



UNIVERSIDAD DE JAÉN

Escuela Politécnica Superior de Jaén

Trabajo Fin de Grado

GEMELO DIGITAL DE LA ESTACIÓN 2 DE CÉLULA DE FABRICACIÓN FLEXIBLE FMS200

Alumno: Guillermo González Moreno

Tutor: Prof. Da. Elisabet Estévez Estévez

Prof. D. Alejandro Sánchez García

Dpto: Ing. Electrónica y Automática



Universidad de Jaén
Escuela Politécnica Superior de Jaén
Departamento de Ingeniería Electrónica y Automática

Don Alejandro Sánchez García y Doña Elisabet Estévez Estévez , tutores del Proyecto Fin de Carrera titulado: Gemelo Digital de la Estación 2 de Célula de Fabricación Flexible FMS200, que presenta Guillermo González Moreno, autoriza su presentación para defensa y evaluación en la Escuela Politécnica Superior de Jaén.

Jaén, Febrero de 2024

El alumno:

Los tutores:

GUILLERMO GONZÁLEZ MORENO

ALEJANDRO SÁNCHEZ GARCÍA
Y ELISABET ESTÉVEZ ESTÉVEZ

Resumen

En el siguiente proyecto se ha realizado una aportación al desarrollo de los modelados digitales de la célula flexible de fabricación FMS-200 dentro de la Plataforma GRAV-DT.

Este proyecto ha tenido dos principales objetivos. En primer lugar, el desarrollo del modelado de un sensor analógico de desplazamiento lineal. Con este desarrollo se contribuye también a asentar las bases para la implementación en el futuro de otros tipos de sensores analógicos.

En segundo lugar, se ha desarrollado el modelado digital de la estación de fabricación FMS-202. Esta estación se encuentra presente en los laboratorios de la Universidad de Jaén y tiene como objetivo la medición y selección de rodamiento y su posterior salida o rechazo.

Por último, se ha establecido su conexionado con el PLC y la programación de este para la simulación de la estación.

Abstract

The following project has contributed to the development of a digital model of the FMS-200 flexible manufacturing cell within the GRAV-DT Platform.

This project has had two main objectives. Firstly, the development of the modeling of an analog linear displacement sensor. This development also contributes to lay the foundations for the future implementation of other types of analog sensors.

Secondly, the digital model of the FMS-202 manufacturing station has been developed. This station is present in the laboratories of the University of Jaén and its objective is the measurement and selection of bearings and their subsequent output or rejection.

Finally, its connection with the PLC and its programming for the simulation of the station has been developed.

Índice

1	INTRODUCCIÓN	1
1.1	Motivación	3
1.2	Objetivos	3
1.3	Estructura	5
2	Plataforma GRAV-DT.....	7
2.1	Modelo 3D	7
2.1.1	Autodesk Inventor	7
2.1.2	Processing	9
2.2	Ficheros de información	10
2.3	Clases Java.....	13
2.3.1	Librerías importadas	13
2.3.2	Clases previamente creadas.....	14
3	Modelado digital de un sensor Analógico.....	18
3.1	Modelo 3D	18
3.2	Ficheros de información	22
3.3	Clases Java.....	24
4	Diseño y Desarrollo del Modelo de la Estación FMS-202	30
4.1	Descripción general de la estación.....	30
4.2	Modelo 3D	33
4.2.1	Base fija	33

4.2.2	Alimentador rodamientos	34
4.2.3	Brazo Giratorio	36
4.2.4	Ascensor	39
4.2.5	Brazo de salida	43
4.3	Modelo de información	50
5	Arranque y Ejecución del Modelo Digital de la Estación FMS 202...	59
6	Conclusiones	66
7	Bibliografía	67
Anexos		68

Figuras

Figura 1. Representacion de un Gemelo Digital	2
Figura 2. Tipos de representaciones digitales	2
Figura 3. Potenciómetros Lineales.	4
Figura 4. Interfaz de Autodesk Inventor 2023	8
Figura 5. Icono Autodesk Inventor	9
Figura 6. Icono de Processing	10
Figura 7. Esquema XML Estación FMS-202.....	11
Figura 8. Definición cilindro en XML	12
Figura 9. Icono NetBeans	13
Figura 10. Icono Snap7.....	14
Figura 11. Clases previamente creadas.	15
Figura 12. Valores del convertidor ADC.....	18
Figura 13. Anclaje sensor analógico con riel.	19
Figura 14. Cuerpo del sensor de desplazamiento lineal.	20
Figura 15. Palpador potenciómetro lineal.	21
Figura 16. Fichero de información XML del sensor analógico.	22
Figura 17. Atributos objeto rodamiento.	23
Figura 18. Declaración atributos de la clase "Escena".....	24
Figura 19. Declaración atributos de la clase "Escena".....	25
Figura 20. Función writeAnalogInput()	26

Figura 21. Representación del valor del Data block.	26
Figura 22. Variables declaradas en SensorAnalogico.	27
Figura 23. Subclase updateEstado()	28
Figura 24. Generación de numero aleatorio.	29
Figura 25. Selección representación modelo3d.....	29
Figura 26. Imagen de la estación FMS-202.....	32
Figura 27. Base fija estación FMS-202.....	34
Figura 28. Vástago cilindro A.....	35
Figura 29. Alimentador de rodamientos.....	35
Figura 30. Soporte brazo giratorio.	36
Figura 31. Brazo	37
Figura 32. Cuerpo de la Pinza	38
Figura 33. Palas brazo giratorio.....	38
Figura 34. Ensamblaje segunda parte estación.....	39
Figura 35. Parte del ascensor incluido en la base fija.....	40
Figura 36. Base elevador y cuerpo del cilindro E.....	41
Figura 37. Centrador.....	41
Figura 38. Vástago cilindro E.....	42
Figura 39. Conjunto completo del elevador más el sensor.	43
Figura 40. Base soporte y cuerpo cilindro.....	44
Figura 41. Brazo giratorio final.....	45

Figura 42. Palas brazo de salida.	46
Figura 43. Seta de emergencia.....	46
Figura 44. Activar sensor presencia rodamiento.....	47
Figura 45. Botón de comienzo.	47
Figura 46. Botón de parada.	47
Figura 47. Selector de modos.....	47
Figura 48. Botón de reseteo de la estación.	47
Figura 49. Rodamiento correcto.	48
Figura 50. Rodamiento incorrecto.....	48
Figura 51. Vista del ensamblaje completo de la estación FMS-202.	49
Figura 52. Vista adicional ensamblaje estación FMS-202.	49
Figura 53. Cilindro anidado.....	50
Figura 54. Datos cilindro 1.....	51
Figura 55. Estructura Cilindro 2.	52
Figura 56. Estructura atributo rodamiento.....	53
Figura 57. Estructura del cilindro 4.	54
Figura 58. Estructura del cilindro 5.	55
Figura 59. Estructura cilindros 6, 7 y 8.....	56
Figura 60. Estructura de los pulsadores.	57
Figura 61. Estructura luces.....	58
Figura 62. Ejecución de la estación FMS-202.	59

Figura 63. Configuración NetToPLCsim.	60
Figura 64. Establecimiento de alta prioridad.....	60
Figura 65. Creación del proyecto en TIA portal.	61
Figura 66. Selección de PLC en TIA Portal.	61
Figura 67. Configuración PLC TIA Portal.....	62
Figura 68. Configuración dirección entradas.	62
Figura 69. Generar Data block.....	63
Figura 70. Ejemplo de uso de Data block como entrada analógica.	64
Figura 71. Carga programa en PLC.....	64
Figura 72. Modelo digital conectado al PLC	65

1 INTRODUCCIÓN

El propósito de este proyecto es establecer una plataforma destinada a la generación de gemelos digitales, con especial atención a la célula flexible de fabricación de la serie FMS-200 de Siemens ubicada en las instalaciones de la Universidad de Jaén. Actualmente, un grupo de estudiantes del grado de ingeniería electrónica industrial de la Universidad de Jaén está trabajando en el desarrollo de esta plataforma.

Gemelos Digitales (GD) (Toala, F.J. et al., 2022) es una tecnología de software con el que se trata de representar digitalmente una realidad física, bien sea esta un producto, proceso o sistema; utilizando como aliados a los recursos tecnológicos tales como: Big Data, IoT, Inteligencia Artificial, Cloud Computing y Machine Learning, entre otras, con la finalidad de proporcionar herramientas para la toma de decisiones, implementando adicionalmente capacidades de simulación y predicción.

Los gemelos digitales proporcionan una gran versatilidad a los usuarios de estos modelos a la hora de implementar y probar modificaciones en los sistemas sin tener la necesidad de estar conectado directamente a él, pudiendo trabajar sobre el modelo en remoto. A parte de esto los gemelos digitales y los modelos digitales tienen muchas más ventajas como permitir trabajar en un entorno menos costoso y más seguro para el implementador. En la Figura 1 (Festo, 2023) se puede ver la representación de un gemelo digital junto a su modelo real.

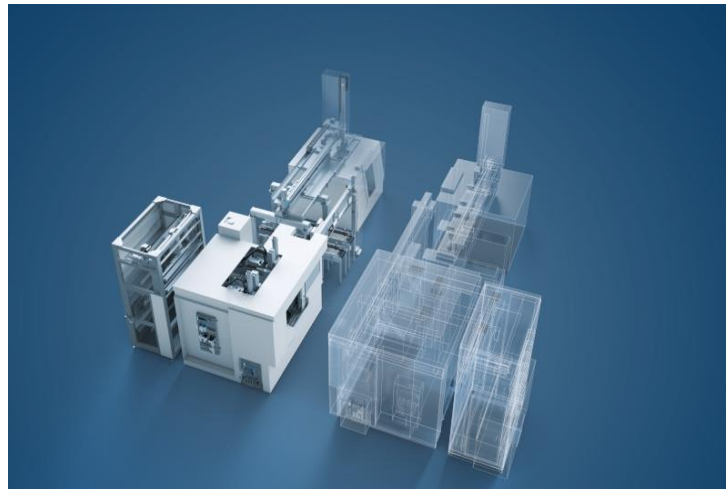


Figura 1. Representación de un Gemelo Digital

Hay que tener en cuenta la diferenciación entre gemelo digital y modelo digital. La principal diferencia radica en que un Gemelo Digital representa una entidad física o un sistema del mundo real con datos actualizados en tiempo real. En otras palabras, refleja el comportamiento y el estado en tiempo real del objeto real dentro de un entorno digital. Por otro lado, un modelo digital es una representación del sistema, sin necesariamente estar conectado a un sistema en tiempo real del cual reciba datos. Ambos modelos son muy versátiles permitiendo el modelo digital la desconexión total del sistema. En la Figura 2 (Medium, 2023) se puede ver la diferenciación entre modelo digital y gemelo digital.

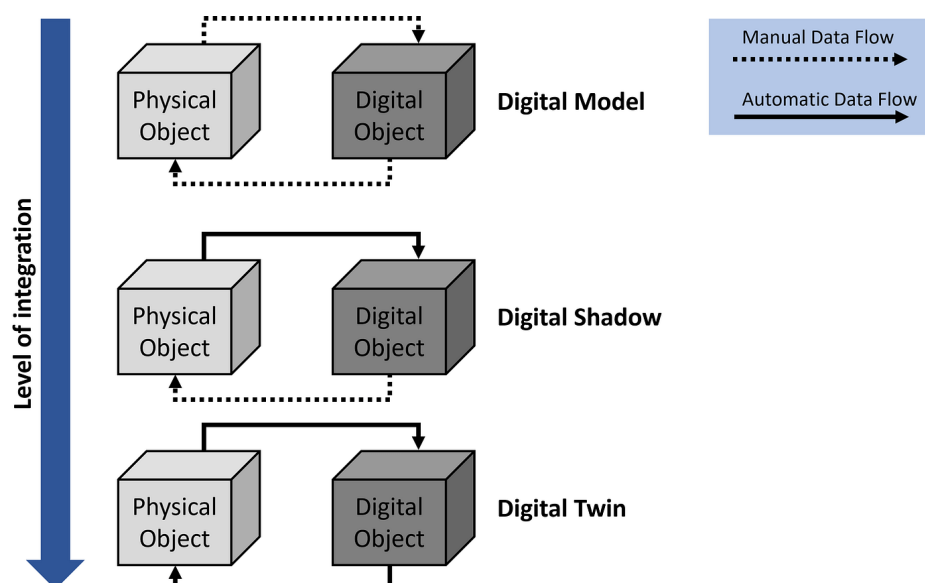


Figura 2. Tipos de representaciones digitales

1.1 Motivación

Este proyecto está motivado debido a la reciente situación con respecto a la pandemia ocasionada por el virus COVID-19. Esta circunstancia impidió la docencia presencial durante mucho tiempo imposibilitando el acceso a aula y consecuentemente la posibilidad de trabajar con las estaciones FMS-200.

A raíz de esta situación se ha visto la necesidad de desarrollar una representación digital de las células de fabricación disponibles en la Escuela Politécnica de Jaén. Este desarrollo también permitiría una mayor eficiencia a la hora de trabajar en las estaciones al poder trabajar varios alumnos simultáneamente en el modelo digital, no siendo necesario estar trabajando presencialmente en el modelo real y evitar la saturación de gente trabajando a la vez en el laboratorio.

A parte este desarrollo permite el estudio de posibles soluciones a diversas situaciones e implementar decisiones con un conocimiento previo previniendo de esta forma la aparición de errores en el modelo real, salvando la integridad del sistema y del propio operador del sistema.

Como conclusión se puede obtener que el desarrollo de este modelo digital puede tener un impacto positivo muy grande en la mejora del aprendizaje permitiendo por ejemplo trabajar desde casa o simular distintas soluciones. Este tipo de representación digital también puede tener un fuerte impacto en la industria ya que se puede observar la tendencia de esta hacia la denominada “Industria 4.0” la cual apuesta por la hiperconectividad y la interconexión de los sistemas.

1.2 Objetivos

El objetivo de este trabajo de fin de grado es el modelado digital de un sensor analógico y la inclusión del mismo dentro de la plataforma GRAV-DT. La implementación de este sensor analógico se hará con la intención de que se pueda usar de manera polivalente dentro de la plataforma. Esto implicara que se

El desarrollo de este trabajo se centrará en la creación de un potenciómetro lineal. Un potenciómetro lineal es un tipo de sensor que se utiliza para medir la posición o desplazamiento lineal de un objeto. Consiste en una resistencia variable con un contacto deslizante que se mueve a lo largo de la resistencia. Algunos de los usos de estos sensores pueden ser control de posición en maquinaria, instrumentación o medición de desplazamiento de las articulaciones y extremidades de robots. En la Figura 3 (Industic, 2024) se puede observar distintos tipos de potenciómetros lineales.



El segundo objetivo de este trabajo es el desarrollo y realización de un modelo digital de la estación FMS-202. Se busca lograr una representación detallada y fiel de la estación, donde el modelo digital sirva como una réplica precisa que permita comprender y simular su funcionamiento en términos visuales y en el código del PLC. Este enfoque tiene como propósito facilitar un análisis exhaustivo y la experimentación virtual de la estación, proporcionando

una herramienta valiosa para diversos propósitos, desde la evaluación de su rendimiento hasta la realización de pruebas y optimización de procesos. El objetivo de la estación es la medición de rodamientos y su posterior salida a la cinta transportadora o descarte en caso de que sea una pieza no válida. Con este desarrollo se consigue que el alumno pueda entender el funcionamiento y la parametrización de un sensor analógico junto al resto de sensores y actuadores de la FMS-202 en el PLC de Siemens.

Como tercer y último objetivo se realizará la simulación del proceso realizado en la estación mediante la conexión al PLCSim de Siemens.

1.3 Estructura

Con el propósito de brindar una comprensión concisa sobre las categorías temáticas del proyecto, se ha elaborado este apartado dedicado a la estructura del mismo. Aquí se delinearán y detallarán los epígrafes fundamentales que conforman la organización y contenido del proyecto.

El capítulo 2 **Plataforma GRAV-DT** describe a la Plataforma de Gemelos digitales junto a todas sus funcionalidades y todas las herramientas utilizadas para el desarrollo de la aplicación.

El capítulo 3 **Modelado Digital de un Sensor Analógico** describe la creación del modelo del Sensor, la generación del fichero XML con los datos necesarios para el sensor y las clases de Java necesarias para el funcionamiento del sensor.

El capítulo 4 **Diseño y Desarrollo del Modelo Digital de la Estación FMS 202** describe la creación de piezas, ensamblaje de estas mediante Processing e implementación de los distintos sensores y actuadores, así como la asignación de las correspondientes entradas y salidas analógicas.

El capítulo 5 **Arranque y ejecución del Modelo Digital de la Estación FMS 202** describe la configuración inicial necesaria y el proceso de arranque y utilización del Modelo Digital.

En el capítulo 6 **Conclusiones** se busca ofrecer un cierre integral al trabajo realizado, destacando los logros obtenidos y proporcionando una perspectiva sobre las direcciones que podrían explorarse en el futuro para ampliar y mejorar aún más el proyecto.

Finalmente se encuentran los Anexos:

Anexo A: Planos eléctricos de la estación

Anexo B: Fichero de Información estación FMS-202

Anexo C: Grafcet con la programación de la estación FMS-202

Anexo D: Video funcionamiento de la estación

2 Plataforma GRAV-DT

La plataforma de gemelos digitales se encuentra integrada dentro del Grupo de Robótica, Automática y Visión por Computador (GRAV). El "Grupo de Robótica, Automática y Visión por Computador (GRAV)" trabaja en la investigación, el desarrollo y la implementación de soluciones tecnológicas avanzadas que permiten resolver problemas complejos de automatización industrial mediante la realización de proyectos, tanto en el ámbito público como en el de la transferencia de tecnología a empresas (Grupo de Robótica, Automática y Visión por Computador (GRAV) - Ujaen, s.f.).

La plataforma para los gemelos digitales tiene como objetivo la digitalización de las estaciones FMS 200 disponibles en la Escuela Politécnica de Jaén. Con ese objetivo una serie de herramientas han sido desarrolladas en distintos programas para facilitar la labor de la digitalización de las estaciones, a continuación, se pasará a enumerar dichas herramientas.

A continuación, en los siguientes apartados se describe las herramientas ya implementadas dentro de la Plataforma GRAV-DT para el modelado 3D, la gestión de los ficheros de información de la estación y las clases de java ya implementadas dentro de Java.

2.1 Modelo 3D

Para el modelado de piezas y la animación de estas se ha usado dos programas Autodesk Inventor y Processing. En los siguientes subapartados se procede a describir el funcionamiento de ambos.

2.1.1 Autodesk Inventor

Como primera etapa tenemos el modelado de las piezas en 3D. Como programa para modelar las distintas partes del proyecto se ha optado por el programa Autodesk Inventor. Autodesk Inventor (Interfaz, Figura 4) consiste un software especializado en el modelado paramétrico de sólidos en tres dimensiones. Este programa proporciona un conjunto de herramientas avanzadas diseñadas específicamente para el diseño mecánico tridimensional,

así como para la documentación y simulación de productos. Este enfoque integral simplifica significativamente el proceso de diseño, facilitando la labor de los profesionales en este campo.

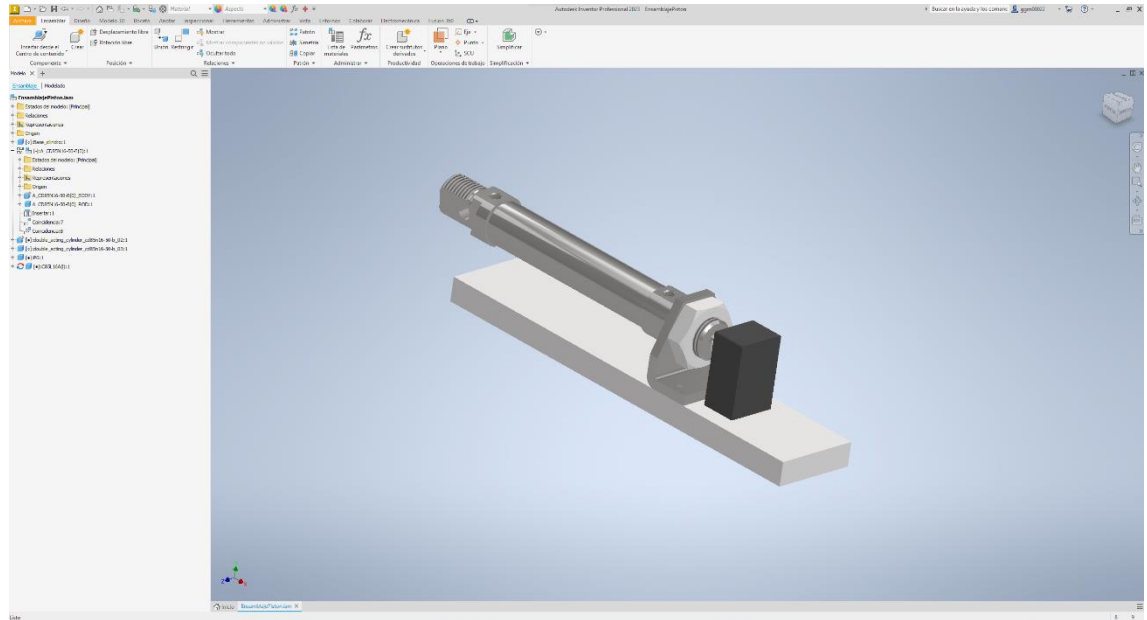


Figura 4. Interfaz de Autodesk Inventor 2023

Las principales funciones de Autodesk Inventor (Inventor, 2024) usadas para el desarrollo de este proyecto son:

- El modelado paramétrico. Autodesk Inventor simplifica notablemente la labor del diseñado, eliminando procesos complejos que podrían ralentizar el trabajo. A través de funciones intuitivas anticipándose a las necesidades del trabajo y ofreciendo un valioso apoyo en el proceso geométrico.
- En lo que respecta al modelado de ensamblajes, Autodesk Inventor optimiza la tarea al facilitar la inserción de componentes específicos en estructuras y ensamblajes. Proporciona opciones para la creación de otros componentes dentro del mismo ensamblaje, permitiendo la finalización eficiente del diseño con componentes tanto conocidos como normalizados.
- En cuanto a la creación de dibujos, Inventor ofrece una amplia variedad de componentes que contribuyen a lograr dibujos perfectos

del diseño del producto. Con vistas de ensamblaje configurables y relacionadas entre sí, este software proporciona una herramienta completa para la representación visual detallada.

Autodesk Inventor (Icono, Figura 5) se presenta como una herramienta integral que mejora la eficiencia y calidad en todas las fases del proceso de diseño y producción.



Figura 5. Icono Autodesk Inventor

2.1.2 Processing

Como herramienta para la realización de las animaciones y el movimiento de las distintas piezas dentro de cada estación se usará Processing (Icono, Figura 6). Processing es una variante de Java de código abierto concebida específicamente para la creación de arte gráfico, animaciones y diversas aplicaciones visuales (ProgramarFacil, s.f.).

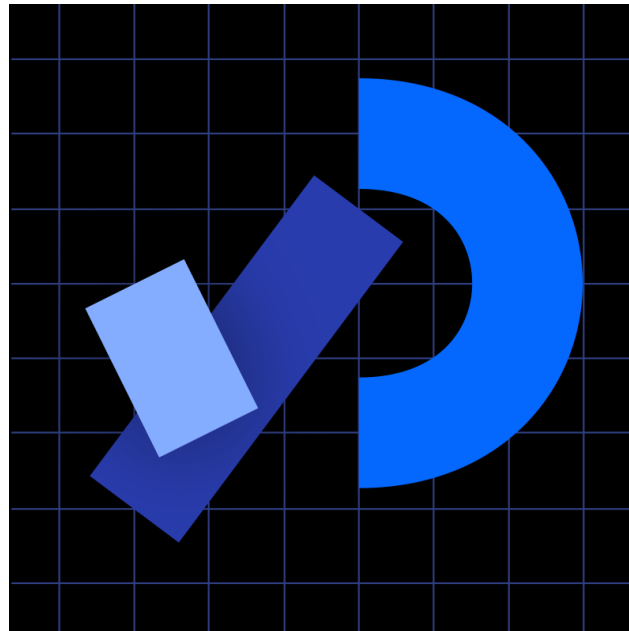


Figura 6. Icono de Processing

Las principales ventajas de Processing son:

- Su aprendizaje es sumamente sencillo y en solo unos minutos se puede comenzar a programar. Processing simplifica esta tarea, ayudando a evitar la frustración que suele acompañar el proceso de aprendizaje de un lenguaje de estas características.
- Pese a su facilidad a la hora de usarlo, Processing es una herramienta muy potente en el desarrollo de entornos gráficos.
- Una característica fundamental es su capacidad de escalabilidad, ya que podemos integrar Processing con aplicaciones Java. En el caso de la Plataforma GRAV-DT se ha optado por la esta opción para integrar esta herramienta en la interfaz de desarrollo NetBeans.

2.2 Ficheros de información

Para la gestión de la información y datos usaremos los archivos de formato XML. El nombre de XML es una abreviación de Extensible Markup Language, cuya traducción sería Lenguaje de Marcado Extensible.

Los archivos XML constan de etiquetas que proporcionan datos e información destinada a ser procesada. Estas etiquetas pueden presentarse de

manera individual o estar anidadas. Generalmente, un archivo XML contiene una cantidad significativa de información que requiere un procesamiento adecuado por parte del usuario (Manuel Alconchel, 2019).

Se usará un fichero de tipo XML llamado “escena.xml” para incluir en él toda la información necesaria para la clase Escena. En este fichero se definirá toda la información relacionada con el entorno del modelo, posicionamiento de las distintas partes o asignación de entradas y salidas de los distintos actuadores y sensores. En la siguiente imagen (Esquema XML Estación FMS-202, Figura 7) se puede ver el esquema completo en XML de la estación FMS-202.

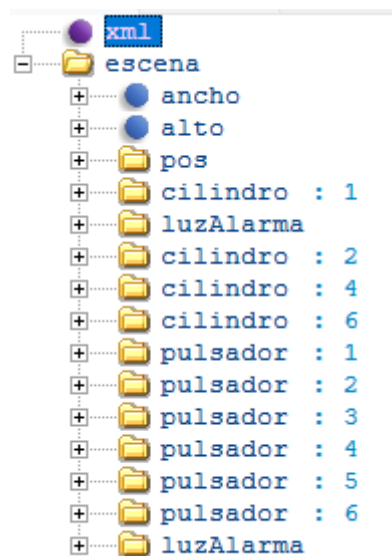


Figura 7. Esquema XML Estación FMS-202

A continuación (Definición cilindro en XML, Figura 8) se muestra a modo de ejemplo la definición de los distintos datos necesarios para la configuración.

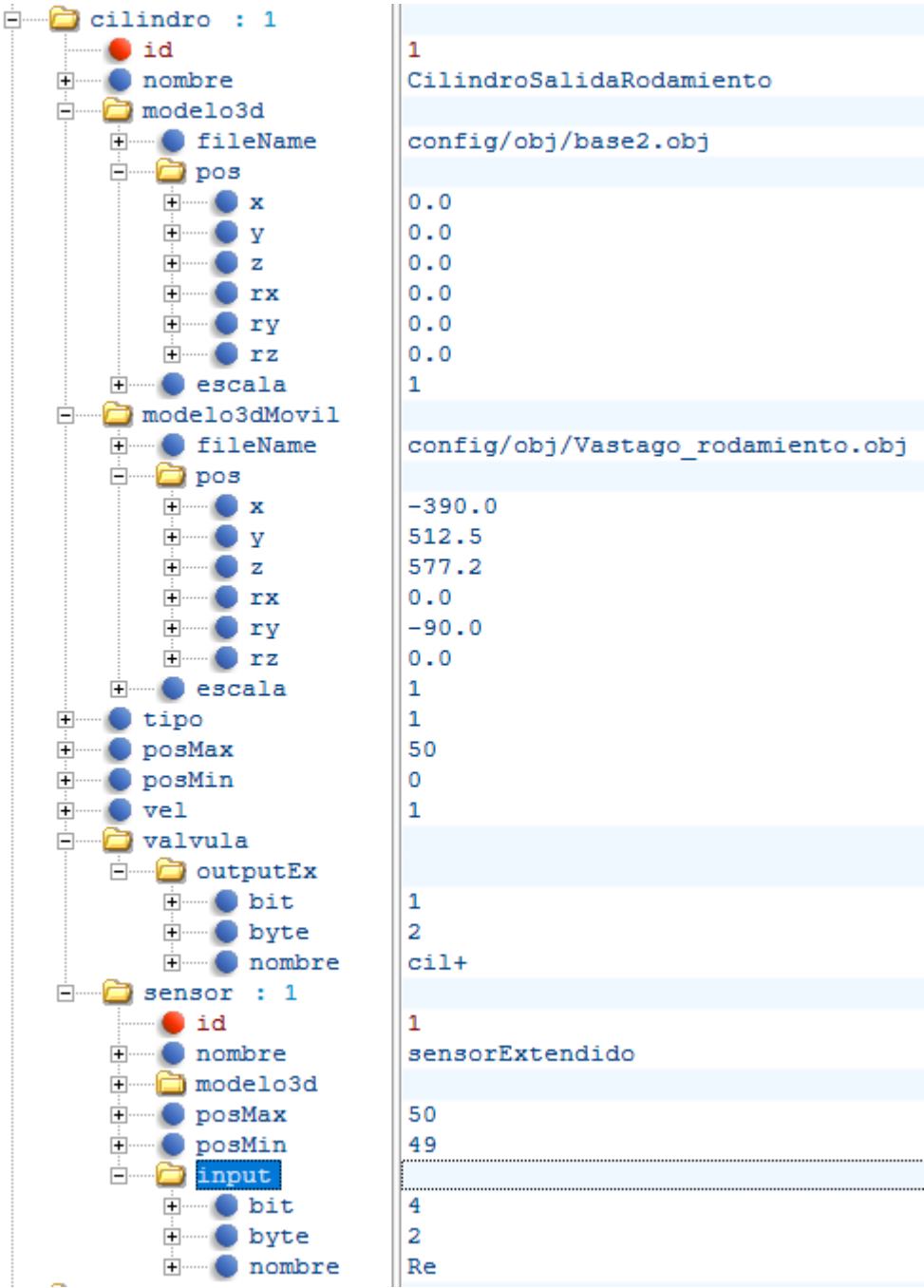


Figura 8. Definición cilindro en XML

En la anterior figura se observa cómo se define un cilindro un objeto cilindro junto a una válvula y un sensor anidados dentro del cilindro. Se puede ver como se indica parámetros como la posición de los modelos3D y los modelos móviles, la asignación de las entradas y salidas de los actuadores y sensores o sus rangos de desplazamiento.

2.3 Clases Java

Como interfaz de desarrollo se ha optado por utilizar NetBeans. NetBeans IDE (Icono NetBeans, Figura 9) es una plataforma de desarrollo integrado de código abierto y gratuito diseñada para la creación de aplicaciones en sistemas operativos como Windows, Mac, Linux y Solaris. Este entorno simplifica el proceso de desarrollo para aplicaciones web, empresariales, de escritorio y móviles, utilizando las plataformas Java y HTML5.



Figura 9. Icono NetBeans

NetBeans IDE proporciona herramientas de alta calidad destinadas al desarrollo de aplicaciones Java en diversas áreas, abarcando aplicaciones móviles, de escritorio, empresariales y web. Destaca como el primer IDE que brinda soporte para las últimas versiones de JDK, Java EE y JavaFX. Además, ofrece resúmenes inteligentes que facilitan la comprensión y gestión de aplicaciones, incluyendo una compatibilidad inmediata con tecnologías populares como Maven (Oracle, s.f.).

2.3.1 Librerías importadas

Netbeans como cualquier IDE de Java tiene la capacidad de importar librerías estándar para facilitar la labor de la programación. En el apartado del procesamiento de los modelos 3D ya se ha hablado de Processing, esta librería permite trabajar y desarrollar distintas aplicaciones gráficas como animaciones o artes gráficas.

También se ha importado dentro de la plataforma la librería Mokka7. Mokka 7 es el porteo en Java de Snap7 (Icono Snap7, Figura 10). Mokka7 se presenta como un conjunto de clases en código fuente que se integran en tu proyecto Java, permitiéndote establecer comunicación con PLCs S7 (Snap7-Sourceforge, s.f).



Figura 10. Icono Snap7

El protocolo S7 es un protocolo propietario de Siemens. Por lo general, la comunicación implementada por Siemens sigue el modelo clásico de maestro-esclavo (o cliente-servidor), en el cual un PC maestro (o cliente) emite solicitudes S7 al dispositivo PLC esclavo (o servidor). El protocolo se centra en la transmisión de funciones y/o comandos, lo que implica que típicamente la comunicación incluye una solicitud S7 seguida de una respuesta correspondiente a dicha solicitud (Inprotech, 2023).

2.3.2 Clases previamente creadas

La plataforma GRAV-DT ha sido creada en base al trabajo desarrollado por tutores y alumnos que han pasado por esta. Consecuentemente muchas clases dentro de NetBeans han sido previamente desarrolladas. A continuación, se procede a la descripción de estas distintas clases (Figura 11, Diagrama Clases previamente creadas).

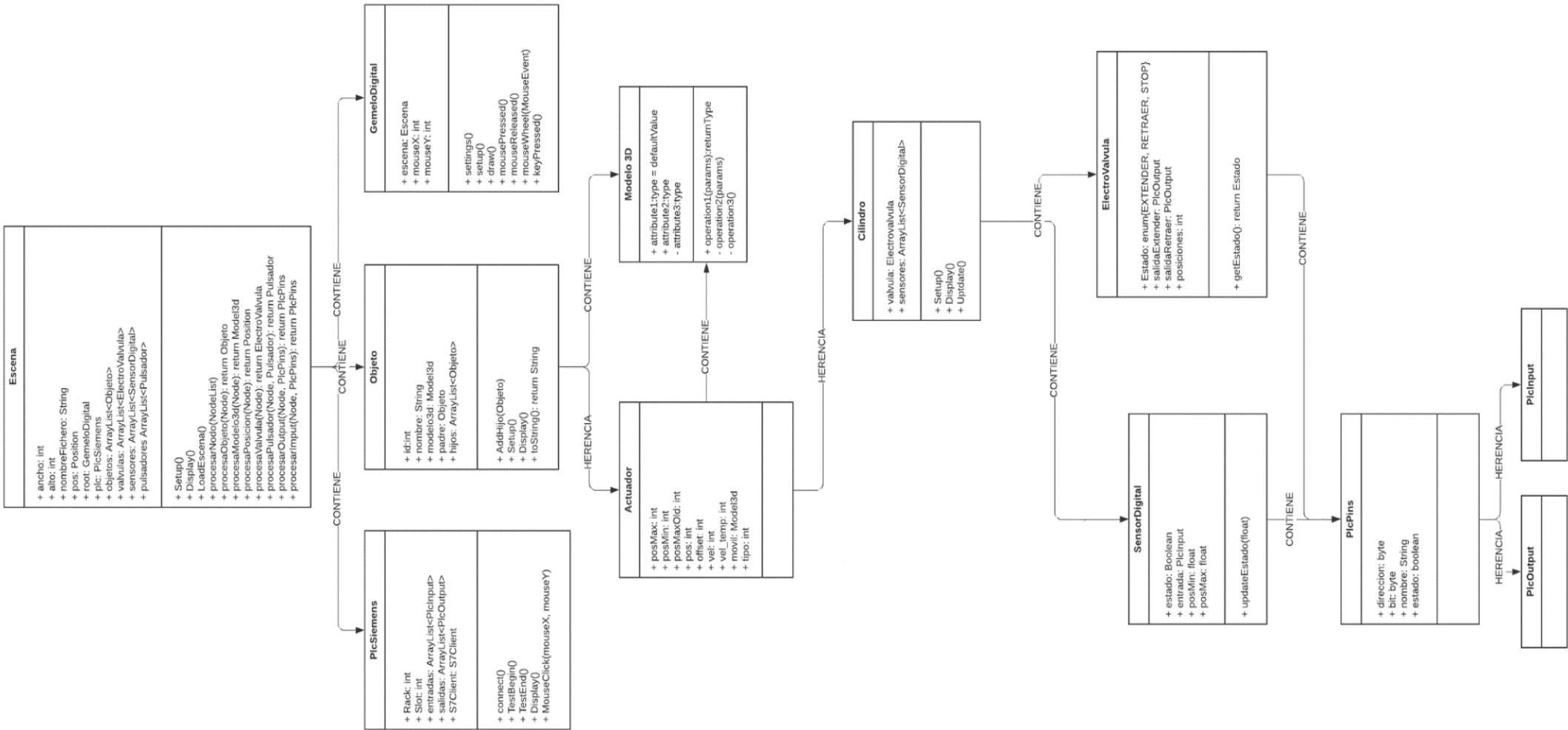


Figura 11. Clases previamente creadas.

La clase GemeloDigital desempeña un papel crucial, siendo la fuerza impulsora detrás de la escena. Esta clase inicializa todos los módulos, incluido Processing, y dibuja toda la estación de manera cíclica, actualizando visualmente la posición de todas las partes.

La clase Escena obtiene todos los parámetros de un archivo XML vinculado a ella y asume la responsabilidad de procesar individualmente todos los elementos del gemelo digital para generar la escena.

La clase Objeto define un objeto con nombre, representación 3D e identificador. Es la clase principal de Actuator.

La clase model3D representa gráficamente todos los objetos. Tiene parámetros propios como nombre, posición, nombre del .obj a representar y escala.

La clase Position determina la posición del objeto con parámetros para los ejes y la orientación del objeto.

La clase Actuator es descendiente de la clase Objeto. Esta clase define a un objeto de tipo actuador. Además de los parámetros de la clase Objeto, se añaden un modelo 3D móvil, velocidad de movimiento, tipo de movimiento, posición máxima y posición mínima.

La clase Cilindro es descendiente de Actuator, definiendo así un actuador específico de tipo cilindro.

La clase Electrovalvula define el manejo del cilindro, ya sea para extender o retraer el pistón.

La clase PLCSiemens simula el comportamiento del PLC, encargándose de gestionar las entradas y salidas del proceso.

La clase PlcPins simula una dirección de entrada o salida del PLC, conteniendo la dirección y su estado, junto con un nombre distintivo. De esta clase se derivan dos clases hijas, PlcInput y PlcOutput, creadas para diferenciar el procesamiento del pin del PLC según si es una entrada o una salida.

La clase Pulsador genera un pulsador para influir en las entradas del gemelo digital. Puede ser de dos tipos, pulsador o interruptor.

La clase SensorDigital representa los sensores de tipo digital en el gemelo digital, con un estado, un bit del PLC y posiciones máxima y mínima para determinar el estado del sensor.

3 Modelado digital de un sensor Analógico

A continuación, se describirá la creación, desarrollo e implementación de un modelo digital de un sensor analógico. Este desarrollo está centrado en el modelado de un potenciómetro lineal, siendo el desarrollo de este adaptable a cualquier otro tipo de sensor analógico.

El esquema de conexionado consiste en un divisor de tensión en el que la entrada analógica es capaz de leer las variaciones de tensión en sus pines de entrada. Es muy importante conocer el valor de resolución de los sensores del PLC, en este caso se usará un PLC Siemens S7-1200 con una resolución de 16 bits. Los rangos de tensión van de 0 a 10 V como se puede ver en la siguiente Figura 12 (Siemens, s.f.). Respecto al rango de medición del sensor este es de 0 a 100 mm, estos datos serán necesarios para futuras operaciones de normalizado y escalado de señales analógicas.

A.4.4.3 Rangos de medida de entradas analógicas de tensión (CPU)

Tabla A- 21 Representación de entradas analógicas para tensión (CPU)

Sistema		Rango de medida de tensión	
Decimal	Hexadecimal	De 0 a 10 V	
32767	7FFF	11,852 V	Rebase por exceso
32512	7F00		
32511	7EFF	11,759 V	Rango de sobreimpulso
27649	6C01		
27648	6C00	10 V	Rango nominal
20736	5100	7,5 V	
34	22	12 mV	
0	0	0 V	
Valores negativos		Los valores negativos no se soportan	

Figura 12. Valores del convertidor ADC.

3.1 Modelo 3D

La representación gráfica del modelo 3D del sensor de medición analógico está integrado dentro de la base fija de la estación FMS 202. Este ha sido

modelado en Autodesk Inventor. A continuación, se muestran las distintas partes del sensor.

Anclaje

El sensor analógico se encuentra montado en una torre que actúa como un ascensor en el que sube el rodamiento y el medido por el palpador del sensor. Este sensor se encuentra en lo alto de la torre conectado a ella mediante el siguiente anclaje de la Figura 13.

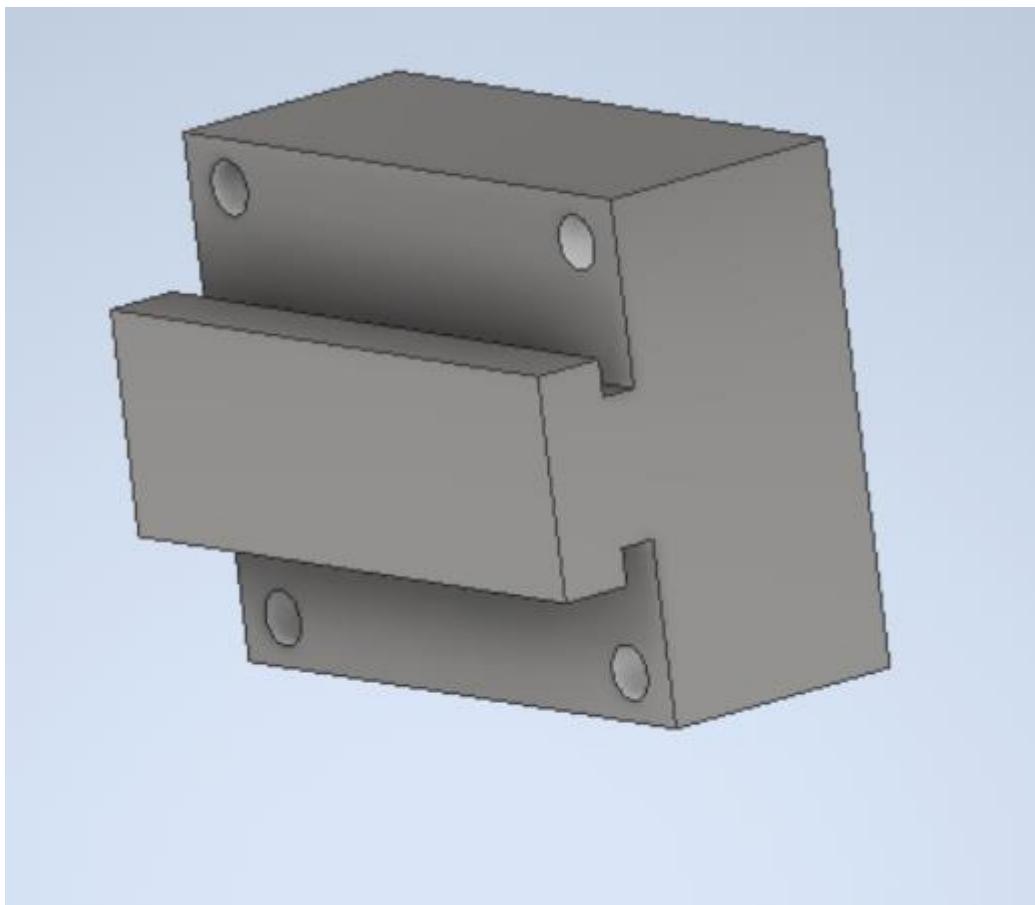


Figura 13. Anclaje sensor analógico con riel.

Cuerpo sensor Analógico

Este soporte anteriormente descrito va unido al cuerpo del sensor, en el cual se encuentra todas las conexiones eléctricas del sensor. En el interior del cuerpo se encuentra el potenciómetro variable que es movilizado por el palpador

y varia la tensión en los pines del ADC para poder así medir la pieza. En la Figura 14 se puede observar el cuerpo del sensor.

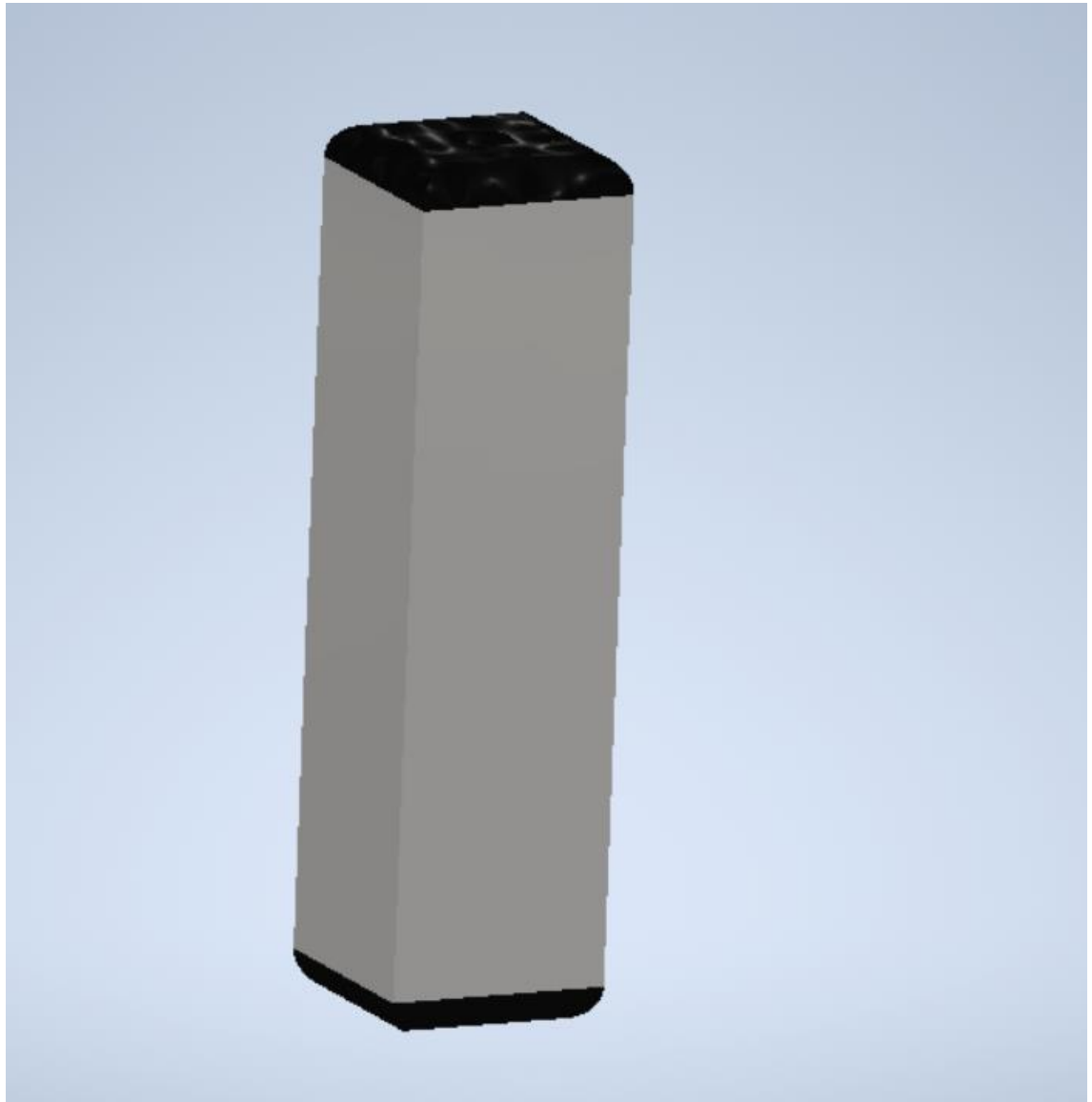


Figura 14. Cuerpo del sensor de desplazamiento lineal.

Palpador

Esta es la parte móvil del actuador. Esta pieza se desplaza al tocar el rodamiento y según la altura de este realiza un recorrido menor o mayor. De esta manera el potenciómetro interior del cuerpo del sensor nos podrá dar una medida que será leída por el PLC. El modelo del palpador se puede observar en la Figura 15.



Figura 15. Palpador potenciómetro lineal.

3.2 Ficheros de información

Sensor analógico

El fichero de información del sensor analógico es mostrado en la Figura 16. Como se puede observar el sensor analógico es un objeto hijo del sensor Cilindro 2 el cual es el encargado de subir y bajar la plataforma en la que se deposita el rodamiento. Consecuentemente la posición medida por el sensor analógico es la posición del Cilindro 2.

cilindro : 2	
id	2
nombre	AscensorSensor
modelo3d	
modelo3dMovil	
tipo	1
posMax	275
posMin	0
vel	1
valvula	
sensor : 2	
sensor : 3	
cilindro : 3	
rodamiento : 1	
sensoranalogico : 1	
id	1
nombre	SensorAnalogico
modelo3d	
pos1	255
pos2	260
input	
DB	6
BytesDB	4
AdressDB	0
StartAddress	0

Figura 16. Fichero de información XML del sensor analógico.

Los atributos definidos para el sensor analógico se pueden observar en la imagen anterior. El objeto “sensoranalogico” contiene los atributos “nombre”, “modelo3d” e “id” como en otros objetos. A parte de estos atributos están los específicos para este objeto, estos son “pos1”, “pos2” y “input”.

Los atributos “pos1” y “pos2” indican la posición en la que el palpador comenzaría a tocar al rodamiento, hay que añadir dos posiciones distintas ya

que puede haber dos tipos de rodamiento de distinto tamaño, impactando uno antes que otro en el palpador.

El atributo “input” contiene a su vez mas atributos descendientes. Estos se usan para indicar la dirección del Data block en el que se va a escribir el valor transmitido por el sensor analógico. Es necesario el uso del Data block para escribir la información transmitida por el sensor analógico debido a la imposibilidad de sobrescribir el valor de la entrada del sensor analógico mediante la librería Mokka 7. Los valores necesarios para definir la entrada es la numeración del Data block, la cantidad de bytes necesarios para transmitir la información, la dirección del Data block y el byte de comienzo de la información en el Data block.

Rodamiento

Además de sensor analógico, es necesario la implementación de un nuevo objeto denominado rodamiento. Esta clase recibirá 2 modelos de rodamiento, uno con la medida correcta y otro con una medida errónea, y mostrará en la escena uno de los dos de manera totalmente aleatoria. El fichero de información en XML de este nuevo objeto se encuentra a continuación en la Figura 17. Como descripción la información más importante es la dirección de los ficheros “.obj” de ambos tipos de rodamiento y la posición asociada a cada uno de ellos.

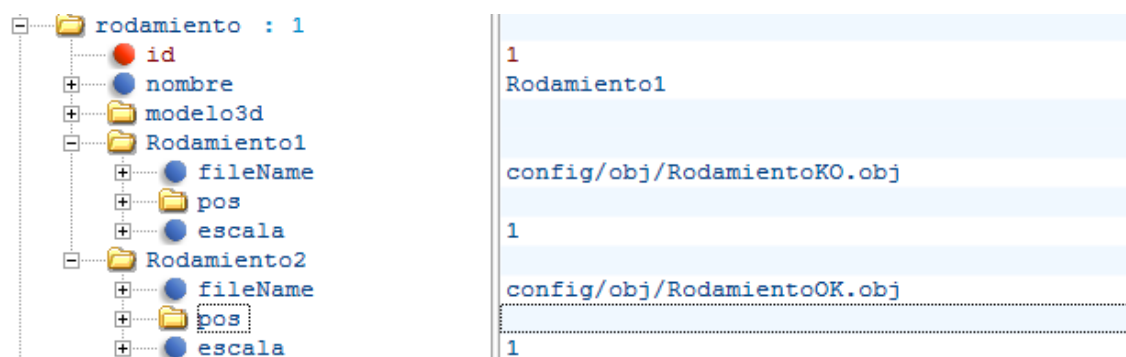


Figura 17. Atributos objeto rodamiento.

3.3 Clases Java

Clase Escena

La clase Escena es una clase ya creada, pero debe de ser modificada para contener la funcionalidad del sensor analógico. Los atributos “modelo3d”, “id” y “nombre” ya se encontraban previamente creados. Además, es necesario añadir un ArrayList en el que se incluya los distintos sensores analógicos.

Dentro de la función “procesaObjeto” es necesario la definición de las funciones “rodamiento1”, “rodamiento2”, “pos1”, “pos2” y “sensoranalogico” como se puede observar en las Figuras 18 y 19.

```
case "Rodamiento1":
    m = procesaModelo3d(childNode,tabs+"\t");
    if(ob instanceof Cilindro){
        Cilindro c = (Cilindro)ob;
        c.setRodamiento1(m);
        System.out.println(tabs+"\t "+childNode.getNodeName());
    }
    else if(ob instanceof Rodamiento){
        Rodamiento r = (Rodamiento)ob;
        r.setRodamiento1(m);
        System.out.println(tabs+"\t "+childNode.getNodeName());
    }
    break;
case "Rodamiento2":
    m = procesaModelo3d(childNode,tabs+"\t");
    if(ob instanceof Cilindro){
        Cilindro c = (Cilindro)ob;
        c.setRodamiento2(m);
        System.out.println(tabs+"\t "+childNode.getNodeName());
    }
}
```

Figura 18. Declaración atributos de la clase "Escena"

```

-----,
case "pos1":
    if(ob instanceof SensorAnalogico){
        SensorAnalogico SA = (SensorAnalogico)ob;
        SA.setPos1(Integer.parseInt(childNode.getTextContent()));
        System.out.println(tabs+"\t "+childNode.getTextContent());
    }
case "pos2":
    if(ob instanceof SensorAnalogico){
        SensorAnalogico SA = (SensorAnalogico)ob;
        SA.setPos2(Integer.parseInt(childNode.getTextContent()));
        System.out.println(tabs+"\t "+childNode.getTextContent());
    }
case "sensoranalogico":
    Objeto SA = new SensorAnalogico();
    sensoresA.add((SensorAnalogico)SA);
    SA = procesaObjeto(childNode, SA,tabs+"\t");
    System.out.println(tabs+"\t ");
    if(ob instanceof Cilindro){
        Cilindro c = (Cilindro)ob;
        c.addSensorAnalogico((SensorAnalogico)SA);
    }else if(ob instanceof Rodamiento){
        Rodamiento r = (Rodamiento)ob;
        //r.addSensorAnalogico((SensorAnalogico)SA);
    }else{
        ob = SA;
    }
break;

```

Figura 19. Declaración atributos de la clase "Escena"

También es necesario la creación de una clase llamada "ProcesarOutputAnalog", esta clase es la encargada de gestionar todos los nuevos valores referentes al Data block que se han incluido en el "input" para generar un pin.

Clase PlcSiemens

Esta clase también ha sido modificada para poder implementar el sensor analógico. Se ha añadido una subclase llamada "writeAnalogInput". Esta clase comprueba que hay conexión con el PLC y en caso afirmativo filtra los pines y en caso de que uno de los pines se analógico obtiene sus datos. Una vez hecho esto se crea un buffer llamado "data" y mediante las funciones proporcionadas

por la librería Mokka 7 se transmite los valores del pin al Data block como se muestra en la Figura 20.

```
public void writeAnalogInput() {
    if (conexion) {
        for(PlcInput pin:EntradasAnalogicas){
            adress = pin.getAdressDB();
            Length = pin.getBytesDB();
            DB = pin.getDB();
            StarAddress = pin.getStartAddress();
            Valor= pin.getValor();
        }
        byte[] data = new byte[Length];
        // Escribir el array de bytes en el PLC
        //System.out.println("fallo");
        S7.SetDIntAt(data, adress, Valor);
        Client.WriteArea(S7.S7AreaDB, DB, StarAddress, data.length, data);
    }
}
```

Figura 20. Función writeAnalogInput()

Durante la función Display() que se ejecuta de manera cíclica se llama a esta función para que este actualizando constantemente el valor de la entrada digital. Además, se ha añadido una función dentro de Display() que muestra la dirección del Data block y el valor que se está enviando en la entrada analógica (ver Figura 21).

```
for(PlcInput pin: EntradasAnalogicas){
    root.fill(0);
    root.pushMatrix();
    root.translate(x+15,y+25);
    root.rotateZ(PApplet.radians(90));
    root.rotateX(PApplet.radians(180));
    root.text("DB"+pin.getDB()+"."+pin.getAdressDB(), 0, 0);
    root.translate(-20,-12);
    root.rotateZ(PApplet.radians(90));
    root.fill(255,0,40);
    root.text(pin.getValor(), 0, 0);
    root.popMatrix();
    x+=size;
```

Figura 21. Representación del valor del Data block.

Clase SensorAnalogico

La clase “SensorAnalogico” es una clase que descende de la clase “Objeto”. Dentro de esta clase se declaran las siguientes variables (véase Figura 22).

```
private PlcInput entrada;  
private float pos1;  
private float pos2;  
int randomValue = Rodamiento.getRandomNumber();  
float valor;  
float valor_desc;  
float valor_desnorm;
```

Figura 22. Variables declaradas en SensorAnalogico.

Dentro de esta clase hay que definir las subclases necesarias get/set para los atributos “entrada”, “pos1” y “pos2”. La principal subclase de la clase “SensorAnalogico” es “updateEstado”. En esta subclase se tiene como input la “pos” del sensor analógico (integrado dentro del Cilindro 2 por lo que se tiene un rango de valores para el eje de desplazamiento entre 275 mm y 0 mm). Con este input se chequea el valor “randomValue” que consiste en un valor aleatorio generado dentro de la clase “Rodamiento” y que nos definirá que tipo de rodamiento tenemos en la estación de manera aleatoria. En caso de que la posición del sensor sea mayor que “pos1” o “pos2” y dependiendo del tipo de rodamiento que haya salido actualizaremos el valor del sensor en tiempo real (véase Figura 23).

```

public void updateEstado(float pos){
    if(randomValue == 1 && pos >= pos2){
        valor =(int) pos - pos2;
        valor_desc =valor/100; //Desescalamos el valor
        valor_desnorm = (int) ((27468)/(1))*(valor_desc); //Desnormalizamos el valor
    }else if(randomValue == 0 && pos >= pos1){
        valor =pos - pos1;
        valor_desc = valor/100; //Desescalamos el valor
        valor_desnorm = ((27468)/(1))*(valor_desc); //Desnormalizamos el valor
    }else {
        valor = 0;
    }
    entrada.setValor((int)valor_desnorm); //Enviamos el mismo valor que enviaria
    //el sensor analogico con la misma resolucion.
}

```

Figura 23. Subclase updateEstado()

Todos los valores de posición antes mencionados están dados en milímetros, pero esta no es la información que se debe de transmitir al PLC, por lo que se ha implementado dos operaciones, una de desescalada y otra de desnormalizado para que los valores enviados al PLC simulen los de un sensor real. Para esto es necesario la información indicada al principio del capítulo al respecto de la resolución del convertidor ADC y el rango de medición del sensor.

La función es llamada dentro de la subclase Display() dentro de las Clases “Rodamiento” y la clase “Cilindro”.

Clase Rodamiento

La clase “Rodamiento” descende de la clase Actuator y es muy similar a la clase “Cilindro”. Esta clase “Rodamiento” recibe dos modelos 3d como input, uno denominado “RodamientoKO” y otro llamado “RodamientoOK”. Para seleccionar uno de los dos modelos a representar se ha tenido que añadir un método de elección aleatoria mediante el cual se obtiene un valor booleano de manera aleatoria y según esta variable se elige si se representa un modelo u otro. En la figura 24 se puede ver la implementación de este método de selección aleatorio.

```

static {
    generateRandomNumber();
}
private static void generateRandomNumber() {
    Random random = new Random();
    randomNumber = random.nextInt(2); // Genera un número aleatorio entre 0 (incluido) y 2 (excluido)
}

public static int getRandomNumber() {
    return randomNumber;
}

```

Figura 24. Generación de número aleatorio.

Tras esto en la clase Display() se comprueba si el modelo3d es de tipo “rodamiento1” o “rodamiento2” y dependiendo del valor aleatorio obtenido se procede a representar uno u otro (véase Figura 25). Este valor aleatorio también se usará a la hora de actualizar el estado del pin del sensor analógico.

```

public void Display() {
    //Update();
    if(modelo3d.getMalla() != null){
        modelo3d.root.shape(modelo3d.getMalla(), 0, 0);
        //TODO Dibuja boundign box, sale desplazado el centro de box no corresponde con el punto de dibujo
        //modelo3d.root.box(modelo3d.malla.getWidth(), modelo3d.malla.getHeight(), modelo3d.malla.getDepth());
    }
    for (Model3d m: rodamiento1){
        if(NúmeroAleato == 0){
            modelo3d.root.pushMatrix();
            modelo3d.root.translate(m.getPos().getX(), m.getPos().getY(), m.getPos().getZ());
            modelo3d.root.rotateX(m.getPos().getRx());
            modelo3d.root.rotateY(m.getPos().getRy());
            modelo3d.root.rotateZ(m.getPos().getRz());
            modelo3d.drawAxis();
            if(m.getMalla() != null) modelo3d.root.shape(m.getMalla(), 0, 0);
            modelo3d.root.popMatrix();
        }
    }
    for (Model3d m: rodamiento2){
        if(NúmeroAleato == 1){
            modelo3d.root.pushMatrix();
            modelo3d.root.translate(m.getPos().getX(), m.getPos().getY(), m.getPos().getZ());
            modelo3d.root.rotateX(m.getPos().getRx());
            modelo3d.root.rotateY(m.getPos().getRy());
            modelo3d.root.rotateZ(m.getPos().getRz());
            modelo3d.drawAxis();
            if(m.getMalla() != null) modelo3d.root.shape(m.getMalla(), 0, 0);
            modelo3d.root.popMatrix();
        }
    }
}

```

Figura 25. Selección representación modelo3d.

4 Diseño y Desarrollo del Modelo de la Estación FMS-202

En este capítulo, se proporciona una visión detallada del proceso llevado a cabo para concretar el modelo digital de la estación FMS-202. Inicia con una breve introducción a la célula flexible de fabricación FMS-200, para posteriormente focalizarse en la estación FMS-202. Se ofrecen detalles exhaustivos los modelos 3D específicos creados para construir su modelo digital. Como cierre, se establecerá el modelo de información implementado en la representación digital de la estación.

Cabe destacar que el desarrollo de este proyecto es la creación de un modelo digital, siendo este distinto del concepto de gemelo digital. El propósito de un modelo digital es proporcionar una representación virtual que pueda ser utilizada para entender, analizar, simular o visualizar aspectos específicos del objeto o sistema en cuestión. En cambio, un gemelo digital es la representación virtual de un objeto al que se le incorporan datos en tiempo real que pueden captarse mediante sensores o tecnologías de análisis de datos (Repsol, s.f).

4.1 Descripción general de la estación

La célula de fabricación flexible FMS-200 de Siemens es un sistema innovador diseñado para automatizar y mejorar la eficiencia en la producción industrial. Esta configuración incluye varias estaciones de trabajo autónomas interconectadas mediante un sistema de transporte, lo que representa un enfoque avanzado en la optimización de procesos manufactureros.

La célula de fabricación flexible FMS-200 está compuesta por varias estaciones individuales. Cada una de estas estaciones realiza un proceso propio y distinto del resto de estaciones. Cada estación cuenta con elementos presentes en la industria de manera que permite trabajar con elementos reales que te darán un conocimiento.

Las estaciones que conforman la célula de fabricación flexible FMS-200 son las siguientes (SMCTraining, s.f.):

- **FMS-201: Alimentación de la base**

Esta estación realiza el proceso de alimentación de la base, que sirve como soporte al producto ensamblado.

- **FMS-202: Inserción del rodamiento**

Esta estación es la encargada de insertar un rodamiento en el alojamiento de la base. El rodamiento podrá ser de diferentes alturas.

- **FMS-203: Prensado hidráulico del rodamiento**

Esta estación realiza el prensado hidráulico del rodamiento insertado en la estación anterior.

- **FMS-204: Inserción del eje**

En esta estación se monta un eje sobre el producto en proceso. Existen dos tipos de ejes, dependiendo del material: aluminio y nylon.

- **FMS-205: Inserción de la tapa**

Esta estación se encarga de insertar una tapa sobre el conjunto. Existen seis tipos de tapa dependiendo del material, color y altura.

- **FMS-206: Inserción de tornillos**

Esta estación inserta cuatro tornillos en la base del producto en proceso.

- **FMS-207: Ensamblaje / desensamblare y atornillado robotizado**

Esta estación supone la integración de la robótica como tecnología. El robot atornillará los cuatro tornillos suministrados por la estación anterior. Además, se podrán realizar operaciones de cambio de ensamblaje de ejes y de tapas.

- **FMS-208: Almacén automático**

Esta estación realiza el almacenamiento de los productos terminados.

- **FMS-209: Secado de pintura en horno**

En esta fase del proceso, se simula el secado de pintura del conjunto mediante un horno de policarbonato.

- **FMS-210: Control de calidad por visión artificial**

Esta estación supone la integración de la tecnología de visión artificial en la familia FMS-200. El producto en proceso es trasladado en esta estación a la posición de inspección, en la que una cámara de visión artificial examina las variables definidas en el control de calidad.

El objetivo de la estación FMS-202, representada en la Figura 26 (FMS, s.f.), es la selección entre dos tipos de rodamiento con diferentes alturas mediante la medición de un sensor analógico. En caso de que el rodamiento sea correcto se depositara en la pieza presente en la cinta transportadora y en caso de que no sea el rodamiento correcto será descartado.

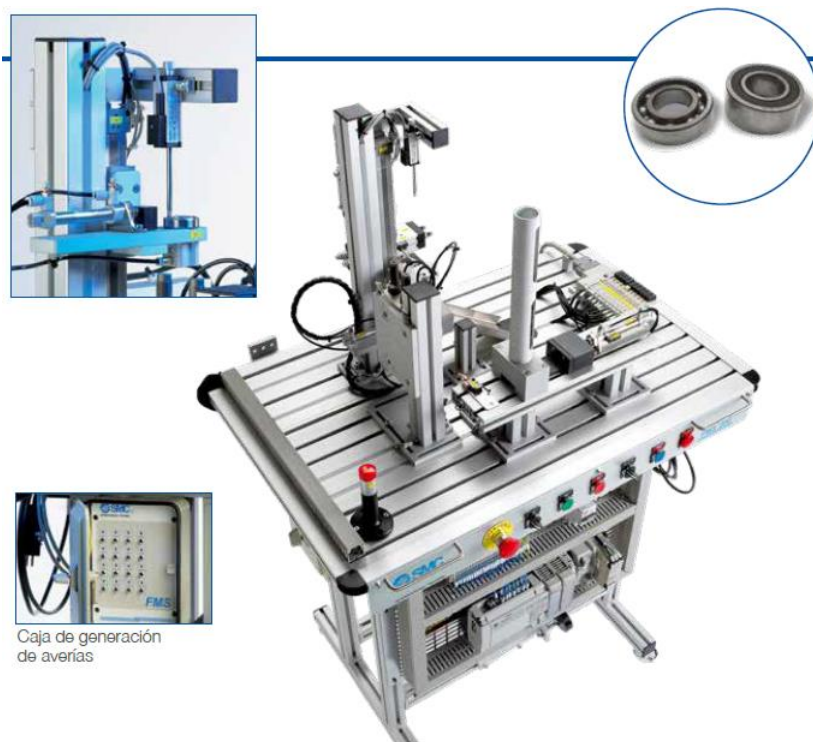


Figura 26. Imagen de la estación FMS-202.

La estación trabaja de la siguiente manera, primero se extrae uno de los rodamientos apilados en la primera parte de la estación mediante la acción del primer cilindro. Al detectarse la presencia de un rodamiento por el sensor, el brazo coge el cilindro y lo transporta hacia el ascensor. Una vez situado en rodamiento en el ascensor es activado el centrador para posicionar correctamente el rodamiento. Tras esto el ascensor subirá para proceder a la medición del rodamiento. Una vez medido descenderá y en caso de no ser correcto será desechado mediante el cilindro situado en la plataforma del ascensor por la rampa. Si el rodamiento tiene la altura deseada, el ultimo brazo de la estación se aproximará descendiendo y sacando el rodamiento a la cinta de transporte.

4.2 Modelo 3D

A continuación, se procede a la descripción de los distintos modelos 3D que han sido desarrollados para la recreación de la estación FMS-202. Todos estos modelos han sido desarrollados en Autodesk Inventor y exportados a formatos .obj para poder trabajar con ellos mediante la herramienta Processing.

4.2.1 Base fija

De primeras se desarrolla un modelado tridimensional que incluye todos los elementos estacionarios en la estación. Se excluyen los componentes móviles como los cilindros y otros casos particulares como, las piezas y los botones, como se evidencia en la Figura 27.

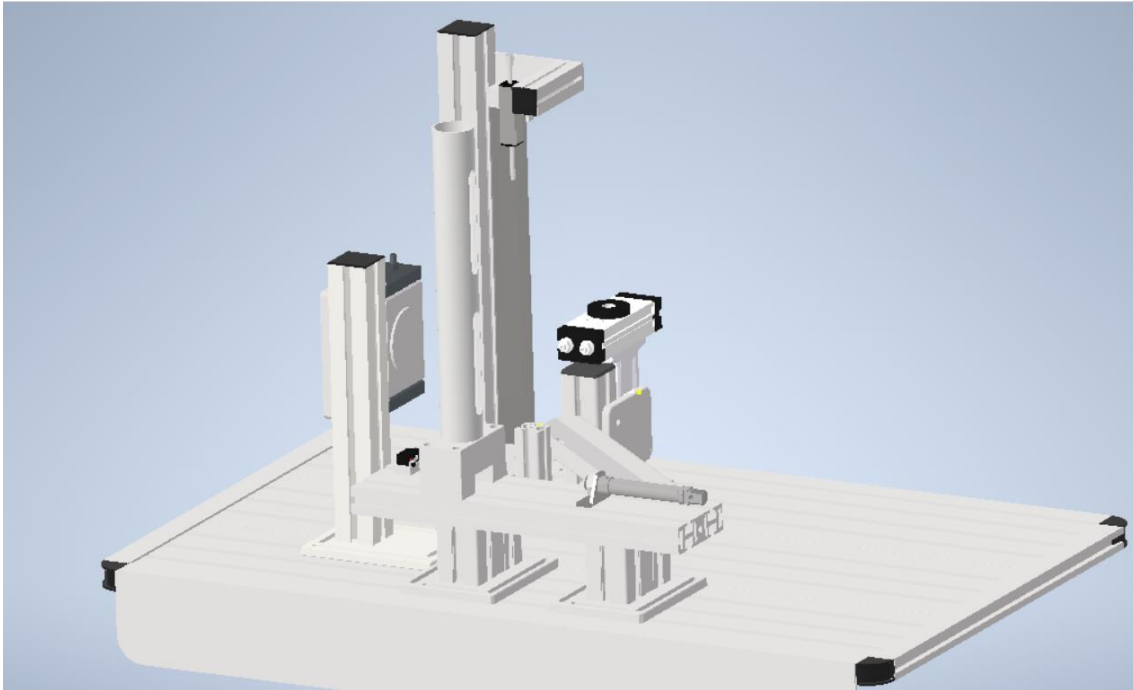


Figura 27. Base fija estación FMS-202.

La figura anterior muestra todas las partes no móviles de la estación. En primer lugar, se encuentran el almacenamiento de rodamientos, el cuerpo del cilindro de expulsión y el sensor de presencia de rodamiento.

La segunda parte cuenta con un soporte y el bloque del motor del brazo giratorio que lleva de los rodamientos de la primera parte al ascensor.

A continuación, se encuentra el soporte del ascensor y el sensor analógico.

Por último, se incluye el soporte y la base de un motor encargado de la extracción del rodamiento al exterior depositándolo en la cinta de transporte.

4.2.2 Alimentador rodamientos

Vástago cilindro A

En la Figura 28 podemos observar el vástago correspondiente al cilindro presente en la primera parte de la estación. Este vástago es la única parte móvil

de esta primera parte de la estación, encargada de la alimentación de rodamientos.

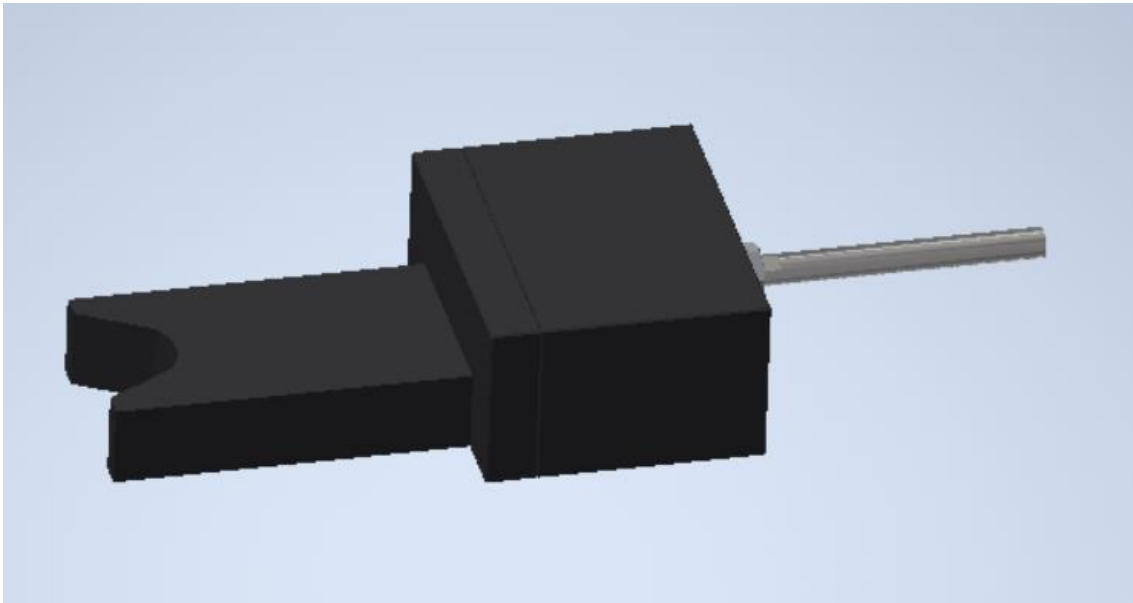


Figura 28. Vástago cilindro A.

A continuación, se muestra en la Figura 29 la representación completa de todo el conjunto que forma la primera parte de la estación.

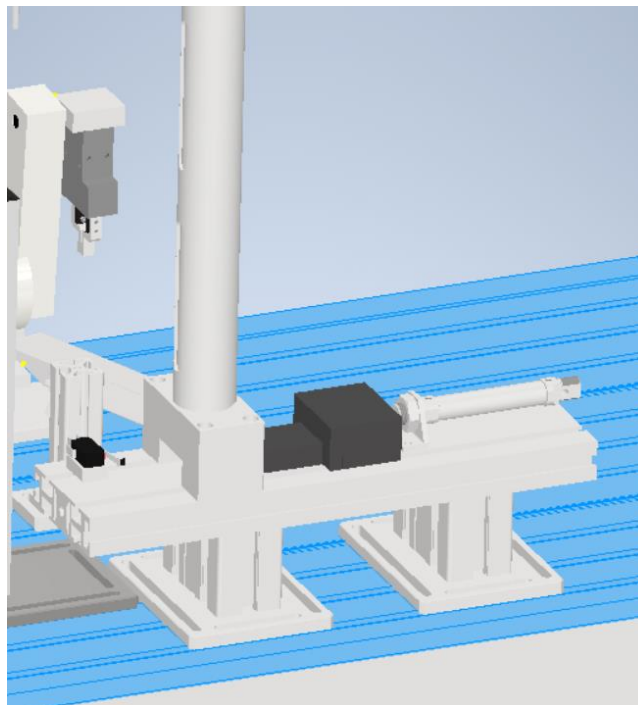


Figura 29. Alimentador de rodamientos.

4.2.3 Brazo Giratorio

Esta parte de la estación es la encargada de transportar los rodamientos desde la parte de la alimentación hasta el ascensor en que será medida la pieza. Se puede observar en la Figura 30 el soporte y el bloque del motor de esta parte de la estación incluido en la base fija.



Figura 30. Soporte brazo giratorio.

Brazo

Esta parte se encuentra unida al motor giratorio por un extremo y por el contrario sostiene el cuerpo de la pinza junto a las palas. En la Figura 31 se

puede observar el modelado de este brazo. Este brazo es rotado por el motor de la base para desplazar los rodamientos

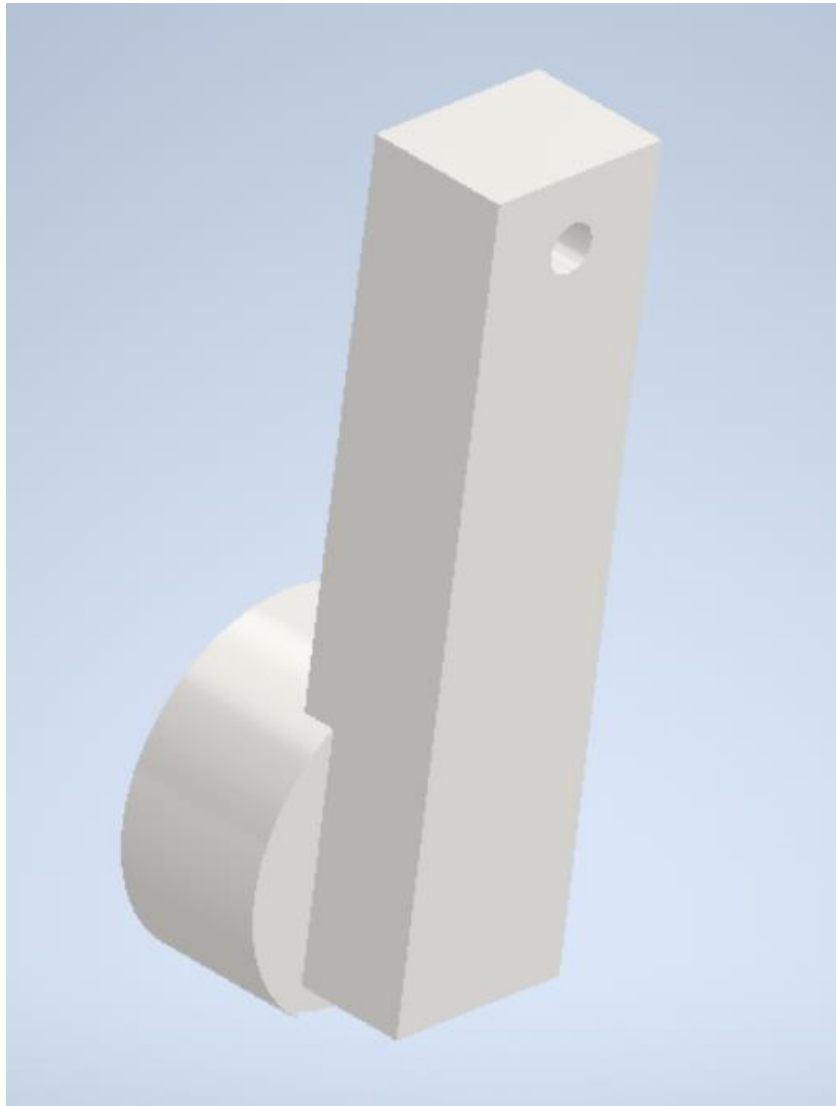


Figura 31. Brazo

Cuerpo de la pinza

Como se ha mencionado anteriormente anexionado al extremo del brazo se encuentra el cuerpo de la pinza. Este es el encargado de abrir o cerrar las palas para coger los rodamientos depositados. La figura 32 muestra el modelo del cuerpo de la pinza. Este siempre se debe de mantener horizontal respecto a la mesa, aunque el brazo este girando.

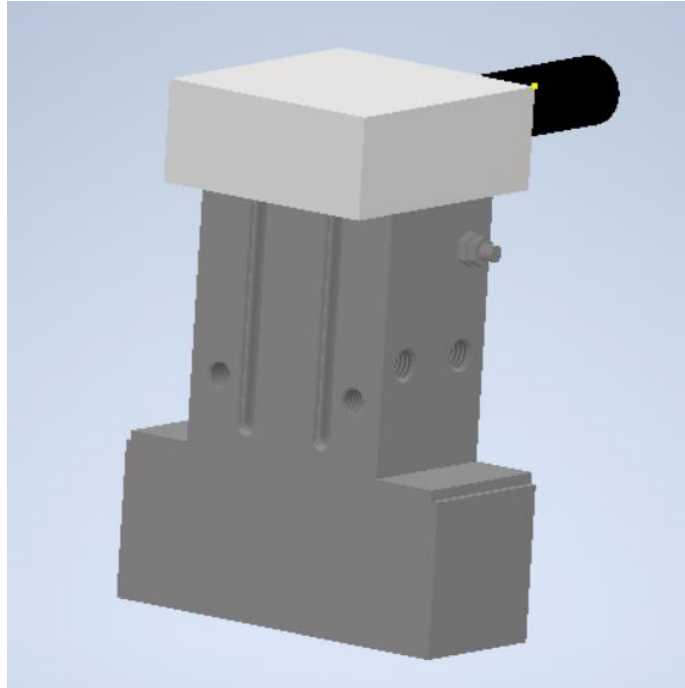


Figura 32. Cuerpo de la Pinza

Palas

Estas son las piezas finales de esta parte de la estación. Son las encargadas de sostener el rodamiento durante el transporte entre las distintas posiciones. Estas son abiertas por el cuerpo del sensor para sostener el rodamiento y cerradas cuando es necesario depositarlo. A continuación, la Figura 33.

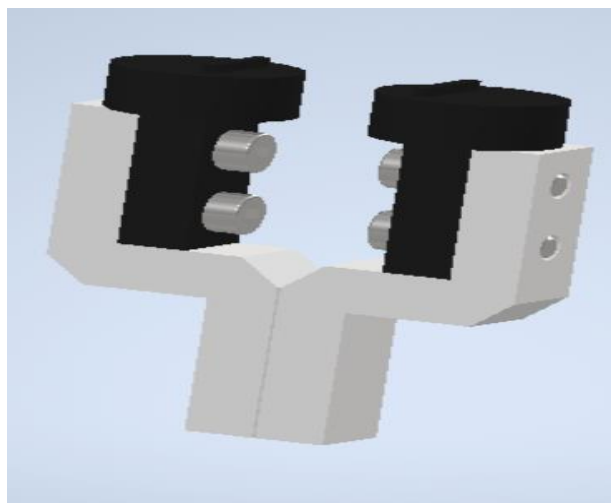


Figura 33. Palas brazo giratorio.

Por último, se mostrará el ensamblaje completo de la segunda parte de la estación FMS-202 representado en la Figura 34.

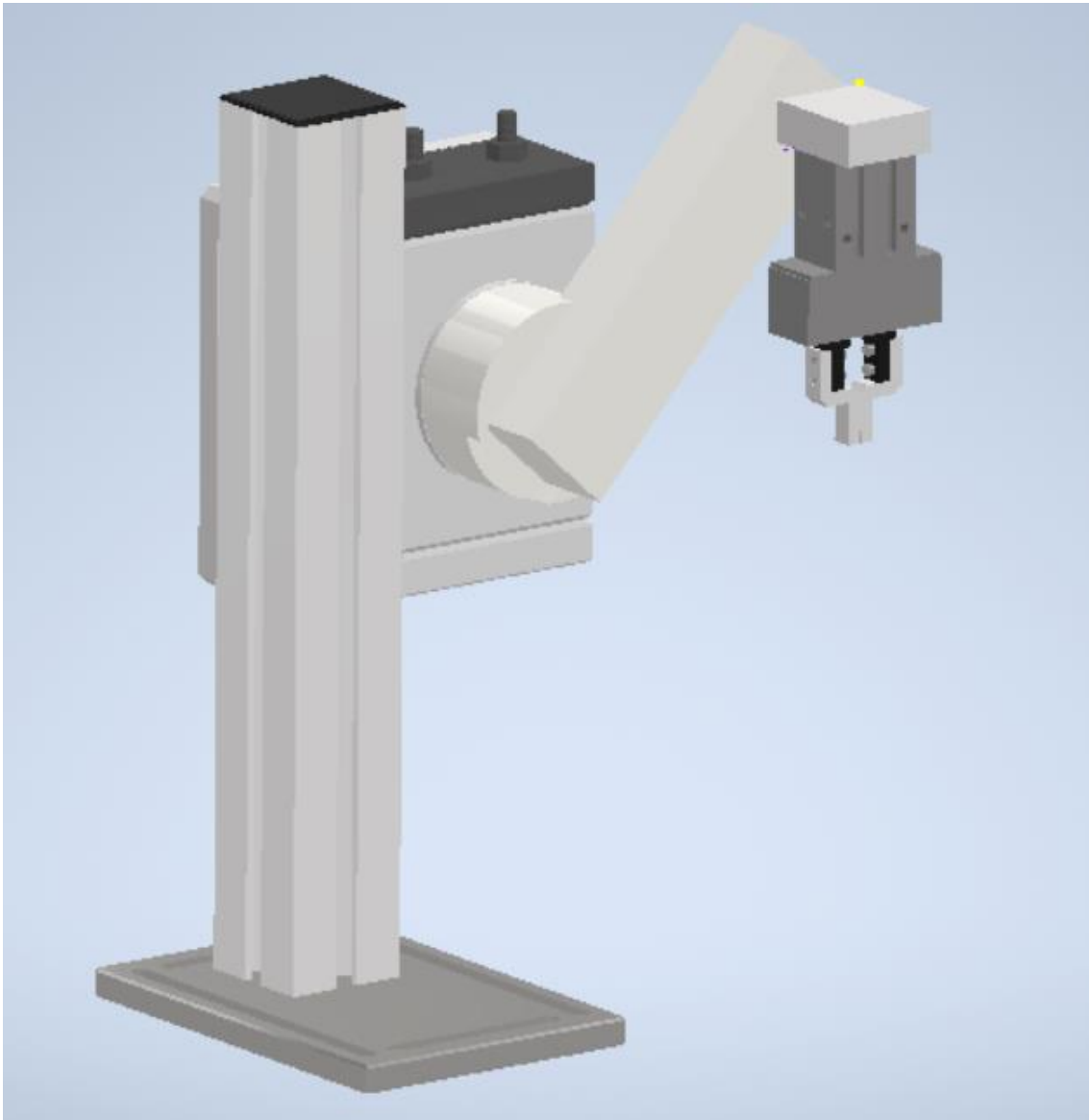


Figura 34. Ensamblaje segunda parte estación.

4.2.4 Ascensor

La siguiente parte dentro de la estación es la encargada de la medición de los distintos rodamientos. Esta parte consta del ascensor, el sensor analógico de medición de desplazamiento lineal, el centrador y el cilindro de expulsión. Dentro de la base fija se encuentra incluida toda la torre junto al bloque que hace como ascensor y el modelado del sensor analógico como se puede ver en la Figura 35.

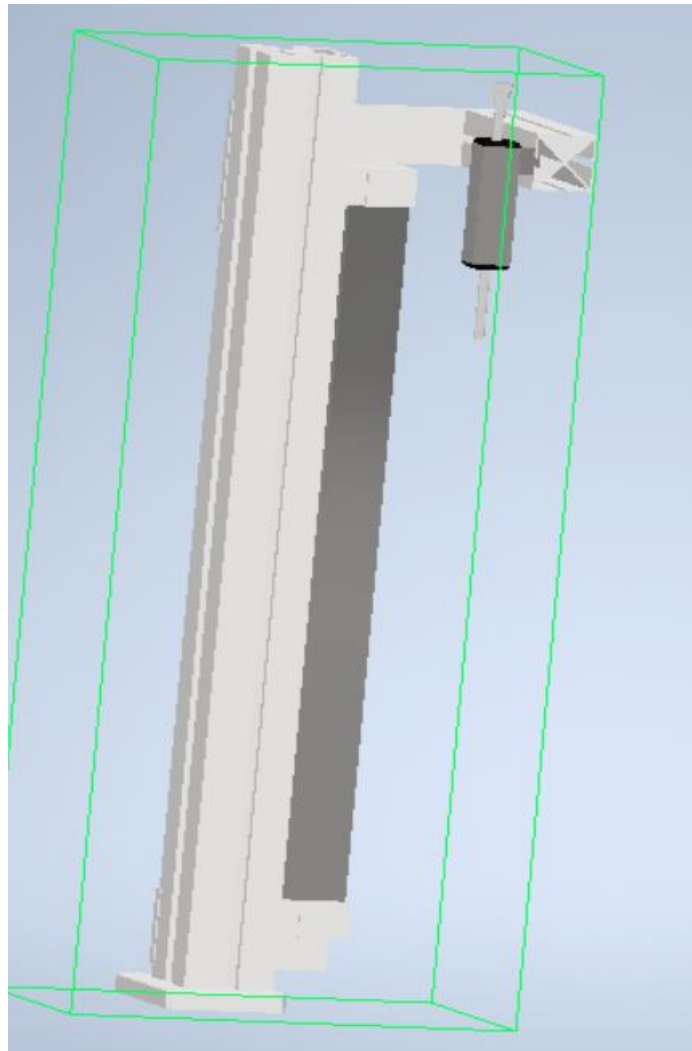


Figura 35. Parte del ascensor incluido en la base fija.

Soporte elevador D

Esta parte consiste en la base que es elevada o bajada junto al cuerpo de cilindro de expulsión. Esta base es en la que es depositada el rodamiento y es subido hasta arriba para su posterior medición. A continuación, se puede ver la representación de este modelo en la figura 36.

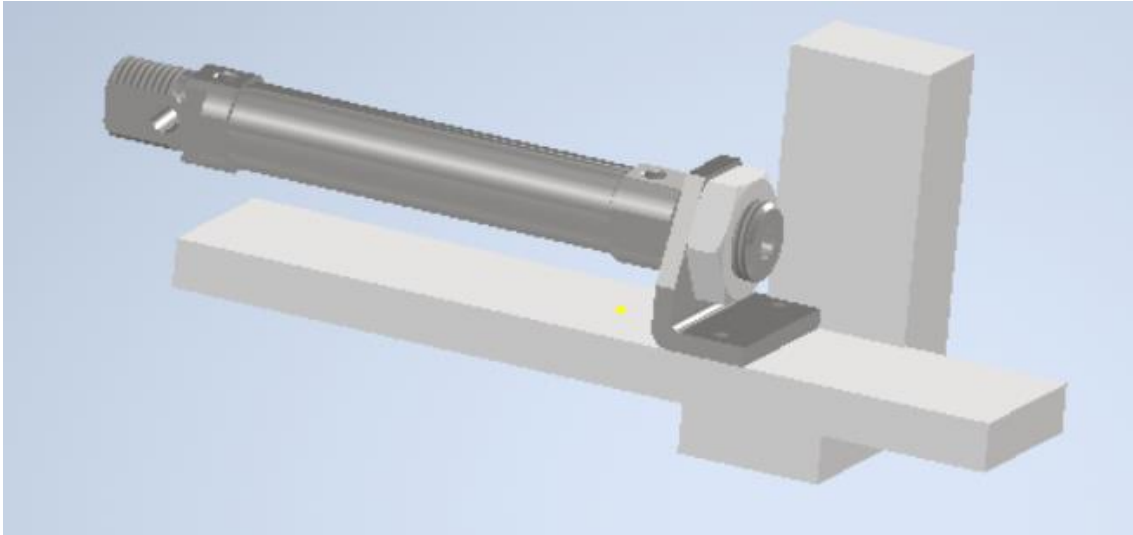


Figura 36. Base elevador y cuerpo del cilindro E.

Centrador

El centrador consiste un pequeño cilindro que mueve un vástago que actúa como centrador de posición para el rodamiento. Este pequeño cilindro se encuentra integrado dentro de la base del elevador y se mueve al unísono con él. En la Figura 37 se puede observar el centrador.

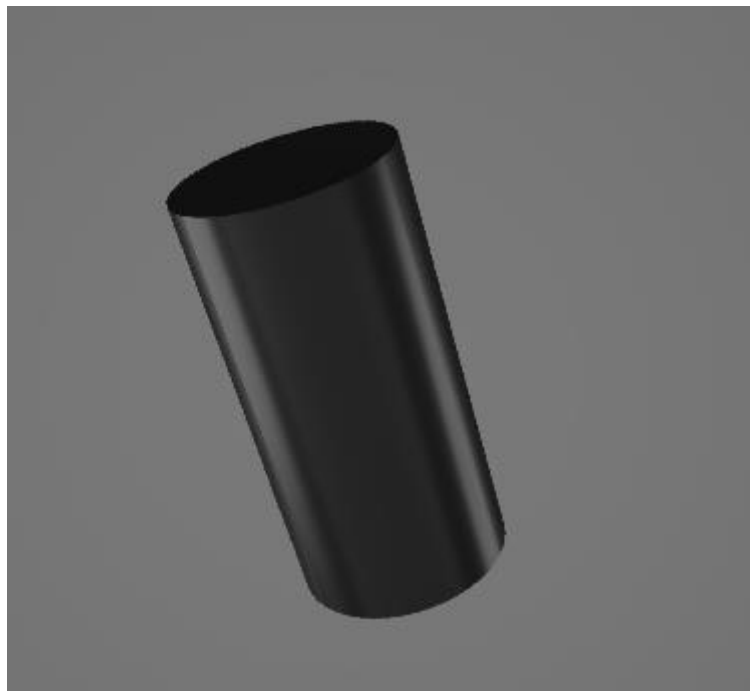


Figura 37. Centrador.

Vástago cilindro E

El vástago del cilindro E consiste en la parte móvil del cilindro E. El cilindro E se encuentra en la base del ascensor y tiene como función la expulsión de los rodamientos no aptos tras su extensión. La Figura 38 muestra el modelado de este vástago.

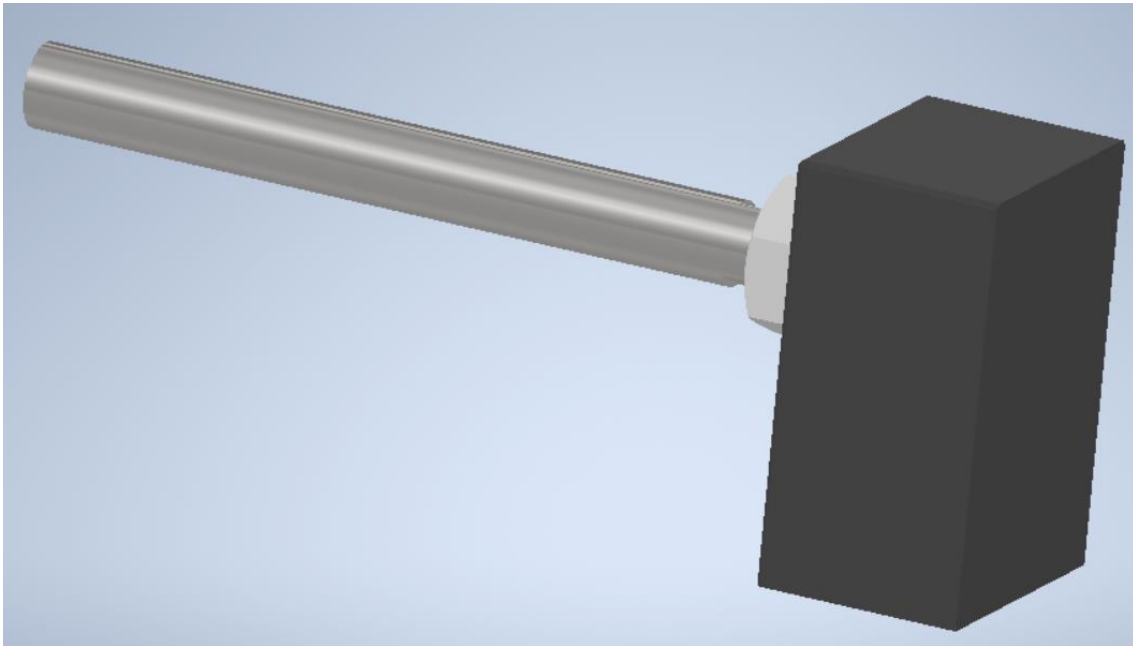


Figura 38. Vástago cilindro E.

Finalmente, con todos los elementos anteriormente descritos se puede observar finalmente como quedaría el conjunto del elevador completo (Figura 39).

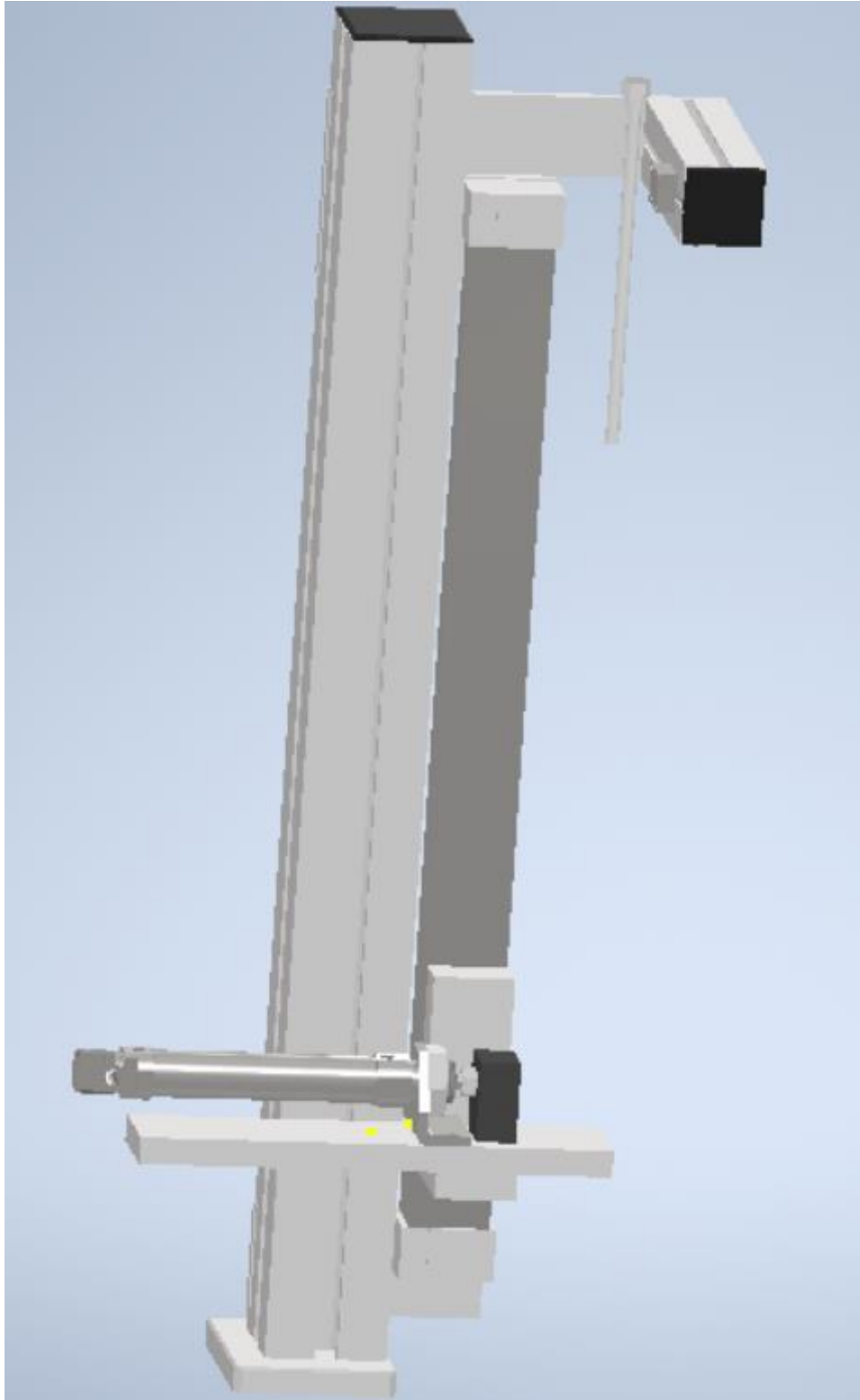


Figura 39. Conjunto completo del elevador más el sensor.

4.2.5 Brazo de salida

Finalmente, esta es la última parte de la estación FMS-202 consistente en un brazo giratorio y elevador capaz de bajar para agarrar el cilindro en la

plataforma del elevador y girar para depositarlo en la línea de transporte. Este cuenta con un soporte y la base del motor del brazo de salida encargado de su rotación y de subir o bajar dicho brazo integrados dentro del modelado de la base fija, como se puede observar en la Figura 40. Además, se tiene el modelado del brazo junto al cuerpo del sensor y el modelado de las palas de agarre que son partes móviles de la estación.

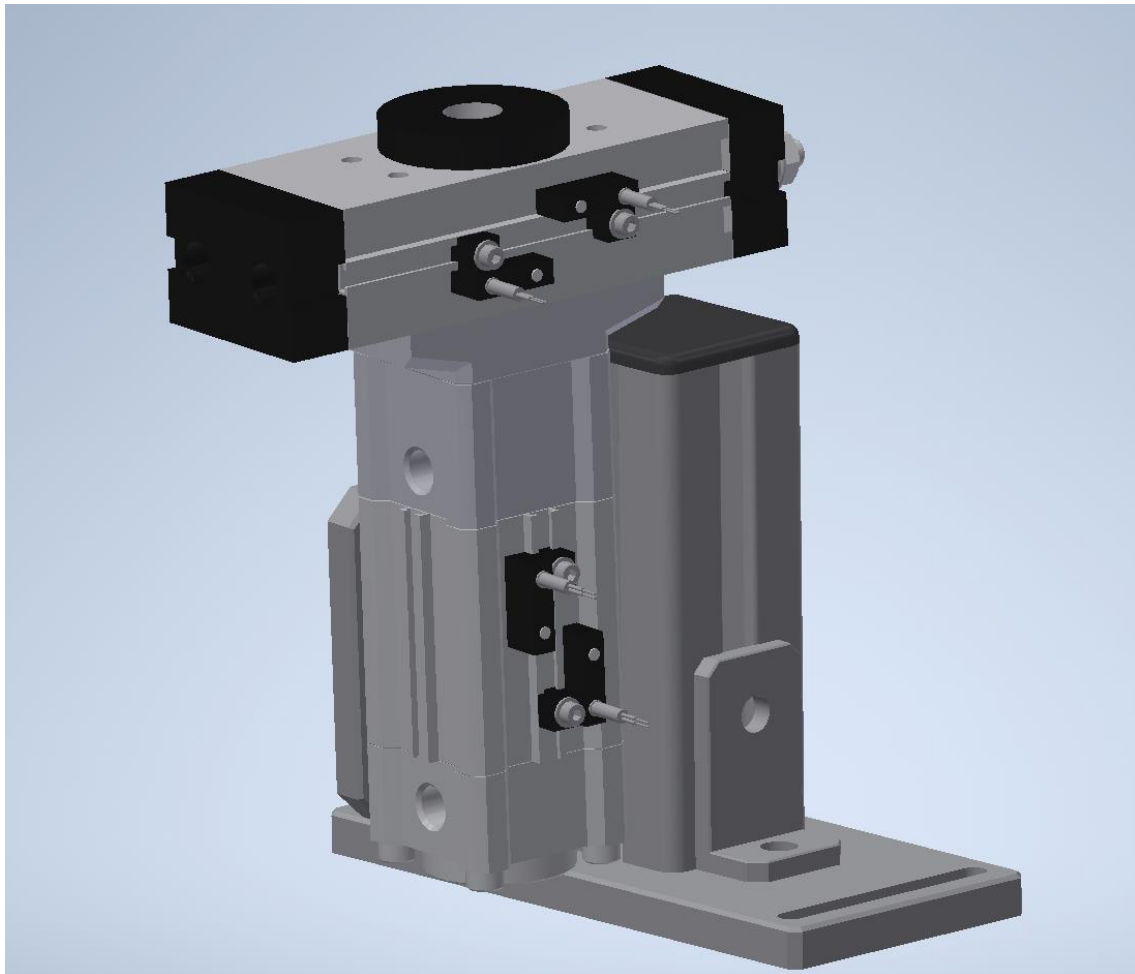


Figura 40. Base soporte y cuerpo cilindro.

Brazo giratorio H

A continuación, se puede observar el modelado 3D del brazo de salida. Este brazo cuenta con el movimiento de rotación y con el movimiento de ascenso y descenso para recoger y depositar la pieza. La Figura 41 muestra el modelado de este brazo.



Figura 41. Brazo giratorio final.

Palas

Por último, se tiene el modelado de las palas, estas son muy similares a las palas del segundo brazo giratorio. Son las encargadas de coger el cilindro y sostenerlo mientras es desplazado. Se puede observar el modelado de las palas en la Figura 42.

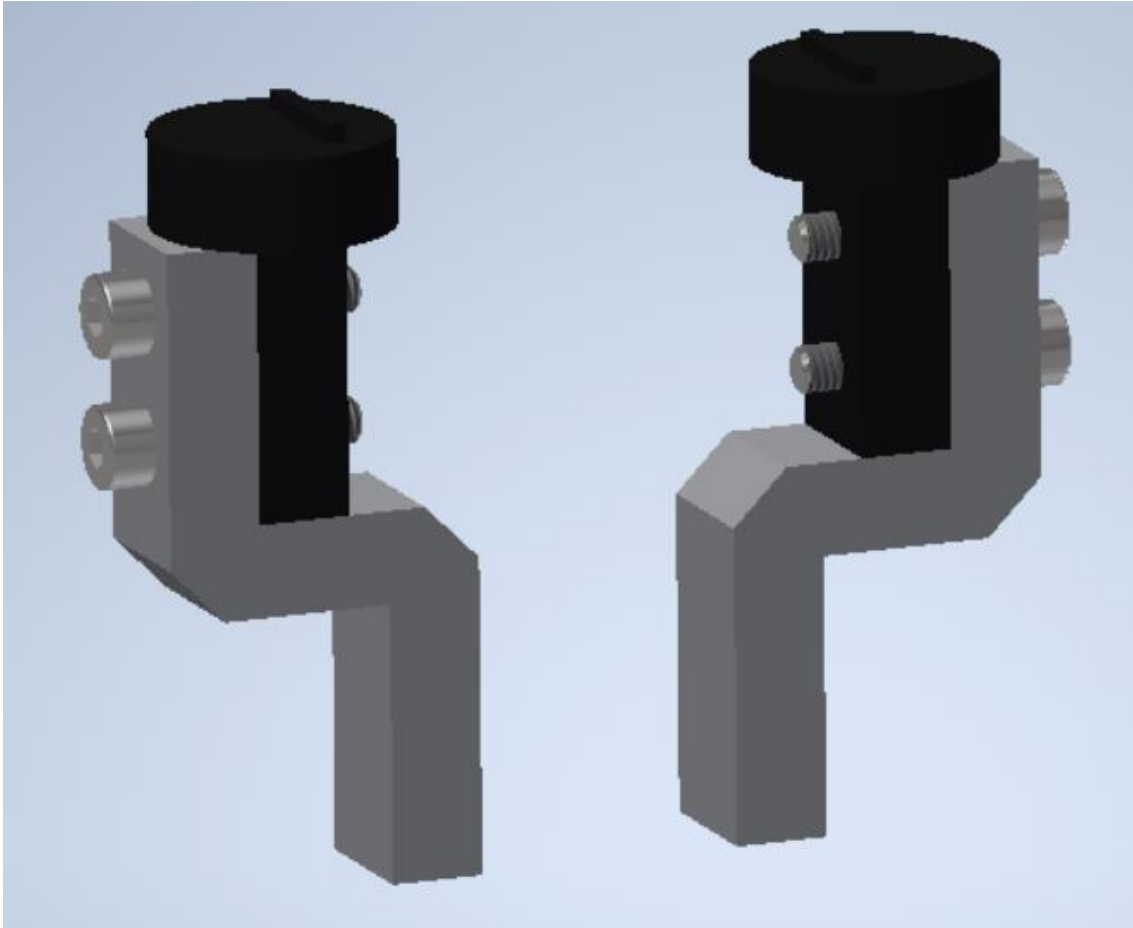


Figura 42. Palas brazo de salida.

Botones

A continuación, en las Figuras 43, 44, 45, 46, 47 y 48 se muestran los distintos botones e interruptores implementados en la estación FMS-202

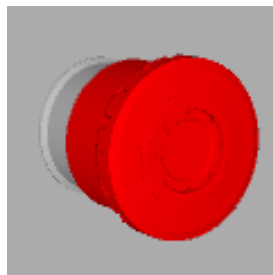


Figura 43. Seta de emergencia.



Figura 44. Activar sensor presencia rodamiento.



Figura 45. Botón de comienzo.

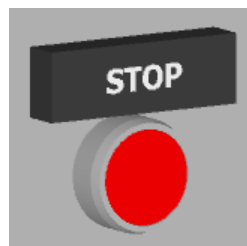


Figura 46. Botón de parada.



Figura 47. Selector de modos.



Figura 48. Botón de reseteo de la estación.

Rodamientos

Se ha tenido que modelar también 2 rodamientos de distinto tamaño. Para un mejor entendimiento del usuario del modelo digital se ha establecido un color de rodamiento verde para los rodamientos con una medida correcta y un rodamiento de color rojo para los rodamientos cuya medida no es la correcta. A modo de información para la futura programación usando el modelo digital, el rodamiento correcto tiene una altura de 15 mm y el incorrecto de 20 mm. En la Figura 49 se puede ver el modelo del rodamiento correcto y en la Figura 50 se puede observar el modelo del rodamiento incorrecto.

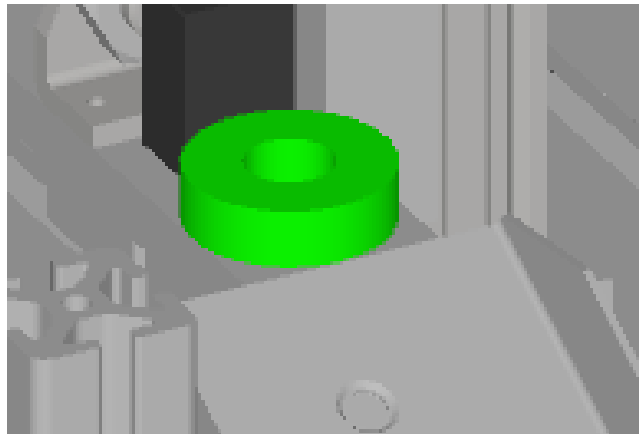


Figura 49. Rodamiento correcto.

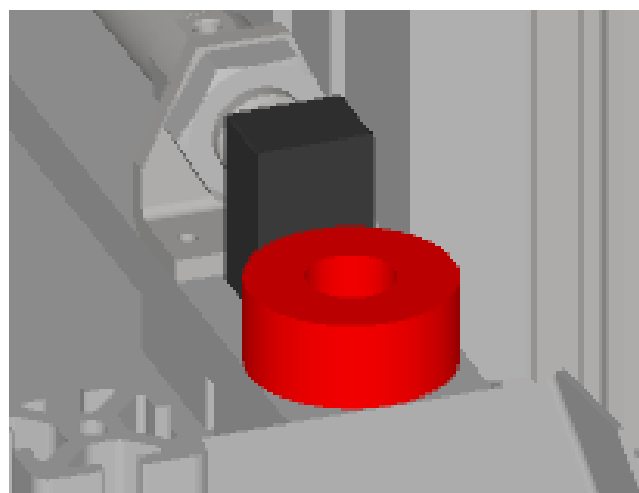


Figura 50. Rodamiento incorrecto.

Estación completa

Por último, se mostrará en las Figuras 51 y 52 el resultado final del ensamblaje de todos los modelos realizados para este proyecto implementados dentro del modelado digital de la estación hecho en Java.

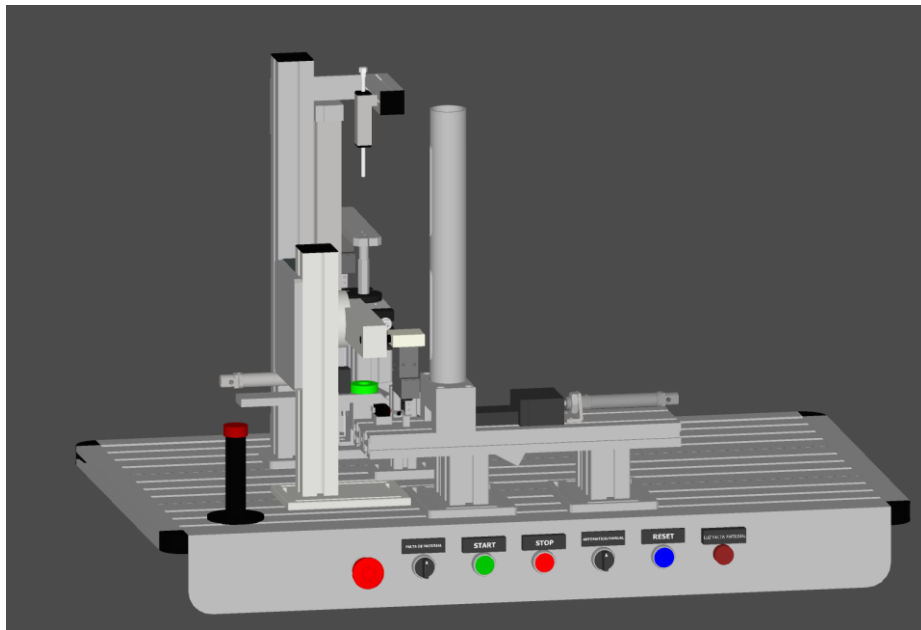


Figura 51. Vista del ensamblaje completo de la estación FMS-202.

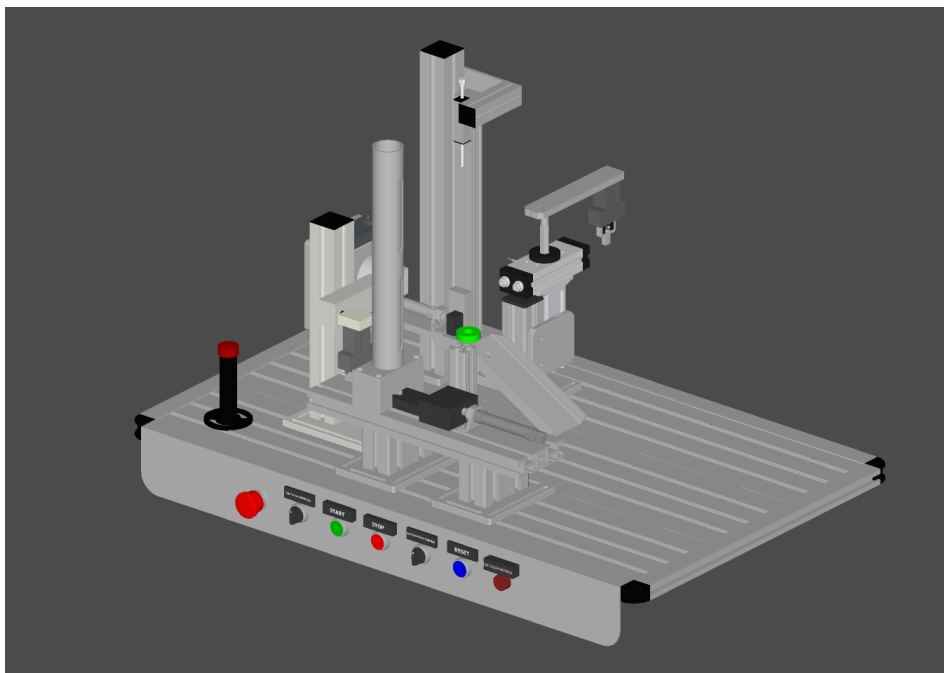


Figura 52. Vista adicional ensamblaje estación FMS-202.

4.3 **Modelo de información**

En este capítulo se procederá a explicar en detalle el fichero de información de tipo XML llamado “escena” en el cual se suministran todos los datos necesarios para el correcto funcionamiento de la estación. Se explicará el funcionamiento de cada cilindro y el sistema de anidamiento y el motivo de porque se ha implementado de esta manera cada cilindro. A tener en cuenta que cada cilindro cuenta con su propia válvula para la activación de estos en los que se indica la entrada que expande el cilindro o lo retrae en caso de los cilindros biestables. La Figura 50 muestra un ejemplo de anidamiento de cilindros dentro del modelo de información de la estación FMS-202.



Figura 53. Cilindro anidado.

Cilindro 1

Este cilindro es el encargado de la expulsión de los rodamientos en la primera etapa. Este cilindro cuenta con un sensor que indica el final de carrera. Podemos observar su fichero de información en la figura 54.

- cilindro : 1 - id - nombre - modelo3d - modelo3dMovil - fileName - pos - escala - tipo - posMax - posMin - vel - valvula - sensor : 1 - id - nombre - modelo3d - posMax - posMin - input	<pre> 1 CilindroSalidaRodamiento config/obj/Vastago_rodamiento.obj 1 1 50 0 1 1 sensorExtendido 50 49 </pre>
---	--

Figura 54. Datos cilindro 1.

Cilindro 2

El cilindro 2 es el que actúa como ascensor en torre encargada de la medición del rodamiento. Este cilindro contiene anidados dentro de él el cilindro 3 y el 9 cada uno encargado de la expulsión del rodamiento en caso de que fuera erróneo y de centrar el rodamiento en la plataforma respectivamente. Además, contiene dos sensores para indicar el principio y el final de la carrera y el atributo rodamiento descrito en el capítulo 3. En la Figura 55 se observa la estructura del cilindro 2.

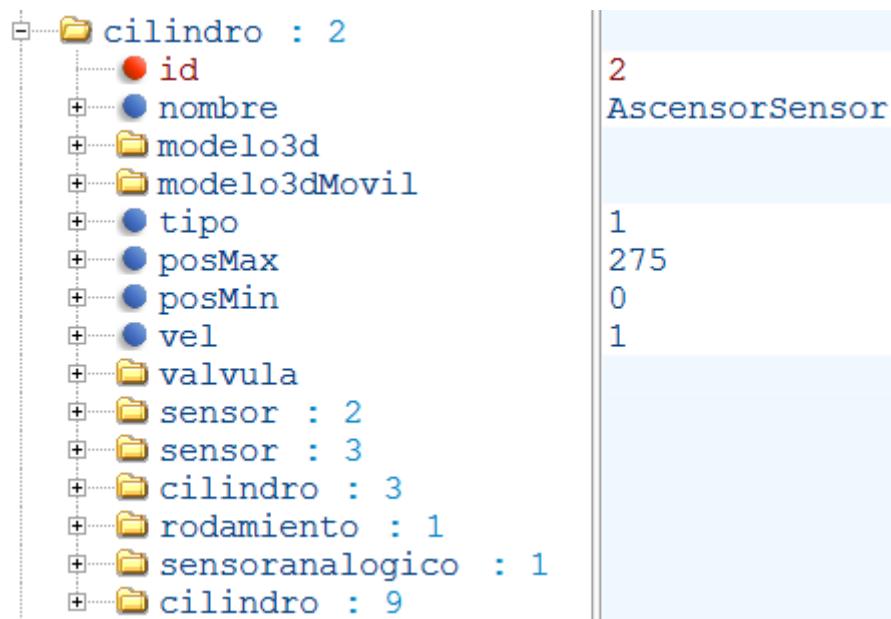


Figura 55. Estructura Cilindro 2.

Cilindro 3

El cilindro 3 tiene como función la expulsión de los cilindros erróneos tras la medición. Cuenta con su válvula para el manejo de este y no tiene ningún tipo de sensor.

Cilindro 9

El cilindro 9 tiene como centrar el rodamiento en la plataforma para que el palpador pueda medir correctamente el contorno del rodamiento. Cuenta con su válvula para el manejo de este y no tiene ningún tipo de sensor.

Rodamiento

El objeto rodamiento ha sido creado para el desarrollo de este proyecto. El rodamiento se encuentra anidado dentro del cilindro 2 para que se mueva sobre la plataforma. Como se ha explicado anteriormente en el capítulo 3 los principales atributos que se definen son los modelos de los dos rodamientos y la posición de estos como se puede ver en la Figura 56.

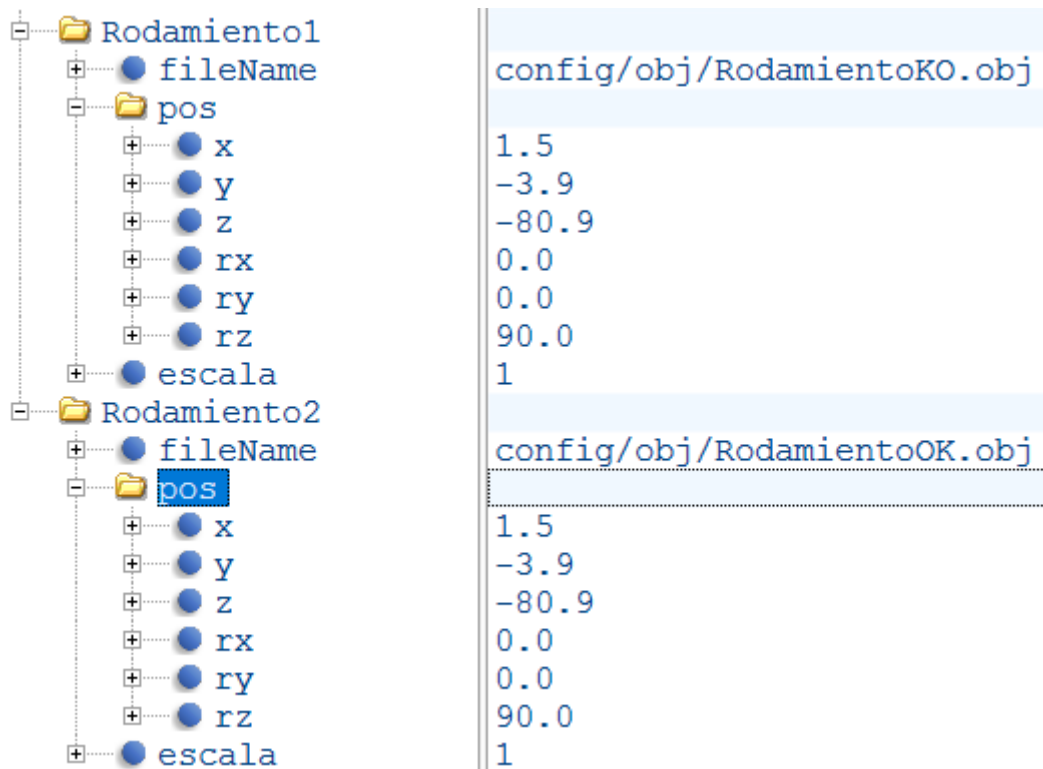


Figura 56. Estructura atributo rodamiento.

Cilindro 4

El cilindro 4 es el encargado de la rotación del brazo giratorio encargado de transportar el rodamiento desde la primera parte de la estación en la que se extrae un rodamiento hasta el ascensor en el que es medido. Este cilindro posee 3 sensores que determinan las posiciones de los extremos y la posición intermedia del cilindro. Además, anidado dentro de él se encuentra el cilindro 5 encargado del movimiento de las palas de la pinza.

Cuando el brazo gira sobre su eje en cualquiera de las posibles posiciones en las que el brazo se encuentre el cuerpo de la pinza siempre se encuentra en horizontal al respecto de la mesa. Para esto ha sido necesario la implementación de dos “modelo3dMovil” uno correspondiente al brazo y otro a la pinza. Tras esto es necesario invertir la dirección del eje en el cuerpo de la pinza respecto al eje del movimiento el brazo para así poder generar el efecto de que el cuerpo de la pinza se mantiene horizontal. Esto puede ser observado en la Figura 57.

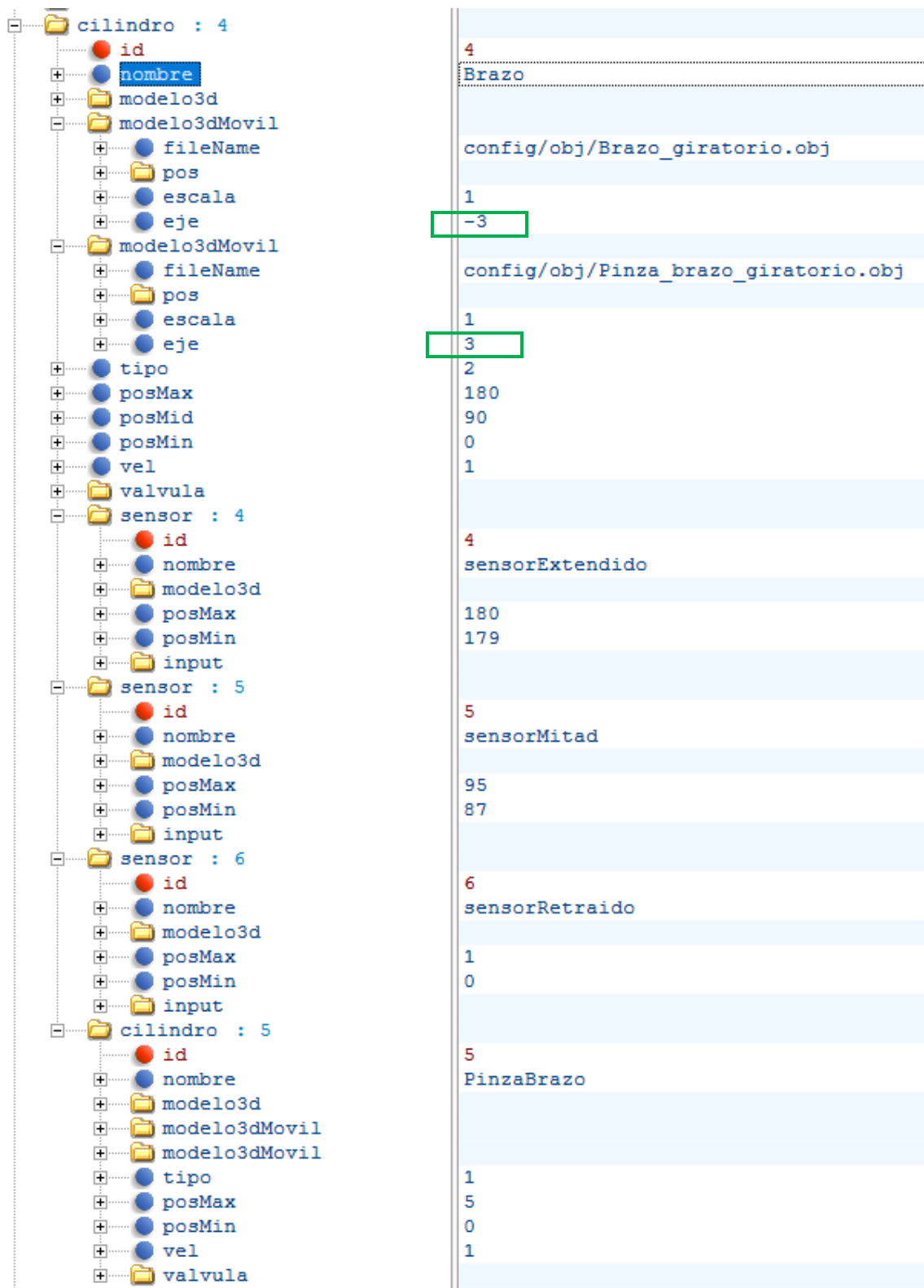


Figura 57. Estructura del cilindro 4.

Cilindro 5

El cilindro 5 es el responsable del movimiento de las palas las cuales son las responsables de coger el rodamiento. Para realizar el movimiento de las palas se ha optado por realizar la misma practica que en el cilindro 4 añadiendo dos “modelo3dMovil” e invirtiendo el sentido de un eje para dar la sensación de apertura de la pinza cuando se activa la válvula. Se puede observar esto en la Figura 58.



Figura 58. Estructura del cilindro 5.

Cilindro 6

El cilindro 6 se encuentra dentro del último subconjunto de la estación, o como se ha mencionado anteriormente “Brazo de salida”. El cilindro 6 tiene como función subir y bajar el brazo giratorio. Además, cuenta con 2 sensores que indican si el brazo se encuentra en la posición alta o baja. Dentro de este cilindro se encuentra el anidado el cilindro numero 7 encargado del giro del brazo. Se puede ver la estructura del cilindro 6 y de todos los cilindros anidados dentro en la Figura 59.

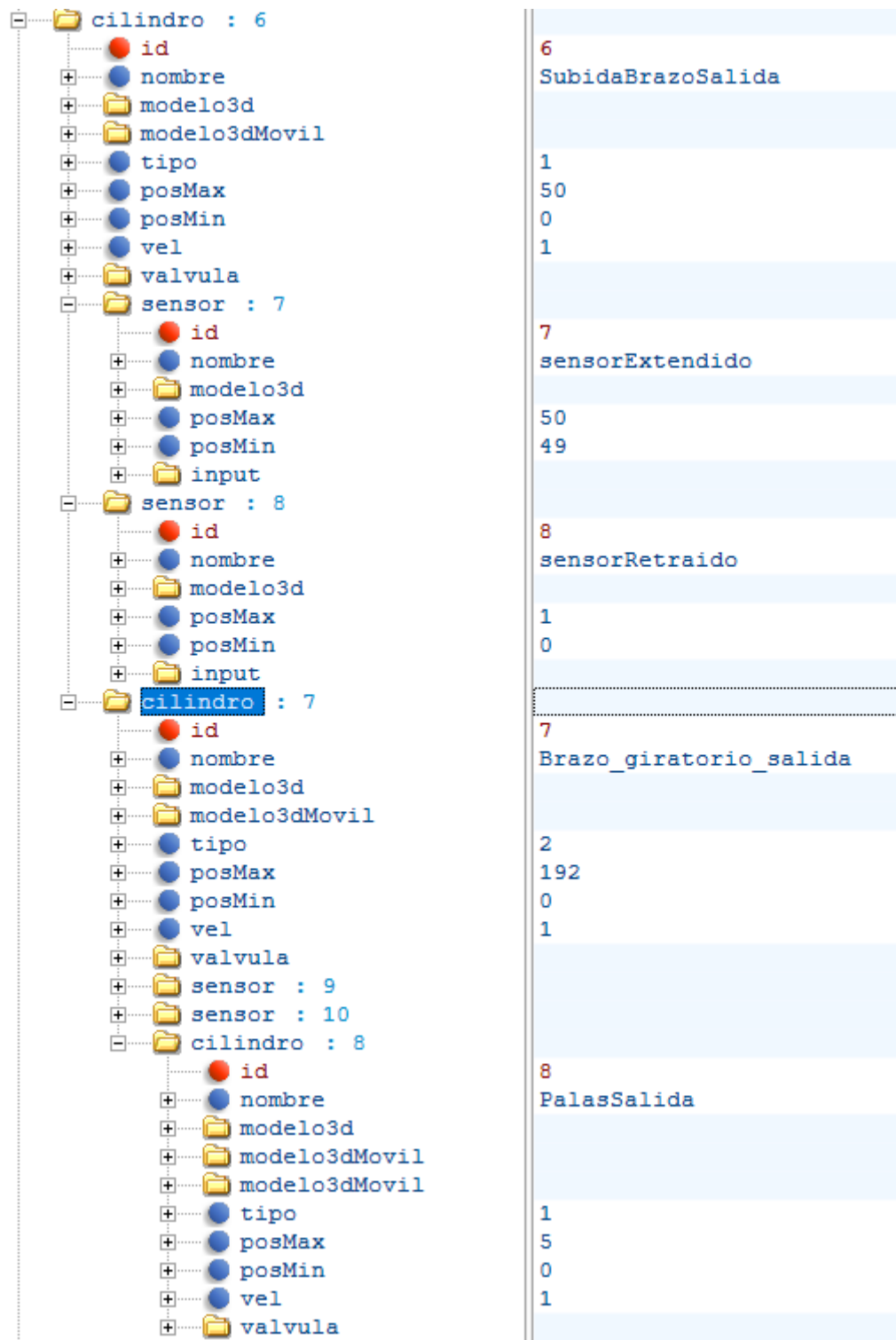


Figura 59. Estructura cilindros 6, 7 y 8.

Cilindro 7

El cilindro 7 es el encargado de la rotación del brazo giratorio de salida. Este se encuentra anidado dentro del cilindro 6 y cuenta con dos sensores para indicar si se encuentra presente en cada uno de los extremos de su rango de movimiento.

Cilindro 8

El cilindro 8 es el encargado de gestionar la apertura de las palas. Se encuentra anidado dentro del cilindro 7 y tiene una estructura similar a la descrita en la Figura 58.

Pulsadores

Todos los pulsadores tienen una estructura similar a descrita en la Figura 59. Esta es consistente en el nombre, el modelo 3D del pulsador, la información relacionada con la entrada y el tipo de pulsador. Es importante destacar el significado del atributo tipo de pulsador. Este establece que el pulsador actúe a modo de botón si está puesto a 0 o que actúe como si fuera un interruptor si vale 1.

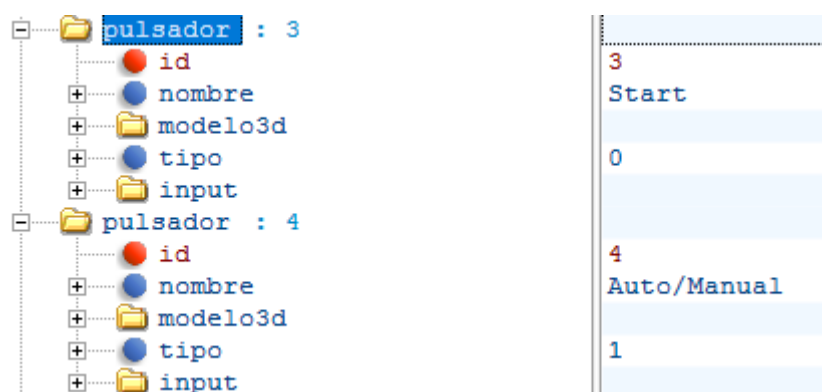


Figura 60. Estructura de los pulsadores.

Luces

Se han añadido dos luces a la estación, una para indicar fallo en la estación y otra para indicar la falta de material. Para definir una luz es necesario indicar los modelos y posición de las luces cuando se encuentran en estado encendido y apagado además de la salida a la que se encuentran asociadas. En la Figura 61 se muestra un ejemplo de cómo se debe implementar.

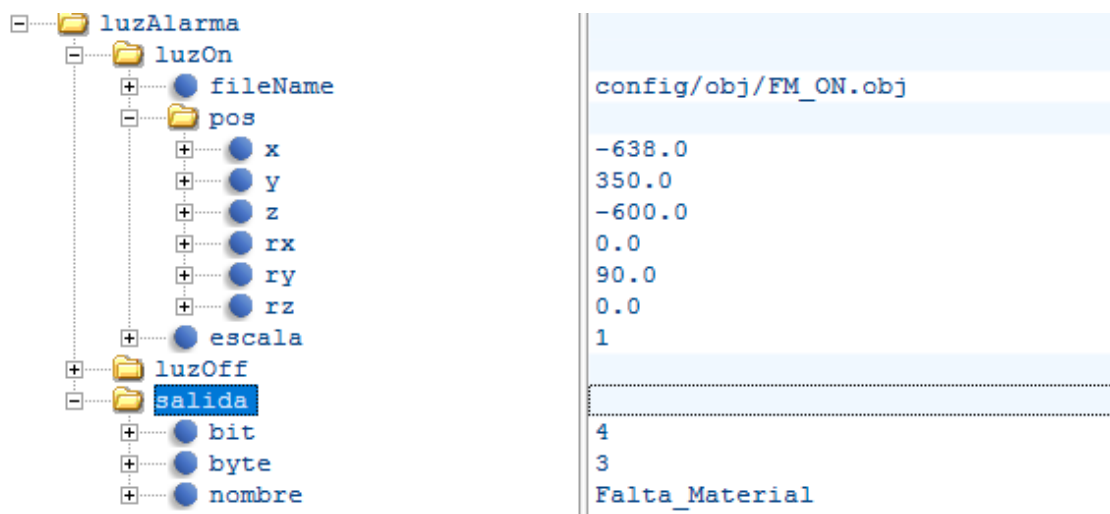


Figura 61. Estructura luces.

5 Arranque y Ejecución del Modelo Digital de la Estación FMS 202

En este capítulo se procederá a explicar el funcionamiento y ejecución del modelo digital de la Estación FMS-202. Como comienzo se empezará abriendo el ejecutable de java de la estación como se ve en la Figura 62. Se comienza abriendo la estación debido a que al ejecutar tantos programas al mismo tiempo es posible que no se abra el ejecutable por falta de memoria del ordenador.

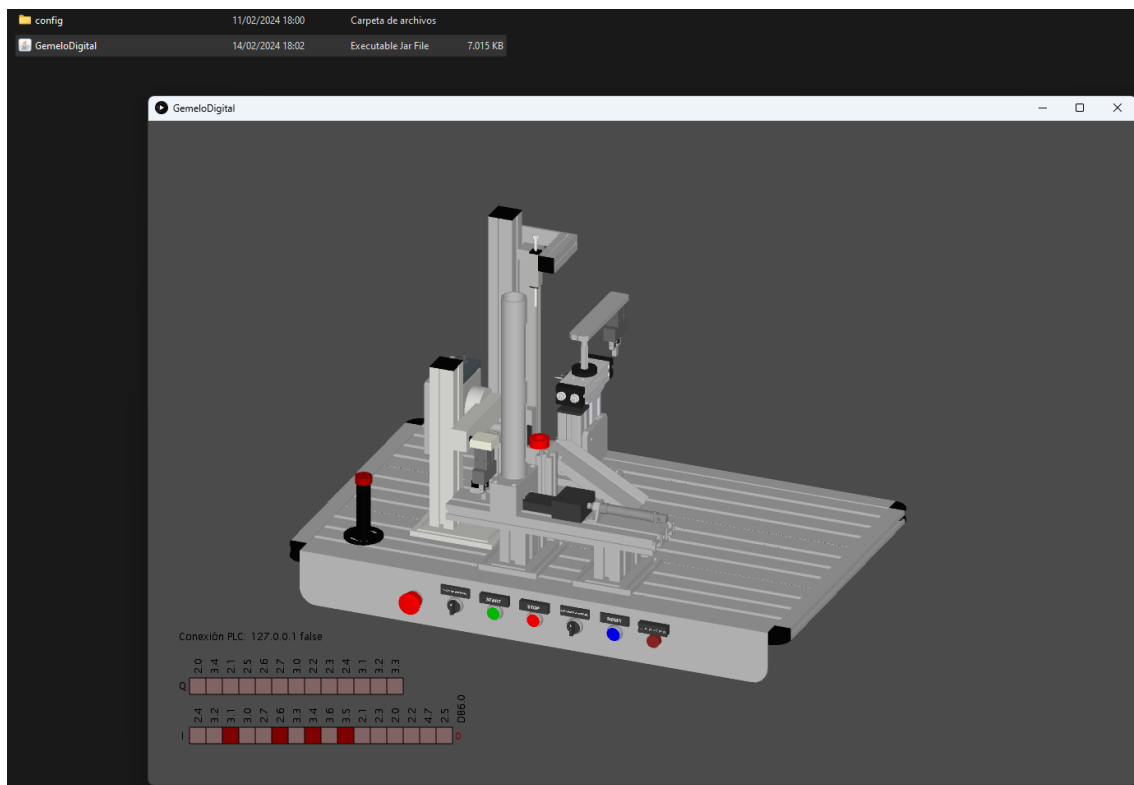


Figura 62. Ejecución de la estación FMS-202.

NetToPLCsim

Tras este paso se dejará de lado el modelo digital y se procederá a crear el server en NEtToPLCsim. Hay que tener en cuenta que para el correcto funcionamiento del programa se tiene que ejecutar como administrador. Para iniciar el server, primero se deberá añadir un nuevo servidor indicando para ello la IP de la network, la IP del PLC y el número del rack y del slot. Podemos observar los valores que se deben de indicar para la conexión del modelo con TIA portal en la Figura 63.

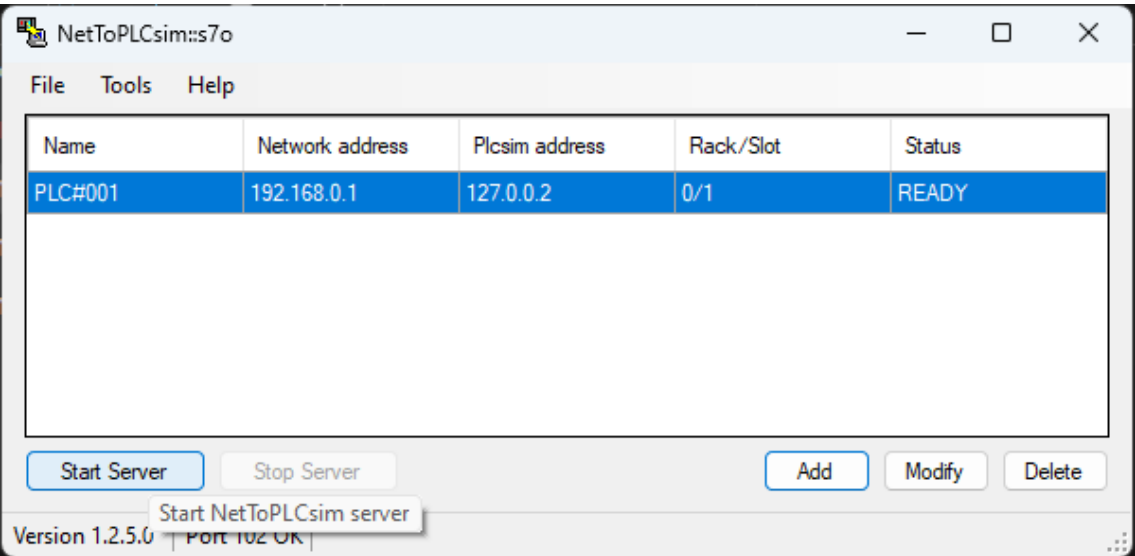


Figura 63. Configuración NetToPLCSim.

Tras esto se puede proceder a iniciar el server y dejarlo en segundo plano funcionando.

TIA portal

Habiendo realizado los dos pasos anteriores se puede proseguir con la creación del proyecto en TIA portal. Es recomendable que a la hora de abrir los programas se establezca una prioridad alta en el administrador de tareas tanto para TIA Portal como para PLCsim. En la Figura 64 se muestra un ejemplo de cómo establecer esta prioridad.

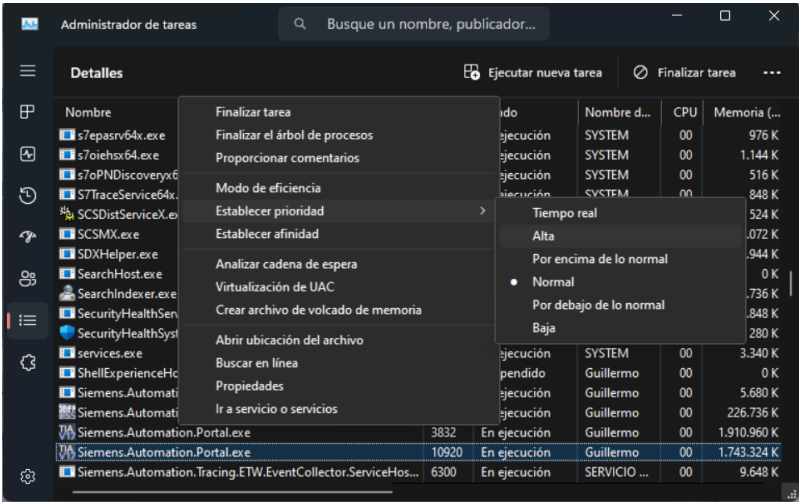


Figura 64. Establecimiento de alta prioridad.

Como se puede ver en la figura 65 debemos seleccionar la opción de crear proyecto y tras esto seleccionar la opción “Open the project view”.

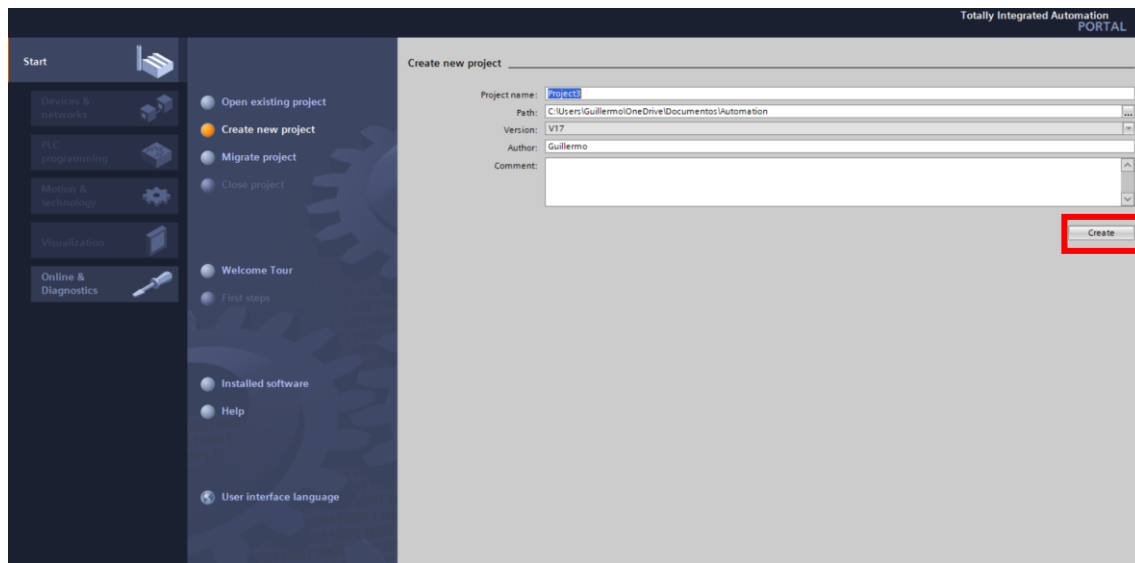


Figura 65. Creación del proyecto en TIA portal.

Una vez creado el proyecto es necesario añadir en el un PLC, para ello se deberá pulsar en la opción “Add new device” y seleccionar uno de los PLC que se encuentra en la lista. El PLC de la estación FMS-202 es el que se ha seleccionado en la Figura 65.

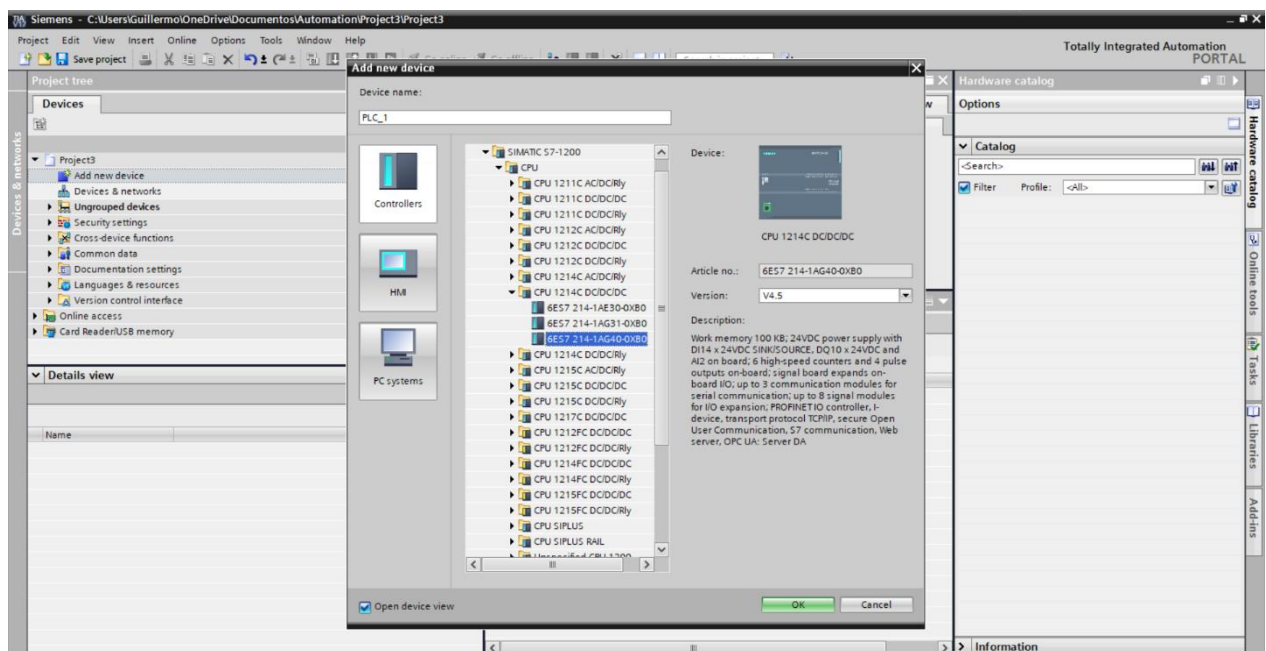


Figura 66. Selección de PLC en TIA Portal.

Es importante que se hagan dos modificaciones dentro de la configuración del PLC además de la comprobación de que la dirección Ip del PLC coincide con la añadida a NetToPLCsim y del rack y slot del PLC. En primer lugar, se procederá a habilitar la comunicación a través de PUT/GET del interlocutor remoto. Para ello, se accederá al apartado “Protection & Security”, según se detalla en la Figura 66.

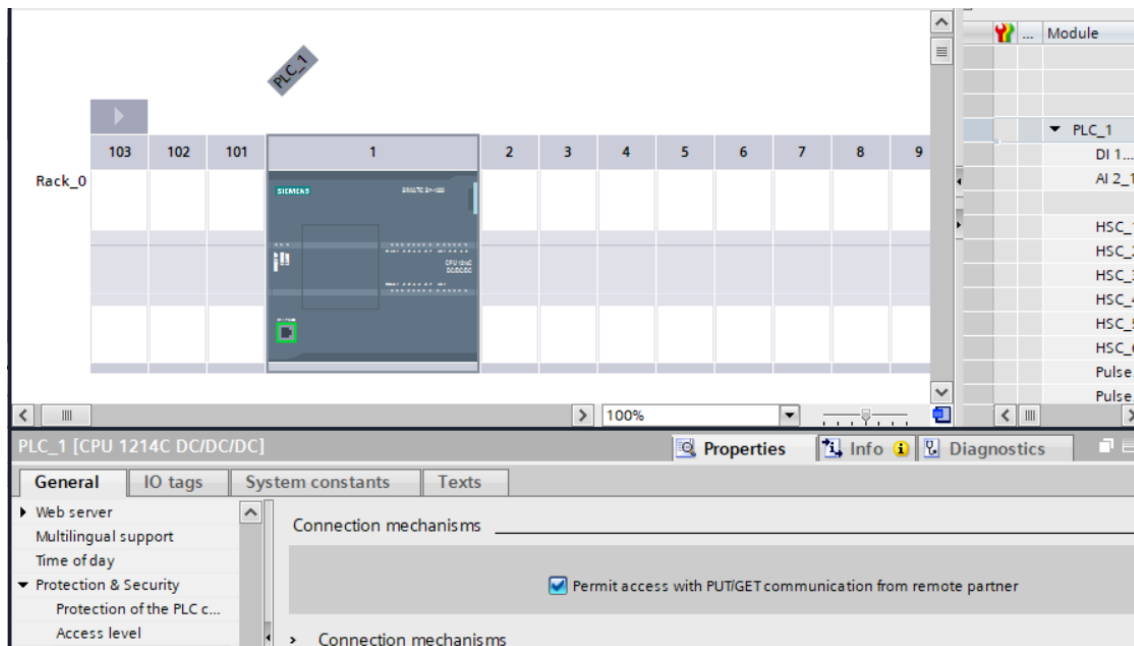


Figura 67. Configuración PLC TIA Portal.

El segundo cambio que se debe realizar es la configuración de la “Start address” como 10 para las direcciones de entrada dentro del apartado “I/O addresses”. En la Figura 68 se muestra cómo se debe realizar esta configuración.

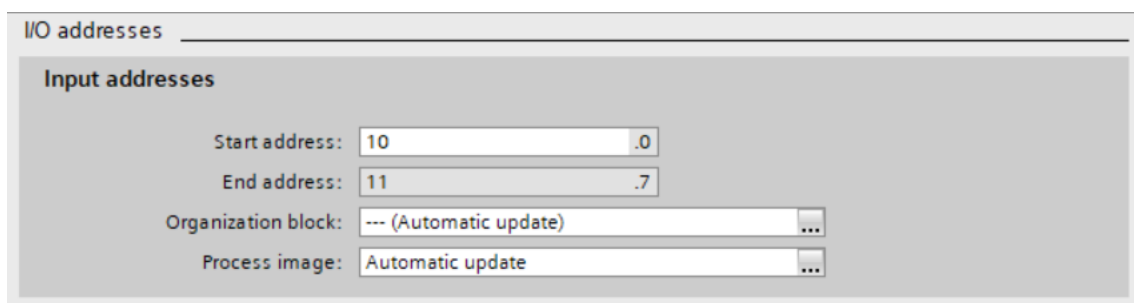


Figura 68. Configuración dirección entradas.

Una vez hecho esto ya se podrá empezar a trabajar con TIA portal de manera normal, lo único a tener en cuenta es la inclusión de un “Data block” como mínimo para poder usarlo como entrada analógica del modelo digital. Para generar el “Data block” se debe hacer click derecho en la pestaña “Program blocks” y esperar que se abra la ventana. Después se ha de seleccionar la opción “Data block” y en “Number” marcar la opción de “Manual” y seleccionar 6, ya que esta es la dirección a la que se encuentra asociada el modelo digital. Nombraremos al bloque “Entrada Analógica” y le daremos a OK como se ve en la Figura 69.

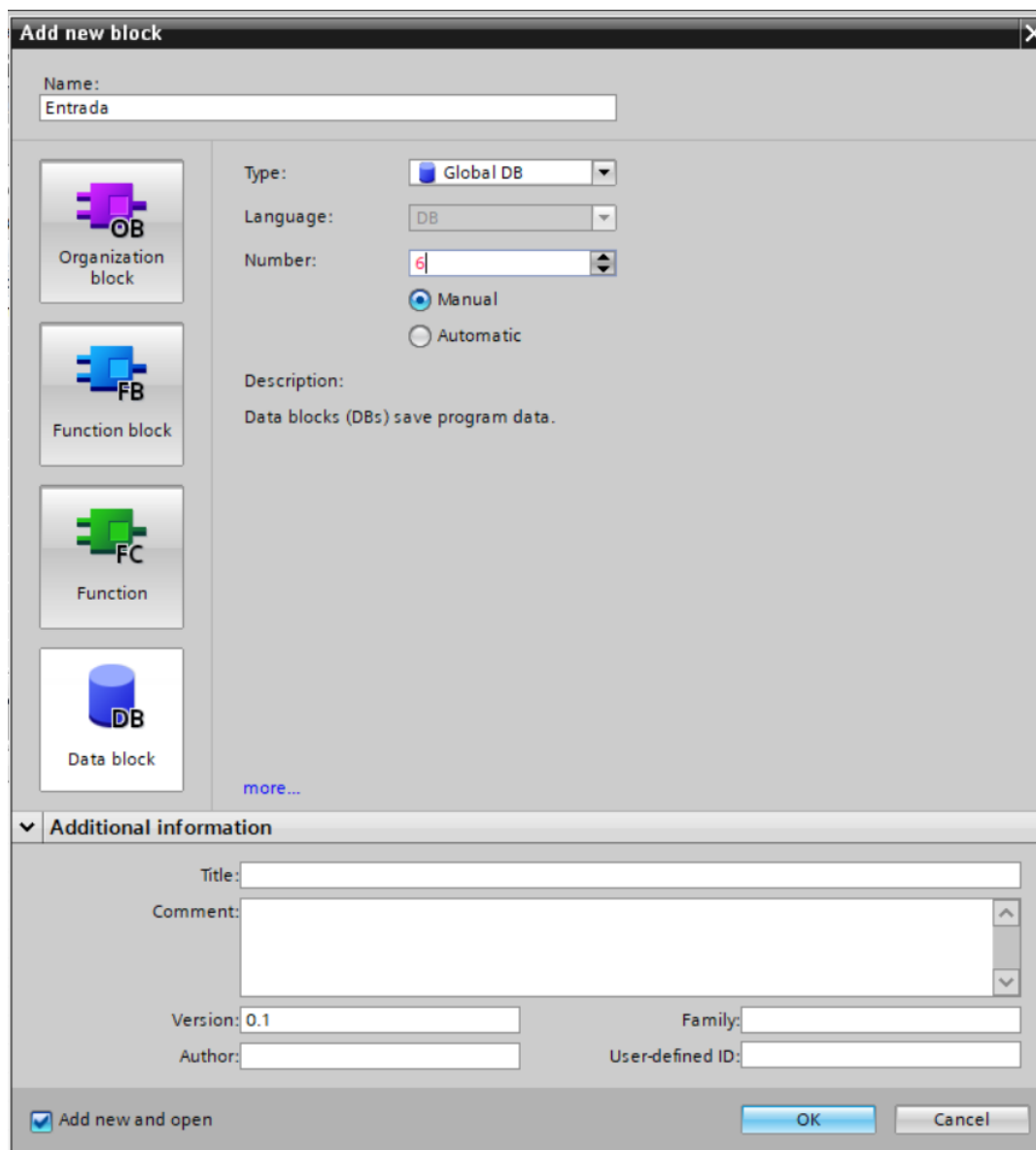


Figura 69. Generar Data block.

Dentro del Data block debe añadirse una nueva fila con el nombre de “Entrada”, tipo de dato “dInt”, y “Default value” igual a 0. Para acabar se ha de hacer clic derecho sobre el Data block, seleccionar el apartado “Propiedades” y dentro de la opción “Atributos” desmarcar las opciones “Optimized block access” y “Data block accesible from OPC UA”. Tras realizar esto podremos llamar a la dirección del Data block que contiene la variable “Entrada” y usarla como si fuera una entrada analógica como se puede ver en la Figura 70.

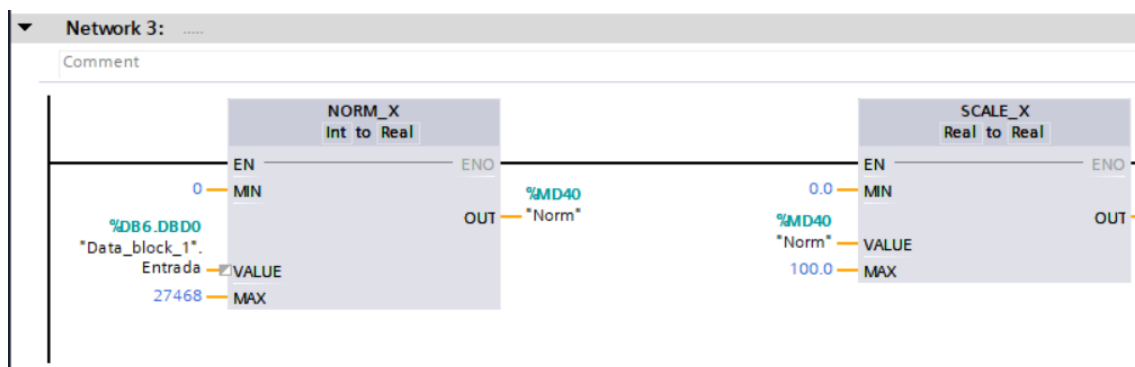


Figura 70. Ejemplo de uso de Data block como entrada analógica.

Una vez realizado todos estos pasos y con el código completado se podrá proceder a conectarnos con el modelo. Para ello se deberá iniciar PLCsim y cargar el programa en el PLC como se realiza de manera habitual. Como se ve en la Figura 71. Tras esto se seleccionará la opción “Go online”

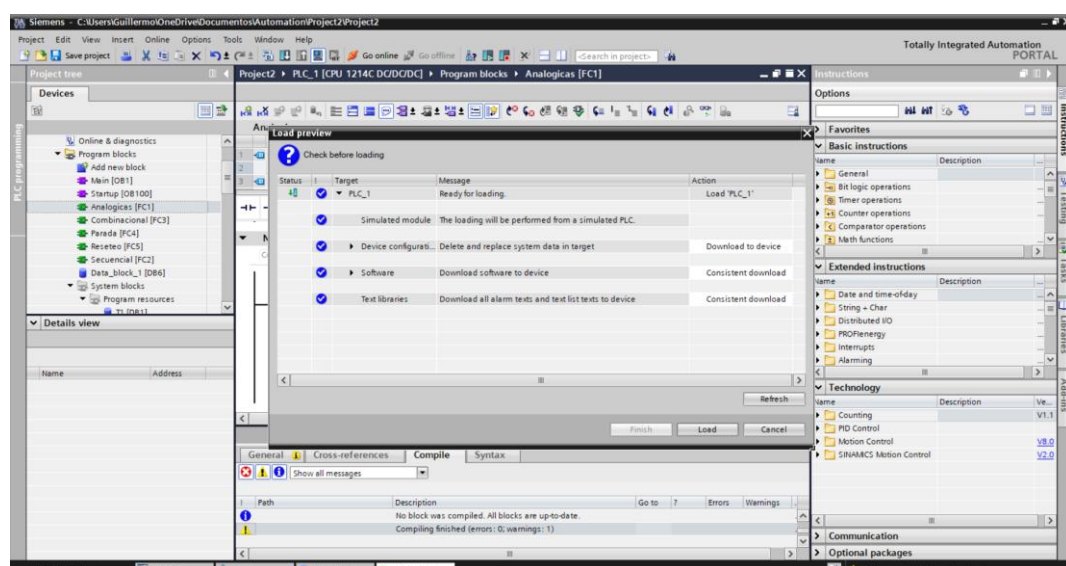


Figura 71. Carga programa en PLC

Finalmente se podrá volver al modelo digital de la estación. Como se ha descrito anteriormente el modelo de la estación cuenta con un sistema de generación aleatorio de rodamientos cada vez que se resetea la estación (Mayus+R). Hay que tener en cuenta que para resetear la estación es necesario que no se encuentre conectada al PLC ya que podría dar lugar a fallo. Para conectar la estación al PLC deberemos pulsar la tecla “p” y ya tendremos la posibilidad de interactuar con el modelo y que responda a la programación generada en TIA portal. En la figura 72 se puede observar cómo se encuentra conectada el modelo de la estación con el PLC.

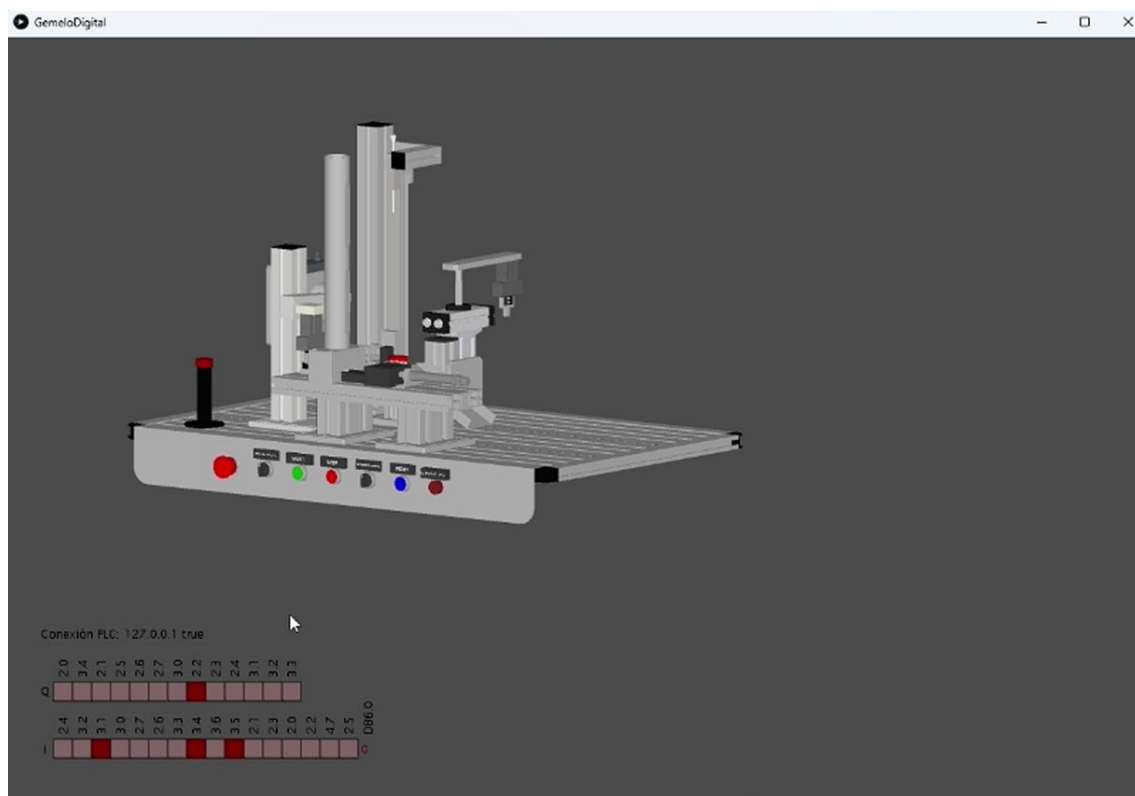


Figura 72. Modelo digital conectado al PLC

6 Conclusiones

Después de finalizar el desarrollo del proyecto del modelo digital, se ha evidenciado el considerable beneficio que aporta una herramienta de este tipo. Este desarrollo aporta una gran cantidad de beneficios y herramientas tanto en el ámbito educativo como en el industrial.

Respecto al objetivo planteado para el proyecto, se ha conseguido desarrollar el modelado digital de un sensor analógico bastante efectivo y que puede ayudar al alumno al entendimiento del funcionamiento de estos. Además, pese a que este proyecto se haya centrado en el desarrollo de un modelo digital para un sensor de desplazamiento lineal también sienta unas bases para poder readaptar todo lo desarrollado en este proyecto e implementarlo para otros tipos de sensores.

En relación a la estación FMS-202 se puede ver que es muy fidedigno respecto a la estación real. Pese a esto a un cuenta con muchas mejoras que pueden ser implementadas, como ejemplo un sistema de movimiento y colisiones dentro de la estación. Durante el desarrollo de esta estación se ha tenido que añadir un switch para controlar una entrada que define la presencia de un rodamiento en la primera plataforma debido a que no es posible actualmente implementar este sensor debido a que no existe el sistema anteriormente descrito.

Aun así considero que el desarrollo de este proyecto de fin de grado ha contribuido positivamente al desarrollo de la Plataforma GRAV-DT y ha dotado al alumnado de la Escuela Politécnica Superior de Jaén de una herramienta muy potente y totalmente versátil para aprender sobre la programación de PLC de Siemens.

7 Bibliografía

Toala, F.J. et al (2022) Obtenido de

<https://editorialalema.org/index.php/pentaciencias/article/view/29>

Inventor (2024). Autodesk Inventor. Obtenido de

<https://www.autodesk.es/products/inventor/features>

Programarfacil. (s.f.). Obtenido de

<https://programarfacil.com/podcast/80-processing-lenguaje-para-graficos/>

Alconchel, M. (2019) Obtenido de

<https://openwebinars.net/blog/que-es-xml-y-para-que-se-usa/>

Oracle. (s.f.). Obtenido de

<https://www.oracle.com/es/tools/technologies/netbeans-ide.html>

Snap7. (s.f). Snap7.Sourceforge. Obtenido de

<https://snap7.sourceforge.net/moka7.html>

Inprotech. (2023). Obtenido de

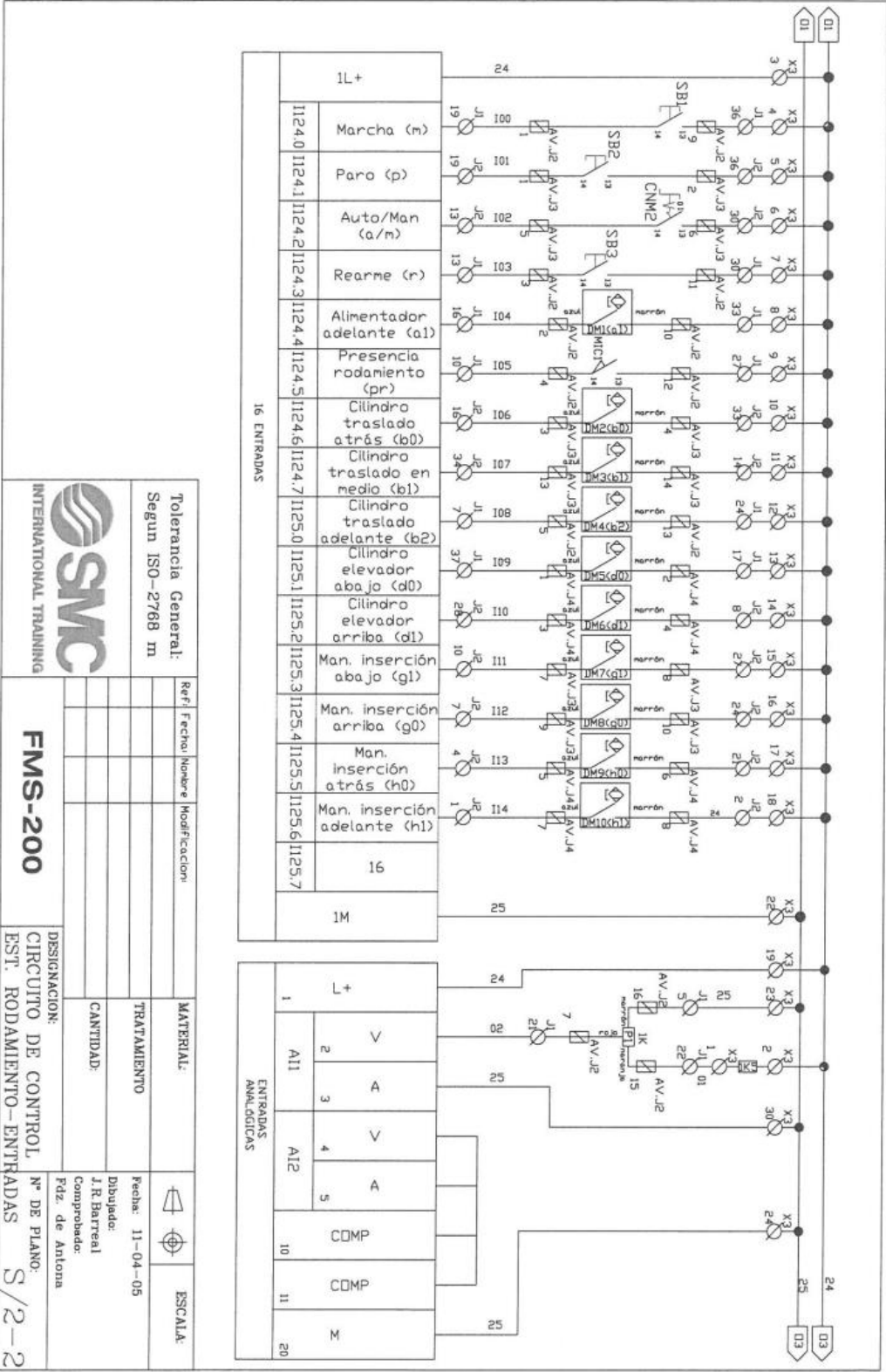
https://inprotech.es/protocolo-s7comm-analizamos-seguridad/#Descripcion_tecnica

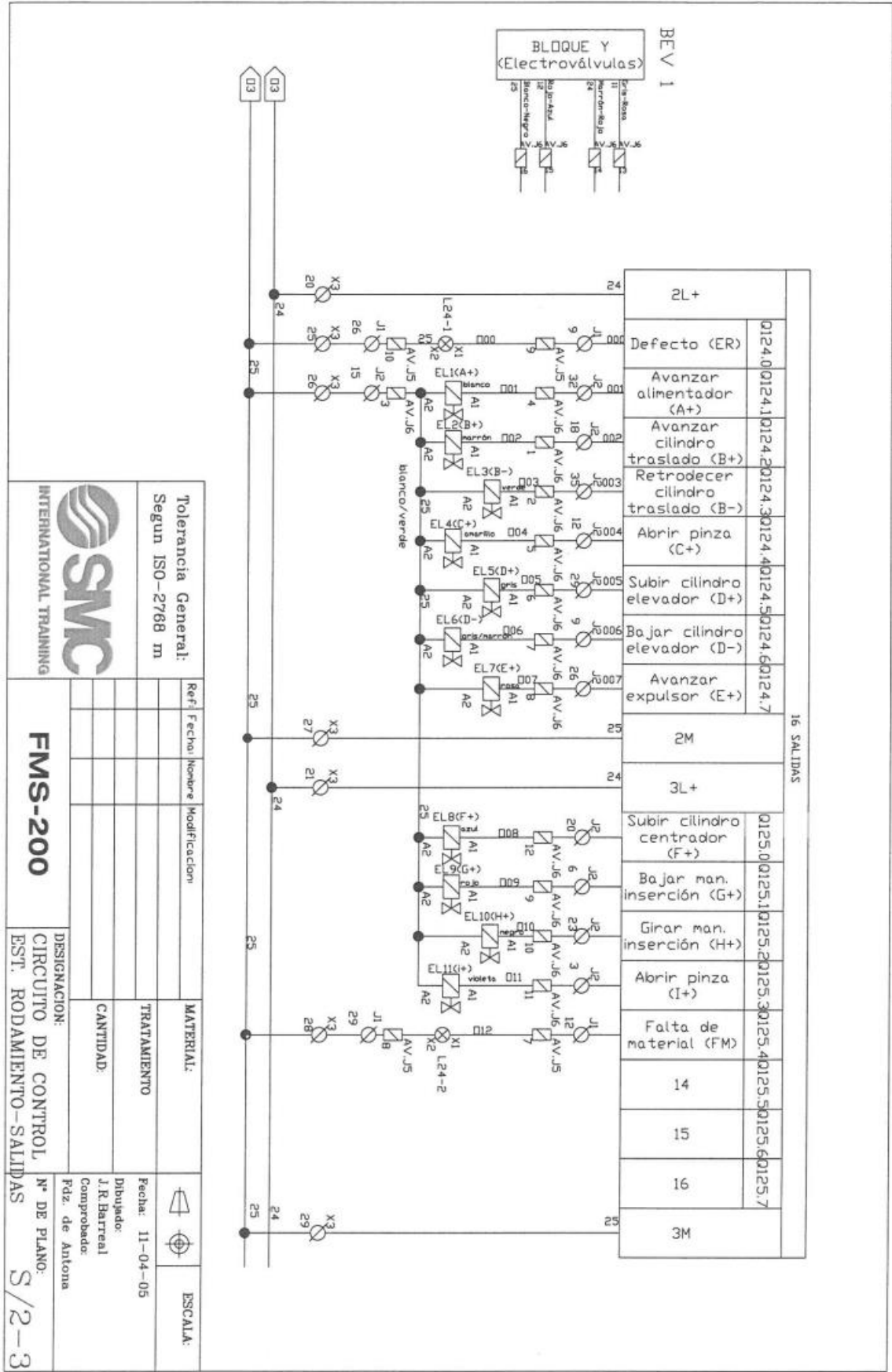
Repsol. (s.f.). Obtenido de

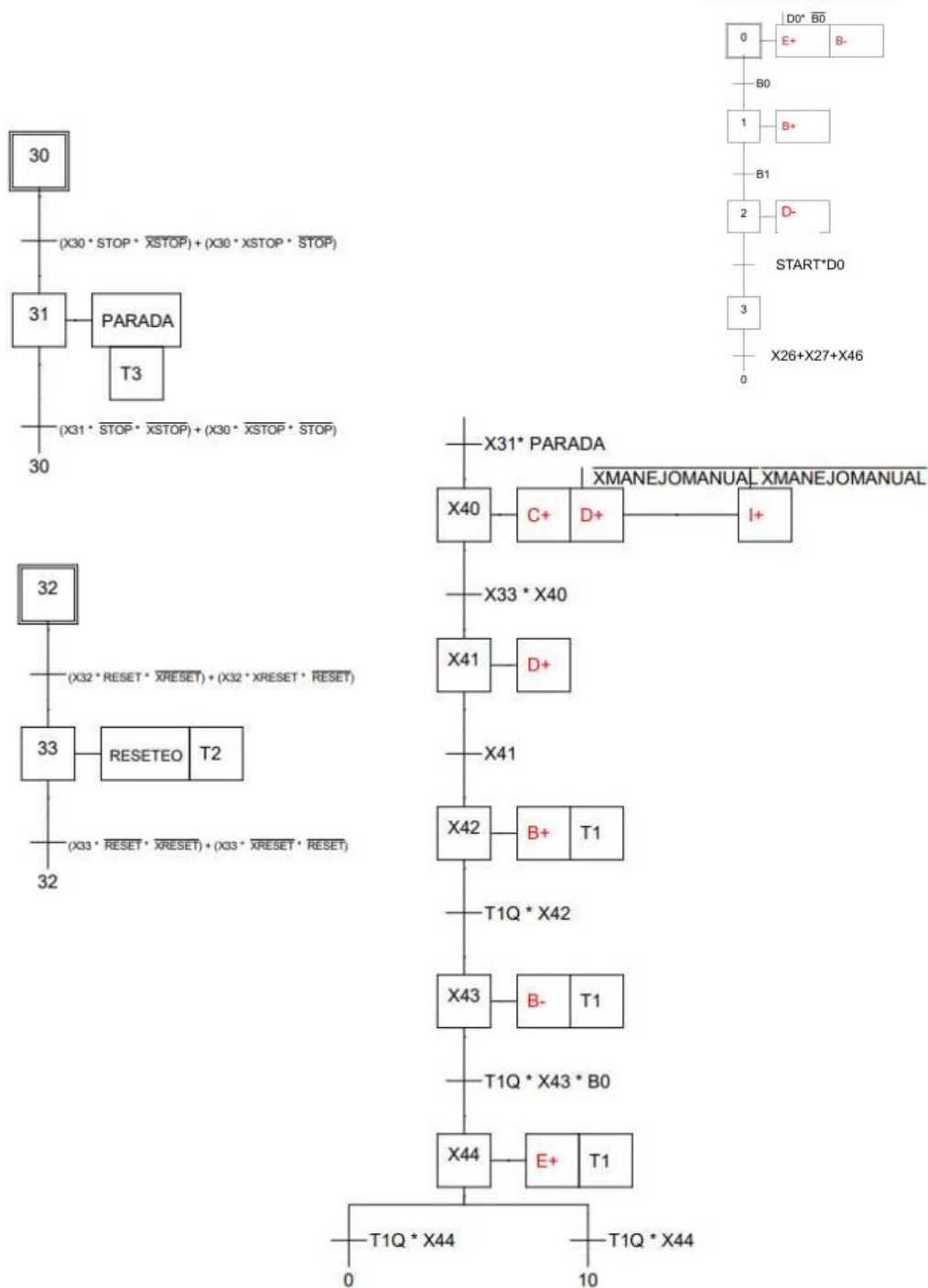
<https://www.repsol.com/es/energia-futuro/tecnologia-innovacion/gemelos-digitales/index.cshtml>

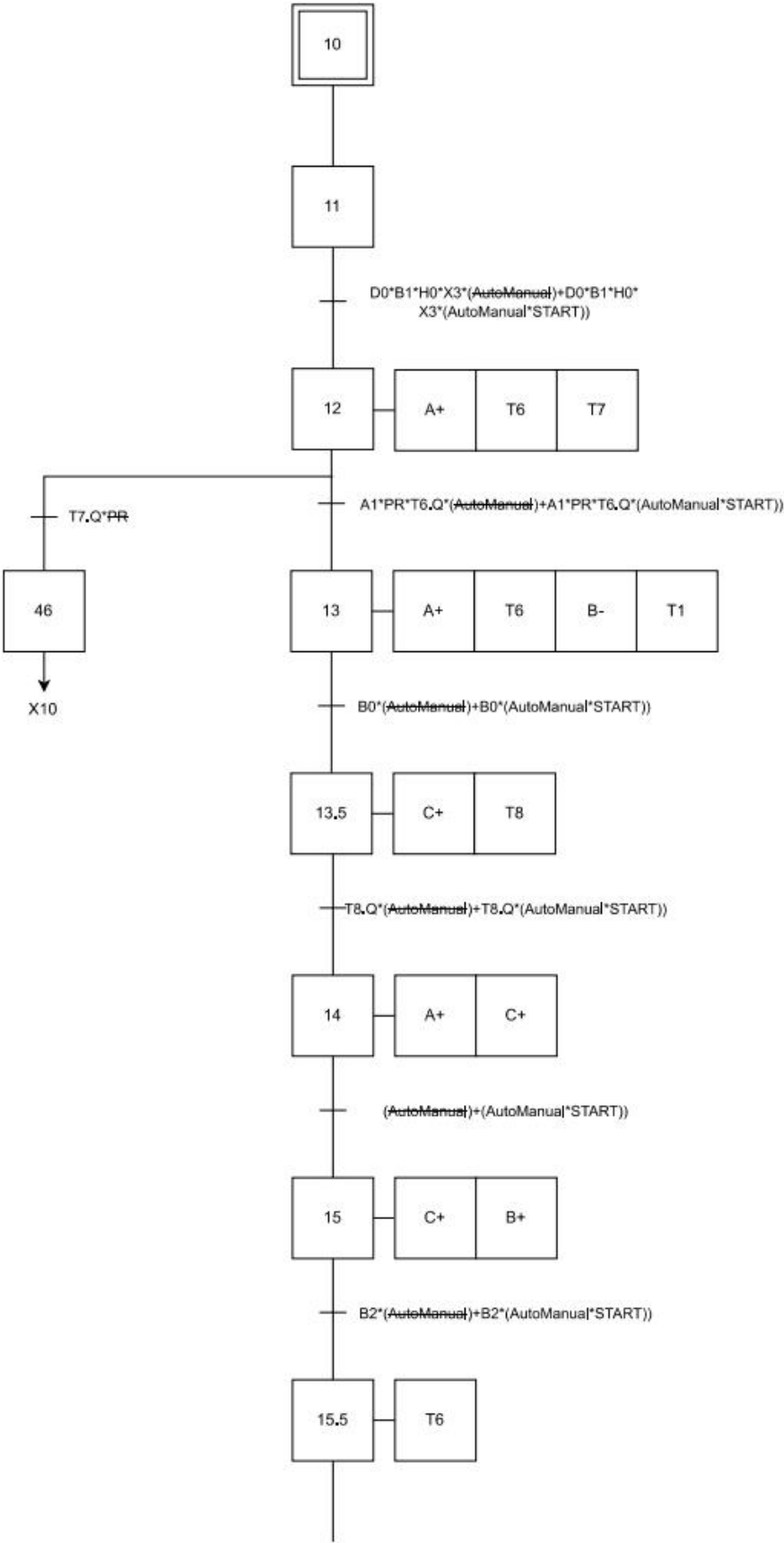
Anexos

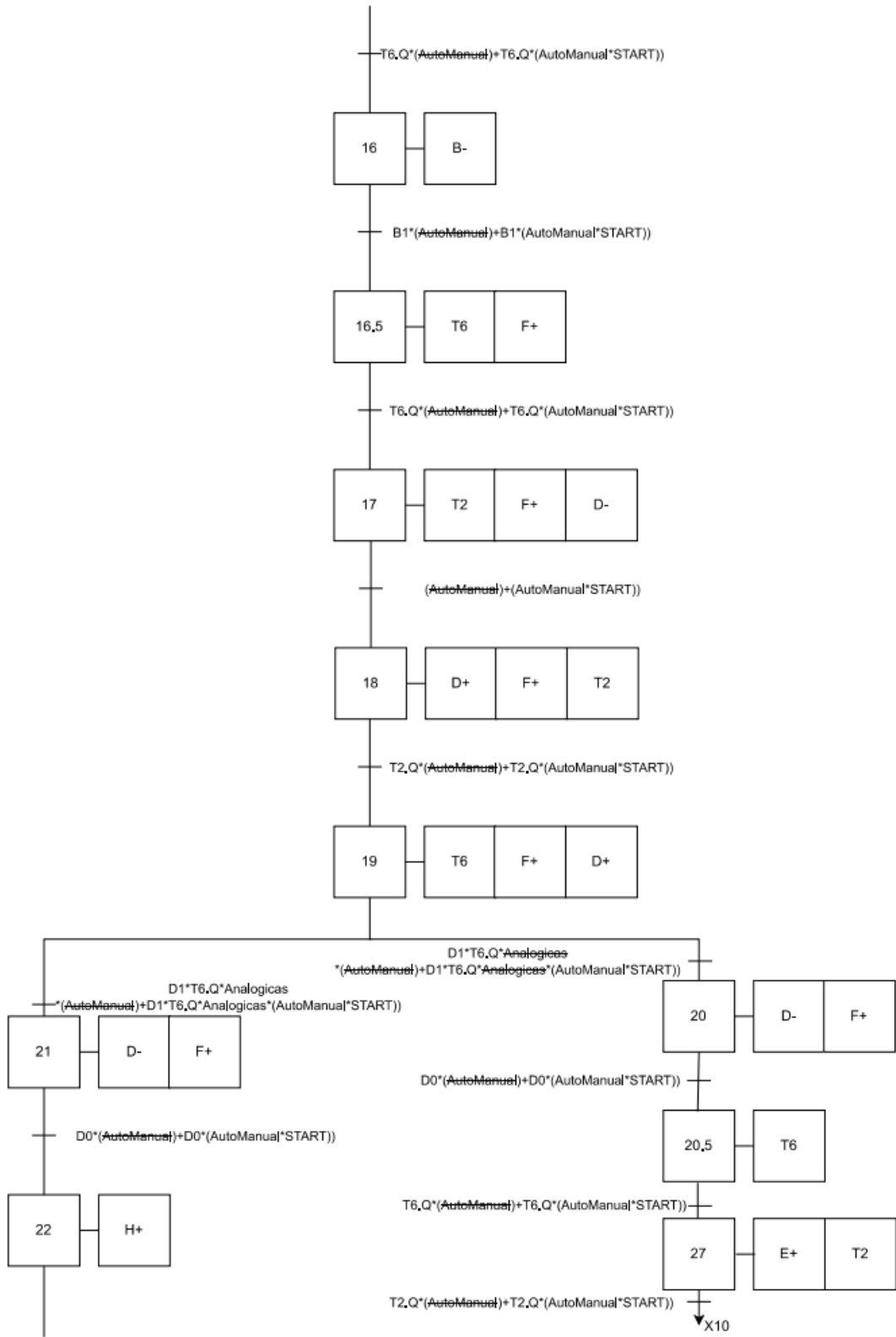
Anexo A: Planos eléctricos de la estación

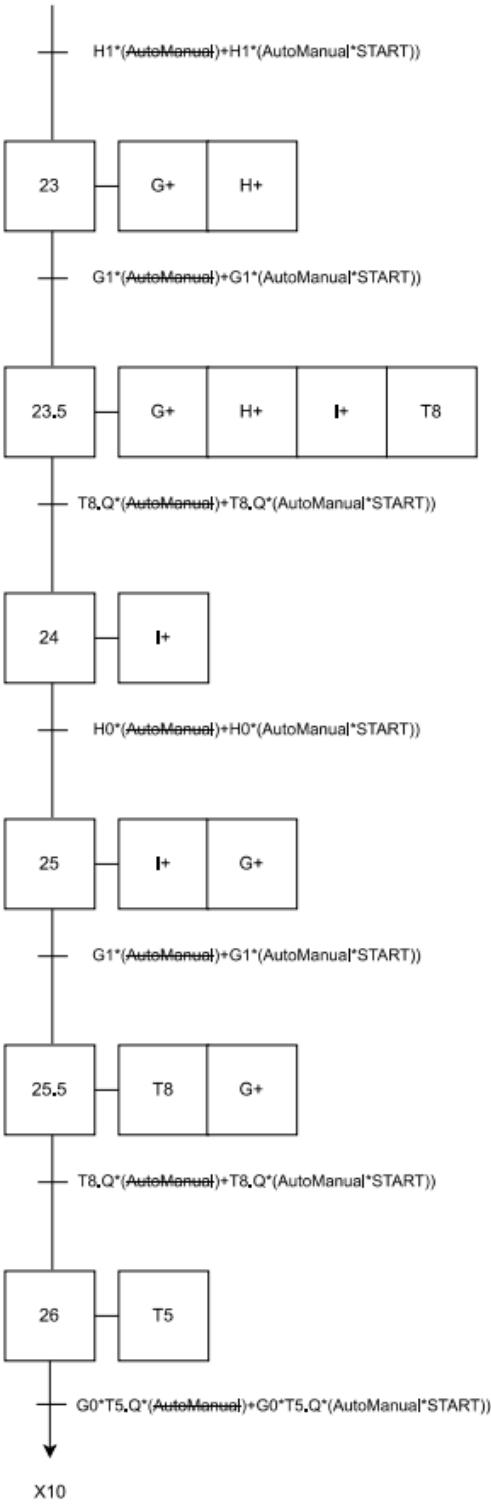




Anexo C: Grafcet con la programación de la estación FMS-202







Anexo D: Video funcionamiento de la estación.



