entre esta tarea y la tarea T_Comunicación, destacando cómo ambas se complementan en la operación del sistema.

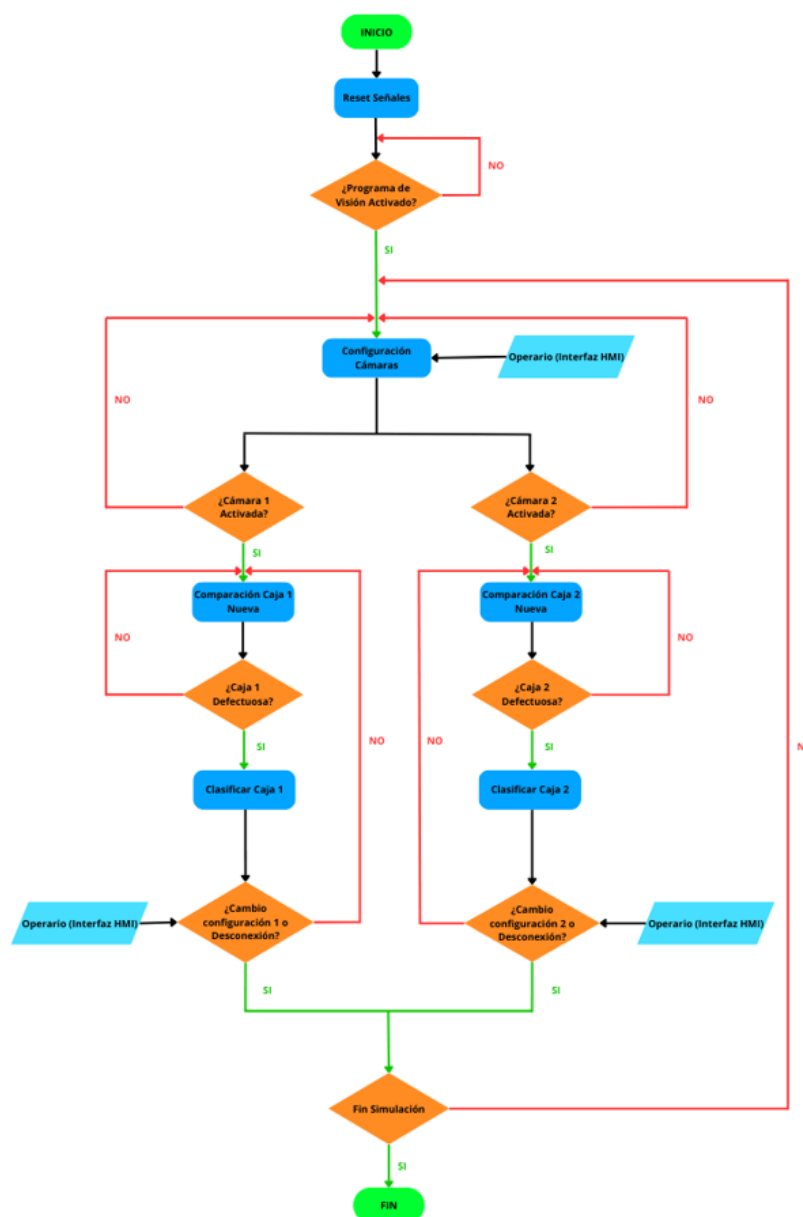


FIGURA 115. Diagrama de flujo para el proceso de clasificado.

Al igual que en la tarea “T_ROB1”, se ha dividido en tres módulos.

Capítulo 5.1.3.2.1. CalibData

Tal y como se ha realizado en “T_ROB1”, se definen las herramientas del robot clasificador, objetos de trabajo que interactúan con él, etc. Por ello, abarca información sobre:

- **Herramientas de trabajo (“TCP_Herramienta_2”)**: establece la configuración de la herramienta que usa el robot clasificador.
- **Objetos de trabajo (“WO_...”)**: en este programa se usan los siguientes objetos de trabajo: “WO_Coge_1”, “WO_Coge_2” y “WO_Pale_Clasificar”. Cada uno con unas coordenadas y orientación diferentes.
- **Procedimientos globales a la tarea (“Succiona_2” y “Suelta_2”)**: mediante estos procedimientos se gestiona la herramienta del robot clasificador para coger o soltar cajas.

Capítulo 5.1.3.2.2. M_Main

“M_Main” es el módulo principal del programa, desde aquí se hace una llamada al módulo “M_Clasificar”. Este último, es el que concentra toda la programación pertinente.

Capítulo 5.1.3.2.3. M_Clasificar

El módulo “M_Clasificar” coordina el proceso de clasificar cajas con los datos que llegan del programa de visión a través de la tarea “T_Comunicacion”.

Desglosando el código, este módulo contiene:

Inicialización y configuración

Comienza con la definición y configuración de jointtargets, robtargets, velocidades y variables. Todos ellos constituyen parámetros básicos para el movimiento del robot y la ejecución del programa.

Destacan las variables “clasifica_C1” y “clasifica_C2”, ya que se encargan de gestionar cuando una caja tiene que ser clasificada o no.

Procedimientos principales

- **Procedimiento “main_M_Clasificar”:** actúa como el bucle principal que asegura la operación continua del robot. Al iniciar la simulación, este procedimiento activa "Inicio_2", y luego entra en un ciclo continuo al llamar repetidamente a "Programa_Principal".
- **Procedimiento “Inicio_2”:** se ejecuta al comienzo de la simulación, reseteando todos los valores de las señales utilizadas en esta tarea y configurando el robot para la clasificación. A su vez, vincula las rutinas TRAP con las señales digitales.
- **Procedimiento “Programa_Principal”:** en este procedimiento se maneja la lógica de clasificación. Dependiendo de las señales del programa de visión, se ajustan las variables "clasifica_C1" y "clasifica_C2", determinando si la caja en cuestión debe ser retirada o continuar en el proceso.

Procedimientos y funciones auxiliares

- **Procedimiento “Resetear_Y_Prog”:** borra todas las configuraciones establecidas y prepara el sistema para una nueva configuración. Se activa cuando el operario modifica la configuración inicial durante la simulación.
- **Función “Posicion”:** calcula y ajusta la posición donde la caja defectuosa debe ser colocada en el palé, basándose en el número de cajas que han sido retiradas.

Rutinas TRAP

- **“Trap_Idi4” y “Trap_Idi5”:** cada vez que se detecta una caja y hay que clasificarla. Indican al robot que la caja está en la zona apropiada de la cinta para recoger cajas defectuosas.
- **“Trap_Idi8” y “Trap_Idi9”:** actúan como punto de inicio para el proceso de clasificación, activando las variables “clasifica_C1” o “clasifica_C2” según corresponda.
- **“Trap_Idi14”, “Trap_Idi15” y “Trap_Idi18”:** gestionan el control del movimiento del robot, incluyendo el inicio, el paro y el reinicio tras una parada.

Capítulo 5.1.3.3. T_Comunicacion

La programación de esta tarea se realiza en el módulo "M_Comunicacion", que está diseñado para optimizar la interacción entre el robot y el sistema de visión artificial mediante conexiones de red. Este módulo se encarga de configurar y mantener la comunicación a través de sockets, además de procesar los comandos recibidos y gestionar eventos de comunicación. A continuación, se detalla cada componente del código:

Variables y configuración importante

Al inicio del módulo, se definen una serie de variables que serán claves para una correcta comunicación:

- **Variables para los porcentajes ("porcentaje_V1" y "porcentaje_V2"):** estas variables almacenan los valores recibidos desde la HMI referentes al porcentaje mínimo aceptable para considerar una caja como adecuada.
- **Strings para los porcentajes ("str_porc_V1" y "str_porc_V2"):** transforman los valores de porcentaje en formatos de cadena de caracteres, lo que facilita su envío a través de la red hacia el programa de visión en Python.
- **Identificadores de interrupción ("Idi6", "Idi7", "Idi13_Conectar", "Idi13_Desconectar"):** estos identificadores corresponden a diferentes señales digitales y se utilizan para gestionar distintos aspectos de la comunicación a través de rutinas TRAP.

Proceso principal para la comunicación ("main_M_Comunicacion")

Dentro de este procedimiento se llevan a cabo las siguientes acciones:

- **Procedimiento "Inicio_Comunicacion":** se ejecuta al comienzo de la simulación y prepara todo lo necesario para iniciar la comunicación. Esto incluye eliminar configuraciones antiguas (interrupciones TRAP previas), resetear señales digitales y conexión de dichas señales digitales con rutinas TRA específicas.
- **Bucle de comunicación:** después de ejecutar el procedimiento inicial "Inicio_Comunicacion", se ejecuta un bucle infinito que se encuentra a la escucha a través del socket. Cuando se reciben datos, se procesan

para activar señales o realizar acciones para clasificar cajas en respuesta a los comandos recibidos.

Manejo de interrupciones TRAP

Dado que las cajas defectuosas tienen que ser apartadas inmediatamente, y que la información manejada por el robot clasificador y el sistema de visión debe ser muy rápida, se ha optado por una programación con rutinas TRAP para disminuir retrasos, paros de cintas y errores.

Las rutinas TRAP que componen este módulo son:

- **“Trap_Idi6” y “Trap_Idi7”:** gestionan las señales de las cintas transportadoras. Si el sistema de visión está conectado, envían un comando a este último para indicar de qué cinta se ha tomado la foto.
- **“Trap_Idi13_Conectar”:** establece conexión con el programa de visión artificial en Python. Crea y configura un socket, espera por una conexión entrante y envía una confirmación una vez establecida. A su vez, envía los valores que el operario ha configurado para el sistema de visión.
- **“Trap_Idi13_Desconectar”:** cierra la conexión de manera ordenada notificando al programa de visión, asegurando que se liberen todos los recursos de la red.

Capítulo 5.2. Desarrollo en TIA Portal

Capítulo 5.2.1. Controlador lógico programable (PLC)

Como se mencionó anteriormente, se ha utilizado el PLC 1214C DC/DC/DC y se ha programado utilizando el lenguaje KOP, también conocido como LADDER, que es el más utilizado en la actualidad.

Para facilitar la programación del PLC, se ha adoptado una estructura modular, lo que permite una mejor organización y simplifica tanto futuros cambios como la comprensión del programa por parte de otros programadores.

Por lo tanto, se han segmentado tanto los bloques del programa como las variables utilizadas en distintos grupos, como se muestra a continuación.



FIGURA 116. Organización de bloques de programa del proyecto en TIA Portal V16.



FIGURA 117. Organización de las variables del proyecto en TIA Portal V16.

Main [OB1]

El bloque Main OB1 es el núcleo del programa en TIA Portal y es fundamental para los programas de usuario. Se ejecuta de manera cíclica, lo que lo hace perfecto para incluir segmentos de programación esenciales y realizar llamadas a otros bloques de función o datos según sea necesario.

El OB1 gestiona el flujo del programa, asegurando que todas las tareas se ejecuten en cada ciclo. Esto incluye leer las entradas, ejecutar el programa del usuario y actualizar las salidas. Al llamar a otros bloques de función, el OB1 mantiene el código modular y organizado, lo cual es crucial para el mantenimiento y la expansión del sistema.

En este proyecto, el OB1 se organiza en segmentos claros y estructurados, como se muestra en la figura 118. Cada segmento realiza llamadas a bloques de función (FB) o funciones (FC) de manera ordenada, lo que facilita la gestión y asegura que todas las funciones se ejecuten correctamente.



FIGURA 118. Organización del bloque OB1 del proyecto en TIA Portal V16.

Un ejemplo de cómo se efectúa la llamada a cada FB o FC desde el bloque OB1 en este proyecto se puede observar en la figura 119.

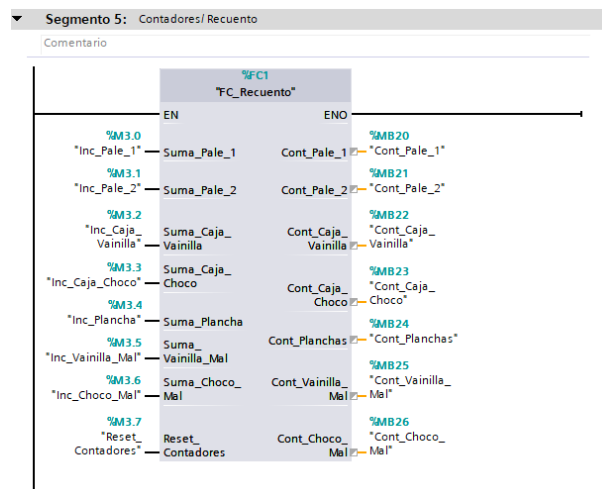


FIGURA 119. Llamada a la función FC_Recuento (FC1) desde OB1 en TIA Portal V16.

Al realizar una llamada a una función, se especifica qué variable del PLC se corresponde con cada variable de la función (FC), o del bloque de datos (DB) en el caso de una llamada a un bloque de función (FB). En la figura 119 se puede observar cómo las variables de entrada se enumeran en la columna de la izquierda ("EN"), mientras que las variables de salida aparecen en la columna de la derecha ("ENO").

Esto asegura que los datos se transmitan y procesen correctamente entre el PLC y los diferentes bloques de función.

FC_Recuento [FC1]

Esta FC se encarga de actualizar el valor de todos los contadores utilizados en el proyecto y de resetearlos cuando se solicite desde la HMI.

La llamada a esta FC se mostró en la figura 119.

Su organización interna se presenta en la figura 120, donde cada segmento ha sido creado para actualizar el valor de cada contador.

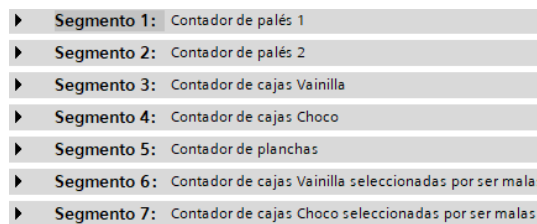


FIGURA 120. Organización de la función FC1 en el proyecto en TIA Portal V16.

Dado que las señales de entrada de la FC son señales digitales, se ha utilizado el bloque CTU (contador ascendente) para gestionar estas entradas y el valor de los contadores, como se muestra a continuación.

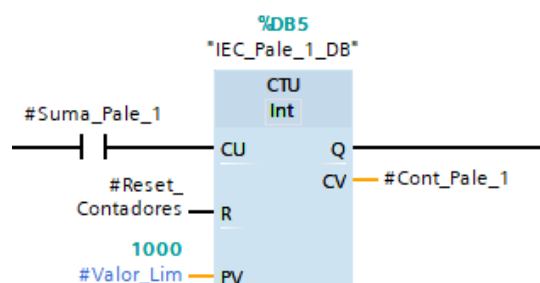


FIGURA 121. Contador ascendente de palés 1 del proyecto en TIA Portal V16.

El funcionamiento de este bloque es el siguiente:

- **Incremento del contador:** la señal “#Suma_Pale_1” activa la entrada CU del contador. Cada vez que “#Suma_Pale_1” se activa, el contador incrementa su valor. Este valor actualizado se almacena en “#Cont_Pale_1”.
- **Reinicio del contador:** la señal “#Reset_Contadores” activa la entrada “R” del contador. Cuando “#Reset_Contadores” se activa, el contador se reinicia a cero, restableciendo el valor de “#Cont_Pale_1”.
- **Valor preseleccionado:** el contador tiene un valor preseleccionado de 1000 “#Valor_Lim”. El contador seguirá incrementándose hasta alcanzar este valor. Cuando el valor actual del contador (“CV”) alcanza el valor preseleccionado, la salida “Q” se activa, indicando que el contador ha alcanzado el límite definido.

La programación de los demás contadores sigue prácticamente el mismo esquema, ajustándose únicamente a las variables específicas de cada contador.

FB_Robot_Clasificador [FB1] y FB_Robot_Paletizador [FB2]

Estos dos bloques de función, diseñados para programar la lógica de los robots desde TIA Portal, están actualmente vacíos porque se ha optado por realizar toda la programación en RobotStudio.

Sin embargo, se han dejado disponibles para permitir futuras mejoras o para trasladar parte de la programación de los robots a TIA Portal si fuese necesario.

FB_Cintas [FB3]

En este FB se ha programado la lógica de las cintas transportadoras de cajas, gestionando tanto su marcha como su paro.



FIGURA 122. Organización de la FB3 del proyecto en TIA Portal V16.

FB_Emergencia [FB4]

Tal como su nombre indica, este FB está destinado a programar el paro de emergencia de ciertos elementos. Cabe destacar que, como se mencionó anteriormente, parte de la programación se ha realizado desde RobotStudio.

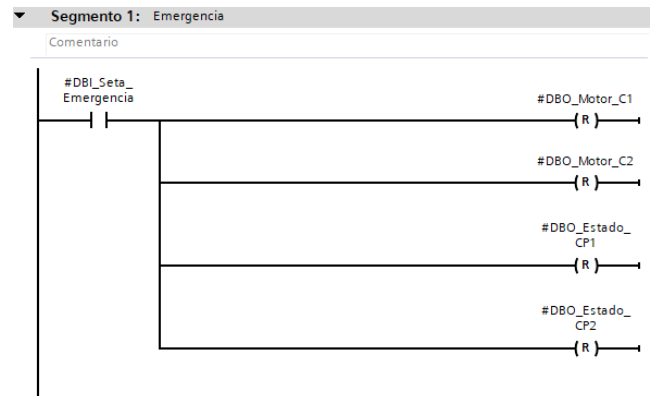


FIGURA 123. Programación FB4 del proyecto en TIA Portal V16.

FB_Programación [FB5]

En este FB, se activa durante dos segundos la variable encargada de actualizar todos los valores programables por el usuario desde la HMI, permitiendo que el componente inteligente de RobotStudio lea y actualice dichos valores en RAPID.

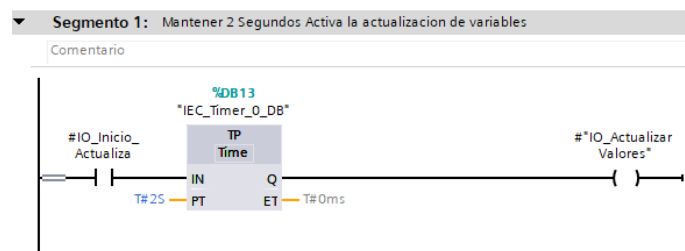


FIGURA 124. Programación FB5 del proyecto en TIA Portal V16.

Capítulo 5.2.2. Interfaz Hombre – Máquina (HMI)

Cabe recordar que la Interfaz Hombre-Máquina (HMI) seleccionada para este proyecto es la HMI KTP1 1200 Basic PN de Siemens.

Capítulo 5.2.2.1. Imágenes

La programación de la HMI en TIA Portal se realiza mediante las denominadas "imágenes", que son las distintas pantallas por las que el usuario puede navegar.

De manera esquemática, la navegación por las distintas pantallas en este proyecto se representa con el siguiente esquema:

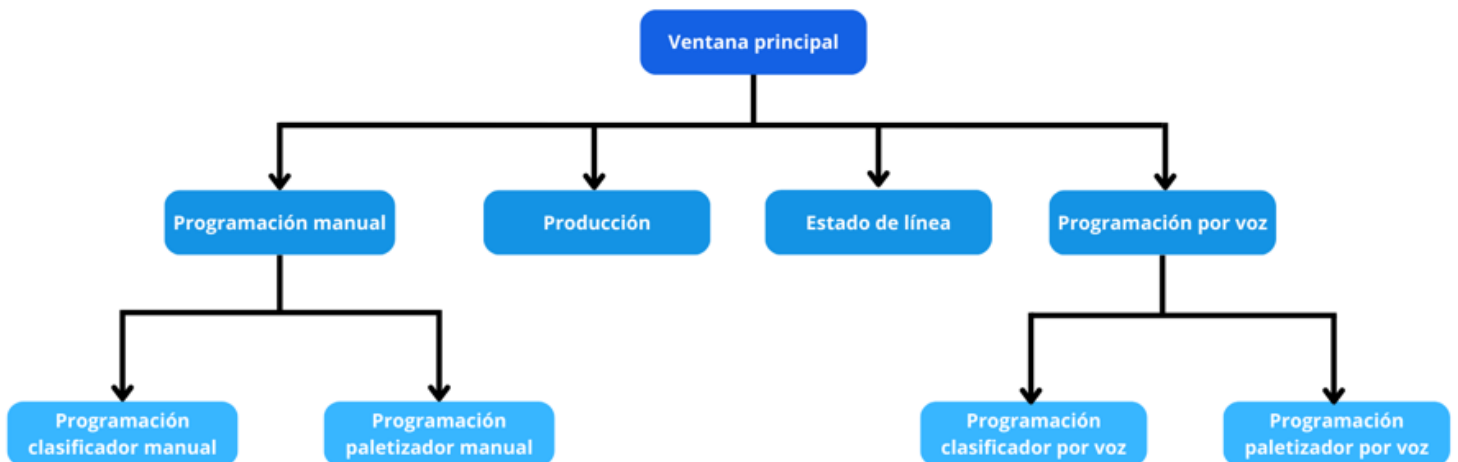


FIGURA 125. Esquema de las pantallas de la HMI del proyecto.

Los elementos comunes a todas las pantallas son los siguientes:

- **Barra superior de navegación:**
 - Es una barra desplegable ubicada en la parte superior permitiendo al usuario acceder directamente a cualquier otra pantalla.
 - Por ejemplo, si la pantalla actual es "Programación clasificador manual" y se selecciona "Estado Línea" desde esta barra, la vista cambiará directamente sin pasar por pantallas intermedias.
 - Esta barra ha sido diseñada para facilitar la navegación y mejorar la usabilidad de la interfaz.
- **Fecha y hora:** todas las pantallas muestran la fecha y hora actuales para mantener al usuario informado del tiempo real.
- **Botón Atrás:**
 - Botón de color gris claro situado en la parte superior izquierda.
 - Permite volver a la imagen anterior según el esquema.
 - Por ejemplo, si la imagen activa es "Programación manual" y se pulsa este botón, se cambiará a la imagen "Ventana principal".
- **Botón de Paro de Emergencia:**
 - Este botón activa la seta de emergencia, deteniendo cualquier movimiento de la estación hasta que el operador lo desactive manualmente.
 - Cuando está activado, el botón se mantiene parpadeando para indicar su estado.
 - La activación de este botón afecta a todas las pantallas, es decir, si se activa y se cambia de vista, la seta de emergencia seguirá activada y podrá desactivarse desde cualquier pantalla.
- **Botones inferiores:** en la parte inferior de cada pantalla se encuentran botones para las siguientes funciones:
 - Volver a la imagen anterior según el esquema anterior.
 - Visualizar avisos no atendidos y pendientes generados durante la simulación.
 - Iniciar sesión con otra cuenta de usuario.
 - Apagar la HMI.

A continuación, se explican detalladamente cada una de las imágenes de la Interfaz Hombre-Máquina (HMI) diseñada, junto con los elementos utilizados y su propósito.

Ventana Principal

Esta es la pantalla principal de la HMI, donde se inicia el programa y desde la cual se puede acceder a cualquier otra pantalla.

Esta pantalla incluye los siguientes elementos:

- Elementos comunes a todas las imágenes ya explicados.
- Logos y textos.
- Botones:
 - **Estado Línea (naranja)**: activa la imagen “Estado de línea”.
 - **Producción (naranja)**: activa la imagen “Producción”.
 - **Programación con voz (naranja)**: activa la imagen “Programación por voz”.
 - **Programación manual (naranja)**: activa la imagen “Programación manual”.
 - **Activar Robots (verde)**: permite dar comienzo al movimiento de los robots. Mientras no se active, ningún robot se moverá. Después de una parada de emergencia, deberá volver a activarse para que los robots reanuden sus movimientos.
 - **Cerrar sesión (gris)**: cierra la interfaz hombre – máquina.



FIGURA 126. Imagen “Ventana Principal” de la HMI implementada.

Producción

Esta imagen surge de la necesidad de contar con indicadores que reflejen el estado de la producción en la estación.

Se ha diseñado a modo de tabla para facilitar la visualización del usuario, y cuenta con:

- **Elementos comunes a todas las imágenes.**
- **Botones:**
 - **Reset contadores (gris oscuro):** actualiza a cero todos los contadores de la tabla.
- **Indicadores numéricos:** reflejan los valores de los contadores programados en el PLC, mostrando el número de cajas apartadas de cada tipo, la cantidad de palés producidos y otros parámetros relevantes. Cada contador tiene su propio indicador, el cual se actualiza continuamente a medida que se producen cambios en la simulación.

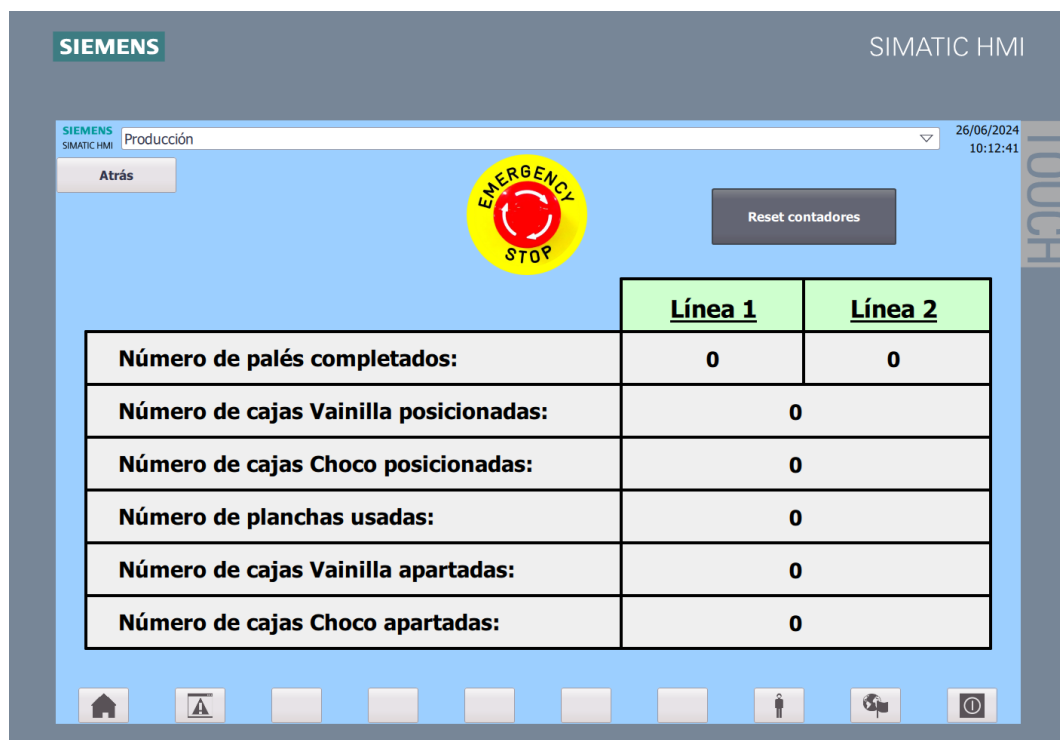


FIGURA 127. Imagen "Producción" de la HMI implementada.

Estado de Línea

En esta ventana se muestra el funcionamiento de las cintas transportadoras, donde se controla el movimiento de las cintas transportadoras de cajas y se visualiza el estado de las cintas transportadoras de palés, cuyo control es automático desde RobotStudio.

Para ello se ha recurrido a los siguientes elementos:

- **Elementos comunes a todas las imágenes.**
- **Botones:**
 - **Marcha (borde verde):** inicia el movimiento de la cinta transportadora de cajas de la línea correspondiente. Para mejorar la interactividad con la interfaz, al pulsarlo, el botón se iluminará en verde.
 - **Paro (borde rojo):** detiene el movimiento de la cinta transportadora de cajas de la línea asociada. Nuevamente, para facilitar la interactividad con la interfaz, al pulsarlo, el botón se iluminará en rojo.
- **Indicadores de estado (fondo blanco):** existe uno para cada cinta transportadora de palés que muestra su estado actual. Si la cinta está activa, se visualizará el texto "Activa" en color verde, mientras que si la cinta está parada, el texto "Inactiva" se mostrará en rojo.
Nota: Estos recuadros no son botones, ya que el movimiento de estas cintas es automático y está controlado en RobotStudio, activándose y desactivándose cuando se completa el palé.
- **Animaciones de cintas:** se ha creado una animación para cada cinta transportadora, que refleja su estado en tiempo real. Si la cinta está en movimiento en la estación, la animación en la interfaz mostrará cómo se mueve la caja o el palé de cajas en la pantalla. Del mismo modo, si la cinta está parada, la animación también estará parada.

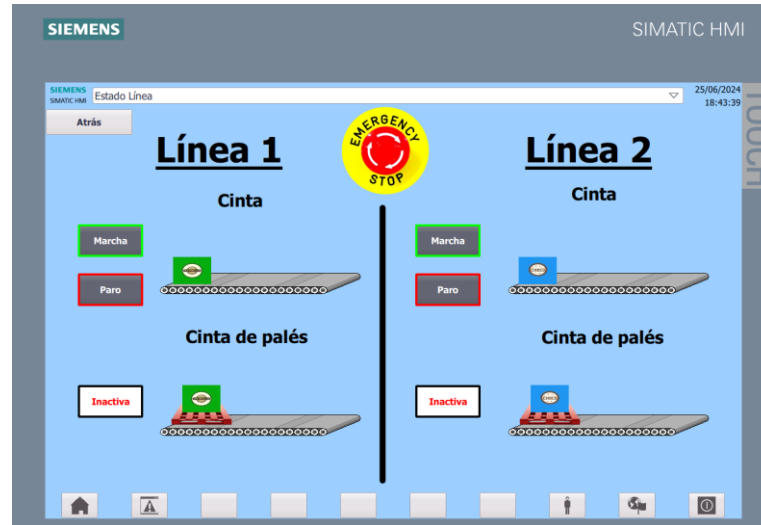


FIGURA 128. Imagen "Estado Línea" de la HMI implementada.

Programación manual

Esta ventana es muy sencilla y actúa como un enlace entre la "Ventana Principal" y las imágenes para programar manualmente de forma individual.

Cuenta con los siguientes elementos:

- Elementos comunes a todas las imágenes.
- Botones:
 - Programación Robot Clasificador (botón central izquierdo gris oscuro): activa la pantalla donde se puede programar manualmente el robot clasificador, utilizando la pantalla táctil.
 - Programación Robot Paletizador (botón central derecho gris oscuro): activa la pantalla donde se puede programar manualmente el robot paletizador, utilizando la pantalla táctil.

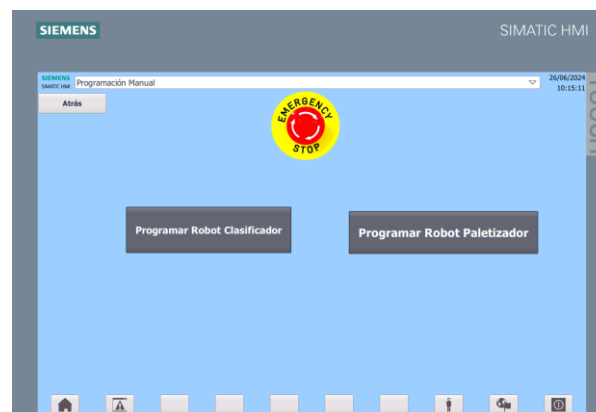


FIGURA 129. Imagen "Programación Manual" de la HMI implementada.

Programación clasificador manual

Desde esta pantalla, el usuario puede programar manualmente el sistema de visión artificial utilizando la pantalla táctil de la HMI.

Para ello, la pantalla cuenta:

- **Elementos comunes a todas las imágenes.**
 - **Botones:**
 - **Conectar (gris oscuro):** conecta el programa de visión artificial creado en Python con la estación de RobotStudio. Cuando la conexión se establece, el texto del botón cambia a "Desconectar", y cuando esta finaliza el texto cambia a "Conectar".
 - **Cámaras de visión (en la tabla aparecen en color blanco sobre fondo rojo):** son botones que permiten activar o desactivar las cámaras de cada una de las dos cintas transportadoras de forma independiente. Cuando la cámara está activada, el fondo del botón cambiará de rojo a verde para indicar su estado.
 - **Porcentaje de similitud (número en la tabla debajo de la cámara):** al pulsar este botón, se despliega un teclado en la pantalla que permite al usuario ingresar el porcentaje de similitud necesario entre las cajas y una caja de referencia ideal para que no sean clasificadas como defectuosas. Este porcentaje es independiente para cada línea de producción.
 - **Indicador de estado del programa de visión (a la derecha del botón "Conectar"):** indicador visual compuesto por dos flechas, cuyo color representa el estado de conexión del programa de visión: verde cuando está conectado y rojo cuando está desconectado.
- Nota: en caso de que el programa de visión no esté conectado, el robot clasificador no retirará ninguna caja defectuosa.



FIGURA 130. Imagen "Programación clasificador manual" de la HMI implementada.

Programación paletizador manual

Gracias a esta interfaz, se define la programación de la disposición deseada para los palés manualmente.

Entre los elementos de esta imagen:

- **Elementos comunes a todas las imágenes.**
- **Desplegables de selección:** para cada parámetro de configuración del palé, existe un desplegable con las opciones disponibles correspondientes. En algunos casos, como en el tipo de cajas o en el modo de colocación de las cajas, dependiendo de la opción seleccionada, aparece una representación ilustrativa al lado que confirma la elección.

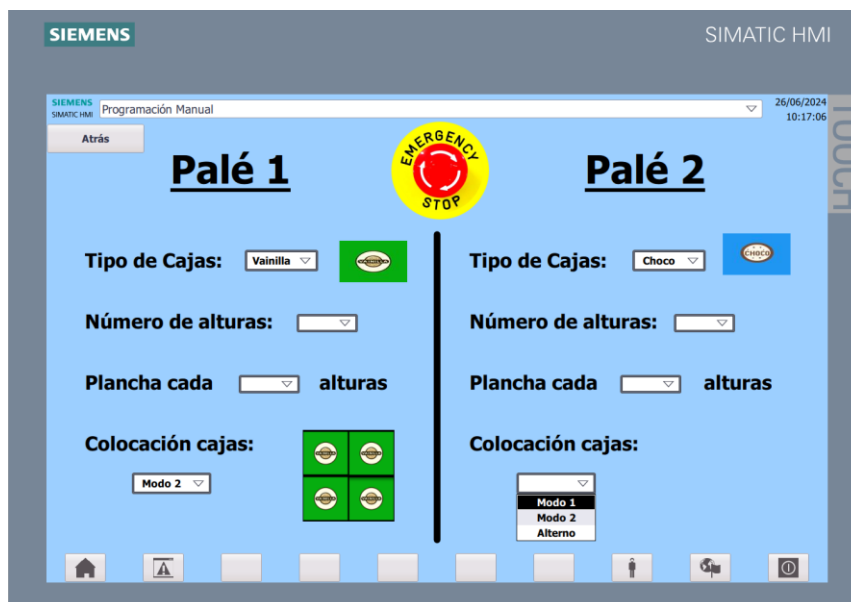


FIGURA 131. Imagen "Programación paletizador manual" de la HMI implementada.

Programación por voz

Esta imagen es prácticamente idéntica a la imagen "Programación manual" en cuanto a su estructura. Sirve como enlace entre las pantallas donde el usuario configura los distintos parámetros y la "Ventana Principal".

Los elementos que la componen son:

- **Elementos comunes a todas las imágenes.**
- **Botones:**
 - **Programar Robot Clasificador (botón central izquierdo gris oscuro):** activa la pantalla donde se puede programar con voz el robot clasificador.
 - **Programar Robot Paletizador (botón central derecho gris oscuro):** activa la pantalla donde se puede programar con voz el robot paletizador.



FIGURA 132. Imagen "Programación por voz" de la HMI implementada.

Programación clasificador por voz

Desde esta pantalla, el robot clasificador se programa mediante comandos de VOZ.

Para realizar esta pantalla se ha recurrido a los siguientes elementos:

- **Elementos comunes a todas las imágenes.**
- **Botones:**
 - **Conectar (gris oscuro):** funciona de la misma manera que el botón “Conectar” de la imagen “Programación clasificador manual” previamente explicado
 - **Iniciar programación (botones gris oscuro):** existe un botón específico para cada línea. Al pulsar este botón, se inicia la programación por voz. El sistema pregunta al usuario cómo desea configurar los parámetros, y el usuario deberá responder a cada consulta para completar la configuración.
- **Indicadores:**
 - **Estado programa de visión (a la derecha del botón “Conectar”):** su funcionamiento es idéntico al explicado anteriormente en la imagen “Programación clasificador manual”.
 - **Escucha (imagen de grabadora):** son imágenes programadas de manera que, cuando el sistema esté en modo de escucha, la figura roja de la grabadora correspondiente cambiará a verde para indicarlo al operario. Si la figura permanece en rojo, significa que el sistema no está en modo de escucha.
 - **Estado de cámara:** existen dos indicadores para cada línea que muestran el estado de la cámara de visión: un texto que indica "Inactiva" o "Activa" y una representación de la cámara. En función de la respuesta del operario a las preguntas del sistema de voz, estos indicadores en la HMI cambiarán, así como el estado de las cámaras en la estación de RobotStudio.
 - **Porcentaje de similitud:** en este caso, se trata de un indicador de texto que muestra el valor que el usuario ha comunicado al sistema mediante su voz. Este indicador se actualiza instantáneamente en cuanto el usuario proporciona la información.



FIGURA 133. Imagen "Programación clasificador por voz" de la HMI implementada.

Programación paletizador por voz

En esta imagen, la configuración para la realización de cada uno de los palés se programa mediante comandos de voz.

Se ha realizado con los siguientes elementos:

- **Elementos comunes a todas las imágenes.**
- **Botones:**
 - **Inicio programación del palé (botones gris oscuro):** hay un botón específico para la programación de cada palé. Al pulsar dicho botón, el sistema comenzará a hacer preguntas al operario hasta completar todos los parámetros necesarios según las respuestas proporcionadas.
- **Indicadores:**
 - **Escucha (imagen de grabadora):** de la misma manera que antes, cuando el sistema está en modo de escucha, la imagen de la grabadora correspondiente cambia a color verde. Si la grabadora está en rojo, significa que el sistema no está en modo de escucha.
 - **Configuración realizada (recuadros blancos):** estos indicadores muestran los parámetros que el usuario ha transmitido por voz a la interfaz. Se actualizan cada vez que se reconoce una opción válida.

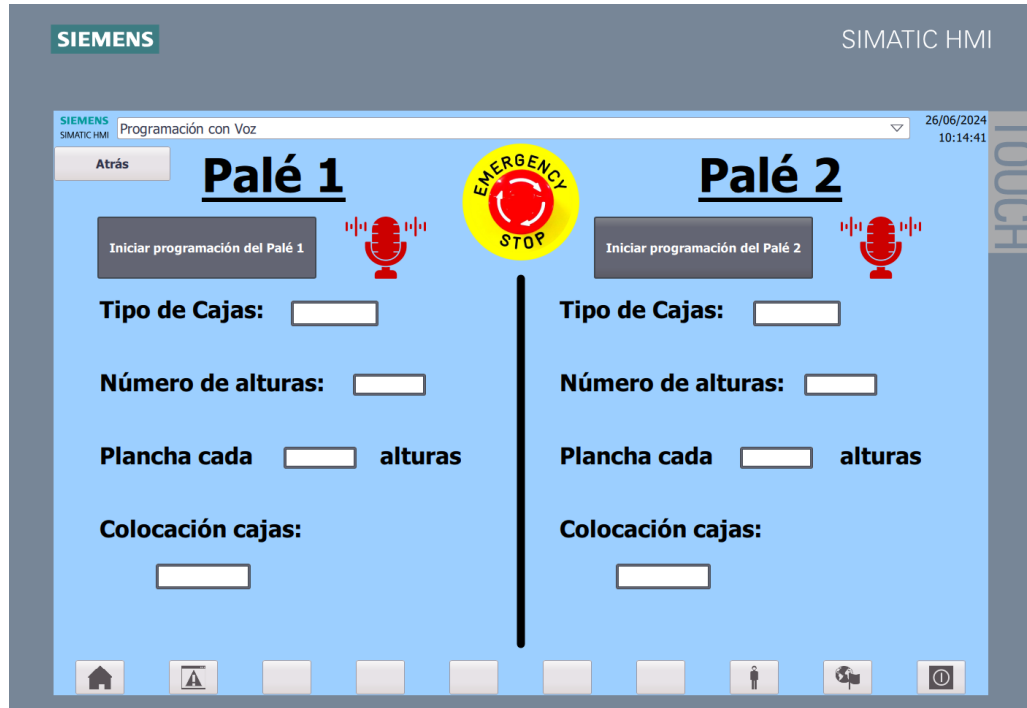


FIGURA 134. Imagen "Programación paletizador por voz" de la HMI implementada.

Capítulo 5.2.2.2. Programación de la HMI

Una vez explicados los elementos presentes en las imágenes de la HMI, se procede a detallar cómo se realiza su programación. Para ello, la programación se organiza en tres grupos principales comunes a todos los elementos. Estos grupos se encuentran en la sección denominada "Ventana de propiedades, información y diagnóstico", cuya explicación completa se ha ofrecido en el capítulo 4.2.1. TIA Portal.

Programación de propiedades

La configuración de propiedades de un elemento permite definir sus características visuales. Para programar un elemento, como un botón, se presenta en la figura 135 la lista de propiedades disponibles y las opciones dentro de cada propiedad. Es importante señalar que esta lista de propiedades varía mínimamente entre diferentes elementos.

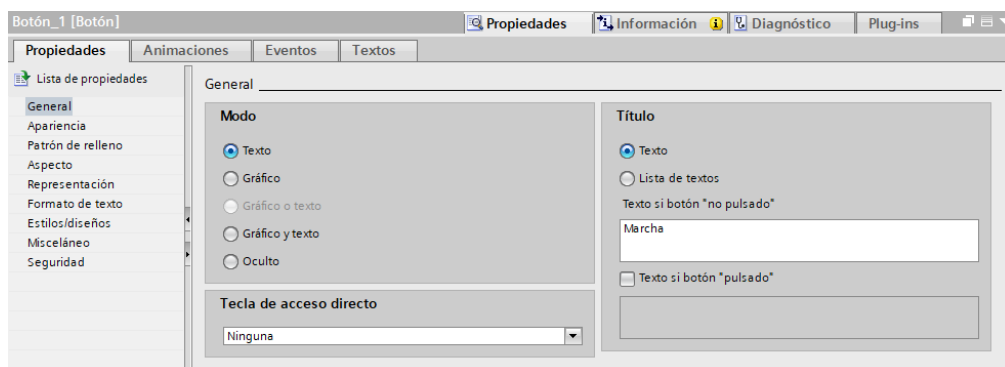


FIGURA 135. Programación de las propiedades de un botón en TIA Portal V16.

De forma sencilla, la finalidad de cada propiedad es:

- **General:** ajusta el modo de visualización (texto, gráfico, ambos u oculto) y asigna teclas de acceso directo si es necesario. Desde esta propiedad se configura el cambio de texto o de gráfico. Por ejemplo, en este proyecto, para cambiar la imagen de la grabadora roja a verde, se requiere la siguiente configuración expuesta:

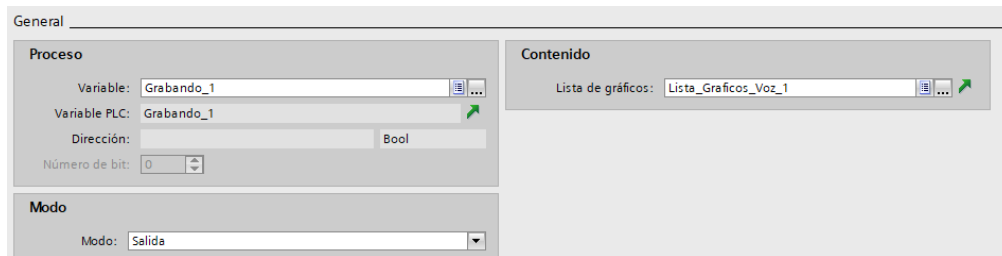


FIGURA 136. Configuración de propiedades para cambio de gráfico en TIA Portal V16.

- **Proceso:** se debe seleccionar la variable que gestionará qué gráfico o texto aparecerá en función del valor de dicha variable.
- **Modo:** indica si el elemento de la interfaz es de entrada, salida o ambos. En el ejemplo de la figura 136 es únicamente de salida, ya que solo sirve para indicar.
- **Contenido:** se debe relacionar el elemento de la HMI con una lista de gráficos o de textos configurados. El acceso a estas listas en TIA Portal se realiza a través de la opción “Listas de textos y gráficos” en el área “Árbol del proyecto”. En este trabajo se han creado las siguientes:

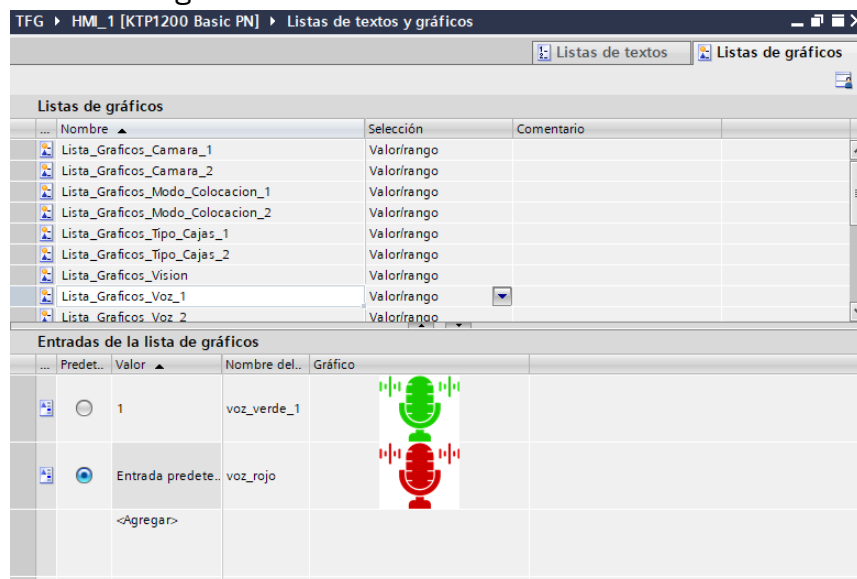


FIGURA 137. Lista de gráficos del proyecto en TIA Portal V16.

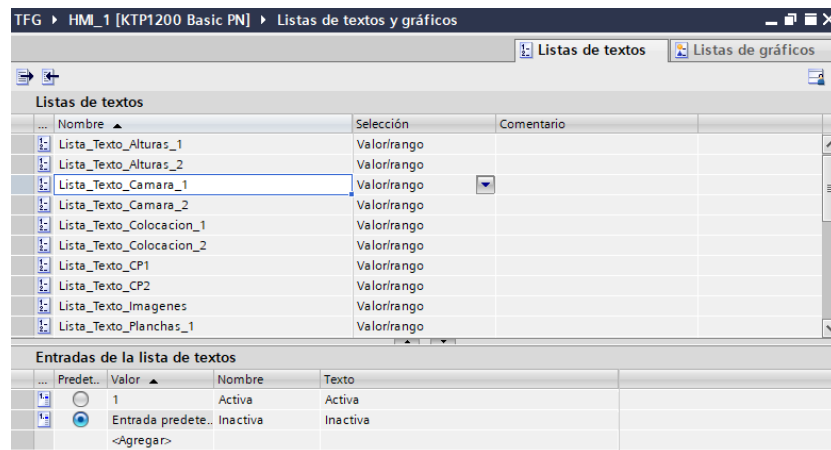


FIGURA 138. Lista de textos del proyecto en TIA Portal V16.

Se observa como cada gráfico o texto tiene un valor asignado. Este valor de la variable asociada es el encargado de gestionar la lista, determinando que gráfico o texto se mostrará.

- **Apariencia:** establece el estilo visual del elemento, incluyendo colores de fondo y bordes, para que se adapte al diseño de la interfaz.
- **Patrón de relleno:** permite seleccionar diferentes patrones de relleno, como sólido o degradado.
- **Aspecto:** ajusta el borde de enfoque del elemento, permitiendo ajustar su grosor y color para resaltar el elemento cuando esté seleccionado.
- **Representación:** define la posición y tamaño del elemento en la pantalla, ajustando las coordenadas (X, Y) y las dimensiones (ancho y alto), adapta el tamaño del objeto al contenido, ajusta los márgenes del texto y la imagen para asegurar una presentación adecuada, etc.
- **Formato de texto:** personaliza la fuente, el tamaño, el color y la alineación del texto que se muestra en el elemento.
- **Estilos/ diseños:** aplica estilos predefinidos para mantener una apariencia consistente en toda la interfaz o crea y aplica diseños personalizados.
- **Misceláneo:** asigna un nombre único al elemento y añade una descripción que detalla su función.
- **Seguridad:** establece los niveles de permiso requeridos para interactuar con el elemento, garantizando que solo los usuarios con autorización puedan llevar a cabo determinadas acciones o acceder a información específica.

Programación de animaciones

La programación de animaciones permite incorporar efectos visuales y movimientos de los elementos de la HMI. Las animaciones disponibles son:

- **Visualización:**
 - **Apariencia:** permite dinamizar colores y efectos de parpadeo en el elemento, haciendo que cambie su aspecto visual en función de ciertas condiciones o eventos.
 - **Visibilidad:** controla la visibilidad del elemento, haciéndolo aparecer o desaparecer según condiciones específicas.
- **Movimientos:**
 - **Directo:** permite mover el objeto de una posición a otra de manera directa.
 - **Diagonal:** permite mover el objeto en una trayectoria diagonal.
 - **Horizontal:** permite mover el objeto de manera horizontal. Esta configuración se utiliza, por ejemplo, para simular el movimiento de las cintas transportadoras.
 - **Vertical:** permite mover el objeto de manera vertical.

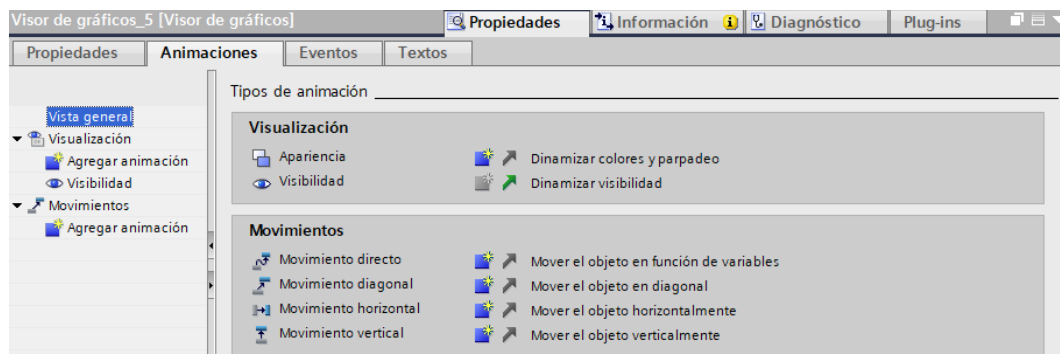


FIGURA 139. Programación de animaciones en TIA Portal V16.

Programación de eventos

La programación de eventos permite definir acciones específicas que se ejecutarán en respuesta a ciertas interacciones del usuario con los elementos de la HMI, generalmente con los botones. Según la activación, se distinguen los siguientes eventos:

- **Hacer clic:** las acciones se ejecutan cuando se hace clic en el elemento.
- **Pulsar:** las acciones se ejecutan cuando el usuario presiona el elemento.
- **Soltar:** las acciones se ejecutan cuando el usuario suelta el elemento después de haberlo presionado.
- **Activar:** las acciones se ejecutan cuando el elemento se activa.
- **Desactivar:** las acciones se ejecutan cuando el elemento se desactiva.
- **Cambio:** las acciones se ejecutan cuando se produce un cambio en el estado del elemento.

Las acciones que puede desencadenar un evento son muy variadas. Entre las más destacadas y utilizadas en este proyecto se encuentran en el grupo denominado "Procesamiento por bits", como se muestra en la siguiente figura:

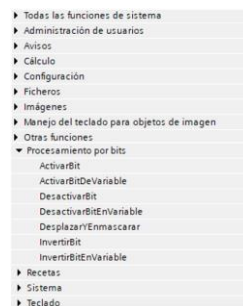


FIGURA 140. Acciones para configurar eventos en TIA Portal V16.

Un ejemplo de cómo se vería la configuración para activar la cinta transportadora al pulsar el botón correspondiente se muestra en la figura 141. En este caso, se utiliza el evento "Pulsar" para activar el bit de la variable de marcha y el evento "Soltar" para desactivar el mismo bit, evitando que permanezca activado constantemente. Aunque existen otras formas de realizar esta programación, esta es una de las más utilizadas.

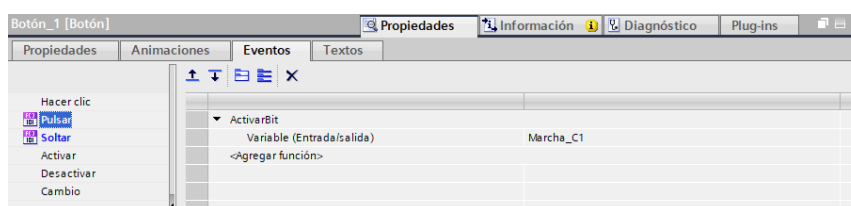


FIGURA 141. Ejemplo programación de eventos en "Botón Marcha" del proyecto en TIA Portal V16.

Capítulo 5.2.3. Conexión entre el PLC y la HMI

La conexión entre el PLC y la interfaz HMI es crucial para asegurar que las instrucciones del operario se ejecuten correctamente considerando la lógica programada en el PLC y lleguen a todos los sistemas y componentes de la estación. En este proyecto, tal como se mencionó anteriormente, la conexión entre el PLC y la HMI se establece mediante PROFINET.

Para hacer esta conexión en TIA Portal, se deben seguir los siguientes pasos:

1. Seleccionar la opción “Dispositivos y redes” dentro del árbol del proyecto.
2. Activar la vista denominada “Vista de redes”.
3. Activar “Conexiones” para observar las conexiones actuales.
4. Hacer clic en uno de los puertos PROFINET de uno de los dispositivos, y arrastrar hasta el puerto PROFINET del otro dispositivo.

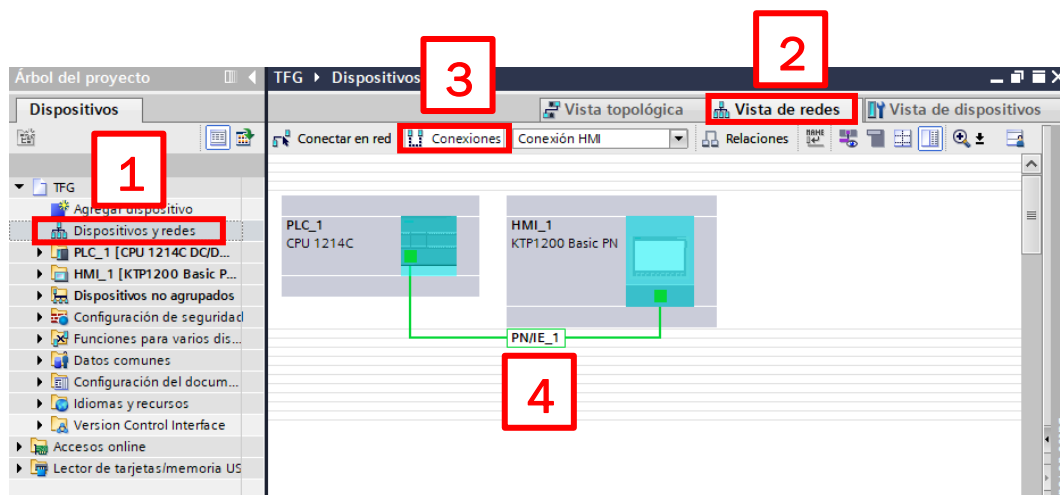


FIGURA 142. Realización de conexión PROFINET entre el PLC y la HMI del proyecto en TIA Portal V16.

Capítulo 5.3. Desarrollo en Python

Todo el desarrollo del código en Python se ha llevado a cabo utilizando el entorno de desarrollo integrado Spyder. Este entorno ha sido empleado tanto para la implementación del sistema de reconocimiento de voz como para el sistema de visión artificial.

Capítulo 5.3.1. Sistema de reconocimiento de voz

El sistema de reconocimiento de voz está integrado por diversas secciones, cada una cumpliendo funciones específicas. En consecuencia, la descripción del diseño y la construcción del sistema se estructura en diferentes apartados, donde se detallan las funciones desarrolladas en el programa, facilitando una exposición clara y comprensible.

Este enfoque resulta esencial, ya que el sistema abarca procesos más complejos que la simple captura e interpretación de voz.

Conexión con el PLC

- **Función “conectar_plc”:** establece una conexión TCP/IP con el PLC utilizando la librería “snap7”, que es una librería de Python creada para comunicarse con hardware Siemens. El proceso de conexión requiere parámetros como la dirección IP, el rack y el slot, para identificar físicamente el módulo del PLC en la red. Una vez establecida la conexión, se devuelve el objeto “plc”, que actúa como el punto de enlace para todas las operaciones siguientes de lectura y escritura.

Lectura y escritura de datos en el PLC

- **Funciones “read_bool_mark” y “write_bool_mark”:** con estas funciones se manipulan marcas booleanas del PLC. Las marcas booleanas son utilizadas para señalar y controlar el flujo lógico del proceso, como iniciar o detener la grabación, etc. Gracias a la función “get_bool” de la librería “snap7.util”, se extrae el valor booleano de un bit específico de un byte. Por otro lado, para escribir en una marca, primero se debe leer el byte completo, a continuación, cambiar el valor del bit específico dentro de ese byte, y por último escribir el nuevo byte de vuelta en el PLC.
- **Funciones “read_byte_mark” y “write_byte_mark”:** operan de manera similar a las funciones booleanas, pero manejan un byte completo de datos, permitiendo leer y escribir múltiples señales o valores en una sola operación. En este proyecto, se usan para leer o modificar variables que ocupan un byte en el PLC.

Procesamiento y síntesis de voz

- **Función “inicializa_voz”:** inicializa el motor de texto a voz con la librería “pyttsx3”. Esta librería convierte texto en habla, permitiendo al sistema comunicarse con el usuario. La configuración incluye la velocidad del habla del sistema para una mejor comprensión y claridad.
- **Función “anuncia_y_devuelve”:** utiliza el motor de voz anterior para anunciar el mensaje que recibe en texto y devolver el valor asociado con ese mensaje.

Reconocimiento y respuesta

- **Función “escucha_y_respuesta”:** es el núcleo del sistema de reconocimiento de voz. Utiliza un micrófono para capturar la entrada de audio, y gracias a la librería “speech_recognition” la convierte en texto. Esta función maneja dinámicamente el estado de escucha del sistema, activando o desactivando el "modo de escucha" basado en la necesidad de capturar audio. A su vez, maneja excepciones específicamente relacionadas con errores de reconocimiento, solicitando al usuario que repita el comando si no se entendió correctamente.

Configuración de parámetros por el usuario

- **Funciones de programación como “prog_tipo_caja”, “prog_num_alturas”, “prog_plancha_cada”, etc:** cada función escucha un comando específico, verifica su validez contra un conjunto predefinido de opciones, y ejecuta los cambios oportunos en función de la entrada recibida. Ejemplos para este proyecto incluyen, la selección del tipo de cajas para el paletizado, la cantidad de alturas para realizar el palé, el control del estado de las cámaras de visión, y demás parámetros programables por el operador.

Monitoreo de emergencias

- **Función “escuchar_comando_stop”:** esta función escucha constantemente comandos de emergencia previamente programados, que requieren detener el sistema inmediatamente. Se ejecuta continuamente en el fondo y utiliza un tiempo de espera corto para responder rápidamente a situaciones críticas, garantizando que el sistema pueda ser detenido de manera segura y eficiente cuando sea necesario.

Capítulo 5.3.2. Sistema de visión artificial

Para la explicación de como se ha llevado a cabo y como es el funcionamiento del sistema de visión artificial diseñado, se sigue un procedimiento de explicación del programa similar a como se ha seguido en el sistema de reconocimiento de voz, ya que también está compuesto por diferentes partes y funciones con fines diferenciados.

Función principal (“Main”)

Esta función actúa como punto de inicio organizando y coordinando el inicio de todas las operaciones y procesos. Aquí las tareas claves que realiza:

- **Preparación de la carpeta de imágenes:** comprueba la existencia de la carpeta “F:\\Fotos” en el ordenador, ya que es la carpeta donde se irán almacenando todas las imágenes que el programa analizará.
- **Carga y configuración de imágenes de referencia:** las imágenes de referencia “Vainilla.jpg” y “Choco.jpg” son cargadas y procesadas para ajustarlas a las dimensiones necesarias. Estas son las imágenes que se utilizarán como base para comparar el grado de similitud de las nuevas imágenes que se vayan guardando en la carpeta.
- **Visualización inicial de las imágenes de referencia recortadas:** se muestran las imágenes de referencia recortadas para comprobar que las imágenes de referencia están listas y configuradas correctamente para el proceso de comparación.
- **Monitoreo de la carpeta de imágenes:** se llama a la función “monitorear_carpeta” para comenzar con el monitoreo de nuevas imágenes en la carpeta designada. Esta acción pone en marcha el observador que detecta y procesa automáticamente cualquier nueva imagen añadida.

Conexión con el servidor

- **Función “conectarse_al_servidor”:** establece la conexión con el servidor de RobotStudio utilizando sockets gracias a la librería de Python “socket”. Facilita la recepción de parámetros claves como los umbrales de similitud para cada imagen de referencia.

Comunicación con el servidor

- **Función “manejar_conexión”:** gestiona continuamente la conexión con el servidor (RobotStudio), recibiendo comandos para cambiar la imagen de referencia activa, detener el monitoreo, etc.

Monitoreo y manejo de eventos

- **Función “monitorear_carpeta”:** por medio de la librería “watchdog”, esta función configura un observador que monitorea continuamente la carpeta específica para nuevas imágenes, activando automáticamente el procesamiento cuando se detecten.
- **Función “al_crear”:** se activa automáticamente cuando se detecta una nueva imagen en la carpeta que se está monitoreando. Guarda la ruta de la imagen para su posterior procesamiento y asegurar que todas sean evaluadas.
- **Función “archivo_listo”:** confirma que la imagen está completamente escrita y lista para ser procesada. Esta verificación es vital para prevenir errores al asegurar que solo se manejen archivos completamente creados y accesibles.

Gestión de la calidad y respuestas automáticas

- **Función “intentar_mostrar_imagen”:** coordina la verificación del archivo, el recorte, la comparación de las imágenes y la visualización en una secuencia lógica. Esta función decide si enviar comandos al servidor (RobotStudio) basándose en los resultados obtenidos tras realizar el análisis. Por ejemplo, si la comparación de imágenes revela que el porcentaje de similitud de una nueva captura es inferior al umbral establecido por el operador, se enviará una notificación al servidor para que clasifique la caja correspondiente.

Análisis y visualización de imágenes

- **Función “recortar_imagen”**: centra el área de análisis de la imagen ajustando el ancho y el largo programado desde su punto medio para realizar las comparaciones.
- **Función “comparar_imagenes”**: evalúa la similitud entre la imagen de referencia y la nueva captura usando la técnica de similitud SSIM entre dos imágenes gracias a la librería “structural_similarity”, calculando el porcentaje de similitud.
- **Funciones “mostrar_imagen”, “mostrar_imagenes” y “mostrar_diferencias”**: gracias a la librería “matplotlib” y la librería “OpenCV” se proporciona una interfaz visual en el panel de gráficos. Donde se muestra la nueva captura, la imagen de referencia junto a la nueva captura, y la comparación de las imágenes destacando las diferencias visuales mediante métodos directos y con un mapa de calor.



CAPÍTULO 6. Pruebas y resultados

Después de haber explicado cómo funcionan los programas utilizados y el proceso seguido para llevar a cabo el proyecto, en esta sección se describirá la simulación del proyecto desarrollado, incluyendo los resultados obtenidos con cada programa empleado. Para clarificar el funcionamiento del proyecto al lector, se incluirán capturas de pantalla representativas de la simulación en los distintos programas, siempre que estas aporten información valiosa.

El primer paso consiste en simular los programas diseñados utilizando los softwares mencionados, estableciendo la comunicación entre ellos conforme a lo explicado en capítulos previos. Posteriormente, se procederá a ajustar los parámetros y se podrán observar los resultados de la simulación.

El procedimiento detallado a continuación, incluyendo capturas de pantalla, cubrirá la mayoría de los aspectos del proyecto. Sin embargo, ciertas funciones, como el sistema de voz, no se pueden representar gráficamente en este informe.

1. Inicio de la simulación:

- **RobotStudio:** la estación comienza con todas las cintas transportadoras y el robot en estado de parada, mientras que los componentes inteligentes correspondientes intentan establecer conexión con el servidor a través del TIA Portal. Este proceso de conexión continúa hasta que se logra establecer satisfactoriamente. Por otro lado, la comunicación con el programa de visión artificial, implementado en Python, se realiza mediante sockets desde RAPID. Sin embargo, es necesario activar una señal específica desde la interfaz hombre-máquina (HMI) para iniciar la comunicación con el programa de visión artificial.

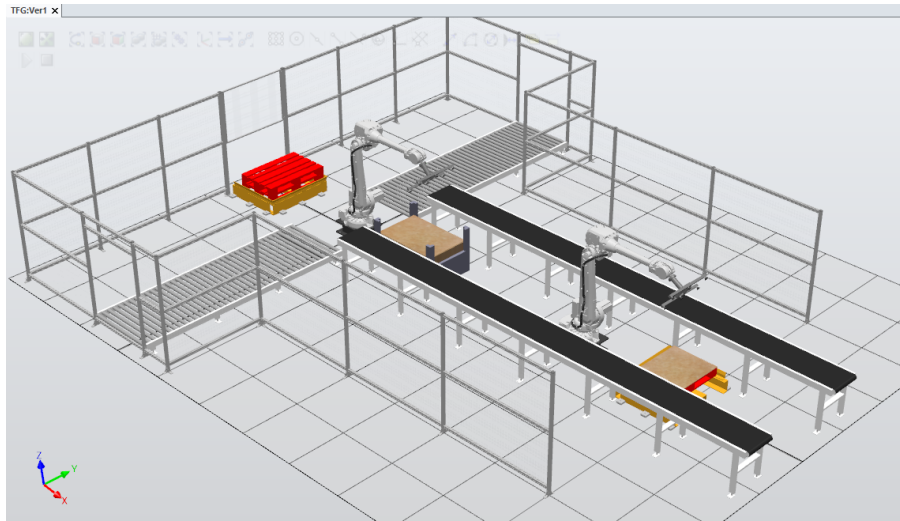


FIGURA 143. Inicio de la simulación en RobotStudio 2023.

- **TIA Portal:** la simulación se inicia desde la “Ventana Principal” de la HMI, que actúa como punto de acceso para las demás ventanas y desde donde se activa el movimiento de los robots.

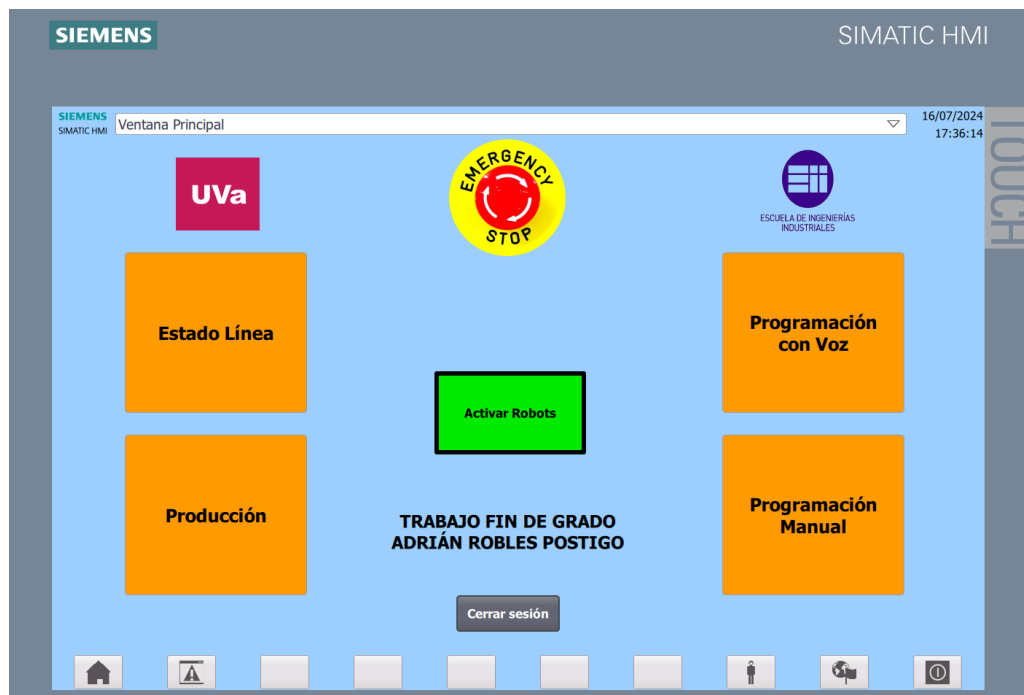


FIGURA 144. Inicio de la simulación en TIA Portal V16.

- **NetToPLCSim:** cuando el servidor está configurado y en funcionamiento, todos los programas correctamente configurados pueden conectarse al PLC que se está simulando.

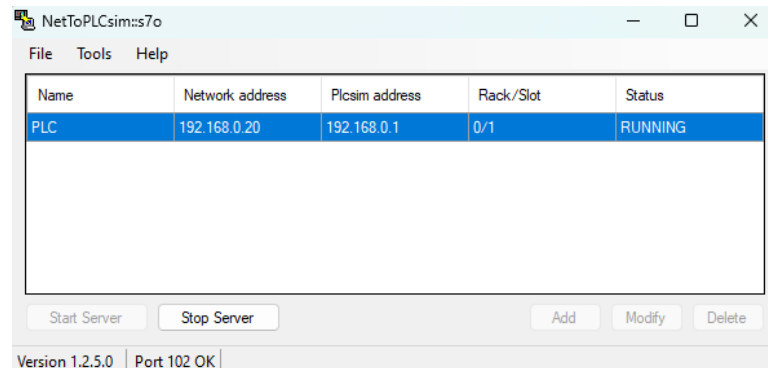


FIGURA 145. Servidor configurado y en funcionamiento en NetToPLCSim.

- **Python (Sistema de reconocimiento por voz):** este sistema se conecta al servidor de NetToPLCSim y permanece en espera hasta recibir órdenes de voz directamente del usuario o solicitudes emitidas desde la HMI.

```
In [4]: runfile('F:/Ingeniería Electrónica Industrial y Automática/TFG/Spyder/Programas Finales/Programacion.py', wdir='F:/Ingeniería Electrónica Industrial y Automática/TFG/Spyder/Programas Finales')
Conectado al PLC
Escuchando comandos para detener el sistema...
Escuchando comandos para detener el sistema...
Escuchando comandos para detener el sistema...
Escuchando comandos para detener el sistema...
Escuchando comandos para detener el sistema...
Escuchando comandos para detener el sistema...
Escuchando comandos para detener el sistema...
Escuchando comandos para detener el sistema...
```

FIGURA 146. Datos en el terminal al inicio de la simulación del sistema de reconocimiento por voz en Python.

- **Python (Sistema de visión artificial):** este sistema intenta establecer conexión con RobotStudio, pero no lo logra exitosamente hasta que el usuario emita la orden desde la HMI para establecer dicha conexión, mientras tanto, lo vuelve a intentar cada 5 segundos.

```
In [3]: runfile('F:/Ingeniería Electrónica Industrial y Automática/TFG/Spyder/Programas Finales/Vision.py', wdir='F:/Ingeniería Electrónica Industrial y Automática/TFG/Spyder/Programas Finales')

Importante
Las figuras se muestran en el panel de Gráficos por defecto. Para que también aparezcan en la terminal, desactive "Silenciar los gráficos en línea" en el menú de opciones de Gráficos.

Error al conectar: [WinError 10061] No se puede establecer una conexión ya que el equipo de destino denegó expresamente dicha conexión, reintentando en 5 segundos...
Error al conectar: [WinError 10061] No se puede establecer una conexión ya que el equipo de destino denegó expresamente dicha conexión, reintentando en 5 segundos...
```

FIGURA 147. Datos en el terminal al inicio de la simulación del sistema de visión en Python.

```
Importante

Las figuras se muestran en el panel de Gráficos por defecto. Para que también aparezcan en la terminal, desactive "Silenciar los gráficos en línea" en el menú de opciones de Gráficos.

Error al conectar: [WinError 10061] No se puede establecer una conexión ya que el equipo de destino denegó expresamente dicha conexión, reintentando en 5 segundos...
Error al conectar: [WinError 10061] No se puede establecer una conexión ya que el equipo de destino denegó expresamente dicha conexión, reintentando en 5 segundos...
Error al conectar: [WinError 10061] No se puede establecer una conexión ya que el equipo de destino denegó expresamente dicha conexión, reintentando en 5 segundos...
Error al conectar: [WinError 10061] No se puede establecer una conexión ya que el equipo de destino denegó expresamente dicha conexión, reintentando en 5 segundos...
Error al conectar: [WinError 10061] No se puede establecer una conexión ya que el equipo de destino denegó expresamente dicha conexión, reintentando en 5 segundos...
Conexión establecida con el servidor.
Conexión establecida correctamente con el Robot
```

FIGURA 148. Datos en el terminal al establecer conexión con el servidor de RobotStudio en Python.

2. Configuración desde la HMI:

- **TIA Portal:** la configuración del paletizado se personaliza según las necesidades específicas, como se ilustra en la figura 149, donde se ha procedido a hacer la configuración manualmente. Mientras que, para el clasificado, se ha procedido con el sistema de voz (en la figura 150, se muestra el sistema de reconocimiento de voz a la escucha ya que el grabador está en verde).

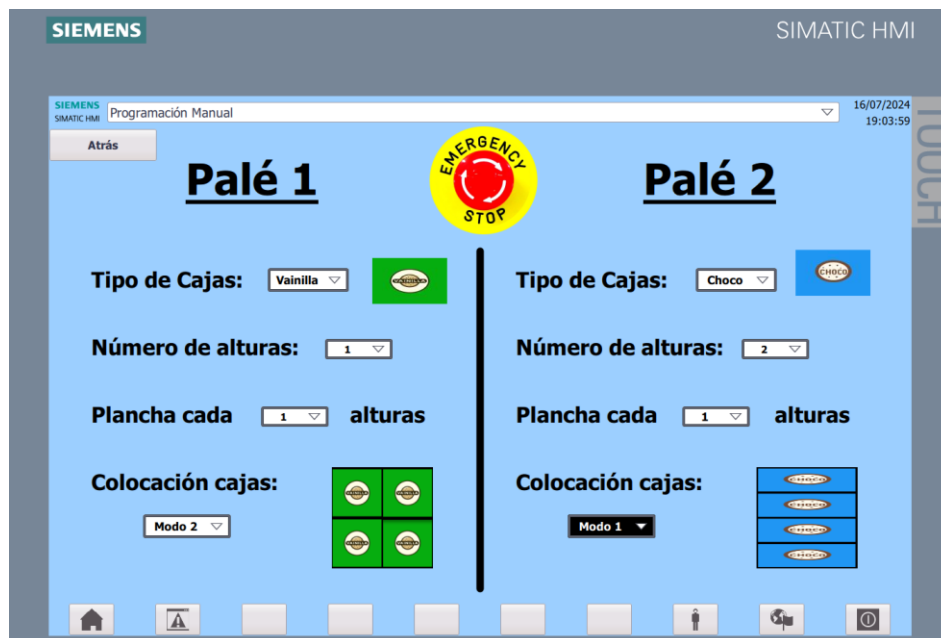


FIGURA 149. Programación del paletizado manualmente desde la HMI en TIA Portal V16.



FIGURA 150. Programación del paletizado por voz desde la HMI en TIA Portal V16.

3. Inicio del movimiento:

- **TIA Portal:** se activan las cintas transportadoras de cajas a través de la imagen "Estado Línea" en la interfaz HMI. Asimismo, el movimiento de los robots se inicia desde la imagen "Ventana Principal".

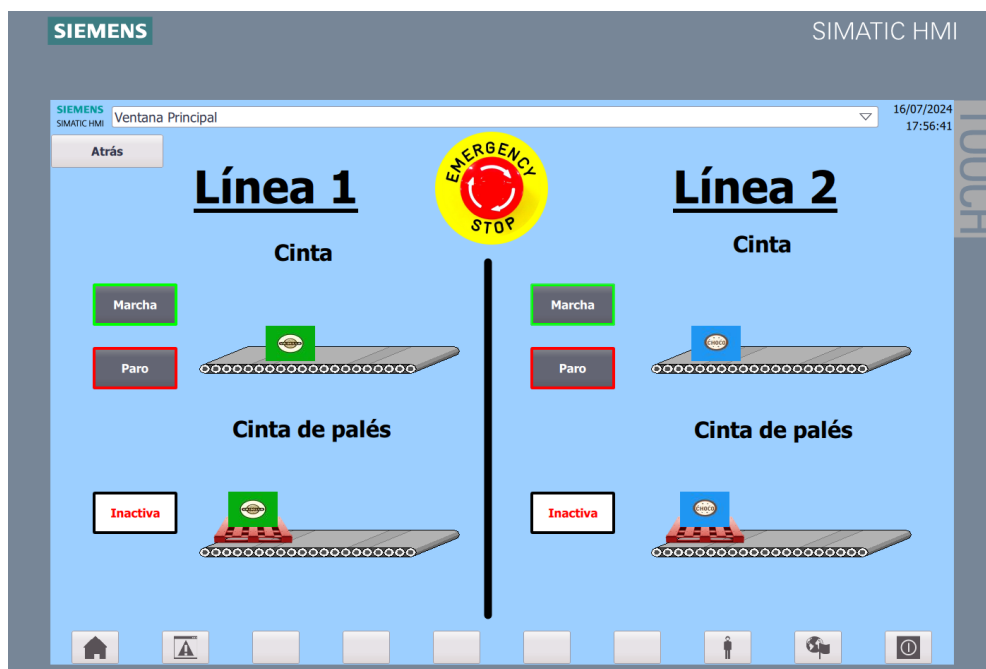


FIGURA 151. Cintas transportadoras de cajas en marcha desde la HMI en TIA Portal V16.

4. Paletizado:

- **RobotStudio:** se observa cómo se crean cajas de ambos tipos y como el robot correspondiente las paletiza con la configuración programada en la figura 154.

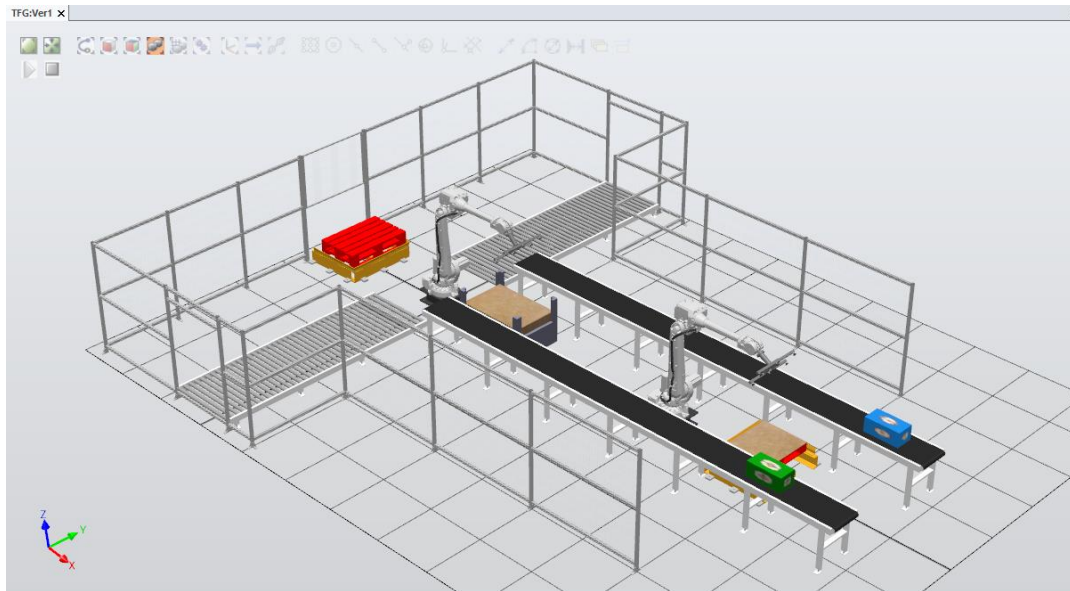


FIGURA 152. Cintas transportadoras de cajas activas en RobotStudio 2023.

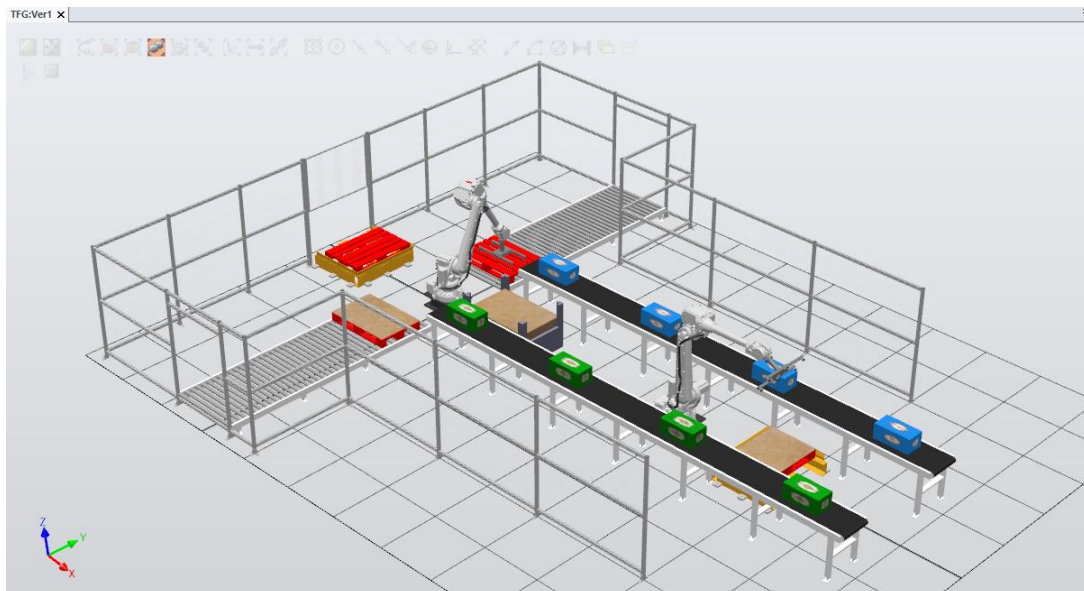


FIGURA 153. Robots y cintas transportadoras de cajas activas en RobotStudio 2023.

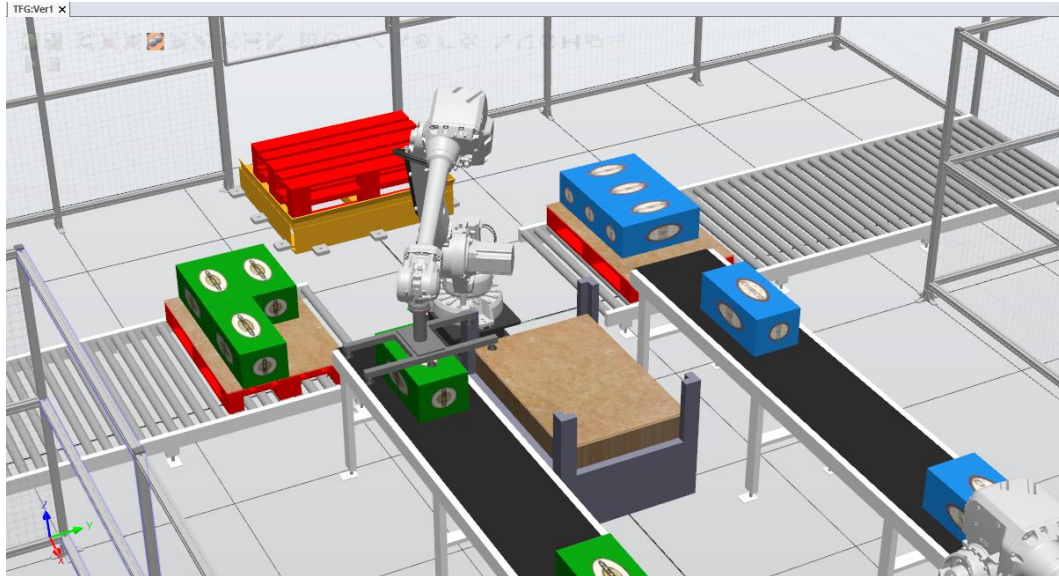


FIGURA 154. Paletizado según la configuración realizada en la figura 149 en RobotStudio 2023.

- **RobotStudio:** al completar el palé, se activan las cintas transportadoras de palets para comenzar uno nuevo.

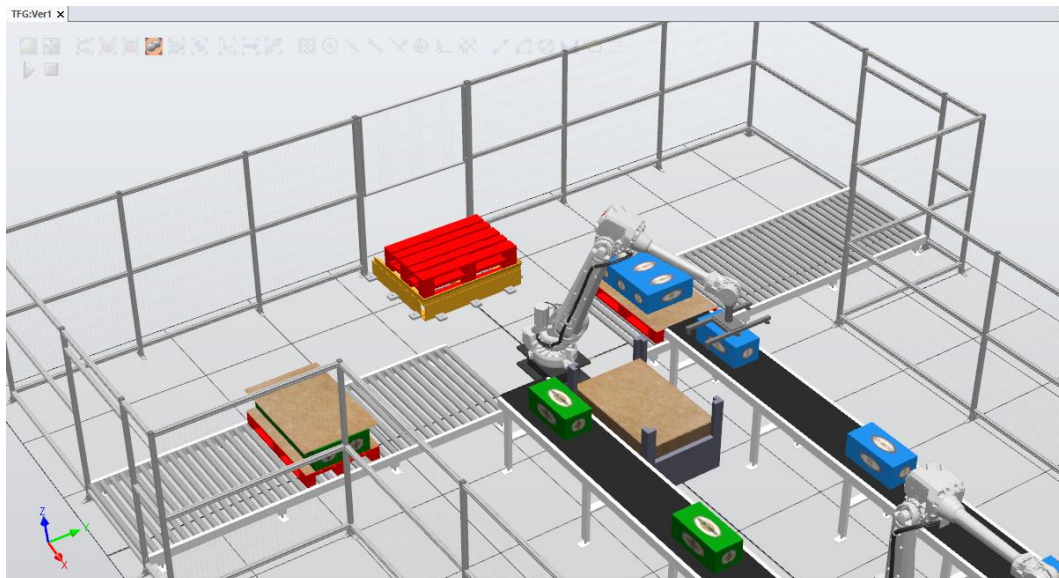


FIGURA 155. Palé completado en RobotStudio 2023.

- **TIA Portal:** en la HMI se actualiza el estado de cada cinta.

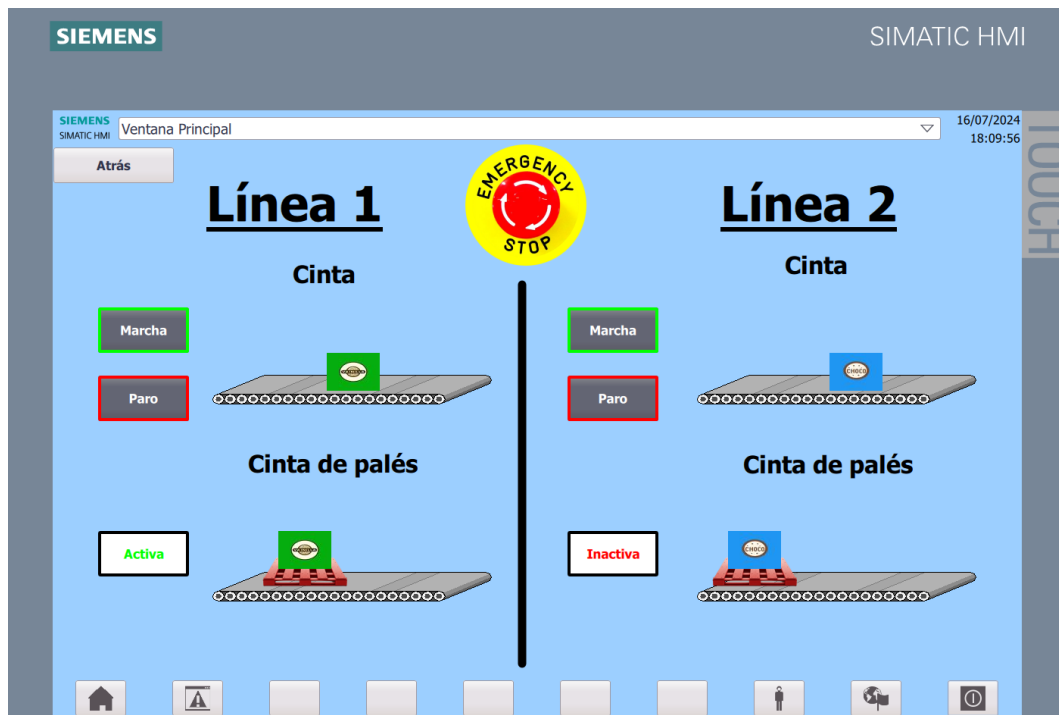


FIGURA 156. Cintas transportadoras de palés activas en HMI en TIA Portal V16.

5. Clasificado:

- **RobotStudio:** continuamente se realizan capturas de pantalla cuando las cajas pasan por debajo de las cámaras, activadas previamente.



FIGURA 157. Captura a cajas de vainilla desde RobotStudio 2023.



FIGURA 158. Captura a cajas choco desde RobotStudio 2023.

- **Python (Sistema de visión artificial):** de forma paralela, el sistema de visión artificial procesa todas las capturas realizadas, comparándolas con las imágenes de referencia y determinando el porcentaje de similitud. Si este es menor al configurado en la figura 150, se desechará dicha caja. Basándose en estos análisis, el sistema envía instrucciones a través del socket sobre si se debe retirar o no la caja analizada.
 - **Capturas sin diferencias:**

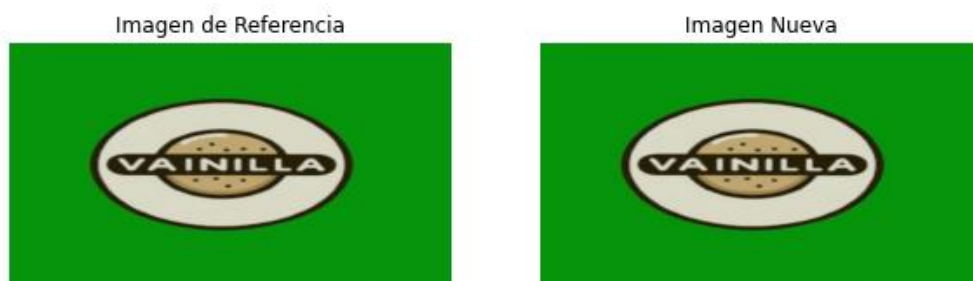


FIGURA 159. Visualización de la caja de referencia y la caja nueva perfecta de vainilla en Python.



FIGURA 160. Diferencias entre la caja de referencia y la caja nueva de vainilla sin diferencias en Python.

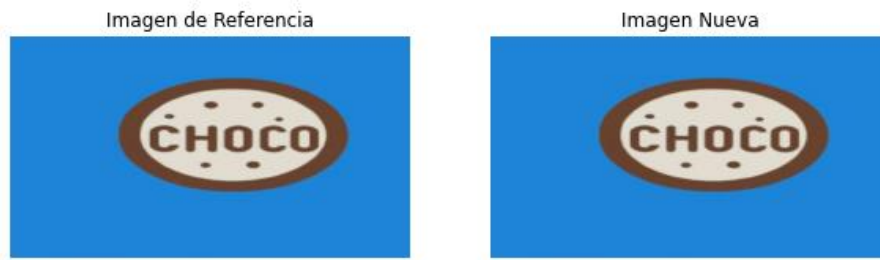


FIGURA 161. Visualización de la caja de referencia y la caja nueva perfecta de choco en Python.



FIGURA 162. Diferencias entre la caja de referencia y la caja nueva perfecta de choco sin diferencias en Python.

○ Capturas con diferencias:

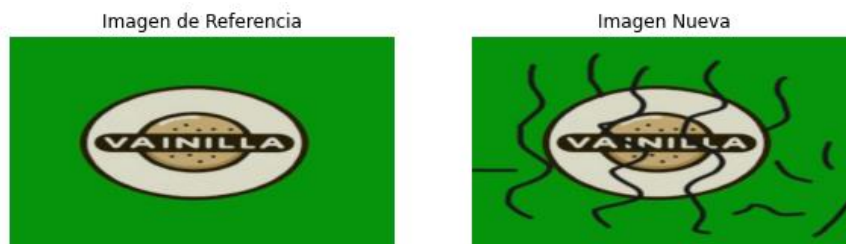


FIGURA 163. Visualización de la caja de referencia y la caja nueva defectuosa de vainilla en Python.



FIGURA 164. Diferencias entre la caja de referencia y la caja nueva defectuosa de vainilla con diferencias en Python.

```
Imagen detectada: F:\Fotos\RobotStudio 7-17_11.JPG  
Comando recibido del servidor para cambiar a imagen de referencia: 1  
Grado de similitud con la imagen de referencia: 88.17
```

FIGURA 165. Datos en el terminal al comparar imágenes de vainilla defectuosas en Python.



FIGURA 166. Visualización de la caja de referencia y la caja nueva defectuosa de choco en Python.



FIGURA 167. Diferencias entre la caja de referencia y la caja nueva defectuosa de choco con diferencias en Python.

```
Imagen detectada: F:\Fotos\RobotStudio 7-17_14.JPG  
Comando recibido del servidor para cambiar a imagen de referencia: 2  
Grado de similitud con la imagen de referencia: 87.13
```

FIGURA 168. Datos en el terminal al comparar imágenes de choco defectuosas en Python.

- **RobotStudio:** si se determina que una caja debe ser clasificada, el robot encargado de este proceso procede adecuadamente, realizando la clasificación basada en los resultados proporcionados por el sistema de visión artificial.

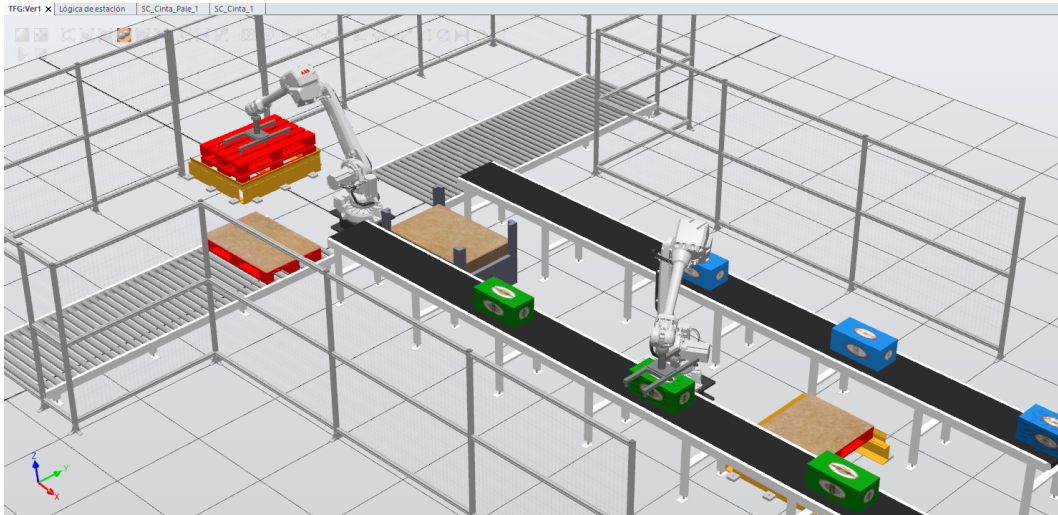


FIGURA 169. Clasificado de caja defectuosa de vainilla según el porcentaje configurado en RobotStudio 2023.

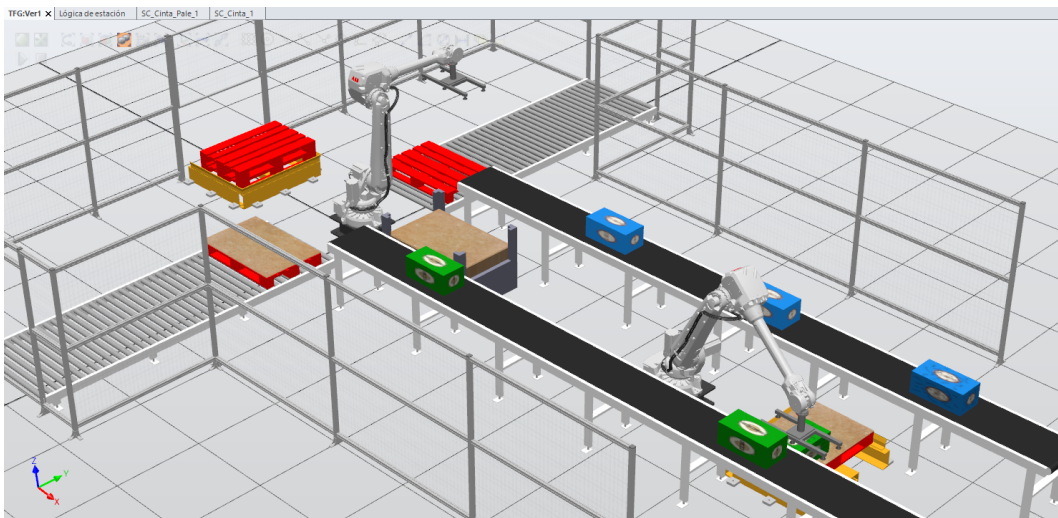


FIGURA 170. Colocación en palé de caja defectuosa en RobotStudio 2023.

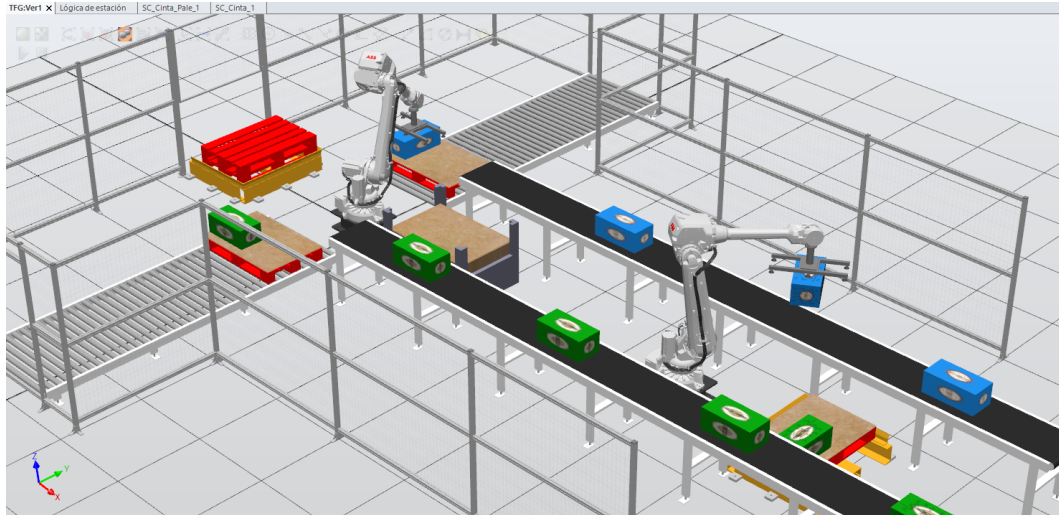


FIGURA 171. Clasificado de caja defectuosa de choco según el porcentaje configurado en RobotStudio 2023.

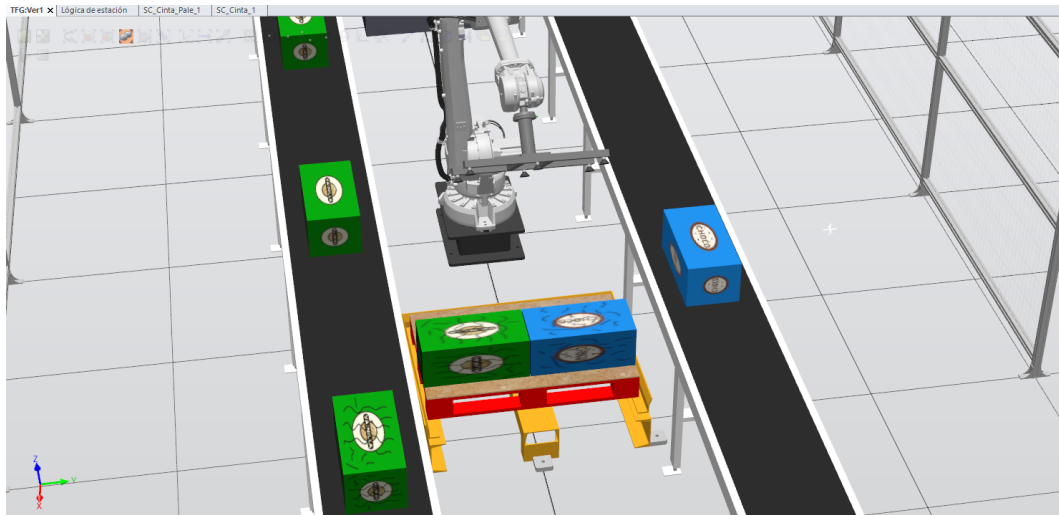


FIGURA 172. Cajas defectuosas paletizadas en RobotStudio 2023.

6. Recuento:

- **TIA Portal:** desde la pantalla “Producción”, se puede observar el recuento de diversas variables representativas de la simulación. Si se desea iniciar un nuevo recuento en ese instante, simplemente se debe hacer clic en el botón "Reset Contadores". Este botón permite reiniciar todos los contadores al estado inicial.

	Línea 1	Línea 2
Número de palés completados:	0	0
Número de cajas Vainilla posicionadas:	2	
Número de cajas Choco posicionadas:	1	
Número de planchas usadas:	4	
Número de cajas Vainilla apartadas:	1	
Número de cajas Choco apartadas:	1	

FIGURA 173. Recuento de la simulación en TIA Portal V16.

7. Paro de emergencia:

- **TIA Portal:** si el usuario presiona el botón de paro de emergencia en la HMI, se detendrá inmediatamente el movimiento de todos los elementos de la estación. Asimismo, este botón comenzará a parpadear, indicando el estado de emergencia, y no se podrá reactivar el movimiento hasta que sea desactivado.



FIGURA 174. Seta de emergencia activada en TIA Portal V16.

- **Python (Sistema de reconocimiento por voz):** de manera similar, el sistema de reconocimiento por voz permanece continuamente a la escucha de comandos que puedan indicar la necesidad de detener el movimiento. Si se pronuncia alguno de estos comandos, se activará la seta de emergencia, desencadenando las mismas consecuencias que si el botón fuera presionado directamente desde la HMI.

```
Escuchando comandos para detener el sistema...  
Escuchando comandos para detener el sistema...  
Comando 'stop' detectado.
```

FIGURA 175. Datos en el terminal al pronunciar “stop” por parte del usuario en Python.

- **RobotStudio:** aunque no se pueda mostrar en una captura de pantalla, se detiene el movimiento de todos los elementos de la estación de forma inmediata tras la activación de la emergencia.

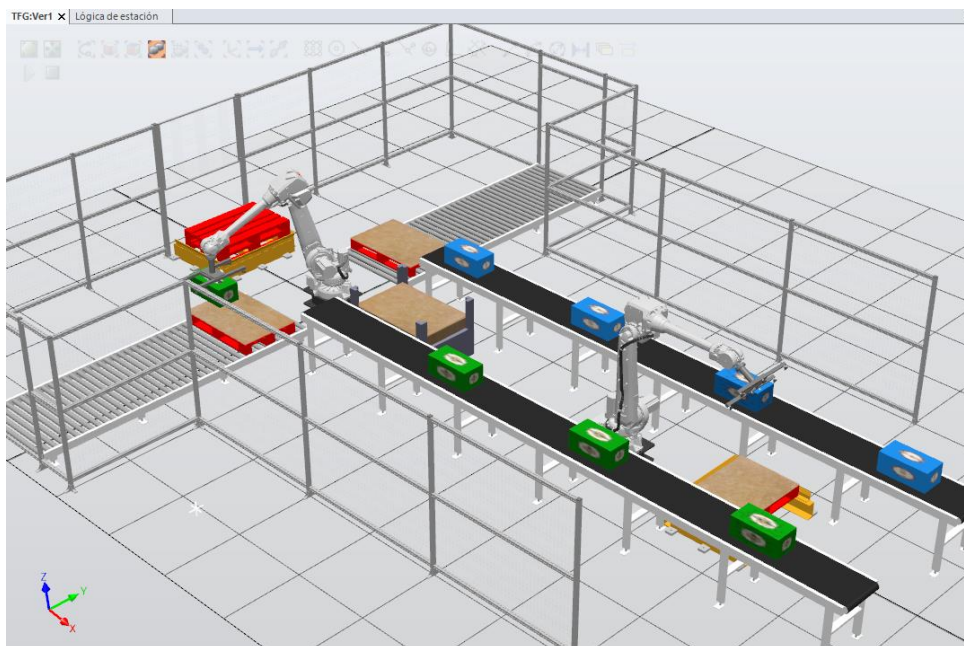


FIGURA 176. Paro de emergencia en RobotStudio 2023.

8. Reset emergencia y comienzo:

- **TIA Portal:** tras un paro de emergencia, haciendo nuevamente clic en el botón de la seta de emergencia en la HMI y pulsando los botones correspondientes se reanuda el movimiento. Esto permite que cada elemento retome su actividad desde la posición exacta en la que se encontraba y continúe realizando la tarea que estaba ejecutando justo antes del paro.

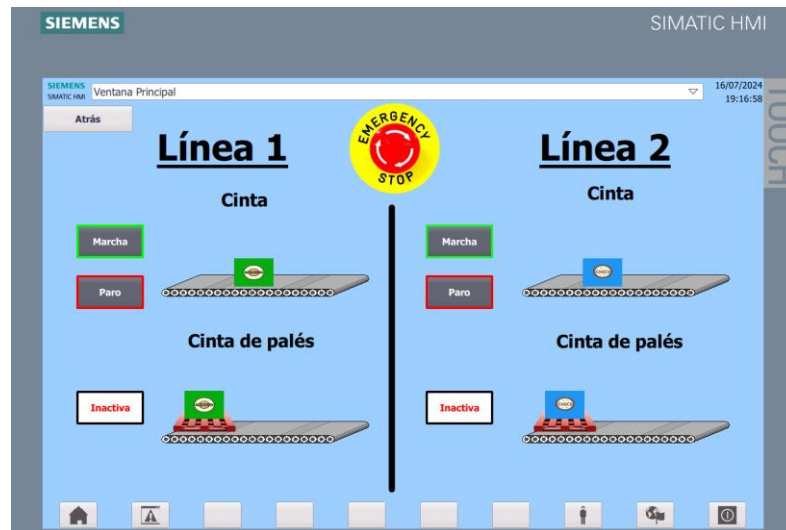


FIGURA 177. Reactivación movimiento después de parada de emergencia en TIA Portal V16.

- **RobotStudio:** al reactivar el sistema después de una parada de emergencia, se reinicia el movimiento de cada elemento desde la posición exacta en la que se detuvo.

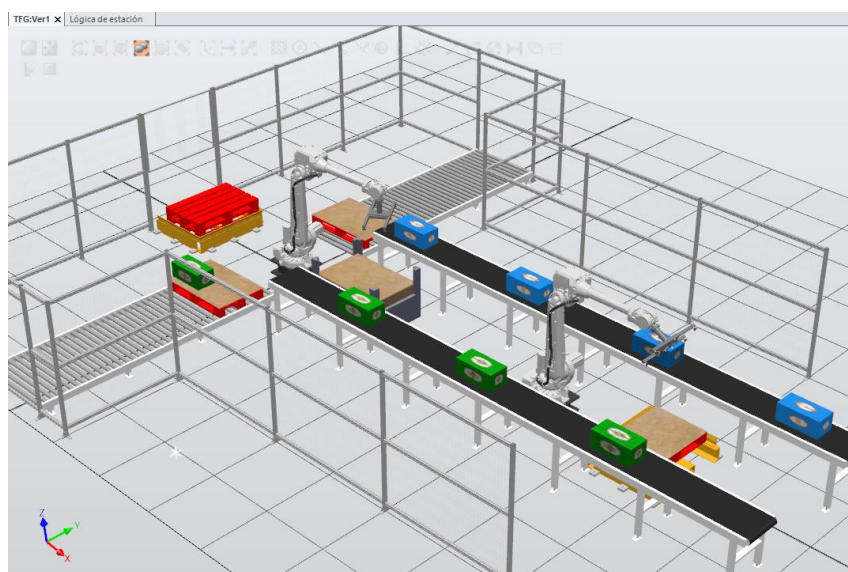


FIGURA 178. Reactivación movimiento después de parada de emergencia en RobotStudio 2023.



Con esta descripción ilustrativa, se ha explicado claramente el funcionamiento de la estación y, por ende, del proyecto realizado. Esto proporciona una comprensión integral de cómo se interrelacionan y operan los diferentes programas dentro del proyecto, asegurando un entendimiento completo de su funcionalidad y propósito.





CAPÍTULO 7. Estudio económico

El propósito de este análisis económico es valorar la viabilidad financiera del desarrollo e implementación del sistema propuesto. Este estudio detalla los costes asociados al desarrollo, implementación y operación del sistema, además de evaluar los beneficios económicos proyectados. Entender estos elementos financieros permitirá calcular la rentabilidad del proyecto y su impacto económico en la empresa que lo adopte. Aunque este análisis constituye una estimación preliminar, para obtener cifras más precisas sería necesario contactar con proveedores y analizar el contexto específico de implementación. No obstante, el estudio incluye una proyección de los gastos y beneficios estimados para un año, facilitando una evaluación inicial de la rentabilidad.

Capítulo 7.1. Gastos del proyecto

El análisis de los gastos del proyecto es fundamental para comprender la inversión necesaria para implementar el sistema. Esta sección desglosa los costes en varias categorías clave. A continuación, se detallan los componentes y el desglose de cada uno de estos costes.

Capítulo 7.1.1. Costes de desarrollo

Los costes de desarrollo comprenden todos los gastos necesarios para la creación y preparación del sistema antes de su implementación. Esto incluye la compra del software y hardware necesarios, y el salario del equipo de desarrollo. Además, se ha considerado la dificultad de cada tarea para determinar el coste de la mano de obra por hora, ajustando así el precio según la complejidad de cada parte del proyecto.

Costes de software			
Concepto (Licencia)	Coste unitario (€/año)	Unidades (año)	Coste total (€)
RobotStudio	2.890,00	1	2.890,00
TIA Portal y módulos necesarios	5.000,00	1	5.000,00
Python y librerías	0,00	1	0,00
NetToPLCSim	0,00	1	0,00
TOTAL			7.890,00 €

Tabla 2. Costes de software del proyecto.

Costes de hardware			
Concepto	Coste unitario (€)	Unidades	Coste total (€)
Robots	40.000,00	2	80.000,00
Herramienta "Vacuum"	15.000,00	2	30.000,00
CPU 1214C DC/DC/DC	462,83	1	462,83
HMI KTP1 1200 Basic PN	1.671,30	1	1.671,30
Grabador y reproductor de voz	200	1	200,00
Cámara de visión artificial Cognex	800	2	1.600,00
Barrera fotoeléctrica	511,48	2	1.022,96
Fotocélula	249	4	996,00
Cinta transportadora	4.500,00	2	9.000,00
Transportador de pales	7.000,00	2	14.000,00
Dispensador de pales	5.000,00	1	5.000,00
Otros componentes	20.000,00	1	20.000,00
TOTAL			163.953,09 €

Tabla 3. Costes de hardware del proyecto.

Costes de mano de obra			
Concepto	Coste unitario (€/Hora)	Unidades (Horas)	Coste total (€)
Diseño de la estación	15	150	2.250,00
Programación de los robots	18	200	3.600,00
Programación PLC	14	150	2.100,00
Desarrollo HMI	18	250	4.500,00
Desarrollo sistema de reconocimiento de voz	18	280	5.040,00
Desarrollo sistema visión artificial	20	320	6.400,00
Comunicación entre elementos del sistema	18	100	1.800,00
TOTAL		1.450	25.690,00 €

Tabla 4. Costes de mano de obra del proyecto.

Con todos estos datos, es posible calcular los costes totales de desarrollo:

TOTAL COSTES DE DESARROLLO = 197.533,09 €

Capítulo 7.1.2. Costes de implementación

Los costes de implementación se refieren a los gastos incurridos durante la fase de instalación y configuración del sistema en el entorno de producción.

Costes de implementación		
Concepto	Detalles	Coste total (€)
Instalación y configuración	Instalación física de robots y sensores, configuración de software y hardware, y pruebas iniciales.	45.000,00 €
Formación	Formación del personal en el uso y mantenimiento del sistema.	6.000,00 €
Integración	Trabajo necesario para asegurar la integración del nuevo sistema con los sistemas existentes en la fábrica, incluyendo ajustes y personalización.	23.000,00 €
	TOTAL	74.000,00 €

Tabla 5. Costes de implementación del proyecto.

Capítulo 7.1.3. Costes operativos

Los costes operativos son los gastos recurrentes necesarios para mantener el sistema en funcionamiento una vez que ha sido implementado.

Costes operativos		
Concepto	Detalles	Coste anual (€)
Mantenimiento	Mantenimiento preventivo y correctivo.	4.000,00 €
Actualizaciones	Mejoras periódicas de software y hardware para mantener el sistema actualizado y óptimo.	2.000,00 €
Soporte técnico	Servicios de soporte técnico para la resolución de problemas y optimización del sistema.	1.500,00 €
Energía	Consumo energético de robots, sensores, ordenadores, etc. (24 horas/día x 224 días/año x 35 kW x 0.15 €/kWh).	28.224,00 €
Operario de supervisión	Salario anual de un operario para la supervisión continua del sistema (15 €/hora x 3 turnos/día x 8 horas/turno x 224 días/año).	80.640,00 €
	TOTAL	116.364,00 €

Tabla 6. Costes operativos anuales del proyecto.

Capítulo 7.2. Beneficios del proyecto

Se estiman los significativos beneficios económicos con la estación robotizada desarrollada. A continuación, se enumeran los principales beneficios previstos.

Beneficios		
Concepto	Detalles	Beneficio anual (€)
Reducción de mano de obra	Ahorros derivados de la automatización de tareas que antes requerían trabajo manual.	75.000,00 €
Aumento de productividad	Incremento en la productividad y reducción de costes de producción debido a la mayor velocidad y precisión del sistema.	25.000,00 €
Reducción de errores	Ahorros derivados de la disminución de errores en la clasificación y manipulación de productos.	10.000,00 €
Mejora en la calidad	Beneficios derivados de una mejor calidad de los productos gracias a la automatización y control de calidad mejorado.	10.000,00 €
	TOTAL	120.000,00 €

Tabla 7. Beneficios anuales del proyecto.

Estos beneficios evidencian cómo la implementación del sistema diseñado puede optimizar las operaciones, reducir los costes y mejorar la calidad de los productos.

Capítulo 7.3. Rentabilidad y análisis económico del proyecto

Con todas las estimaciones anteriores, se procederá a calcular la rentabilidad del proyecto, así como el tiempo necesario para amortizar la inversión inicial. El primer paso es determinar los costes iniciales requeridos para realizar la inversión:

$$\text{Inversión inicial} = \text{Costes de desarrollo} + \text{Costes de implementación}$$

$$\text{Inversión inicial} = 197.533,09 \text{ €} + 74.000,00 \text{ €}$$

$$\text{Inversión inicial} = 271.533,09 \text{ €}$$

La inversión inicial requerida para el proyecto asciende a 271.533,09 €.

A lo largo del año, el proyecto logra generar unos beneficios netos anuales de 120.000,00 €. Este resultado se obtiene después de deducir los costes operativos anuales, que ascienden 116.364,00 €.

Para calcular los años necesarios para amortizar la inversión inicial, se empleará la siguiente expresión:

$$\text{Años de amortización} = \frac{\text{Inversión inicial}}{\text{Beneficios anuales}}$$

$$\text{Años de amortización} = \frac{271.533,09}{120.000,00}$$

$$\text{Años de amortización} = 2,26 \text{ años}$$

Para evaluar la rentabilidad del proyecto, se recurrirá al método ROI (Retorno sobre la Inversión). Este indicador financiero mide el beneficio o la ganancia obtenida en relación con el capital invertido, proporcionando una visión clara del rendimiento de la inversión.

Utilizando la siguiente fórmula:

$$\text{ROI} = \left(\frac{\text{Beneficios anuales}}{\text{Inversión inicial}} \right) \cdot 100$$

Sustituyendo datos:

$$ROI = \left(\frac{120.000,00}{271.533,09} \right) \cdot 100$$

$$ROI = 44,19 \%$$

El cálculo indica que la tasa de retorno sobre la inversión (ROI) para este proyecto es aproximadamente del 44,19%. Esto implica que, por cada euro invertido, el proyecto genera cerca de 0,45 euros en beneficios netos anuales.

La evaluación confirma que, a pesar de que los datos sean aproximados, tanto el período de amortización como la rentabilidad del proyecto son excepcionales. La mayoría de los beneficios provienen de la reducción de costes en personal y del incremento en la productividad, lo que refleja claramente algunos de los objetivos principales de la robótica y la automatización. Tampoco se debe dejar pasar por alto el impacto positivo que la reducción de errores y la mejora en la calidad pueden tener en otras áreas de la empresa, contribuyendo también significativamente a los beneficios globales del proyecto.

CAPÍTULO 8. Conclusiones y líneas futuras

Capítulo 8.1. Conclusiones

La realización de este Trabajo de Fin de Grado ha resultado en una serie de logros significativos, alcanzando los objetivos planteados al inicio del proyecto. Estos logros incluyen tanto el manejo de herramientas avanzadas como la incorporación de tecnologías innovadoras, mostrando un avance considerable en los procesos industriales.

Uno de los logros más destacados ha sido el manejo avanzado de RobotStudio de ABB. A través de este proyecto, se ha profundizado en el uso de esta herramienta para la simulación y programación avanzada de robots utilizando el lenguaje RAPID. Este conocimiento ha permitido optimizar las tareas de paletizado y clasificación, asegurando procesos precisos y eficientes en un entorno controlado.

La creación de una Interfaz Hombre-Máquina (HMI) con control por voz ha sido otro avance notable. Desarrollada en TIA Portal y utilizando Python para el reconocimiento de voz, esta HMI ha mejorado significativamente la interacción del usuario con el sistema. Los usuarios ahora pueden operar los robots de manera más intuitiva y eficiente, mejorando la accesibilidad y operatividad del sistema.

En el ámbito de la visión artificial, se ha implementado un sistema avanzado utilizando librerías de Python para identificar y clasificar cajas defectuosas automáticamente. Este sistema se ha integrado completamente con la estación robótica, elevando la eficiencia en el proceso de clasificación. La capacidad de detectar y rechazar productos defectuosos de manera automática supone una mejora crucial en los estándares de calidad.

La comunicación eficaz entre los distintos componentes del sistema ha sido esencial para el éxito del proyecto. Se ha establecido una comunicación robusta y fiable entre RobotStudio, TIA Portal y Python mediante el protocolo TCP/IP, y mediante sockets para la conexión del sistema de visión. Esta infraestructura ha garantizado una coordinación fluida y un flujo de datos constante y estable, elementos esenciales para el funcionamiento eficiente de la estación.

Durante el proyecto, se realizaron numerosas simulaciones y pruebas exhaustivas para validar la funcionalidad y eficacia del sistema. Estas pruebas permitieron identificar y corregir errores, optimizando el rendimiento del sistema. Este proceso de validación fue fundamental para asegurar que el sistema esté listo para una posible implementación en un entorno industrial real.

El impacto del proyecto en términos de productividad, calidad y económicos ha sido significativo. La estación robotizada ha demostrado su capacidad para mejorar notablemente la eficiencia de los procesos industriales, reduciendo errores y aumentando la precisión en la clasificación de productos. Además, este sistema ha tenido un impacto positivo en las condiciones laborales de los trabajadores, mejorando su seguridad, salud y comodidad. La automatización de tareas repetitivas y físicamente exigentes disminuye el riesgo de lesiones y enfermedades ocupacionales, mientras que la interfaz intuitiva facilita el uso del sistema, reduciendo el estrés y la carga de trabajo. La integración de tecnologías avanzadas también ha establecido una base sólida para futuras investigaciones y desarrollos en el campo de la Industria 4.0, promoviendo un entorno de trabajo más seguro y eficiente.

En resumen, los logros alcanzados en este proyecto no solo reflejan la aplicación práctica de los conocimientos técnicos adquiridos durante el grado, sino que también establecen una base sólida para futuras investigaciones y desarrollos como se mostrará en el siguiente capítulo 8.2.

Capítulo 8.2. Propuestas para futuras investigaciones y desarrollos

A partir de los logros alcanzados en este Trabajo de Fin de Grado, se identifican varias áreas prometedoras para futuras investigaciones y desarrollos. Estas propuestas buscan ampliar las capacidades del sistema, mejorar su rendimiento y explorar nuevas aplicaciones industriales. A continuación, se presentan algunas direcciones sugeridas para el avance del proyecto:

Mejoras del sistema de visión artificial

- **Algoritmos de aprendizaje automático:** incorporar algoritmos avanzados de aprendizaje automático para mejorar la precisión y velocidad en la identificación y clasificación de cajas defectuosas o de cualquier otro objeto que se desee clasificar.
- **Análisis multidimensional:** ampliar las capacidades del sistema de visión para analizar parámetros adicionales como peso y volumen, optimizando aún más el proceso de clasificación.

Optimización de la Interfaz Hombre-Máquina (HMI)

- **Mejorar el reconocimiento de voz:** investigar en otras tecnologías avanzadas de reconocimiento de voz para aumentar la precisión y reducir el tiempo de respuesta.
- **Dinamizar la interfaz gráfica:** desarrollar interfaces gráficas aún más intuitivas y adaptativas, logrando que se puedan ajustar automáticamente a las necesidades de los operarios y al contexto operativo.

Mejorar la comunicación y coordinación del sistema

- **Protocolos de comunicación avanzados:** explorar nuevos protocolos de comunicación que ofrezcan menor latencia y mayor fiabilidad de transmisión de datos entre los distintos componentes.

Aplicaciones en nuevos sectores

- **Industria alimentaria:** adaptar el sistema para la clasificación y empaquetado de productos alimentarios, teniendo en cuenta los requisitos específicos de higiene y manipulación. Esto podría incluir la detección de defectos en productos frescos, el control de calidad en líneas de envasado y la automatización de tareas repetitivas.
- **Industria electrónica:** implementar el sistema en la clasificación y ensamblaje de componentes electrónicos, donde la precisión y la gestión cuidadosa son cruciales. Esto podría incluir la detección de defectos en placas de circuitos impresos (PCB), la colocación automática de componentes y la inspección final de productos.
- **Automoción:** aplicar el sistema en la industria automotriz para la inspección y clasificación de piezas, la detección de defectos en componentes críticos y la automatización de procesos de ensamblaje, contribuyendo a mejorar la calidad y reducir tiempos.

Evaluación y mejora continua del sistema

- **Análisis en tiempo real:** añadir herramientas que permitan la evaluación continua del sistema, proporcionando datos en tiempo real para realizar ajustes y mejoras constantes.
- **Retroalimentación de usuarios:** establecer mecanismos sólidos para recopilar y analizar la retroalimentación de los usuarios, utilizando encuestas, entrevistas y análisis de uso. Esta información será vital para identificar áreas de mejora y guiar el desarrollo de futuras actualizaciones del sistema.

Integración de la inteligencia artificial y de redes neuronales

- **Optimización del HMI:** integrar la inteligencia artificial en la Interfaz Hombre-Máquina para mejorar el reconocimiento de voz y personalizar las interacciones basadas en el comportamiento del usuario.
- **Predicción y mantenimiento preventivo:** emplear algoritmos de inteligencia artificial para predecir fallos en el sistema y realizar mantenimientos preventivos, aumentando la fiabilidad y longevidad de los equipos.
- **Bases de datos inteligentes:** crear una base de datos para almacenar y analizar datos, utilizando inteligencia artificial o redes neuronales para identificar patrones y optimizar el rendimiento del sistema.



Estas propuestas no solo buscan mejorar y expandir las capacidades del sistema desarrollado en este proyecto, sino también abrir nuevas posibilidades y aplicaciones en diversos sectores industriales. El objetivo es avanzar en la mejora y eficiencia de los sistemas industriales, contribuyendo al desarrollo de soluciones innovadoras y adaptadas a las necesidades cambiantes.





Bibliografía

- Barrientos, A., Peñín, L., Balaguer, C., & Aracil, R. (2007). *FUNDAMENTOS DE ROBÓTICA*. Madrid: McGraw-Hill.
- Brunete, A., San Segundo, P., & Herrero, R. (2024). *Introducción a la Automatización Industrial*. Madrid.
- DenisFR. (4 de Julio de 2018). *GitHub*. Recuperado el 10 de Mayo de 2024, de <https://github.com/DenisFR/RSConnectGIOToSnap7>
- EDS Robotics. (5 de Mayo de 2020). Recuperado el 3 de Junio de 2024, de <https://www.edsrobotics.com/blog/sistemas-de-vision-artificial-tipos-aplicaciones/>
- Elecproy. (11 de Marzo de 2024). Recuperado el 3 de Junio de 2024, de <https://elecproy.com/es/blog/vision-artificial-que-es-tipos-aplicaciones/>
- FasterCapital. (18 de Abril de 2024). Recuperado el 4 de Junio de 2024, de <https://fastercapital.com/es/contenido/Rompiendo-barreras-con-FFDLC-avances-en-el-reconocimiento-de-voz.html#Tipos-de-sistemas-de-reconocimiento-de-voz>
- Fernández, D. G. (Julio de 2017). *uvadoc*. Recuperado el 12 de Junio de 2024, de <http://uvadoc.uva.es/handle/10324/24682>
- Fernández, R. P. (Junio de 2016). *uvadoc*. Recuperado el 12 de Junio de 2024, de <http://uvadoc.uva.es/handle/10324/18998>
- Fraile Marinero, J. C. (Octubre de 2023). *Campus Virtual UVa, Dpto. Ingeniería de Sistemas y Automática, Introducción a la robótica*. Recuperado el 5 de Junio de 2024
- García, J. D. (18 de Mayo de 2020). *uvadoc*. Recuperado el 12 de Junio de 2024, de <http://uvadoc.uva.es/handle/10324/41133>
- García, R. S. (Junio de 2021). *uvadoc*. Recuperado el 12 de Junio de 2024, de <https://uvadoc.uva.es/handle/10324/47254>
- IBM. (s.f.). Recuperado el 3 de Junio de 2024, de <https://www.ibm.com/es-es/topics/computer-vision#citation2>
- InfoPLC. (10 de Enero de 2021). Recuperado el 18 de Junio de 2024, de <https://www.infoplcn.net/descargas/107-siemens/software-step7-tiaportal/tia-portal/3195-nettoplcsim-extension-red-simulador-plcsim-siemens>



- Logicbus*. (17 de Junio de 2019). Recuperado el 10 de Junio de 2024, de <https://www.logicbus.com.mx/blog/protocolos-de-comunicacion-industriales/>
- ningenia*. (2 de Diciembre de 2015). Recuperado el 10 de Junio de 2024, de <https://ningenia.com/comunicaciones-industriales-i/>
- Robotics, A. (29 de Septiembre de 2012). *ABB*. Recuperado el 18 de Junio de 2024, de <https://library.e.abb.com/public/5240940e78c27430c1257b67004c6759/3HAC032104-es.pdf>
- Robotics, A. (06 de Septiembre de 2019). *ABB*. Recuperado el 07 de Julio de 2024, de <https://new.abb.com/products/robotics/robots/articulated-robots/irb-4600>
- Sanchis Llopis, R., Romero Pérez, J. A., & Ariño Latorre, C. V. (2010). *Automatización Industrial*. Castellón de la Plana.
- Santos, Á. G. (Enero de 2016). *uvadoc*. Recuperado el 12 de Junio de 2024, de <http://uvadoc.uva.es/handle/10324/17056>
- Sanz, D. (s.f.). *Juegos Robótica*. Recuperado el 14 de Julio de 2024, de <https://juegosrobotica.es/diagrama-de-flujo/>
- Sicma21*. (22 de Abril de 2021). Recuperado el 10 de Junio de 2024, de https://www.sicma21.com/que-son-las-redes-de-comunicacion-industrial/#Que_son_las_redes_de_comunicacion_industrial