

MÁSTER UNIVERSITARIO EN INGENIERÍA INDUSTRIAL

TRABAJO FIN DE MÁSTER

PUESTA EN MARCHA Y MONITORIZACIÓN DE UN BANCO DE ENSAYOS PARA ESTUDIO DE RESPUESTA AL FALLO DE SISTEMA BASADO EN CILINDROS NEUMÁTICOS



Estudiante: Castro del Cid, Jon

Directora: Romero Anton, Naiara

Codirector: Bediaga Escudero, Iñigo

Curso: 2023-2024

Fecha: Bilbao, 27 de febrero de 2024



Resumen

En la presente época caracterizada por avances tecnológicos acelerados, el presente estudio se adentra en el impacto transformador de las aplicaciones de la inteligencia artificial (IA) en el ámbito de la fabricación industrial. Este sector, con sus procesos complejos y numerosas operaciones, está en vías de experimentar cambios significativos debidos a la adopción de soluciones impulsadas por el desarrollo de aplicaciones de aprendizaje automático, orientadas a mejorar la eficiencia, productividad y la toma de decisiones en el sector.

Con el propósito de abordar esta inminente transformación, el presente trabajo introduce innovaciones en el campo del aprendizaje automático que comprenden modelos de inteligencia artificial generativa, entre otros. Los tres modelos concebidos abren las puertas a la implementación de un plan de mantenimiento predictivo en componentes del sector de la máquina herramienta, aunque cabe destacar que estas soluciones son altamente adaptables y escalables a diversos sectores.

Sin embargo, antes de la aplicación de estos modelos, es imperativo llevar a cabo la puesta en marcha y validación de un banco de ensayos, cuyas tareas se definen también en el escrito presente. Este banco sirve, sobre todo, como plataforma para el montaje de los sensores que proporcionan los datos para el estudio posterior, relacionado con las implementaciones de modelos basados en redes neuronales.

Los resultados obtenidos en este estudio son sumamente satisfactorios; no solo se han logrado cumplir con los objetivos establecidos para la generación de modelos de detección de anomalías, sino que además se ha orientado a parte del grupo del centro de investigación del centro tecnológico hacia el estudio más detenido de uno de los modelos presentados.

Palabras clave:

Industria 4.0, Mantenimiento predictivo, Inteligencia Artificial generativa, IoT, *Smart Manufacturing*.



Laburpena

Aurrerapen teknologiko azeleratuak nagusi diren garai honetan adimen artifizialeko (IA) aplikazioek industria-fabrikazioan duten eragin eraldatzailea aztertzea izan da helburu azterketa honetan. Fabrikazio industrialean aldaketa ugari eta nabarmenak gertatzen ari dira, ikaskuntza automatikoko aplikazioen garapenak bultzatutako soluzioak inplementatzen baititu, horien helburua delarik sektorean eraginkortasuna, produktibitatea eta erabakiak hartzea hobetzea.

Gain-gainean den iraultza honetan murgildu ahal izateko, lan honek ikaskuntza automatikoan oinarritutako aurrerapenak aurkezten ditu, besteak beste adimen artifizial sortzailearen ereduak sortuz eta azalduz. Garatutako hiru ereduak makina-erremintaren sektorean nonahi erabiltzen diren elementuen mantentze-lan prediktiboaren plan bat ezartzeko baliabideak eskaintzen dituzte. Hala ere, soluzio horien irismena eta potentziala makina-tresnaren sektoretik haratago doa, askotariko sektoreetan aplikatu daitezkeen ereduak baitira, ez industriaren arloan soilik.

Nolanahi ere, eredu hauek aplikatu aurretik, ezinbestekoa da proba-banku bat muntatzea eta balidatzea. Hori kontuan hartuta, ereduaren emaitzak aurkeztu aurretik, beharrezkoak izan diren kontrol-elementuak, materialak, softwarea eta esperimenduaren diseinua azalduko dira dokumentuan zehar. Banku horrek, batez ere, ondorengo azterketarako datuak ematean dituzten sentsoreak muntatzeko plataforma gisa balio du, zeina lotuta baitago sare neuronaletan oinarritutako ereduaren inplementazioekin.

Ikerketa honetan lortutako emaitzak oso pozgarriak izan dira; izan ere, anomaliak atzemateko ereduak sortzeaz gain, ikerketa honetan azalduetako eredu sortzaileak posible egin du ikerketa-zentroan ikerketa-arlo berri bat sortzea.

Hitz gakoak:

Industria 4.0, Mantentze-lan prediktiboa, Adimen Artifizial sortzailea, IoT, *Smart Manufacturing*.



Abstract

In the epoch characterized by accelerated technological progress, this study explores the transformative impact of artificial intelligence (AI) applications within the realm of industrial manufacturing. The intricacies of this sector, with its complex processes and multifaceted operations, stand poised for substantial modifications owing to the adoption of applications propelled by the evolution of machine learning. These applications are explicitly designed to enhance efficiency, productivity, and decision-making processes within the industrial manufacturing sector.

In pursuit of addressing this imminent transformation, the present work introduces advancements in the field of machine learning, incorporating generative artificial intelligence models, among others. The three models developed offer avenues for implementing a predictive maintenance plan applicable to components widely employed in the machine tool sector. It is noteworthy that these solutions exhibit a high degree of adaptability and scalability, extending their applicability to diverse industrial sectors.

Prior to the deployment of these models, it is imperative to conduct the implementation and validation of a test bench, the details of which are delineated in this document. This test bench serves as a pivotal platform for the assembly of sensors, providing data essential for the subsequent examination pertaining to the implementations of neural network models.

The results derived from this study are exceptionally satisfactory; not only have the predefined objectives for the generation of anomaly detection models been achieved, but also the generative model elucidated in this study has paved the way for further research within the research centre's team.

Key words:

Industry 4.0, Predictive Maintenance, Generative Artificial Intelligence, IoT, Smart Manufacturing.



Índice de contenido

1. Introducción.....	1
2. Contexto.....	4
2.1. DANOBATGROUP	5
2.2. Rectificadora vertical VG.....	7
2.3. Cilindros neumáticos.....	9
2.4. Sistema de Control (VTEM).....	11
2.5. Captación de datos (Savvy).....	13
2.5.1. Savvy Smart Box	14
2.5.2. Servicio <i>cloud</i> de Savvy.....	15
2.5.3. Ciclo de vida del dato.....	18
3. Objetivo y alcance.....	21
4. Beneficios	22
4.1. Beneficios técnicos.....	22
4.2. Beneficios económicos.....	23
5. Descripción de requerimientos	24
5.1. Requerimientos materiales.....	24
5.2. Requerimientos de software.....	26
6. Estado del arte.....	27
6.1. Modos de fallo de los cilindros neumáticos.....	28
6.1.1. Mecanismo de fallo en los cilindros	30



6.2.	Motion Terminal VTEM de Festo	32
6.2.1.	Motion App	34
6.2.2.	Válvulas piezoeléctricas	37
6.3.	Mantenimiento predictivo en la industria.....	44
7.	Exposición de alternativas.....	48
7.1.	Criterios de selección.....	49
7.2.	Sensores de distancia láser.....	50
7.3.	Sensores de cable	52
7.4.	Sensores de efecto Hall – Elección final.....	53
8.	Descripción de la solución propuesta.....	55
8.1.	Topología.....	56
8.1.1.	Circuito neumático.....	57
8.2.	Control.....	58
8.3.	GUI de la HMI	60
8.3.1.	Pantalla principal.....	60
8.3.2.	Pantalla de válvula.....	62
8.3.3.	Pantallas secundarias.....	64
8.4.	GUI del portal de acceso de Savvy.....	65
9.	Descripción tareas, fases y procedimientos	67
10.	Diagrama de Gantt	70
11.	Casos de aplicación de los modelos predictivos.....	71
11.1.	Representación de los datos.....	73



11.1.1.	El modo de transferencia.....	74
12.	Algoritmos desarrollados.....	79
12.1.	Autoencoder	79
12.2.	Modelo VAE	81
12.3.	Modelo TCN	82
13.	Resultados.....	83
13.1.	Autoencoder	83
13.2.	Autoencoder variacional.....	85
13.3.	TCN	88
14.	Descripción del presupuesto.....	89
14.1.	Análisis de rentabilidad.....	90
	<i>Conclusiones y líneas futuras</i>	92
	ANEXO I: CÓDIGO DE CONTROL DE MOVIMIENTO	96
	ANEXO II: RESUMEN DE OPERACIONES DE MECANIZADO DE LAS RECTIFICADORAS DE DANOBAT	140



Índice de tablas

Tabla 1. Resumen de productos de la gama VG.....	8
Tabla 2. Resumen de los cilindros a instalar.....	9
Tabla 3. Listado y descripción de las MA.	34
Tabla 4. Características de los distintos tipos de transductores.....	43
Tabla 5. Comparativa de sensores láser de medición.....	51
Tabla 6. Comparativa de sensores de medición de cable.....	53
Tabla 7. Modos de estado de las válvulas.....	61
Tabla 8. Desglose de costes de los elementos de control.....	89
Tabla 9. Desglose de costes de los cilindros y sus componentes.	90
Tabla 10. Análisis de retorno de inversión.	91
Tabla 11. Rectificadoras de exteriores.....	141
Tabla 12. Rectificadoras de interiores.....	141
Tabla 13. Rectificadoras sin centros.....	142
Tabla 14. Rectificadoras verticales.....	142



Índice de figuras

Figura 1. Trampilla de estudio (parte del sistema ATC).....	4
Figura 2. Esquema corporativo de DANOBATGROUP.	5
Figura 3: Cilindro DSBC-63-500-PPSA-N3.....	11
Figura 4: Cilindro CP96SDB63-500C.....	11
Figura 5. VTEM instalado en el BEC.	13
Figura 6. Dispositivo Savvy Smart Box instalado en el BEC.....	15
Figura 7. Portal de visualización de Savvy cloud.....	16
Figura 8. Gráfica por defecto del portal de visualización.	17
Figura 9. Portal de gestión de Savvy cloud.	18
Figura 10. Diagrama conceptual de las Savvy Smart Box.....	19
Figura 11. Ciclo de vida del dato.....	20
Figura 12. Elementos de unión de los cilindros: laminas, perfil y escuadras.....	25
Figura 13. Desglose del cilindro de la marca Festo.....	28
Figura 14. FTA de los cilindros neumáticos.....	29
Figura 15. Mecanismo de fallo por abrasión de la superficie.....	30
Figura 16. Mecanismo de fallo por oxidación.	31
Figura 17. Diagrama de beneficios del VTEM.....	34
Figura 18. Electroválvula de la marca Tameson.....	37
Figura 19. Material de partida.....	38
Figura 20. Adición de superficie conductora.	38



Figura 21. Polarización del material cerámico.....	39
Figura 22. Deformación de componente piezoeléctrico.....	39
Figura 23. Creación del actuador de flexión.....	40
Figura 24. Diferencia energética entre tecnologías de válvulas.....	41
Figura 25. Estado del transductor sin voltaje.....	42
Figura 26. Estado del transductor con voltaje.....	42
Figura 27. Proporcionalidad de la tecnología piezoeléctrica.....	44
Figura 28. Evolución temporal del mantenimiento.....	46
Figura 29. Jerarquía de los modelos de predicción de RUL.....	46
Figura 30. Propuestas piloto de sensores de posición.....	48
Figura 31. De izquierda a derecha sensores Baumer, Keyence y Wenglor.....	51
Figura 32. De izquierda a derecha los sensores de Baumer, TSM y Positek.....	52
Figura 33. Transmisores de posición de Festo (izda.) y de SMC (dcha.).....	54
Figura 34. Solución final del banco de ensayos.....	55
Figura 35. Topología de elementos conectados a los cilindros.....	56
Figura 36. Tanque de aire presurizado.....	57
Figura 37. Segunda llave de paso del sistema. A la entrada del filtro de aire.....	58
Figura 38. Bloque de control para la válvula 0.....	59
Figura 39. Pantalla principal de la GUI.....	60
Figura 40. Ejemplo de aparición del botón Acknowledge.....	61
Figura 41. Zona táctil para acceder a las pantallas de válvula.....	62
Figura 42. Pantalla de válvula.....	63



Figura 43. Ajustes de grabación de la válvula 0.....	64
Figura 44. Ejemplo de mensaje de la GUI.....	64
Figura 45. GUI de la plataforma Savvy.	65
Figura 46. Cámara IP instalada en el banco.....	66
Figura 47. Salida de la junta rascadora.	71
Figura 48. Evolución de parámetros temporales en el cilindro dañado.....	72
Figura 49. Datos en crudo de la presión.....	73
Figura 50. Datos en crudo del caudal.....	74
Figura 51. Valores disponibles con la MA 1.....	75
Figura 52. Definición de una válvula virtual para lectura de parámetros.....	76
Figura 53. Información de obtención del caudal mediante el modo de transferencia.	76
Figura 54. Definición de parámetros para la lectura del caudal.	77
Figura 55. Datos sin tratar de la señal de posición.	78
Figura 56. Esquema de arquitectura de AE.....	80
Figura 57. Arquitectura del VAE.....	81
Figura 58. Arquitectura de modelo TCN.....	82
Figura 59. reconstrucción con datos normales (autoencoder).....	83
Figura 60. Reconstrucción con datos anómalos (autoencoder).....	84
Figura 61. Histogramas de comportamientos normales y defectuosos.....	84
Figura 62. Reconstrucción con datos normales (VAE).....	85
Figura 63. Reconstrucción con datos anómalos (VAE).....	86



Figura 64. Datos sintéticamente generados por el VAE.....	87
Figura 65. Histogramas de comportamientos normales y defectuosos (VAE).....	87
Figura 66. Predicción con datos normales (TCN).....	88
Figura 67. Predicción con datos anómalos (TCN).....	88



Acrónimos

ALT	<i>Accelerated Life Testing</i>
ATC	<i>Automatic Tool Changer</i>
BEC	Banco de Ensayos de Cilindros
CNC	<i>Computer Numerical Control</i>
C.U.R.S.	<i>Cycles Until Recording Starts</i>
FTA	<i>Fault Tree Analysis</i>
GUI	<i>Graphical User Interface</i>
IIoT	<i>Industrial Internet of Things</i>
MA	<i>Motion App</i>
MAE	<i>Maximun Absolute Error</i>
M2C	<i>Machine to Cloud</i>



PLC	<i>Program Logic Controller</i>
RUL	<i>Remaining Useful Life</i>
TCN	<i>Temporal Convolutional Network</i>
TTF	<i>Time To Failure</i>
VAE	<i>Variational Autoencoder</i>



1. Introducción

El aumento gradual e incesante de las necesidades técnicas de las piezas realizadas mediante fabricación automatizada ha desembocado en un cambio categórico en cuanto a los requisitos y a la complejidad demandada en el sector de la máquina herramienta.

Pese a que históricamente la responsabilidad del mecanizado dependía en gran medida del buen hacer del operario, cada vez es mucho menos común poder encontrar máquinas que no estén gobernadas por algún tipo de controlador. Este hecho es fácilmente comprobable si se realiza un rápido estudio a los catálogos de empresas de renombre en el sector de la máquina herramienta (tales como Danobatgroup, DMG Mori o GMTG), puesto que uno se puede percatar de que es difícil, si no imposible, encontrar maquinaria que no tenga incorporada algún sistema de control numérico (CNC) que permita manipular la máquina en cuestión.

A todas luces, el nivel de complejidad y sofisticación de las máquinas herramientas ha aumentado sustancialmente en respuesta a los altos estándares que exige el mercado contemporáneo. Esto resulta en que en las máquinas coexistan una infinidad de componentes embebidos en distintos sistemas; como el sistema de control, el sistema hidráulico, el sistema de refrigeración o el sistema de evacuación entre varios otros.

Como en todo sistema, al aumentar el número de elementos se corre el riesgo de que la tasa de fallos del sistema global aumente, lo cual supone una disminución en el tiempo en el que la máquina se encuentra disponible, que resultará tarde o temprano en un desembolso económico por parte del cliente final en gastos de mantenimiento.

Con el fin de reducir al máximo los periodos en los cuales la máquina no se encuentra operativa, es común que quede definido en los manuales de mantenimiento el realizar ciertas tareas de prevención de fallos. Estas tareas pueden comprender desde la realización de ciertos chequeos de los parámetros significantes del sistema hasta el intercambio de piezas que, realmente, se encuentran en buen estado.

Como se puede suponer, el realizar intercambios de piezas en buenas condiciones repercutirá en gran medida en el coste final de mantenimiento y puede alargar en demasía el tiempo de amortización de la máquina, el cual de por sí es de un periodo



considerable. Las prácticas de mantenimiento más disruptivas prescriben el realizar un mantenimiento predictivo de los componentes. Esta metodología se basa en la recopilación de datos de la máquina y en su posterior análisis para detectar; pero, sobre todo, predecir cuándo y cómo se va a dar el siguiente fallo.

Este tipo de metodologías se ven reforzadas por la inminente (incluso quizás ya presente) digitalización total de la industria: la denominada industria 4.0. Gracias a la implementación de sensores y otros dispositivos de toma de datos que parametrizan de manera continua el comportamiento de las máquinas, se es capaz de recopilar cantidades ingentes de información a la cual se puede acceder de forma inmediata. Asimismo, las opciones de interconectividad que ofrecen los elementos de medida con los elementos de control abren la posibilidad de realizar correcciones a tiempo real que eviten averías o paradas inminentes de la máquina. En definitiva, el campo de estudio y aplicación de estos nuevos sistemas es infinito y muy prometedor.

De todos modos, de poco sirve la toma continua de datos, la interconectividad total de la máquina herramienta o la implementación de numerosos y precisos sistemas de medida si no existe un modelo predictivo o preventivo detrás debidamente ajustado que sea capaz de poner el valor toda la información de una manera interpretable por el operario o usuario final. Además, no se ha de descartar la posibilidad de que no sea un único modelo con el que se predigan las rupturas o la vida útil de los componentes, sino que es de gran conveniencia que se entrenen modelos de distintas naturalezas con el fin de comparar entre sí los resultados obtenidos, para poder predecir de mejor manera escenarios de distinta naturaleza.

Pese a que la vida útil y las garantías de los componentes más cruciales (tales como los motores de accionamiento, sistemas de transmisión, electro husillos etcétera) de una máquina herramienta cumplan con las exigencias suficientes como para no causar largas paradas, muchas de las veces son los componentes de los sistemas auxiliares los que causan el mayor número de imprevistos en la operación. La categoría de este tipo de componentes comprende a piezas como cilindros guiados, rodamientos, compuertas automáticas o elementos de medida mecánicos, los cuales debido a su bajo coste en comparación con los elementos considerados como principales, su mantenimiento suele estar descrito dentro de un programa de mantenimiento preventivo.



Justamente, el objeto del presente documento radica en realizar un ejemplo de los primeros pasos de implementación de un sistema de mantenimiento predictivo para uno de estos componentes auxiliares: los cilindros neumáticos. Estos componentes (o sus análogos hidráulicos) se encuentran en infinidad de configuraciones de máquinas herramientas y abarcan un innumerable número de funcionalidades, por lo que el estudio es ampliable a un sinfín de máquinas herramientas.

Como puede resultar lógico, la manera de efectuar el estudio de la predicción de fallos de los cilindros neumáticos no se realizará directamente en las costosas y herméticas máquinas comerciales, si no que se deberá plantear un sistema que simule el modo de actuar de dichos componentes. Consecuentemente, el proyecto que describe el documento consistirá en el desarrollo de un banco de ensayos de cilindros neumáticos sobre los cuales se realizarán los pertinentes ensayos de ciclo de vida. El banco no es más que una estructura metálica que sirve de soporte para una disposición de varios cilindros en paralelo similares a los que se emplean en las máquinas de Danobat. Sobre ellos se dispondrán varios sensores de posición, caudal y presión que servirán para monitorizar el comportamiento de los cilindros.

Puesto que el montaje del nombrado banco de ensayos de cilindros (en adelante, BEC) excede de la carga de trabajo de un proyecto académico y; además, el autor no cree conveniente una explicación exhaustiva del mismo, el presente documento centrará los esfuerzos en las tareas referidas a la puesta en marcha (control) así como esbozará algunas directivas acerca de la utilización de los datos que se obtengan del BEC. Asimismo, por no corresponderse con las competencias adquiridas en el máster en el que se evalúa el presente escrito, no se entrará en demasiado detalle en lo referido a las técnicas de depuración y tratamiento de datos y de inteligencia artificial que se empleen en las últimas etapas del proyecto, aunque sí se mencionen generalidades y resultados de la aplicación de estas técnicas. No obstante, se pone a disposición del lector [1] el trabajo de fin de máster, del mismo autor, en el que se esclarecen las mismas.

Por tanto; y muy resumidamente, en lo referido a este escrito se describirá una puesta en marcha del banco de ensayos. En grandes rasgos, se describirán los componentes empleados, los métodos de control que actúan sobre el movimiento de los cilindros, el plan de pruebas que se llevará a cabo y el flujo de trabajo y dispositivos que conllevan la generación de datos.

2. Contexto

La motivación del estudio de la vida útil de los cilindros neumáticos viene de la problemática causada por las averías en los sistemas de apertura y cierre de las trampillas que se encuentran en algunas de las configuraciones de las rectificadoras verticales VG de Danobat; más concretamente, en los sistemas de cambio de herramienta automáticos (ATC) de las mismas.



Figura 1. Trampilla de estudio (parte del sistema ATC).

En los ATC de estas máquinas se encuentra la trampilla que funciona como elemento intermediario entre el almacén de herramientas y el volumen de trabajo, donde se encuentra la bancada sobre la que se apoya la pieza a mecanizar. Tal y como se alcanza a ver en la **Figura 1**, la trampilla abre y cierra el paso al brazo robótico (a la dcha. de la imagen) que es el encargado de acceder al portaherramientas y realizar el cambio pertinente. Una vez descrita la funcionalidad de la trampilla, queda en evidencia que, pese a no ser funcionalmente imprescindible, el correcto funcionamiento del mecanismo cumple la función de proporcionar una mayor seguridad, pero sobre todo hermeticidad, al conjunto máquina-herramienta.

Para poder fijar el caso de estudio, y sobre todo conocer qué funciones es capaz de realizar la máquina de estudio, se ha decidido comenzar por presentar el esquema organizativo de Danobatgroup, grupo en cuya pertenencia recae la máquina en la cual se sitúa el componente de estudio.

2.1. DANOBATGROUP

Es un grupo industrial cuya creación data del año 1983 [1] y surge para dar una respuesta conjunta de varias de las empresas emergentes del alto y bajo Deba a la demanda regional de la máquina herramienta. Hoy día, el grupo está compuesto por 4 empresas (además de varias filiales), entre las cuales destacan Soraluze y Danobat, cuyos mercados en los que operan exceden del estatal. Además de desarrollar y proveer al cliente de máquinas herramienta, desarrollan líneas llave en mano y soluciones para aplicaciones específicas destinadas a sectores de alta tecnología que dan respuesta a demandas con requisitos técnicos elevados.

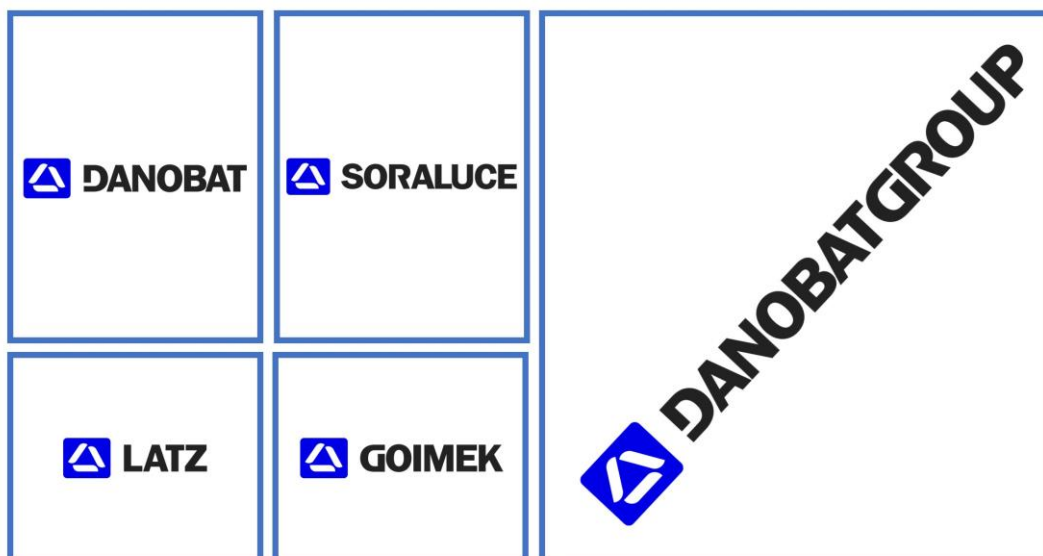


Figura 2. Esquema corporativo de DANOBATGROUP.



Junto con Danobat, Soraluze es la empresa de mayor peso de la organización. Soraluze proporciona “soluciones de mecanizado de vanguardia” [2] y se centra en operaciones de mecanizado complementarias a las que se especializa Danobat, tales como fresado, mandrinado y torneado (principalmente torneado vertical). De la empresa se destaca su amplia gama de productos, soluciones y servicios que proporcionan al mercado, puesto que ofrece desde máquinas individuales hasta líneas de fabricación completas en las que pueden verse incorporadas todo tipo de tecnologías emergentes entre las que se destacan las máquinas multifunción, los sistemas automatizados o centros de mecanizado que incorporan en su ser los denominados “Packs Digitales Soraluze”.

Goimek, por su parte, centra su trabajo en realizar los mecanizados y/o los estudios metrológicos completamente a medida a los requisitos del cliente. Ellos se describen a sí mismos como un “taller de mecanizado que ofrece soluciones integrales” [3]. Poseen dos plantas productivas sobre las cuales se reparten los distintos trabajos de mecanizado, según el tamaño de las piezas a mecanizar y de la precisión que se requiera en la pieza final.

Latz centra su línea de producción en las herramientas de corte. Su catálogo es tan extenso como los procesos que se puedan encontrar en sector de la máquina herramienta, puesto que ofrecen herramientas para realizar taladrado, escariado, fresado, roscado, ... En definitiva, un sinfín de operaciones.

Danobat es otro de los grandes nombres asociados al grupo y se considera como el “líder en el mercado de máquina herramienta de la fabricación de rectificadoras” [4]. No obstante, su mercado no solo se centra el proceso de rectificado, si no que, además, junto con Soraluze, son fabricantes de varios modelos de tornos de alta precisión. Aquí se haya la primera divergencia en la gama de productos de Danobat, puesto que la gama se distingue claramente entre tornos y rectificadoras.

Cabe recordar que los componentes de estudio forman parte de una rectificadora y con el fin de conocer las distintas máquinas de rectificado que ofrece Danobat, se ha decidido realizar un resumen de su catálogo en lo que a esta operación de mecanizado respecta. Dicho resumen se encuentra adjunto a este escrito en el **ANEXO II: RESUMEN DE OPERACIONES DE MECANIZADO DE LAS RECTIFICADORAS DE DANOBAT.**

2.2. Rectificadora vertical VG

Una vez se tiene una visión general del catálogo de las rectificadoras de DANOBAT, resulta de una más sencilla comprensión el tipo de función que cumple la gama a la cual pertenece la máquina de estudio. Comenzando por sus siglas (VG), se deduce que estas guardan el significado de la operación de mecanizado de la máquina herramienta en inglés (*Vertical Grinder*). Además, este hecho nos permite diferenciar la gama con su análoga competidora: las rectificadoras verticales DVG (*Double Vertical Grinder*). Esta segunda opción que se encuentra en los catálogos de DANOBAT, dispone de dos columnas de rectificado, lo cual posibilita el rectificado de componentes de mayor tamaño.

La gama en cuestión ha sido diseñada desde su concepción con un esquema modular en mente, para cubrir las necesidades individuales de cada cliente, de manera que el producto ofrecido es versátil y ampliable desde una configuración base. Es por este motivo por el cual no en todos los modelos adquiridos por los clientes se puede hallar el sistema ATC del cual forma parte el cilindro al cual se le va a realizar el estudio de ciclo de vida.

Algunas características globales que definen a esta gama de productos son:

- Su fabricación en hierro fundido.
- El accionamiento directo de los ejes (empleo de electro husillos).
- Su sistema de refrigeración activo de circuito cerrado.
- Reglas ópticas incorporadas en cada uno de los ejes de la máquina.
- Ampliable/Configurable (Citado anteriormente).

La gama VG está compuesta por 5 modelos que toman el nombre de VG-X, donde la X determinará el tamaño de piezas que la máquina es capaz de rectificar (a mayor valor de X, mayor será el tamaño de pieza rectificable).



Tabla 1. Resumen de productos de la gama VG.

	VG-600	VG-800	VG-1000	VG-1500	VG-1700
Diámetro máximo rectificable	600 mm	800 mm	1000 mm	1500 mm	1700 mm
Altura rectificable	600 mm	600 mm	600 mm	700/1000 mm	700/1000 mm
Carga máx. soportada	1.000 kg	2.000 kg	3.000 kg	4.000 kg	4.000 kg
Núm. de herramientas	4+1 ¹	4+1 ¹	4+1 ¹	4+1 ¹	4+1 ¹

Los campos de aplicación de este tipo de máquinas suelen estar restringidos a las últimas operaciones de mecanizado de las piezas y suelen realizarse sobre piezas cuyas prestaciones han de ser del más alto nivel, en las cuales las rugosidades han de ser mínimas (del orden de micras). Es por ello, por lo que no resulta extraño que se empleen en piezas tales como: discos de turbina, ejes de rotor, ejes de tren de aterrizaje, estatores o rodamientos de grandes dimensiones.

¹ Brazo de medición

2.3. Cilindros neumáticos

Debido a que se trata del elemento principal de estudio, se ha decidido realizar una pequeña introducción con el fin de recoger las principales características de estos componentes. Toda la información técnica básica alrededor de los cilindros (dimensiones básicas, indicaciones de montaje y accesorios que se puedan añadir a los cilindros) puede encontrarse en la norma ISO 15552:2018. Al tratarse de un producto cuyas características están normalizadas, en el mercado existe una amplia oferta de fabricantes y a modo experimental, aprovechando la oportunidad que brinda el hecho de realizar el montaje de BEC desde un comienzo, se ha decidido realizar un montaje de dos marcas de cilindros (SMC y Festo) para comprobar si se pueden observar diferencias significativas entre el comportamiento de estos. Para conocer más acerca de la disposición, métodos de anclaje y control, se puede consultar el Descripción de la solución propuestas, en el cual se describe en mayor detalle la propuesta final.

A modo de resumen, se desea indicar que se instalarán 4 cilindros (todos del mismo diámetro, pero de distintas longitudes) en el BEC: dos de la marca SMC y otros dos de la marca Festo. Sirvan las siguientes tablas como resumen que recoge las características principales de cada uno.

Tabla 2. Resumen de los cilindros a instalar.

	DSBC-63-1150-PPSA-N3	DSBC-63-1400-PPSA-N3	CP96SDB63-1150C	CP96SDB63-1400C
Diámetro	63 mm	63 mm	63 mm	63 mm
Carrera	1.150 mm	1.400 mm	1.400 mm	1.150 mm
Marca	Festo	Festo	SMC	SMC



Con objeto de familiarizarse con las particularidades de los cilindros, resulta conveniente destripar el significado de las siglas y códigos que se encuentran en la nomenclatura de los productos. En el caso de Festo, el desglose de la nomenclatura es tal y como sigue [5]:

- **DSBC**: Nombre que Festo adjudica al diseño de los cilindros. El cilindro es normalizado y de doble efecto.
- **63**: Tamaño (en mm) del diámetro sobre el cual desliza el émbolo.
- **1150/1400/...**: Longitud de la carrera máxima realizable por el émbolo.
- **PPSA**: Indica que el producto integra una solución de amortiguación hermética autorregulable a ambos lados.
- **N3**: Hace referencia a que el modelo sigue las indicaciones adscritas a la norma ISO 15552.

En el caso de la marca SMC, sigue la siguiente nomenclatura [6]:

- **CP96S(D)**: Designación que SMC adjudica al diseño de los cilindros que siguen el estándar ISO 15552. La letra D indica que el cilindro posee la funcionalidad de *auto switch* montada de serie, la cual permite detectar si el émbolo se encuentra en alguno de los finales de carrera.
- **B**: Se refiere al tipo de anclaje con el que se corresponde el cilindro. En el caso estudiado, es un montaje básico (mediante una escuadra en L).
- **63**: Tamaño (en mm) del diámetro sobre el cual desliza el émbolo.
- **1150/1400/...**: Longitud de la carrera máxima realizable por el émbolo.
- **C**: Indicativo de que el cilindro posee un acolchado doble (por aire y goma) en ambos lados.

A continuación, se muestran las imágenes de cilindros similares² a los que han sido testeados en el BEC en la **Figura 3**: Cilindro DSBC-63-500-PPSA-N3. y en la **Figura 4**: Cilindro CP96SDB63-500C..



Figura 3: Cilindro DSBC-63-500-PPSA-N3.



Figura 4: Cilindro CP96SDB63-500C.

2.4. Sistema de Control (VTEM)

Se trata de unos de los componentes clave de la instalación, no solo por su versatilidad, sino también por la novedad tecnológica que ofrece al usuario. A grandes rasgos, el VTEM es un conjunto de elementos de control (tanto de software como de hardware) que ofrecen al usuario la posibilidad de controles neumáticos sofisticados, con la posibilidad de programación mediante aplicaciones intuitivas. Estas aplicaciones son las que Festo ha denominado como Motion Apps (MA).

Pese a que se cree conveniente describir todas las novedades que aporta la solución de servopilotaje mediante válvulas piezoeléctricas que propone Festo en el **Introducción**; seguidamente, se realizará una breve presentación de este conjunto clave en la instalación.

Según lo define la propia empresa proveedora [7], el Motion Terminal VTEM es una solución innovadora para sistemas neumáticos que ofrece ventajas frente a la competencia en el nivel de control que es capaz de ofrecer al usuario. El sistema, cuya ilustración se muestra seguidamente, ofrece ventajas como: funciones de regulación de presión y/o caudal para cada una de las válvulas, funciones de

² La no correspondencia de las imágenes con los componentes reales es la debida a la variación de longitud que existe entre las imágenes y los componentes montados en el BEC.



comportamiento modificable, detección de fugas, etcétera. En definitiva, se trata de una extensión en cuanto a funcionalidad y configuración de las tan ampliamente empleadas electroválvulas neumáticas. Pese a existir un sinfín de configuraciones del VTEM, hecho que se respalda en que las necesidades de cada instalación varían notablemente, en el presente documento se tiene la configuración que se muestra en la **Figura 5**. Teniendo dicha imagen como referencia, se pide al lector que diferencie 5 secciones (de izquierda a derecha):

- En el lado de más a la izquierda se tiene el nodo de bus de campo EtherCAT, encargado de interconectar todos aquellos dispositivos que empleen dicho protocolo. En este montaje en concreto, son dos los dispositivos que emplean este protocolo: el conjunto de HMI y PLC, al cual Festo denomina CDPX y el sistema de automatización CPX-AP-I (sistema I/O remoto), del cual hasta el momento de redacción no se ha hecho uso.
- A su derecha, se tienen dos módulos de entradas analógicas idénticos. Estos módulos serán los empleados para conectar los sensores de medición de cada uno de los cilindros.
- Seguido a los módulos de entrada analógica se encuentra el controlador del *Motion Terminal* (no confundir con el controlador de nivel superior que se encuentra junto con el HMI).
- Junto al controlador es donde se encuentran las válvulas piezoeléctricas programables. En el caso de estudio se tienen 4 válvulas idénticas que se encargarán de distribuir adecuadamente la presión a los cilindros de ensayo.
- Por último, se tienen los espacios reservados a otros módulos de entrada. A diferencia de los que ya han sido mencionados, estos módulos de entrada son los empleados para el manejo de ciertas MA. En caso de que no se empleen todos los módulos de los que se dispone (como es el caso) es posible encajar una placa ciega en su lugar.



Figura 5. VTEM instalado en el BEC.

Tómese estos apuntes de la configuración del VTEM como una pequeña introducción a los componentes que han sido empleados en el proyecto. Para un mayor entendimiento del esquema de los componentes empleados, así como de la función que cumple cada uno de ellos y las interacciones que se dan, se aconseja al lector 8.1.

Se desea incidir en que al referenciar el VTEM no se está únicamente refiriendo a las válvulas programables, sino que también a todos los distintos módulos de entrada y salida que hacen posible la interconectividad de los distintos protocolos empleados en el ámbito industrial (tales como EtherCAT, TCP/IP o Profinet entre otros), así como se hace referencia a todos los componentes que sirven de enlace neumático entre el filtro de aire y las cavidades de los cilindros.

2.5. Captación de datos (Savvy)

Una de las piedras angulares en las que se fundamenta la línea de investigación es el tratamiento de los datos que se obtienen del banco de ensayos. Siempre con el objetivo de implementar una solución fundamentada en la aplicación de técnicas de mantenimiento preventivo, es necesario una correcta creación y almacenamiento de los parámetros. Por ello, se hace imprescindible dar con una infraestructura y un sistema capaz de procesar de manera rápida y eficaz los datos que se quieran manipular del BEC.



En este aspecto es donde entra en juego la bien establecida relación del grupo DANOBAT con la empresa *Savvy Data Systems* (Savvy de ahora en adelante). Savvy ofrece “soluciones industriales de transformación digital a empresas y está especializada en servicios avanzados de monitorización, análisis de datos e inteligencia artificial” [8]. Para ello, ofrece a sus clientes acceso tanto a su plataforma Savvy Industrial Cloud como a su dispositivo *edge* Savvy Smart Box de los cuales se hará una breve introducción en los próximos párrafos.

2.5.1.Savvy Smart Box

Se trata del aparato a instalar en las máquinas sobre las que se desea realizar el análisis. En otras palabras más técnicas, se trata del dispositivo *edge computing*. En resumidas cuentas, este dispositivo aporta una capacidad de computación extra a la máquina, o máquinas, que se quieran analizar.

Internamente, incorpora consigo la opción de captura de datos, así como la posibilidad de ejecución de algoritmos y la transformación de variables en tiempo real. No obstante, las herramientas más concernientes al trabajo en cuestión son las que guardan relación con el almacenamiento y gestión de datos y; por ello, son las que se expondrán con más detalle en apartados posteriores.

Para acceder a estos dispositivos Savvy desarrolla su propio entorno *cloud*. En él, sus funcionalidades permiten al usuario orquestar todos los dispositivos que tenga en su pertenencia y acceder así a la información que se requiera. Este entorno *cloud* pone a la disposición del usuario una gran cantidad funcionalidades entre las que se encuentran la gestión remota de activos (de gran importancia para el trabajo realizado), despliegue de algoritmos en máquina, control de versiones y más.

La mayor ventaja de la implementación de estos aparatos en cualquier tipo de máquina (y en concreto, la del estudio) viene dada por la capacidad de computación *edge* que proveen las *Savvy Smart Boxes* sin necesidad de delegar la computación a servidores conectados a través de internet. No obstante, se desea declarar que la elección de implementar dicho dispositivo y captura de datos en el BEC viene fuertemente impulsado por la relación del grupo Danobat con la más recientemente fundada Savvy.



Figura 6. Dispositivo Savvy Smart Box instalado en el BEC.

2.5.2. Servicio *cloud* de Savvy

Como ya ha sido comentado recientemente, además del dispositivo *edge* que ofrece Savvy a sus clientes, la empresa dispone de servidores en los que se almacena y gestiona toda la información recogida por las *Savvy Smart Box*. Todo lo vinculado con estos servidores, desde la visualización de los parámetros de las máquinas hasta el posible despliegue de los algoritmos predictores, es a lo que la empresa denomina entorno *cloud*.

Puesto que a lo largo del proyecto se empleará en repetidas ocasiones algunas de las herramientas desarrolladas por la propia Savvy para acceder y realizar unas rápidas visualizaciones de los datos que se obtengan del BEC, seguidamente se realizará una pequeña introducción de lo que herramientas del entorno *cloud* que se emplearán más adelante.

Para acceder a las máquinas, y a sus correspondientes datos, existen dos portales online. Por un lado, existe un portal ciertamente más restringido en cuanto a la manipulación de la *Smart Box* se trata, mucho más centrado a la visualización. Como se puede observar en la **Figura 7**, en la pantalla principal de este portal se encuentran todos aquellos dispositivos *edge* vinculados al usuario con el cual se accede.

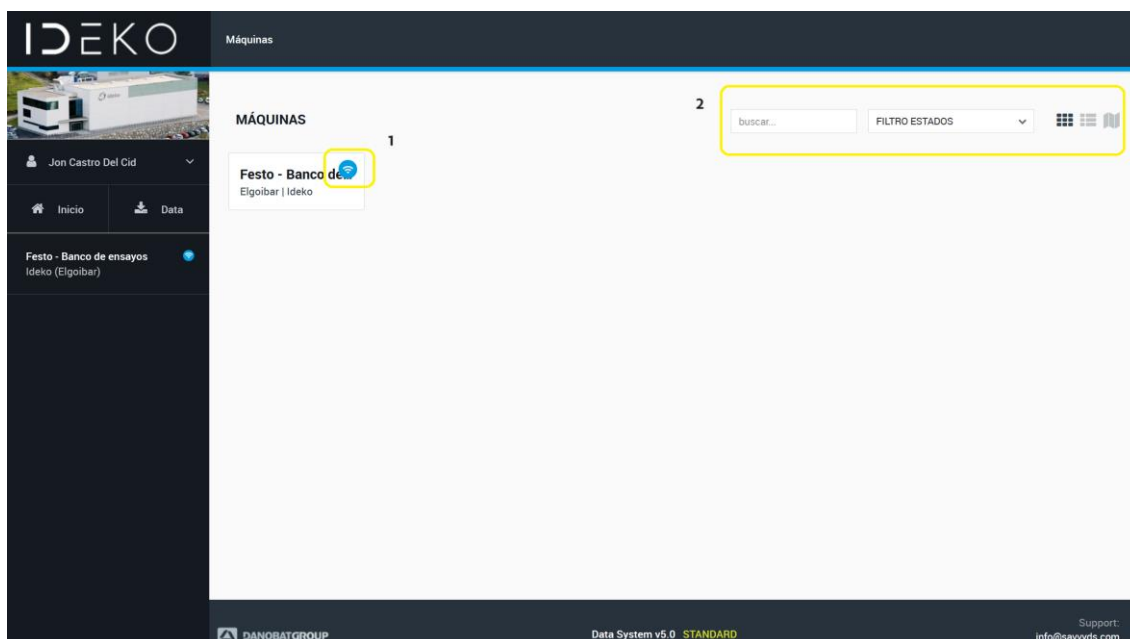


Figura 7. Portal de visualización de Savvy cloud.

De un primer vistazo, esta GUI permite conocer el estado de todas las máquinas³, además de que facilita un filtrado de máquinas por estado. Una vez el usuario seleccione la máquina, podrá acceder a todas aquellas visualizaciones que se hayan desarrollado para esta. En su debido momento, se expondrán al lector las visualizaciones que se han creído convenientes a lo largo del trabajo, pero a modo de avance, y de forma que pueda ayudar a la comprensión del lector, se muestra un pantallazo de una de las visualizaciones que ofrece este portal por defecto. Se desea anotar que las variables que se pueden observar en la **Figura 8** son las variables reales con las que se trabajará más adelante.

³ Azul: El captador (*Savvy Smart Box*) se encuentra conectado (operativo). Ámbar: captador conectado con errores. Gris: Captador desconectado.

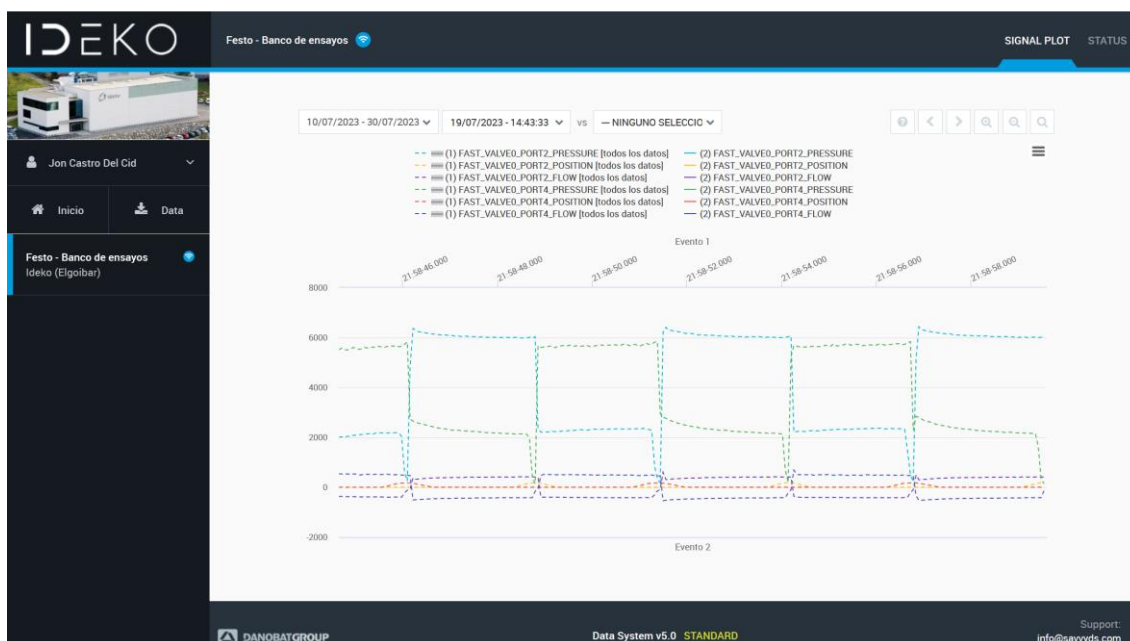
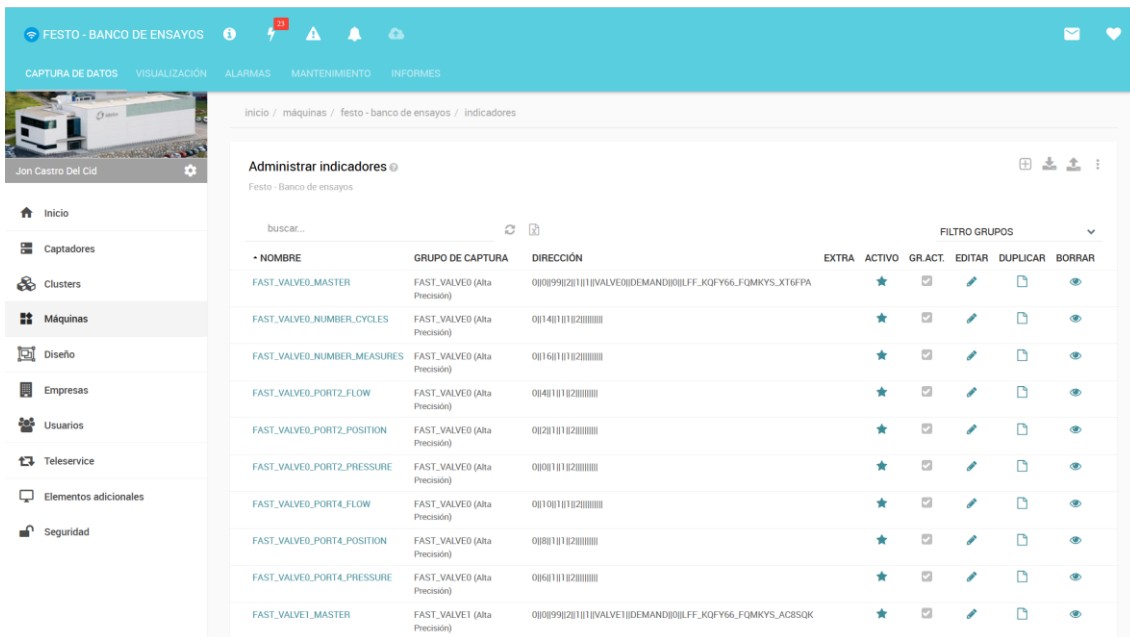


Figura 8. Gráfica por defecto del portal de visualización.

Por otro lado, se encuentra el portal de gestión (denominado de esta manera por la empresa Savvy). Las posibilidades que este portal ofrece al usuario son infinitamente mayores a las del portal de visualización. Este hecho se puede comprobar de un rápido vistazo con la GUI del portal.

Como se puede ver en la **Figura 9** existe muchas más opciones, desplegables y botones en pantalla. Este portal será el empleado en la mayor parte del estudio, puesto que permite acceder a todos los datos obtenidos⁴ a lo largo de la vida del captador y permite la creación de pantallas de visualización personalizadas por el usuario, entre otras muchas más funcionalidades.

⁴ Realmente, es una herramienta útil para acceder manualmente a los datos, pero como se desea automatizar dicha captura, la pantalla de gestión se empleará mayoritariamente para la creación de indicadores y la visualización.



The screenshot shows a web interface for managing indicators. The main content is a table with the following data:

NOMBRE	GRUPO DE CAPTURA	DIRECCIÓN	EXTRA	ACTIVO	GR.ACT.	EDITAR	DUPLICAR	BORRAR
FAST_VALVE0_MASTER	FAST_VALVE0 (Alta Precisión)	010099 20 1 1 VALVE0 DEMAND 0 LFF_KGFY66_FQMKYS_X16FPA		★	☑	✎	📄	👁
FAST_VALVE0_NUMBER_CYCLES	FAST_VALVE0 (Alta Precisión)	01140 1 1 2		★	☑	✎	📄	👁
FAST_VALVE0_NUMBER_MEASURES	FAST_VALVE0 (Alta Precisión)	01160 1 1 2		★	☑	✎	📄	👁
FAST_VALVE0_PORT2_FLOW	FAST_VALVE0 (Alta Precisión)	0140 1 1 2		★	☑	✎	📄	👁
FAST_VALVE0_PORT2_POSITION	FAST_VALVE0 (Alta Precisión)	0120 1 1 2		★	☑	✎	📄	👁
FAST_VALVE0_PORT2_PRESSURE	FAST_VALVE0 (Alta Precisión)	0100 1 1 2		★	☑	✎	📄	👁
FAST_VALVE0_PORT4_FLOW	FAST_VALVE0 (Alta Precisión)	01100 1 1 2		★	☑	✎	📄	👁
FAST_VALVE0_PORT4_POSITION	FAST_VALVE0 (Alta Precisión)	0100 1 1 2		★	☑	✎	📄	👁
FAST_VALVE0_PORT4_PRESSURE	FAST_VALVE0 (Alta Precisión)	0100 1 1 2		★	☑	✎	📄	👁
FAST_VALVE1_MASTER	FAST_VALVE1 (Alta Precisión)	010099 20 1 1 VALVE1 DEMAND 0 LFF_KGFY66_FQMKYS_ACBSQK		★	☑	✎	📄	👁

Figura 9. Portal de gestión de Savvy cloud.

A pesar de que quede mucho por informar en lo que respecta a la plataforma IloT que ha creado Savvy, no se cree conveniente desencaminar más el análisis hacia la descripción de la plataforma, debido a su extensión e irrelevancia en algunos aspectos en lo que a la línea de investigación del presente trabajo respecta.

2.5.3.Ciclo de vida del dato

Uno de los aspectos fundamentales del trabajo es cómo se trata el dato desde su concepción hasta su extracción y manejo por la parte del autor. Nuevamente, el dispositivo *edge* Savvy Smart Box juega un papel fundamental. Este dispositivo multifunción desempeña diversas tareas, siendo una de sus responsabilidades principales el acceso a los datos del elemento de control a través de bibliotecas de desarrollo propias. Estos datos son posteriormente enviados a los servidores de Savvy mediante lo que Savvy denomina el protocolo M2C.

Como ya ha sido expuesto, la infraestructura en la nube no solo proporciona un espacio para almacenar datos, sino que también brinda al cliente acceso a una API que permite el desarrollo de aplicaciones personalizadas, entre otras funcionalidades.

La relevancia de esta funcionalidad radica en su conexión directa con el acceso a los datos laborales. Aunque no fue desarrollado por el autor, se ha creado un programa que facilita el acceso a los servidores de Savvy y la transferencia de datos a un repositorio local para su manipulación posterior.

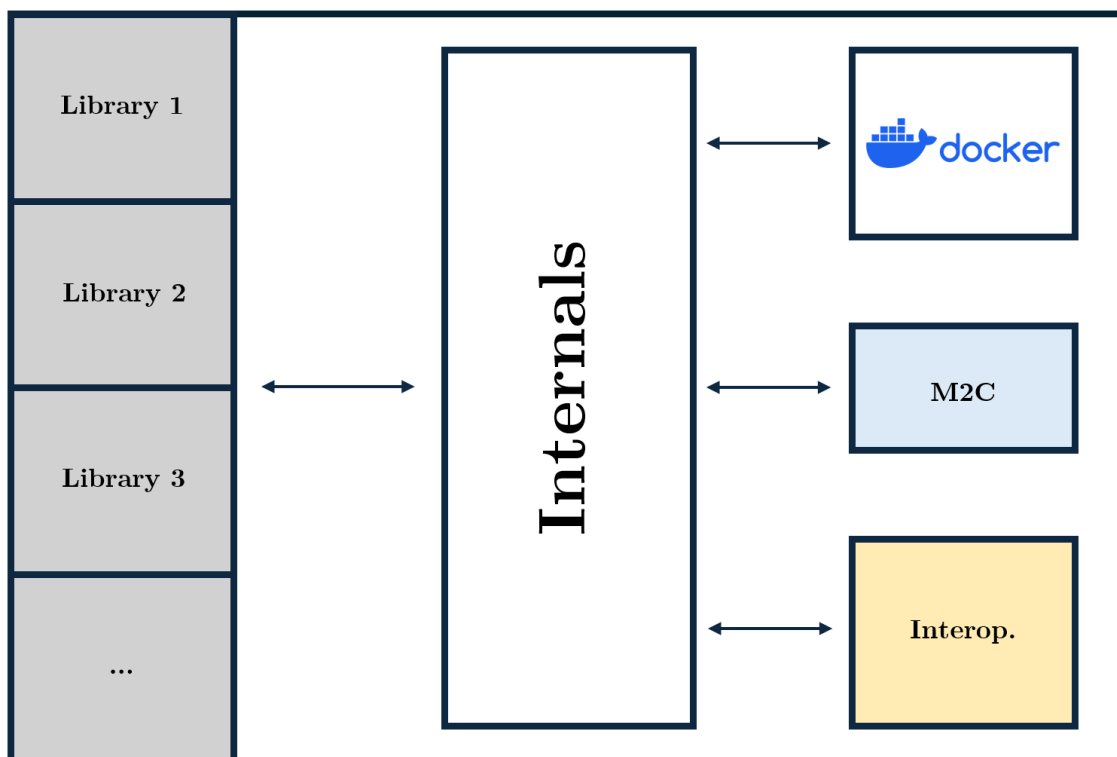


Figura 10. Diagrama conceptual de las Savvy Smart Box.

No obstante, las funcionalidades del dispositivo *edge*, también conocido como captador, van más allá. Al examinar el esquema de la arquitectura del dispositivo en la **Figura 10**. Diagrama conceptual de las Savvy Smart Box., se identifican elementos como las bibliotecas de acceso a los elementos de control y el área dedicada al protocolo M2C.

Además de estas funciones, se reserva un espacio para los contenedores de Docker. Este espacio engloba todas las capacidades proporcionadas por los contenedores utilizados para implementar aplicaciones que puedan ser desarrolladas para la máquina. Aunque no se haya utilizado durante el curso de este trabajo, se considera altamente conveniente en caso de que se lleven a producción los modelos de detección de anomalías que se presentarán en las secciones siguientes.

De manera introductoria, es importante señalar que la tecnología ofrecida por la plataforma Docker empaqueta tanto el código como las dependencias, permitiendo que la aplicación se ejecute de manera rápida, confiable e independiente de otras aplicaciones que puedan ejecutarse simultáneamente en el captador.

Por último, se encuentra el módulo de interoperabilidad *RestStreaming*. Aunque no se haya utilizado a lo largo del trabajo, su potencial radica en la posibilidad de acceder a los datos en tiempo real que la caja es capaz de leer de la máquina a la que se conecta. A modo informativo, se menciona que a este módulo a menudo se le denomina API local. En resumen, se presenta un esquema breve (**Figura 11**. Ciclo de vida del dato.) que muestra el ciclo de vida de los datos en el presente trabajo.

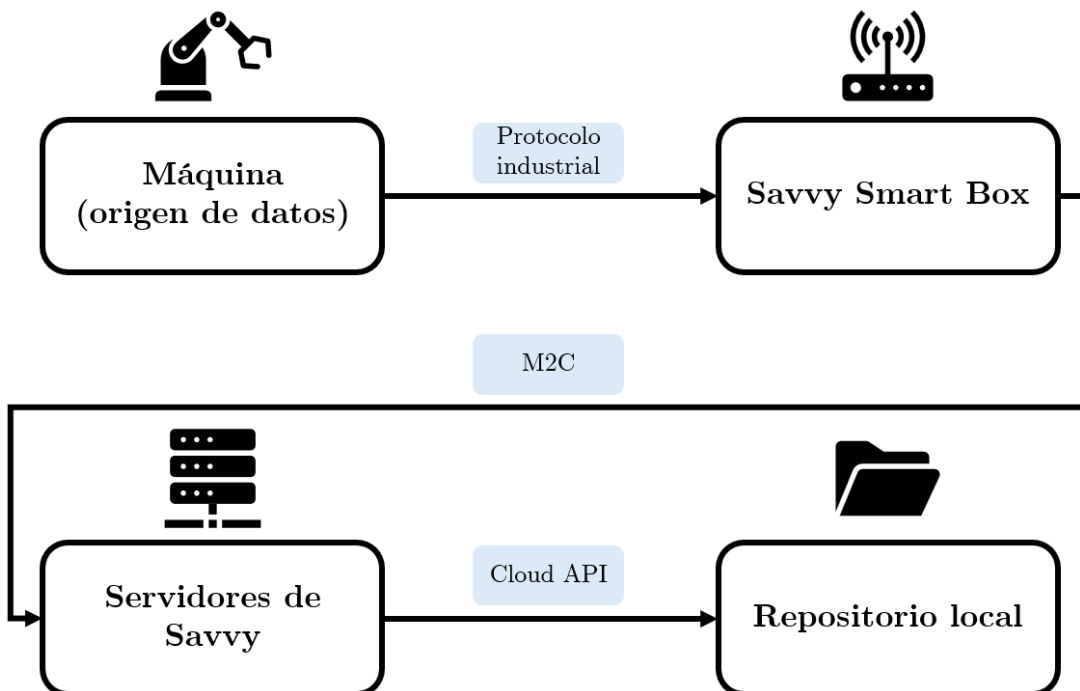


Figura 11. Ciclo de vida del dato.

3. Objetivo y alcance

El objetivo del presente trabajo fin de máster es realizar una puesta en marcha y monitorización de un banco de ensayos de cilindros neumáticos con el fin de estudiar la respuesta al fallo de estos elementos. Tal y como se ha comentado en la introducción, este estudio pertenece a una de las dos partes que componen el estudio de una posible implementación de un sistema de mantenimiento predictivo en un sistema basado en cilindros neumáticos. Por ello, el desglose del objetivo de este trabajo se hará de tal forma que únicamente contemple las tareas imprescindibles para asentar las bases del posterior análisis.

Tómese la siguiente lista de tareas como desglose del objetivo:

- Supervisión del diseño del banco de ensayos.
- Supervisión y recomendación del acopio de materiales.
- Definición del plan de pruebas.
- Coprogramación del software de control del banco.
- Desarrollo de pantallas de monitorización en el portal de visualizaciones.
- Estudio de calidad de los datos.
- Validación de los datos.

La puesta en marcha y monitorización del banco se darán por concluidas una vez que se verifique que los datos obtenidos se correspondan con la realidad, que no existan pérdidas significativas de datos y cuando tanto la frecuencia como la cantidad de información generada por el banco de ensayos sean del agrado de los supervisores del proyecto.

Se quiere recalcar que, que debido a la gran carga de trabajo que supone la puesta en marcha de un banco de ensayos desde cero, existen algunas tareas en las que el autor no participará. Más concretamente, el listado de tareas las cuales exceden el alcance del proyecto son las siguientes:

- Diseño del banco de ensayos.
- Elección y acopio de materiales necesarios.
- Elección del mecanismo de reducción de vida útil de los cilindros.
- Elección de los sensores a instalar en el banco de ensayos.
- Montaje del banco de ensayos.



4. Beneficios

4.1. Beneficios técnicos

El estudio de la respuesta al fallo de elementos pertenecientes a las máquinas de la industria manufacturera es de gran importancia debido a varios motivos técnicos. Primero que todo, una posible implementación de algoritmos que permitan predecir la rotura de los componentes es uno de los desafíos que se está llevando a cabo en la industria y es uno de los primeros pasos para dar con estos algoritmos es el de comprender cuáles son las respuestas al fallo que se dan en estas mismas máquinas.

La aplicación de algoritmos para implementar metodologías de mantenimiento preventivo en la industria permite una detección temprana de posibles fallas en los equipos y maquinaria. Esta capacidad de anticipación posibilita la realización de mantenimiento preventivo en el momento óptimo, evitando así paradas no planificadas y optimizando la programación del mantenimiento.

Al evitar paradas técnicas no programadas, estos algoritmos contribuyen directamente a la mejora de la eficiencia operativa. La reducción del tiempo de inactividad se traduce en una mayor disponibilidad de los activos, lo que impacta positivamente en la productividad y, por ende, en la rentabilidad. Además, la extensión de la vida útil de los activos, facilitada por la intervención oportuna, resulta en un retorno de inversión a largo plazo al minimizar los costos asociados con la adquisición y reemplazo de maquinaria.

Cabe destacar que la seguridad laboral también experimenta mejoras, ya que la detección temprana y resolución de problemas contribuyen a minimizar situaciones de riesgo en el lugar de trabajo. En resumen, la implementación efectiva de algoritmos de mantenimiento preventivo no solo garantiza la confiabilidad y eficiencia de los activos, sino que también impacta positivamente en la seguridad, los costos operativos y la rentabilidad global de la empresa manufacturera.



4.2. Beneficios económicos

A nivel financiero, la implementación de algoritmos de mantenimiento preventivo conlleva ahorros significativos. La reducción de paradas técnicas no planificadas implica menos gastos asociados con reparaciones de emergencia y pérdida de producción. Estos ahorros operativos se traducen en una mejora de la rentabilidad global de la empresa. Además, la eficiencia energética se ve favorecida, ya que los equipos mantenidos en condiciones óptimas operan de manera más eficiente, reduciendo el consumo de energía y los costos asociados.

De acuerdo con una investigación de IBM Analytics [10], las ventajas del mantenimiento predictivo superan considerablemente a los métodos de mantenimiento convencionales, generando un retorno de inversión que es diez veces superior. Según este estudio, la implementación del Mantenimiento Predictivo resulta en una reducción de los costos de mantenimiento en un rango del 20 al 25% más significativa en comparación con otros enfoques de mantenimiento. Además, se logra una disminución de las averías en un intervalo del 70 al 75%, y los tiempos de inactividad se reducen entre un 35 y un 45% más.



5. Descripción de requerimientos

Para llevar a cabo este trabajo se diferencian dos grandes grupos de requerimientos. Por un lado, se detallarán todos los componentes necesarios para la materialización del banco de ensayos y por otro las herramientas ofimáticas y software empleado para la monitorización y análisis del estado del banco.

5.1. Requerimientos materiales

- 4 cilindros normalizados del tipo DSBC siguiendo la ISO 15552.
- 2 perfiles de aluminio que actúan como elemento intermedio entre el suelo y los propios cilindros.
- Escuadras de acero fundido que se colocan como elementos de unión entre los perfiles de aluminio y el suelo. 8 han sido el número de escuadras empleadas para el BEC.
- Escuadras de aluminio que realizan la conexión entre los cilindros y la lámina.
- 4 láminas de aluminio que sustentan juntan a los cilindros con los perfiles.

A modo ilustrativo, se facilita una imagen (**Figura 12.** Elementos de unión de los cilindros: laminas, perfil y escuadras.) en la que se observan los elementos de unión que se han descrito en estos puntos.



Figura 12. Elementos de unión de los cilindros: laminas, perfil y escuadras.

- Conexiones neumáticas:
 - Alimentación de aire a presión. El baremo de la presión ha de ser de entre 6 y 8 bares para cumplir con los requisitos de las normas.
 - Tubos de conexión. Cada uno de los cilindros posee dos orificios que los conectan con la alimentación. Además, el abastecimiento de las válvulas se ha de realizar mediante un cable de conexión de un mayor diámetro.
 - Llaves de paso. Es esencial por lo menos disponer de una llave que sea capaz de cortar el flujo de aire a todo el banco. En este caso particular son dos las llaves que se tienen.
- 8 sensores de proximidad de efecto Hall.
- Juegos de pesas metálicas que hagan de elemento agravante en la degradación del cilindro.
- Motion Terminal VTEM.
- Pantalla CDPX (hace las labores de HDMI y PLC).
- Capturador de datos Savvy Smart Box.
- Cámara de vigilancia de conexión mediante IP.
- Conexiones y alimentaciones para todos los dispositivos electrónicos.



- Elementos de unión tanto para la colocación de las pesas, la fijación de los cilindros o la de los elementos de control como el VTEM. Siempre se ha dado preferencia por la utilización de tornillos allen.

5.2. Requerimientos de software

- Entorno de programación de Python v3.10 (se han empleado tanto Visual Studio como Spyder)
- Paquetes de Python para la creación de modelos neuronales como TensorFlow, PyTorch y paquetes relacionados con el tratamiento de datos como SciPy.
- Notepad++ v8.6.4 para abrir documentos
- Herramientas ofimáticas de Microsoft 365 (Power Point, Excel, Word y Teams)
- CODESYS V3.5 SP18 Patch 4 para la programación en texto estructurado del PLC que gobernaba el accionamiento de las válvulas.
- Designer Studio v4.5.0.224.



6. Estado del arte

Pese a que el primer objetivo del largo proyecto de la implementación del mantenimiento predictivo en cilindros neumáticos doblemente accionados sea la puesta en marcha de un banco de ensayos, no se ha de perder de vista el marco global en el que se encuentra el BEC. Más concretamente, no se ha de olvidar que la etapa primordial por la que ha de pasar el proyecto es el entendimiento del funcionamiento de la pieza clave: el cilindro. Aunque, más que el entendimiento del correcto funcionamiento quizás sea más interesante entender los modos y los caminos que llevan al elemento neumático al fallo.

Como resulta lógico, los cilindros neumáticos están diseñados para ofrecer una alta fiabilidad y una larga vida útil. En múltiples [9] [10] [11] [12] de los estudios en los que se identifica la vida útil de los cilindros neumáticos en condiciones de operación normales⁵ la vida útil de estos supera fácilmente el millón de ciclos (sea un ciclo un vaivén del cilindro). Incluso, no es algo fuera de lo común encontrar ensayos que superan el umbral de los mil millones de ciclos. Se quiere realizar un pequeño inciso en lo que a la unidad de la vida útil de cilindros neumáticos respecta. Puesto que estos elementos muy comúnmente realizan operaciones cíclicas es habitual encontrar el tiempo de funcionamiento del cilindro en unidades de, lógicamente, ciclos [11].

Para dar respuesta a esta problemática tan habitual en el estudio de tiempo hasta el fallo de componentes tan duraderos es por lo que existe una rama de investigación denominada ALT (*Accelerated Life Testing*). Esta rama es la encargada de proponer planes para probar un producto sometiéndolo a condiciones superiores a las que operaría normalmente; ya sea tensión, deformación, temperatura, voltaje, presión, etc. El objetivo de someter a las piezas a estas condiciones es el de descubrir errores de comportamiento y posibles modos de fallo en un periodo muy reducido en el tiempo. La gran problemática que se esconde detrás de estos ensayos es la de proponer planes que no alteren los mecanismos de fallo [9] que se darían en las condiciones normales.

⁵ Se define como condiciones de operación normales a los cilindros accionados que no arrastren carga adicional, en condiciones atmosféricas inalteradas (T^a y presión) y con niveles de presión de activación cercanos a los 6 bares.

6.1. Modos de fallo de los cilindros neumáticos

Es innegable suponer que si se desean conocer los modos de fallo de los cilindros neumáticos será de gran ayuda conocer los componentes que forman el conjunto de estos elementos; para poder así conocer en qué elementos es más probable que se dé la rotura.

Por ello, se muestra a continuación la figura de un cilindro neumático de la marca Festo. Más concretamente se muestra el desglose de un cilindro normalizado del tipo DSBC, ISO 15552.

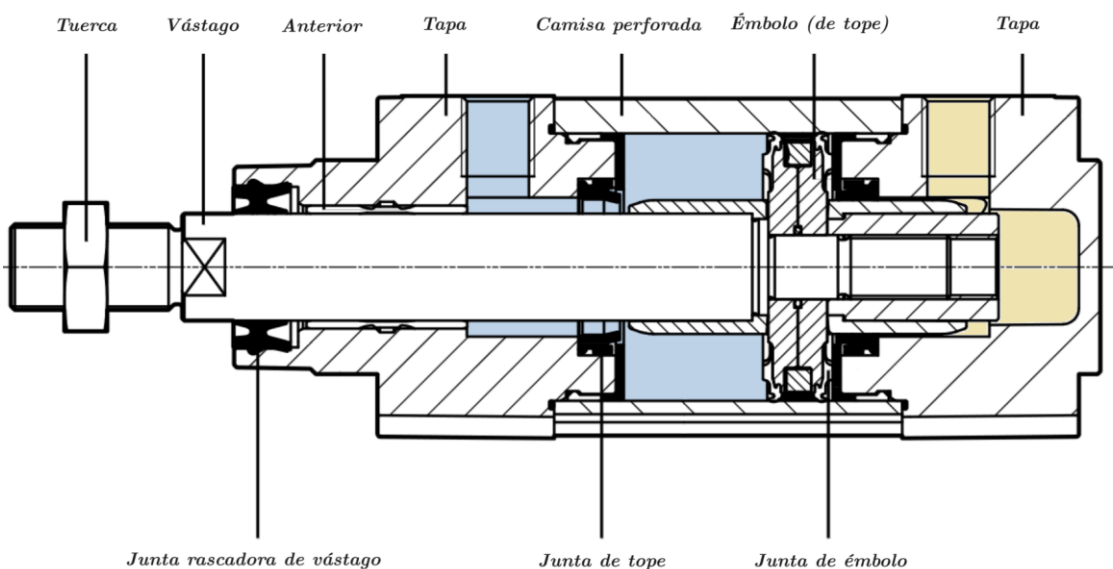


Figura 13. Desglose del cilindro de la marca Festo.

Sin conocimiento previo de los posibles modos de fallo se puede intuir que los elementos que se deben tener en el punto de mira serán las juntas: 3 en el caso de la **Figura 13**. Desglose del cilindro de la marca Festo.. Tanto la junta rascadora de vástago, como la junta de tope como la junta de émbolo se supone que serán los más afectados por la degradación con el paso del tiempo. Para conocer los posibles modos de fallo de los cilindros se recomienda la lectura ***Accelerated Life Test for Reliability Evaluation of Pneumatic Cylinders*** [15]. En este trabajo se diferencian 3 modos de fallo generales con sus consecuentes causantes. De este trabajo es de donde se extrae el FTA que se presenta en la **Figura 14**. FTA de los cilindros neumáticos.

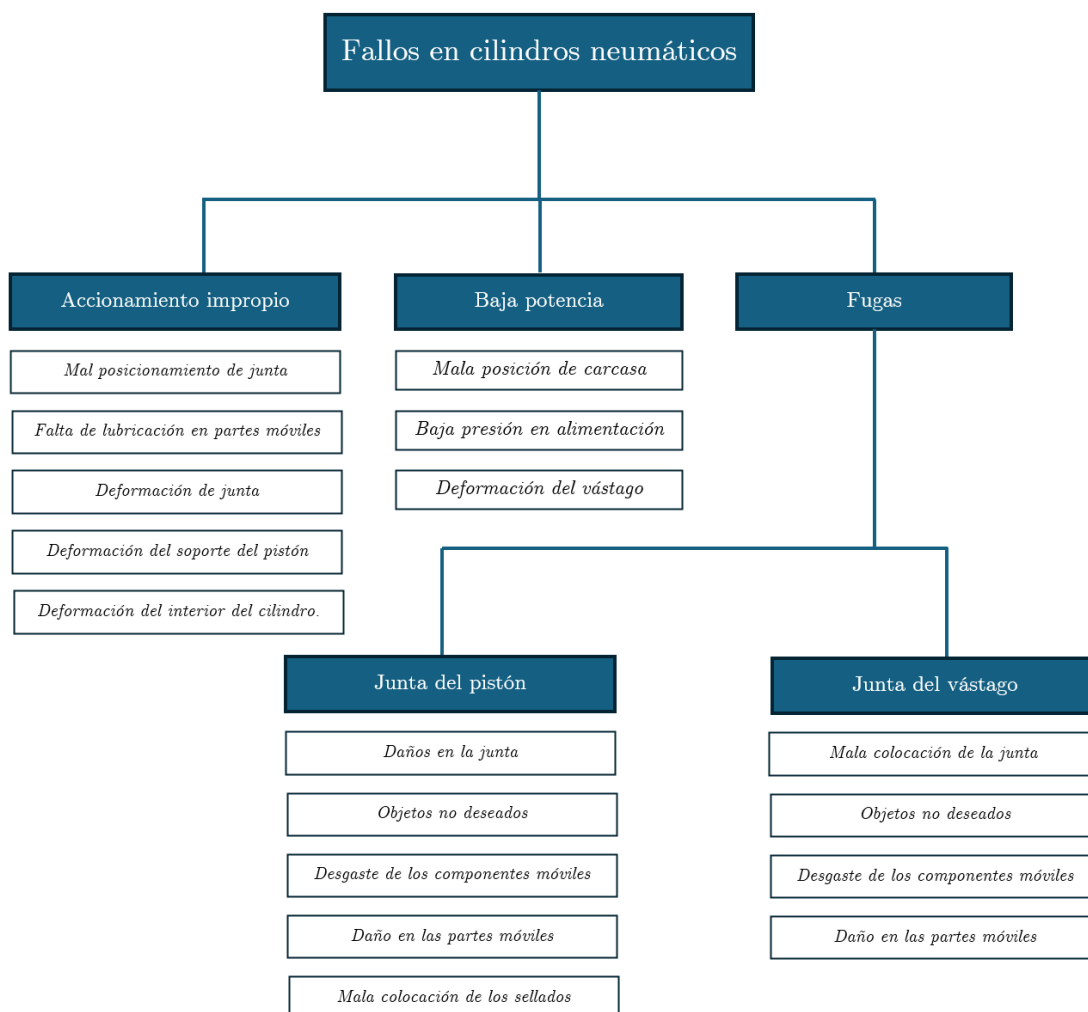


Figura 14. FTA de los cilindros neumáticos.

Con este *fault tree analysis* se vuelve a corroborar lo que se sospechaba al comienzo: las partes móviles están, lógicamente, más involucradas en el fallo. Según este estudio también se puede razonar que los tres principales modos de fallo son el accionamiento impropio, la baja potencia y las fugas de aire. Estos modos de fallo son el resultado de los posibles mecanismos de fallo que ocurren en los cilindros, entre los que se incluyen fatiga, envejecimiento y desgaste. El análisis también identifica al anillo de sellado del pistón como el componente débil y su desgaste como la principal causa de fallo. No obstante, existen varios modos de fallo que han de ser estudiados en un nuevo apartado. El estudio de estos modos de fallo es realmente interesante para poder realizar un plan de aceleración de la vida del cilindro acorde con estos métodos de degradación.

6.1.1. Mecanismo de fallo en los cilindros

En las primeras etapas de degradación, debido a la presencia de asperezas entre las superficies de contacto, se comienzan a dar desprendimientos puntuales de material⁶. Por lo tanto, se produce un desgaste abrasivo de dos cuerpos. Cuando las partículas o residuos producidos tienen libertad para rodar y deslizarse sobre las superficies de contacto, se produce un desgaste abrasivo de tres cuerpos.

Los efectos sinérgicos debidos a la ferocidad del movimiento tienden a elevar la temperatura, lo que da como resultado una película de lubricante menos espesa, que suele concluir en el contacto directo de las piezas móviles. Este contacto directo crea a su vez una mayor cantidad de residuos provenientes del desgaste. Este mecanismo de fallo es el que se ve representado en la **Figura 15**. Mecanismo de fallo por abrasión de la superficie.

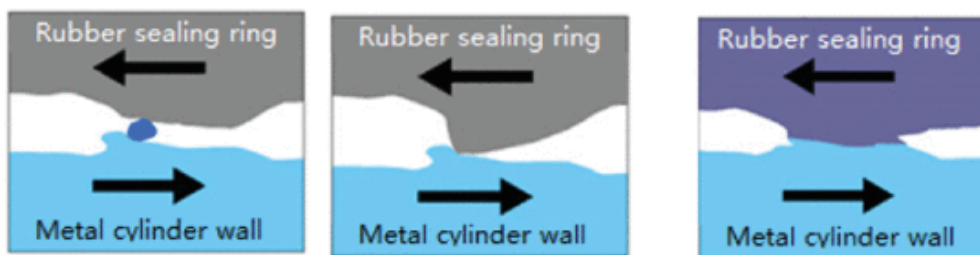


Figura 15. Mecanismo de fallo por abrasión de la superficie.

Otro de los mecanismos de fallo es el que se da también debido a la virulencia del movimiento que se da en el interior de los cilindros. Concretamente, la alta frecuencia del movimiento relativo puede llevar al cuerpo a temperaturas elevadas que provoquen un alto nivel de oxidación en la capa interna del cilindro. La oxidación trae consigo una capa de menor adherencia que puede verse desprendida del cuerpo principal y provocar un mayor material indeseado dentro del cuerpo del cilindro. Este mecanismo de fallo es el que se muestra en la **Figura 16**. Mecanismo de fallo por oxidación.

⁶ Sobre todo, pertenecientes a las juntas.

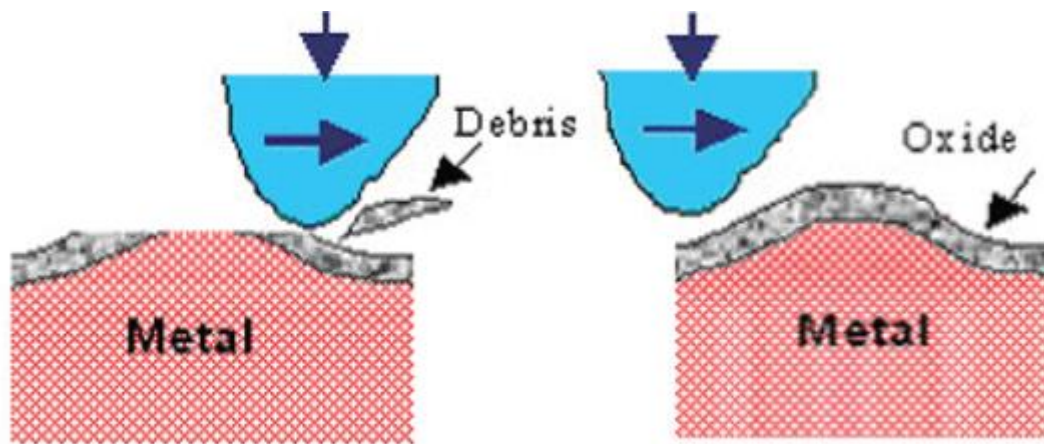


Figura 16. Mecanismo de fallo por oxidación.

No obstante, también puede darse otro mecanismo de fallo. Cuando el cilindro y las partes móviles del cilindro se encuentran dañadas o carecen de grasa, es probable que se produzcan ligeras deformaciones y movimientos no deseados. Entonces, los cilindros fallarían debido a una actuación incorrecta. El daño de las partes móviles del vástago del pistón provoca fácilmente daños en las juntas lo que resulta en una fuga de aire, por lo que los cilindros pierden la capacidad de proporcionar la fuerza de salida requerida.

No obstante, el estudio nos muestra que el mecanismo de fallo que prevalece es el de la abrasión por fricción y es el que se dará en la mayoría de los casos. Por ello, es este mismo mecanismo de fallo el que teóricamente se tendrá que vigilar en las etapas de monitorización en los componentes que se monten en el banco.

6.2. Motion Terminal VTEM de Festo

Tal y como ha sido contextualizado, el Motion Terminal de Festo es la pieza clave de la instalación, puesto que ocupa del control del movimiento de los cilindros. Se trata de una solución novedosa en el ámbito de la neumática en varios aspectos. En primer lugar, las válvulas que se encuentran en el VTEM son válvulas piezoeléctricas, capaces de ser pilotadas electrónicamente. No obstante, el mayor distintivo de este dispositivo es que se trata del primer controlador neumático cuyas válvulas son gobernadas por aplicaciones [14] (las denominadas Motion Apps).

Además de la novedad tecnológica de las aplicaciones, el VTEM permite controlar y monitorea⁷ algunas de las variables más importantes de los sistemas neumáticos como la presión, temperatura y caudal entre otros. Ciertamente, este fue uno de los puntos a favor que llevó a instalar este aparato en el banco de ensayos: la implementación de varios sensores que permitía monitorizar a tiempo real las variables más decisivas del sistema neumático. Este punto es uno de los beneficios que se anuncia en los documentos informativos de Festo [14]; y es que, según la propia marca, "con el VTEM preparas a tu sistema para la industria 4.0". A modo de resumen y de manera esquemática se recogen los atributos más beneficiosos del empleo del novedoso dispositivo de Festo, según queda recogido en la documentación de la propia empresa [15].

Reducción de costes de ingeniería:

- Dado que el comportamiento neumático está vinculado a las MA en lugar de a hardware mecánico, se consigue reducir entre un 25 y 30 % el coste total de la instalación gracias a la disminución de gastos de ingeniería durante la fase de diseño.
- Además, puesto que cualquier modificación futura del comportamiento de la válvula se puede realizar mediante software en vez de hardware, los costes de modificación de proyecto disminuyen notablemente.

⁷ Más adelante, cuando se expongan los datos que se obtienen del BEC, se observará que en ciertas condiciones la calidad del dato deja mucho que desear y que la simultaneidad de los parámetros no siempre se da de manera tan satisfactoria como lo puede requerir el estudio.

Aumento de la productividad:

- Puesto que para cambiar el comportamiento de la máquina no es necesario una modificación física de los componentes, no son necesarias las paradas de las líneas de producción de en las que se instale el VTEM. Este hecho repercute directamente en un aumento de la productividad de los sistemas.

Eliminación de las configuraciones manuales:

- La implementación de algunas tareas en las válvulas convencionales, como por ejemplo el ajuste de las velocidades de desplazamiento de los cilindros, suele ser un proceso que requiere mucho tiempo en los sistemas mecatrónicos convencionales. Este tiempo queda drásticamente (de 3 a 4 minutos por válvula) recortado con soluciones como la del VTEM.

Disminución del tiempo de reparación de errores:

- Gracias al diseño modular de las válvulas del VTEM, se pueden reemplazar (incluso intercambiar) las válvulas entre sí. Este hecho permite acortar en gran medida los tiempos de reparación.

En muy resumidas cuentas, los beneficios de la implementación del VTEM son el resultado de la combinación de los beneficios neumáticos y de automatización eléctrica tal y como se muestra en el siguiente diagrama de venn (**Figura 17. Diagrama de beneficios del VTEM.**).

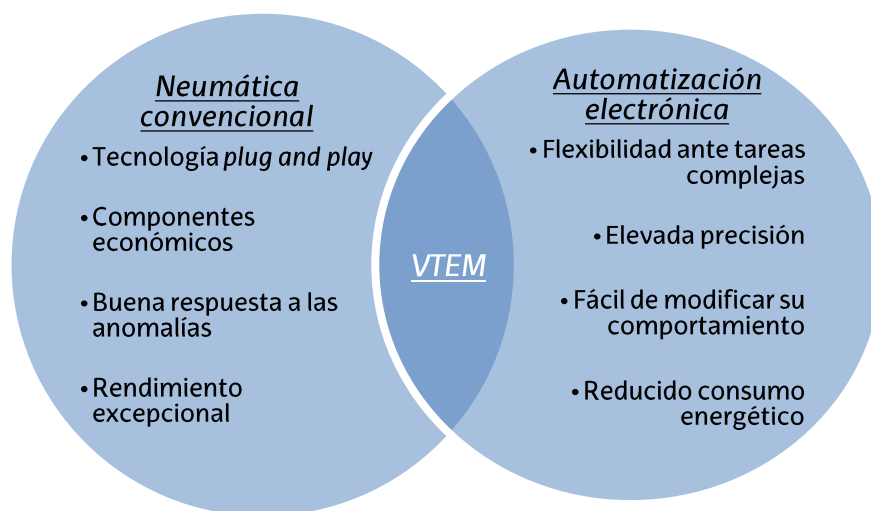



Figura 17. Diagrama de beneficios del VTEM.

6.2.1. Motion App

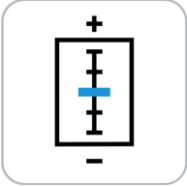


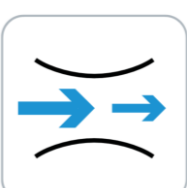


Las Motion App desarrolladas íntegramente por Festo son aplicaciones que permiten comandar al VTEM. Estos desarrollos permiten interactuar con el firmware de tal manera que la programación se pueda realizar de una manera más amable al usuario.


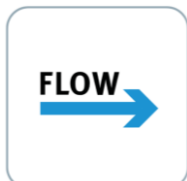
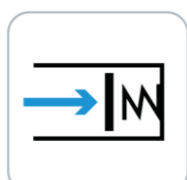
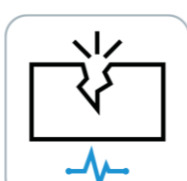

Las MA se pueden descargar desde Festo App World, aunque algunas de estas estén sujetas a licencia y se tenga que realizar un aporte económico extra para activarlas en la unidad de control que se haya adquirido. Las MA permiten controlar la presión, el flujo, la velocidad, la posición y el movimiento de los actuadores neumáticos conectados al Motion Terminal y son únicas en el mercado debido a la facilidad que aportan al usuario con respecto las demás soluciones neumáticas. En el caso del BEC, al ser una compra ordinaria, se disponían de 4 de las 12 MA que se presentan en la **Tabla 3**. Listado y descripción de las MA.

Tabla 3. Listado y descripción de las MA.

	<i>Motion App 01</i>	<i>Directional control valve function</i>	Permite la apertura cierre de los puertos. Operación más sencilla.
---	----------------------	---	--



	<i>Motion App 02</i>	<i>Proportional directional control valve</i>	Permite la operación de la válvula de manera proporcional.
	<i>Motion App 03</i>	<i>Proportional pressure regulation</i>	Permite regular la presión de los puertos 2 y 4.
	<i>Motion App 04</i>	<i>Model-based proportional pressure regulation</i>	Permite regular la presión de los puertos 2 y 4 sin necesidad de definir manualmente parámetros de ajuste.
	<i>Motion App 05</i>	<i>Supply and exhaust airflow control</i>	Permite ajustar el grado de apertura de los puertos.
	<i>Motion App 06</i>	<i>ECO drive</i>	Permite el movimiento de los actuadores neumáticos con presiones más bajas que la de la red de alimentación.
	<i>Motion App 07</i>	<i>Presetting of travel time</i>	Posible control del tiempo de la dinámica del actuador.

	<i>Motion App 08</i>	<i>Selectable pressure level (ECO)</i>	Permite la actuación neumática con niveles de presión más inferiores que la presión nominal.
	<i>Motion App 10</i>	<i>Flow control</i>	Posibilidad de controlar el caudal de los puertos.
	<i>Motion App 11</i>	<i>Soft Stop</i>	Automáticamente ajusta la dinámica de los cilindros para implementar una parada suave en los finales de recorrido.
	<i>Motion App 12</i>	<i>Leakage Diagnosis</i>	Modo predefinido que clasifica las fugas que se puedan dar en los circuitos en base a 3 niveles.
	<i>Motion App 33</i>	<i>Motion profile and positioning</i>	Permite indagar en los ajustes elementales de las dinámicas de movimiento y posicionamiento,

Lamentablemente, el acceso a las licencias de la mayoría de las aplicaciones está precedido por un pago extra que desbloquearía la licencia. A pesar de todo, con cualquiera de las aplicaciones se puede tener acceso a controlar las válvulas mediante el PLC del sistema.

6.2.2. Válvulas piezoeléctricas

Las válvulas piezoeléctricas son una alternativa más moderna a las clásicas electroválvulas. Este tipo de válvulas, las cuales se encuentran en el interior del VTEM, basan su dinámica en el efecto piezoeléctrico, efecto estudiado desde los análisis de los hermanos Pierre y Jacques Curie [17].

La piezoelectricidad se puede definir como la carga eléctrica que se acumula en ciertos materiales en consecuencia de la aplicación de un estrés mecánico. Las aplicaciones modernas de este comportamiento se encuentran en innumerables aparatos, llegando incluso a formar parte en los relojes internos de los procesadores que se pueden encontrar en cualquier chip. Más concretamente, se emplean cristales de cuarzo para para fabricar los osciladores encargados de general la señal de reloj en innumerables dispositivos electrónicos, incluidos los procesadores.

Volviendo al campo de aplicación neumático, el efecto piezoeléctrico abre unas posibilidades de automatización enormes en los sistemas que emplean neumática convencional y para entender la diferencia con las válvulas convencionales empleadas en este ámbito primero se ha decidido realizar una pequeña presentación de estas.

El funcionamiento de las electroválvulas se puede resumir por la acción del campo electromagnético que se induce en la bobina del elemento, el cual magnetiza un émbolo, encargado de abrir y cerrar el paso del fluido de control.



Figura 18. Electroválvula de la marca Tameson.

Es en la caja negra, denominada armadura, de la **Figura 18**. Electroválvula de la marca Tameson. es donde se encuentran los elementos citados. Además, de modo que la válvula tenga un estado de reposo, dentro de la armadura se sitúa un muelle. Dependiendo de si este elemento se encuentra naturalmente contraído o retraído las electroválvulas podrán ser del tipo normalmente cerradas (NC) o normalmente abiertas (NA).

Volviendo a la tecnología piezoeléctrica aplicada en las válvulas del Motion Terminal esta se ve aplicada para la creación de elementos piezoeléctricos que serán la piedra angular de las válvulas piezoeléctricas. El proceso es el que se muestra a continuación.

1. Preparación del material cerámico.

- El primer paso es la laminación de un material cerámico de un espesor mínimo (de unos 150µm). Este material es naturalmente no conductor, por lo que deberá de ser tratado seguidamente para seguir con el proceso.



Figura 19. Material de partida.

2. Añadir la superficie conductora.

- Para poder realizar el proceso de polarización que sigue, es necesario aplicar una capa de material conductor, normalmente en forma de polvo, que permita crear un elemento capacitivo formado por dos placas conductoras de electricidad y el material cerámico que funciona como elemento dieléctrico de la capacitancia.



Figura 20. Adición de superficie conductora.

3. Proceso de polarización.

- Una vez que se haya atentado la superficie conductora se puede dar comienzo al proceso de polarización. Bajo la influencia de fuertes campos eléctricos, es posible polarizar permanentemente los materiales cerámicos que se encuentran entremedio de las placas conductoras. En otras palabras, es posible "alinear" las moléculas del material de modo que se tenga un material con estructura no isotrópica.

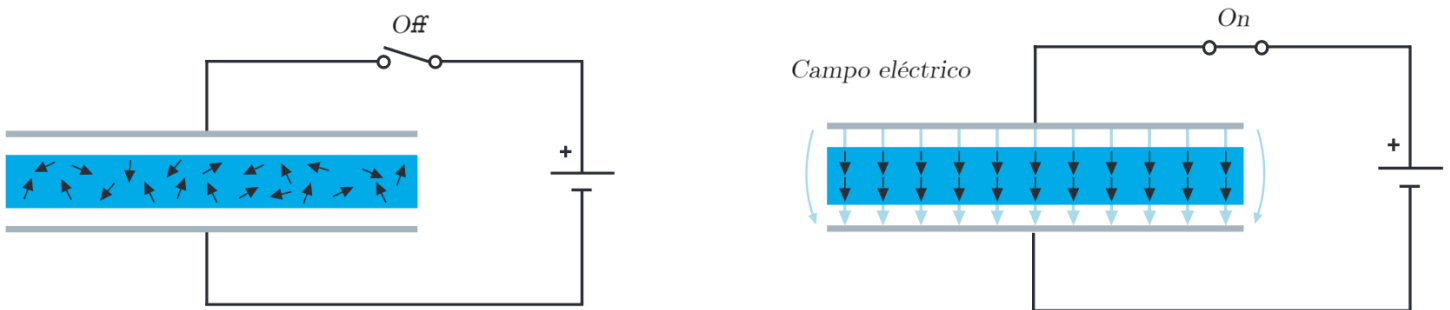


Figura 21. Polarización del material cerámico.

- Como resultado de la aplicación del campo magnético el material cerámico posee propiedades piezoeléctricas y es capaz de cambiar de forma cuando se le aplica un voltaje.

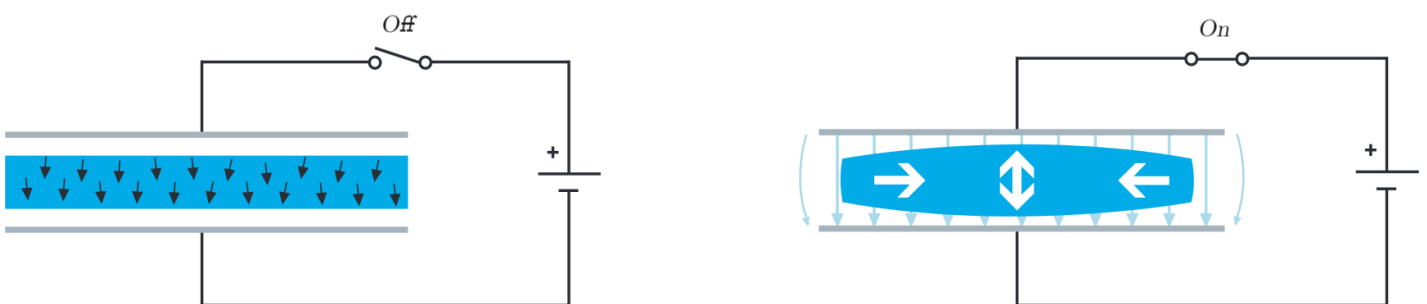


Figura 22. Deformación de componente piezoeléctrico.

- Al aplicársele un voltaje en el material no tardan en aparecer tensiones que siguen las direcciones tridimensionales del campo magnético generado. El resultado es la deformación (contracción) del componente.

4. Unión de la capa conductora.

- El último paso de la fabricación del elemento piezoeléctrico es la unión de la lámina cerámica energizada con una lámina conductiva a la que se le denomina sustrato. La unión da como resultado el actuador de flexión que se desee. Como se verá en seguida, dependiendo de la forma de la lámina y la cantidad de actuadores que se apilen, se tendrán actuadores de distintas aplicaciones y características.

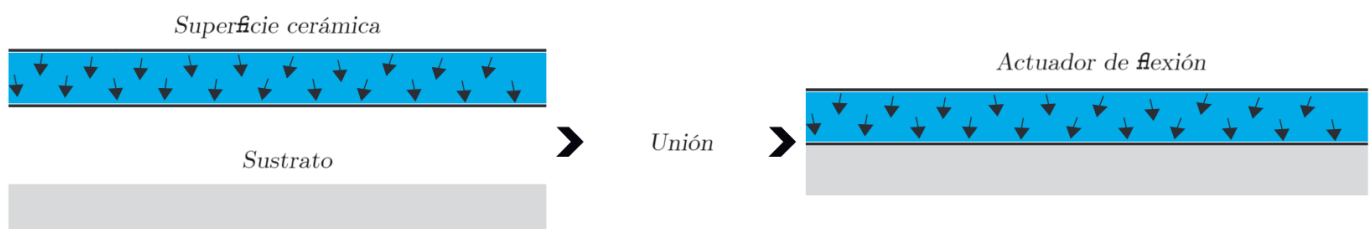


Figura 23. Creación del actuador de flexión.

Una vez aclarado el proceso de fabricación de este tipo de piezas se va a dar pie a resaltar la mayor ventaja que ofrece esta tecnología frente a las ampliamente utilizadas electroválvulas.

La ventaja de los mecanismos piezoeléctricos radica en el hecho de que pueden activarse con un muy menor gasto energético. Este hecho se debe a que, en términos eléctricos, un elemento piezoeléctrico no es más que un condensador que consta de dos placas conductoras separadas por el material piezoeléctrico cerámico que funciona como elemento dieléctrico. La corriente solo fluye mientras el capacitor se está cargando y el flujo cae a cero cuando este se carga al completo.

Este mecanismo difiere en gran medida con el funcionamiento de las electroválvulas, las cuales necesitan de la generación constante del campo magnético generado por la acción de la corriente que se les alimenta para su continuo funcionamiento en su estado de encendido. Se dirige al lector a **Figura 24. Diferencia energética entre tecnologías de válvulas.** en la que se intenta mostrar gráficamente esta tan tangible diferencia. En esta figura se emplea el color azul para graficar la energía (potencia en el tiempo) consumida por las válvulas

piezoeléctricas y el color gris para graficar el comportamiento energético de las electroválvulas.

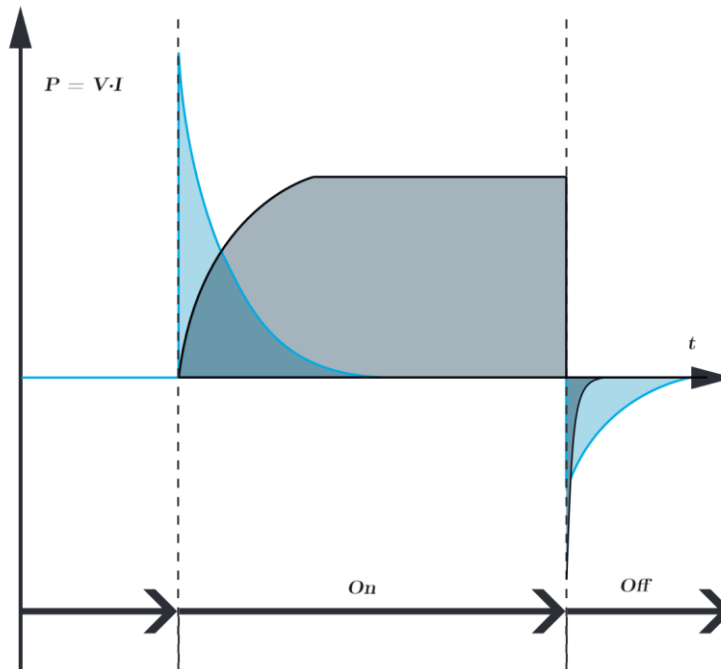


Figura 24. Diferencia energética entre tecnologías de válvulas.

Volviendo a uno de los aspectos mencionados en la fabricación de los elementos piezoeléctricos, se quiere seguir con la clasificación de los distintos tipos de transductores piezoeléctricos. Según la clasificación realizada por la empresa neumática Festo [18] existen 3 tipos básicos de transductores básicos. El **transductor de flexión**, el **transductor de disco** y el **transductor apilado**.

Como norma general todos siguen el mismo proceso de fabricación y funcionamiento, pero difieren en su geometría o en el número de elementos piezoeléctricos que incorporan en su ser. Se extenderá en el funcionamiento del primero de los actuadores que ha sido expuesto: el transductor de flexión, puesto que los demás transductores tienen un funcionamiento análogo a este.

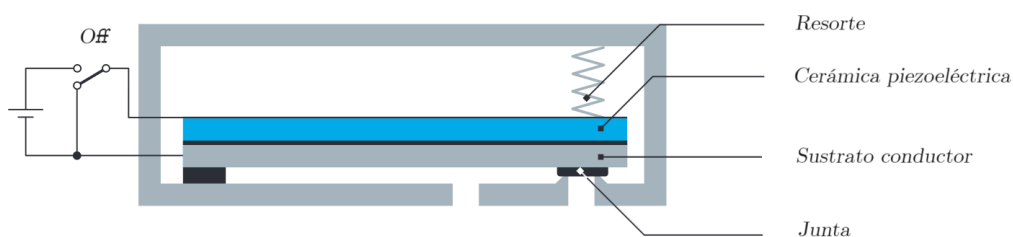


Figura 25. Estado del transductor sin voltaje.

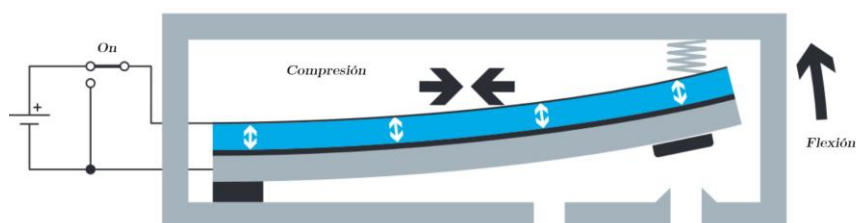


Figura 26. Estado del transductor con voltaje.

Como se puede observar en las figuras de los dos estados del transductor, al cerrar el circuito y aplicarle la tensión al transductor este se flexiona, debido al efecto piezoeléctrico. La flexión da como resultado un movimiento del transductor y con él una liberación de uno de los orificios neumáticos de los que se disponga en la válvula.

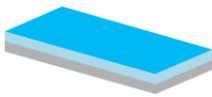
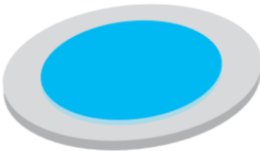

Este mismo mecanismo es aplicable para todos los transductores que se han dado a conocer a lo largo de este apartado. En el caso de los transductores de disco, el voltaje aplicado deforma al disco de tal forma que el disco se vuelve más grueso a la vez que su diámetro disminuye.

Los transductores de pila (también denominados pilas piezoeléctricas) son el resultado del apilado de discos piezoeléctricos que están conectados mecánicamente en serie y eléctricamente en paralelo. Esta configuración se caracteriza por posibilitar desplazamientos cortos, pero con enormes fuerzas de accionamiento de hasta varios kN.

Para diferenciar el rango de aplicación de cada uno de los elementos es para lo que se realiza la **Tabla 4**. Características de los distintos tipos de transductores., donde además de mostrar la forma de cada uno de los dispositivos se da una orden de magnitud de la fuerza que son capaces de aplicar y los desplazamientos que son capaces de proporcionar.

Hay que destacar que son los transductores de flexión los que se encuentran en el VTEM que se instala en el banco de ensayos, debido a su empleabilidad en los elementos de las válvulas proporcionales que permitan el correcto control de presión y flujo de los sistemas neumáticos que se instalan.

Tabla 4. Características de los distintos tipos de transductores.

	Transductor de flexión	Transductor de disco	Transductor apilado
			
Desplazamiento	100-1.000 μm	10-1.000 μm	10-1.000 μm
Fuerza	0,1-2 N	1-10 N	1.000-10.000 N

Para dar por finalizado el apartado dedicado al estudio de las válvulas basadas en tecnología piezoeléctrica, se quiere mencionar otro de las características intrínsecas de esta tecnología: la proporcionalidad que ofrece la tecnología piezoeléctrica. Como queda retratado en la **Figura 27**. Proporcionalidad de la tecnología piezoeléctrica., aspectos intrínsecos al comportamiento dinámicos de los actuadores como el caudal son prácticamente linealmente proporcionales a la tensión que se le aplica al circuito.

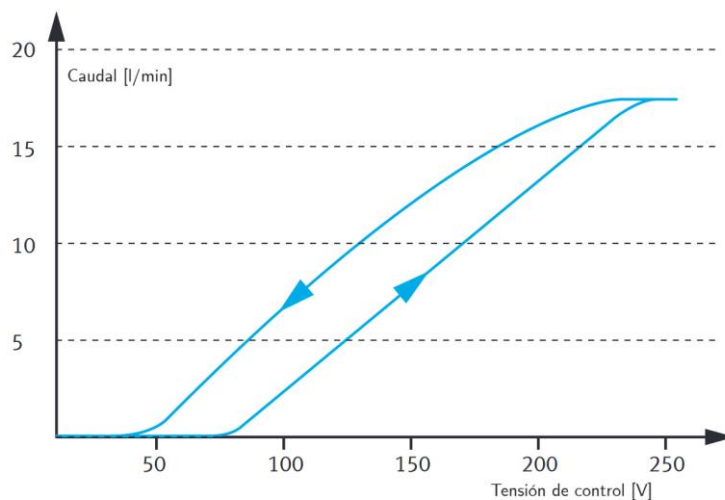


Figura 27. Proporcionalidad de la tecnología piezoeléctrica.

Esta proporcionalidad, junto con las demás características que han sido mencionadas hacen de las válvulas piezoeléctricas una opción más que razonable para ser las posibles sucesoras de las más primitivas electroválvulas.

6.3. Mantenimiento predictivo en la industria

Una vez estudiado la novedad tecnológica que ofrece el VTEM y las válvulas piezoeléctricas se quiere conocer de qué manera pueden ser detectados los posibles fallos que se puedan dar en el BEC. Pese a que teóricamente se hayan descrito los modos de fallo más comunes que se pueden dar en los cilindros neumáticos, se ha de tener en cuenta de que otro de los objetivos es barajar los posibles modelos que se pueden implementar. Se vuelve a recordar que el estudio que se muestre en este trabajo no será más que una introducción a los modelos que se vayan a implementar en la práctica y que, para un complemento se debe dirigir al lector al trabajo complementario.

El concepto del mantenimiento predictivo hace referencia a la idea central que subyace en la motivación de la investigación. La idea en sí no es difícil de entender, ya que se puede describir como la utilización de datos generados por la implementación de sensores para diagnosticar y monitorear en tiempo real los estados de las máquinas. A través del uso de algoritmos, se busca identificar métodos predictivos para prever roturas o fallos en las máquinas. Como se explicará en los próximos párrafos, esta metodología surge de manera natural a partir de las técnicas más tradicionales de mantenimiento.



Desde los albores de la revolución industrial, se concibió la noción de mantenimiento como las acciones necesarias para restablecer el funcionamiento de los equipos industriales tras sufrir algún tipo de avería o detenerse por completo. Con el tiempo, este enfoque frente a las fallas se denominó **mantenimiento correctivo**.

La necesidad de adoptar estrategias más sofisticadas en máquinas y equipos industriales se vincula estrechamente a la intensa competitividad experimentada en las economías de la posguerra durante la segunda mitad del siglo XX. La imperativa reconstrucción económica, un mercado global cada vez más competitivo y el florecimiento de sectores que ofrecían productos de alto valor agregado llevaron a los fabricantes a modificar sus planes de mantenimiento, incorporando indicadores de salud de las máquinas y dando origen al concepto de **mantenimiento preventivo**.

En este tipo de mantenimiento, se implementan planes rutinarios para evitar costosas interrupciones en la línea de producción que podrían derivar de la inoperatividad de la maquinaria. Estos planes incluyen la verificación de parámetros clave y la sustitución planificada de componentes fundamentales con el objetivo de prolongar la vida útil general de la maquinaria.

A pesar de sus ventajas competitivas frente al mantenimiento tradicional, este enfoque resulta insuficiente en los entornos industriales modernos más exigentes. Por ende, surge el **mantenimiento predictivo** como la forma más innovadora de abordar el mantenimiento, prometiendo maximizar la vida útil de los equipos. Esta metodología no solo se presenta como la más rentable, sino también como la opción más responsable con el medio ambiente, gracias a una mejor optimización de los recursos industriales. A continuación, se presenta un diagrama resumido que destaca los aspectos más relevantes de los diversos tipos de mantenimiento.

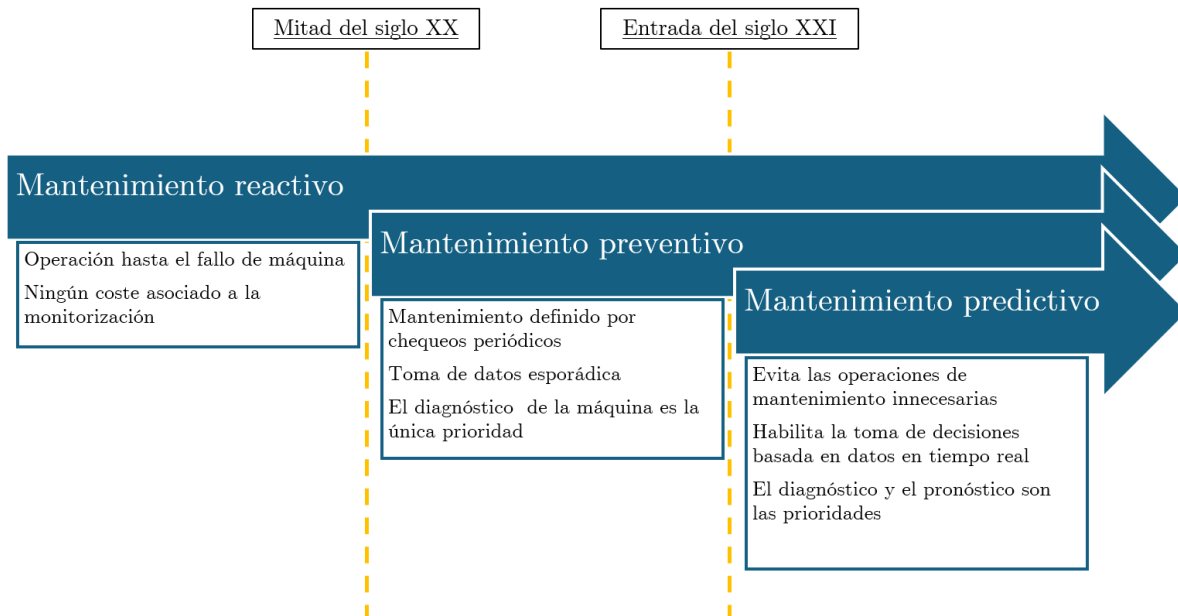


Figura 28. Evolución temporal del mantenimiento.

Existen innumerables técnicas para la implementación de métodos de mantenimiento predictivo entre los que se pueden destacar todos aquellos métodos que tienen como objetivo el cálculo del RUL. Estos modelos, como su acrónimo indica tienen como objetivo el cálculo de la vida útil restante de los componentes y es una metodología muy dependiente de la cantidad de datos que se tenga. En la siguiente figura se muestra la jerarquía de estos modelos.

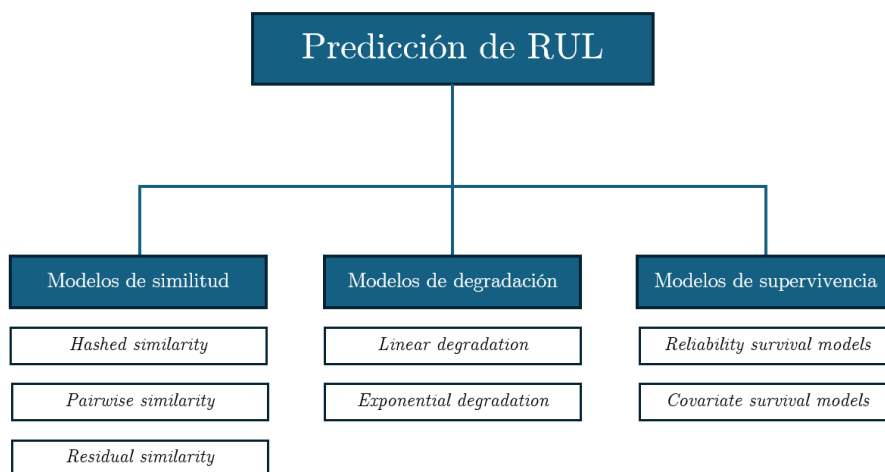


Figura 29. Jerarquía de los modelos de predicción de RUL.



Específicamente, los modelos de similitud se utilizan cuando se dispone de todo el historial de datos, desde el inicio de la operación del equipo hasta el momento en que experimenta una falla. En cambio, los modelos de degradación estiman el momento en el cual una señal monitoreada cruzará un valor umbral predefinido, indicando que se han alcanzado las condiciones de fallo. Además, es necesario contar con un sistema de monitorización que permita registrar la evolución de estos indicadores de degradación.

Finalmente, se encuentran los modelos de supervivencia, los cuales basan su estimación únicamente en muestras de las que se conoce un marcador temporal del tiempo de vida de los componentes. Estos últimos se emplean principalmente en situaciones en las que no es posible la monitorización directa de los componentes estudiados, aunque también pueden utilizarse como una estimación complementaria a las técnicas mencionadas anteriormente.

No obstante, por falta de datos, no se va a poder hacer uso de los modelos de predicción de rotura que se han esquematizado. A la hora de finalización de las prácticas en el centro tecnológico, únicamente se han tenido 2 ejemplos de rotura de los cilindros, lo cual es insuficiente a todas luces para el ajuste de cualquier modelo.

Por tanto, la alternativa que se planteó en vez la del cálculo del RUL fue la de aplicar métodos vanguardistas para la detección de anomalías en las señales que se tuvieran. Más concretamente, se han implementado métodos basados en inteligencia artificial. Toda la teoría que ha sido desarrollada en lo que al aspecto más técnico de estos métodos respecta (interpretación, arquitectura, validación) se detalla con mucha concreción en el trabajo complementario.

7. Exposición de alternativas

Desde que se presentó el proyecto al autor, eran varios los parámetros del trabajo que estaban completamente cerrados. Entre ellos se encuentran, el número de cilindros máximo que el BEC podrá testear simultáneamente, la elección del dispositivo de control, las marcas de los cilindros sobre los que se realizarán los ensayos o el método de acortamiento de vida escogido para forzar la degradación de los cilindros. No obstante, todavía quedaba por zanjar un elemento muy importante que concierne a los datos obtenidos del banco: los sensores de medida. A diferencia de los datos de presión y caudal, el VTEM no es capaz de proporcionar mediciones de posición, y se debió buscar solución en el mercado.

Tal y como se ha ido exponiendo a lo largo del trabajo, uno de los puntos clave de la monitorización del banco es el poder conocer la dinámica de los cilindros, además de ser punto de partida para el cálculo de indicadores termodinámicos tan descriptivos como el trabajo realizado en cada uno de los ciclos. Por ello, en un primer momento se planteó la instalación de sensores de distancia láser y/o codificadores de cuerda (a veces también denominados potenciómetros de cuerda) que fueran capaces de trazar el movimiento completo de estos. Esta idea llegó incluso a ser modelada en la propuesta piloto que se realizó del banco de ensayos tal y como se muestra en la **Figura 30**. Propuestas piloto de sensores de posición.

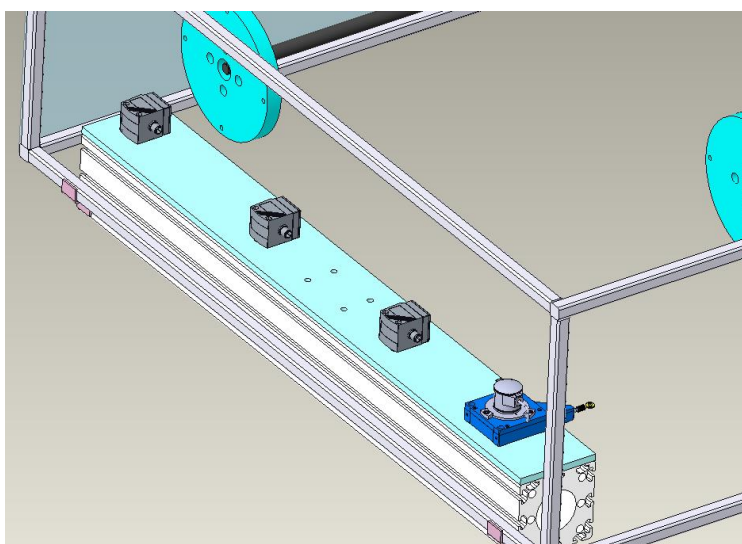


Figura 30. Propuestas piloto de sensores de posición.

Como se observa en la figura, los dispositivos se instalarían en un perfil de aluminio individual muy cercano a la posición de fin de carrera de los cilindros. Para el caso de los codificadores de cuerda, sería además necesario algún método de anclaje con el cilindro (sistema que no queda representado en la representación en CAD).

7.1. Criterios de selección

Para el acopio de estos sensores, en el proceso de selección se ha tenido muy en cuenta que cumplan con los requisitos básicos que se listan seguidamente:

- **Rango de detección:** Suficiente para captar toda la carrera del cilindro. Teniendo en cuenta que la cota superior de este requisito la imponen los cilindros de mayor carrera con 1400mm y que debe existir una separación mínima entre sensor y pistón, no se aceptarán sensores con rango de detección menores a los 1450mm⁸.
- **Rango de movimiento:** Además de que se puedan colocar los sensores a una distancia mayor de los de la expresada con anterioridad, es necesario que el dispositivo sea capaz de medir en un rango mayor a los 1400mm (carrera máxima de los cilindros).
- **Resolución:** Con el objetivo de no perder demasiada información de movimiento se ha decidido descartar cualquier sensor con resoluciones menores a 1mm.
- **Frecuencia de medida:** El objetivo de este criterio es el de garantizar que no se pierda información de la cinemática del cilindro, dado que este pueda moverse con una velocidad elevada. Los cálculos que se han de realizar para la deducción de este parámetro se basan en varias suposiciones en cuanto a comportamiento del sistema se trata.
 - Para empezar, se debe tener algún dato de referencia de la velocidad de cilindros que operen en condiciones similares a las que se someterán las muestras del banco de ensayos. Gracias a los datos de ensayos parecidos que se han llevado a cabo en la empresa Danobat, se conoce que para cilindros de carreras similares se puede fijar una velocidad media normal de unos 0.8 metros por segundo.

⁸ Valor arbitrario a elección del autor.

- No obstante, puesto que se quiere llevar a los cilindros a funcionar de manera anómala, este valor se multiplicará por un coeficiente de seguridad de 1,2. Esto supone dimensionar los sensores de manera que se suponga que la velocidad máxima de los cilindros sea de 1m/s.
- Una vez fijada la máxima velocidad a la que se moverán los cilindros se aplicará la sencilla fórmula que se encuentra a continuación para calcular la frecuencia necesaria de los medidores. Para que siga vigente el criterio de resolución de 1mm se tendrán que emplear sensores con frecuencias de:

$$f \text{ [Hz]} = \frac{1\text{m}}{\text{s}} \cdot \frac{1}{0,001\text{m}} = 1 \text{ kHz}$$

- **Precio:** Como puede resultar obvio, a igualdad de condiciones prevalecerá aquel sensor que menor coste de adquisición tenga.

7.2. Sensores de distancia láser

A diferencia de los codificadores de cuerda, este tipo de dispositivos ópticos pueden medir la distancia del objeto determinado utilizando un rayo o punto láser no invasivo. A muy grandes rasgos, son sensores muy polifacéticos en cuanto al rango de medida, puesto que el mercado ofrece sensores de medida desde rangos de unos pocos milímetros hasta decenas e incluso decenas de metros.

La precisión de estos dispositivos está muy ligada a la frecuencia de medición que posean y a la velocidad de movimiento del objeto que se quiera medir (en el caso de estudio, los cilindros). No obstante, en lo que a la frecuencia de medida usual de estos dispositivos respecta, no es difícil encontrar dispositivos con valores de algunos cuantos kHz.

Los 3 sensores de distancia láser que fueron seleccionados para la comparativa son: OM70-P1700.HV1700.EK de la marca Baumer, IL-2000 de Keyence y el sensor OY2P303A0135 de la marca Wenglor. Como se expondrá en la tabla de comparativas debido al proceso de fabricación necesario de estos sensores es demandante debido a su complejidad técnica y eso repercute en el coste final del producto. Como se acabará incidiendo en las conclusiones de este apartado, no se trata de la solución más económica, pero sí de la más precisa y debido a la incorporación de tecnologías más vanguardistas que ofrecen prestaciones de alto nivel se podría buenamente justificar su inversión económica.



Figura 31. De izquierda a derecha sensores Baumer, Keyence y Wenglor.

En esta figura **Figura 31**. De izquierda a derecha sensores Baumer, Keyence y Wenglor. se muestran los sensores sobre los que se realizará la comparativa. Cabe resaltar que tanto en factor de forma como en tamaño estos son muy parecido entre sí y que todos cumplen con los requisitos marcados en los criterios de selección.

Tabla 5. Comparativa de sensores láser de medición.

Modelo	Rango de movimiento	Resolución	Frecuencia de medida	Precio
OM70-P1700.HV1700.EK	1.550 mm	▲ 0,17 mm	■ 2.000 Hz	▼ 2.528,40 €/ud.
IL-2000	2.500 mm	■ 0,3 mm	▲ 3.000 Hz	■ 1.745,20 €/ud.
OY2P303A0135	2.800 mm	▼ 0,5 mm	▼ 1.000 Hz	▲ 522,10 €/ud.

Como se puede apreciar en la tabla, se evidencia una notable disparidad, especialmente en lo que respecta a los costos de los productos. En caso de que se deba adoptar una postura particularmente influenciada por consideraciones económicas, sería suficiente optar por la implementación de los sensores de la marca Wenglor (modelo OY2P303A0135), los cuales satisfacen todos los requisitos.

Por otro lado, los sensores de la marca Keyence, los cuales ocupan la segunda posición en la tabla, poseen una frecuencia de medida excepcional a la vez que mantienen un precio competente. Este producto podría, por ende, categorizarse como un producto de gama media.

Finalmente, se encuentran los sensores de la empresa Baumer, los cuales proporcionan una resolución excepcional a cambio de un valor de mercado más elevado. Cabe mencionar que justamente para el caso de estudio tal resolución no se contempla vitalmente necesaria, por lo que se podría optar por alternativas más económicas.

7.3. Sensores de cable

Los sensores de distancia por cable representan una opción realmente competente a los sensores de láser para la medición de posiciones lineales y distancias. También suelen denominarse, aunque de manera menos frecuente, como sensores o codificadores de sirga. Este tipo de sensores se diferencian con los sensores láser en tanto en lo que a la metodología de medición respecta. En vez de ser un haz de luz el encargado de detectar el objeto a medir, se debe acoplar un cable al objeto sobre el que se desea medir y realizar un acople físico al objeto mediante un fino cable que se enrollará sobre el codificador digital.

El proceso de montaje es sencillo y poco complejo puesto que tan solo se requiere fijar el sensor de distancia en uno de los perfiles fijos del banco y asegurar el extremo del cable a unos elementos móviles del cilindro. Con el movimiento subsiguiente, el cable se extraerá y retraerá interior del sensor, induciendo una rotación en el elemento de medida que podrá ser potenciométrico, magnético u óptico, entre otros.

El modo de operación de este tipo de sensores hace que estos elementos sean fácilmente integrables, además de ser una alternativa poco costosa. No obstante, uno de los grandes problemas que se dan con este tipo de elementos de medición es que es son ciertamente intrusivos. Como se expondrá en la tabla de comparativas, existen fuerzas mínimas para la retracción y extensión del cable de medida que deberán tenerse en cuenta en conjunto con los demás criterios de selección de los que ya se ha hecho mención en el 7.1. En esta nueva comparativa se vuelve a confiar en los sensores de la marca alemana Baumer con su modelo GCA3-PM. Además, estos sensores se comparan con el modelo CET5 CANopen de la marca TSM y el modelo SR1A de la marca Positek.



Figura 32. De izquierda a derecha los sensores de Baumer, TSM y Positek.

Tabla 6. Comparativa de sensores de medición de cable.

Modelo	Rango de medida	Resolución	Pull-in force	Pull-out force	Precio
GCA3-PM	2.300 mm	0,561 mm	▲ 1,5 N	▲ 8,0 N	▼ 624,36 €/ud.
CET5 CANopen	5.000 mm	0,1 mm	▼ 2,0 N	▲ 8,0 N	■ 353,20 €/ud.
SR1A	1.575 mm	0,384 mm	▼ 2,0 N	▼ 10,0 N	▲ 106,72 €/ud.

Nuevamente, la línea de calidad de los sensores de la marca Baumer destacan por su precio, aunque pueda parecer difícilmente justificable observando únicamente las características que se muestran en la tabla. Una de las respuestas al precio tan dispar con respecto a los demás sensores es la fiabilidad que proporciona la marca, la cual promete 500.000 ciclos de operación correcta [17].

7.4. Sensores de efecto Hall – Elección final

A pesar de todo el estudio económico-técnico que se ha realizado, no se han elegido ninguno de los componentes de medición que se han presentado hasta el momento. Es más, la solución ha venido impuesta por los superiores del autor y ha quedado muy influenciada por la sencillez y el bajo coste de los equipos de medición que se han querido montar. Sin más dilaciones, se quiere constatar que se han empleado lo que se denomina en la industria como transmisores de posición.

Debido a que estos elementos se montan sobre el propio cilindro, son elementos de tamaño reducido y fabricados exclusivamente para un tipo de geometría en concreto. Esto resulta en que cada uno de los fabricantes pueda producir transmisores de geometrías dispares. Este es el caso de estudio, en el cual se montan los dos sensores que se muestran en la **Figura 33**. Transmisores de posición de Festo (izda.) y de SMC (dcha.).

Sin embargo, el principio de medición es idéntico en los dos, puesto que ambos emplean el efecto Hall para realizar las mediciones. En estos elementos se aprovechan de los materiales de composición del propio conjunto émbolo-vástago para inducir el campo electromagnético en el dispositivo de medición. Por tanto, también se trata de una técnica de medición no invasiva.



Figura 33. Transmisores de posición de Festo (izda.) y de SMC (dcha.).

El mayor problema que presentan los medidores es que realmente no son del todo aptos para la finalidad en la que se les quiere implementar. El problema data en que este tipo de sensores tienen un rango de medición muy menor al que se ha expresado en los criterios de selección y prácticamente solo son capaces de detectar los finales de carrera de los cilindros. Esta es una traba que se hará visible sobre todo al final del trabajo, cuando se haga reflexión acerca de la calidad de los datos de la posición.

Pese a que se comentará más adelante con la solución propuesta, sí que se quiere dejar por escrito que el rango de medición de los sensores difiere con el modelo. Concretamente, los medidores de Festo tienen un rango de medida de 160mm, mientras que el rango de medición de los sensores de SMC es de 200mm.

8. Descripción de la solución propuesta

Finalmente, después de haber comparado las posibles alternativas aplicables a la solución final y posteriormente a analizar los riesgos del proyecto se pasará a la descripción de la solución propuesta, cuyo montaje se llevó a la realidad finalmente a los cuatro meses de haberse planteado el trabajo al autor.



Figura 34. Solución final del banco de ensayos.

En los siguientes apartados se ahondará más en todos los aspectos de la composición del banco, pero a muy grandes rasgos se pueden realizar varias observaciones. Primero, se observa que las fijaciones de los cilindros se han realizado mediante los perfiles de aluminio que se anclan al suelo con la ayuda de las escuadras de hierro fundido de color negro que se atisban en la imagen.

Además, como se puede comprender, en la **Figura 34**. Solución final del banco de ensayos. no se muestra al completo el BEC, puesto que quedan por enseñar los indispensables componentes de control que gobiernan el movimiento de los cilindros. Paulatinamente a lo largo de este nuevo apartado se presentarán todos estos elementos del banco, describiendo al detalle las funciones que realizan.

8.1. Topología

En este breve apartado se quiere mostrar esquemáticamente cómo se sitúan los elementos más importantes del banco.

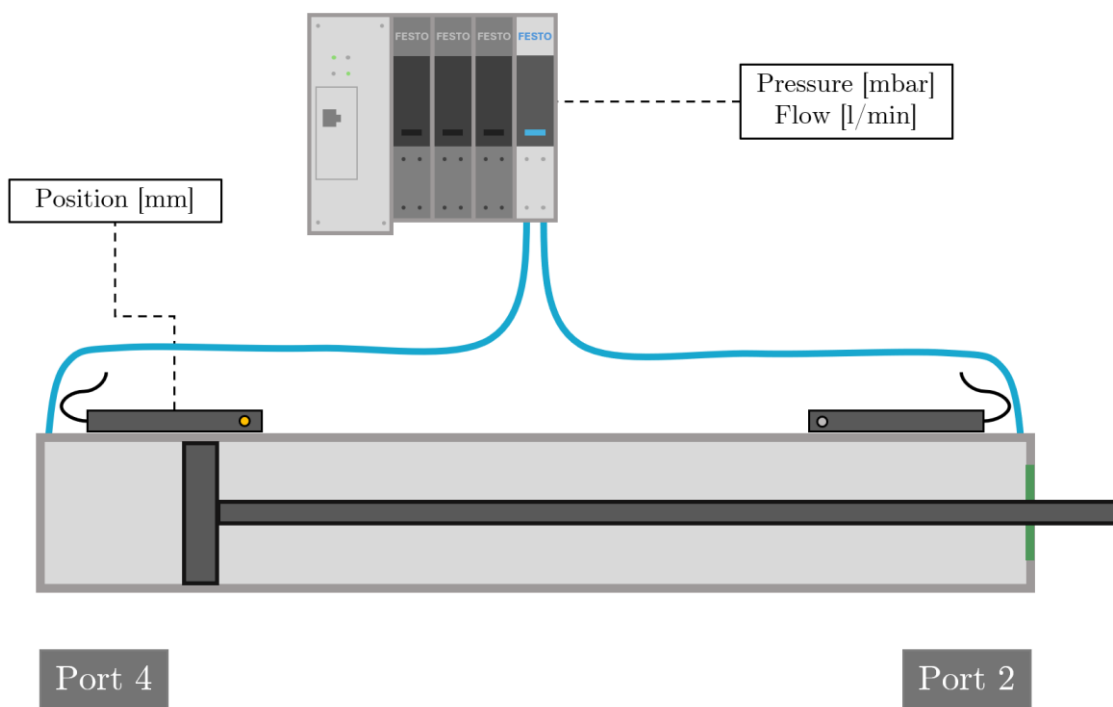


Figura 35. Topología de elementos conectados a los cilindros.

Como se puede observar en la figura, las conexiones neumáticas de los cilindros se realizan directamente a las válvulas. Además, pese a que no se muestre en la imagen, los sensores de posición se conectan a las entradas de 8 pines que pone a la disposición los periféricos del Motion Terminal.

También, mediante esta figura se incide en la proveniencia de las señales. La lectura de la presión y del caudal las realiza las válvulas y la posición proveniente de los sensores de posición que se han descrito ya con anterioridad.

8.1.1. Circuito neumático

El centro tecnológico cuenta con un tanque de aire presurizado que estaba alimentado por un compresor alternativo de 2 cilindros en disposición de V cuyo fluido termodinámico era el diésel. El aire atmosférico comprimido se almacenaba en el tanque que se muestra a continuación.



Figura 36. Tanque de aire presurizado.

Del manómetro del tanque se puede conocer que la presión a la que se encuentra el aire alimentado a los cilindros es de alrededor de 7,6 bar. Se conoce que este valor es correcto para la operación de los cilindros, puesto que se tiene que garantizar que la presión de operación se encuentre en los 6 y 8 bares. Por otro lado, también se desea fijar la mirada en la llave de paso que se encuentra próxima al tanque. Esta primera llave de paso es la que cerrará o abrirá el caudal de aire que se alimente a los cilindros. Complementariamente a esta, en el filtro de acceso del aire también se puede encontrar otra llave de paso.



Figura 37. Segunda llave de paso del sistema. A la entrada del filtro de aire.

8.2. Control

En este apartado se hará un especial hincapié en la algoritmia desarrollada para dar respuesta a los requisitos de movimiento de los cilindros. Pese a no comentar línea a línea el código desarrollado en texto estructurado en conjunto con el ingeniero senior envuelto en el proyecto, sí que se detallarán las decisiones más fundamentales que fueron tomadas y se expondrán las condiciones y funciones fundamentales del proyecto.

La primera idea en cuanto al control de las válvulas es conocer el modo de control que Festo ha desarrollado para interactuar con las válvulas. Realmente, la empresa ha desarrollado unas librerías en las que se encuentran las funciones que se emplean para controlar cada una de las válvulas que se encuentran en el VTEM. En estas librerías se encuentran las funciones de control de las válvulas.



Mediante este ejemplo del bloque de función para la válvula 0 se pueden realizar varias observaciones. Primero, se emplea una configuración de la válvula del tipo *4/3 normally exhausted*. Esta decisión se tomó para salvaguardar a los operarios en caso de fallo, para que cuando se pueda dar un error las válvulas liberen toda la presión. Por otro lado,

```
(*****)  
(* Control valvula 0 *)  
Valve0_FB_ValveControl(  
  awBusDataFromVTEM := Valve0_Inputs,  
  awBusDataToVTEM := Valve0_Outputs,  
  xEnable := Valve0_Enable,  
  bySetValveMode := Valve0_ValveMode,  
  bySetAppControl := Valve0_AppControl,  
  bySetAppOption := Valve0_AppOption,  
  iSetpointValue1 := Valve0_SetpointValue1,  
  iSetpointValue2 := Valve0_SetpointValue2,  
  xManualAcknowledge := Valve0_ManualAcknowledge,  
  byActualValveMode => Valve0_ActualValveMode,  
  byActualValveState => Valve0_ActualValveState,  
  byActualAppState => Valve0_ActualAppState,  
  iActualValue1 => Valve0_ActualValue1,  
  iActualValue2 => Valve0_ActualValue2,  
  iResponseToValveModeSet => Valve0_ResponseToValveModeSet,  
  iErrorCode => Valve0_ErrorCode);
```

Figura 38. Bloque de control para la válvula 0.

Por otro lado, se ha empleado en todo momento la MA número 1 para el control del movimiento de los cilindros. Esto es debido a que esta opción permite abrir y cerrar el paso del aire a la presión de alimentación del banco. Como el objetivo no es la longevidad de los cilindros, se quieren emplear presiones cuanto más grandes mejor (siempre y cuando no sobrepasen los umbrales de operación definidos en los manuales de operación).

La forma en la que se realiza el cambio de sentido de los cilindros es mediante el parámetro AppControl. Si el cambio de su valor se realiza de manera correcta, este parámetro invierte el movimiento del vástago en el momento en el que el vástago del cilindro llega a su final de carrera. Lógicamente, dentro de código también se realizará un conteo de los ciclos de los cilindros. Este valor es el que se visualizará en las GUI que han sido programadas.

8.3. GUI de la HMI

La interfaz de usuario creada para este proyecto constituye un componente central en la gestión integral del banco. Durante su desarrollo, se ha mantenido un enfoque constante en tres principios fundamentales: la atracción visual, la simplicidad y el control seguro de las operaciones bancarias. La concepción de esta interfaz se ha guiado por la premisa de proporcionar a los usuarios una experiencia atractiva y eficiente, priorizando la claridad en la presentación de información y garantizando un manejo seguro y preciso.

8.3.1. Pantalla principal

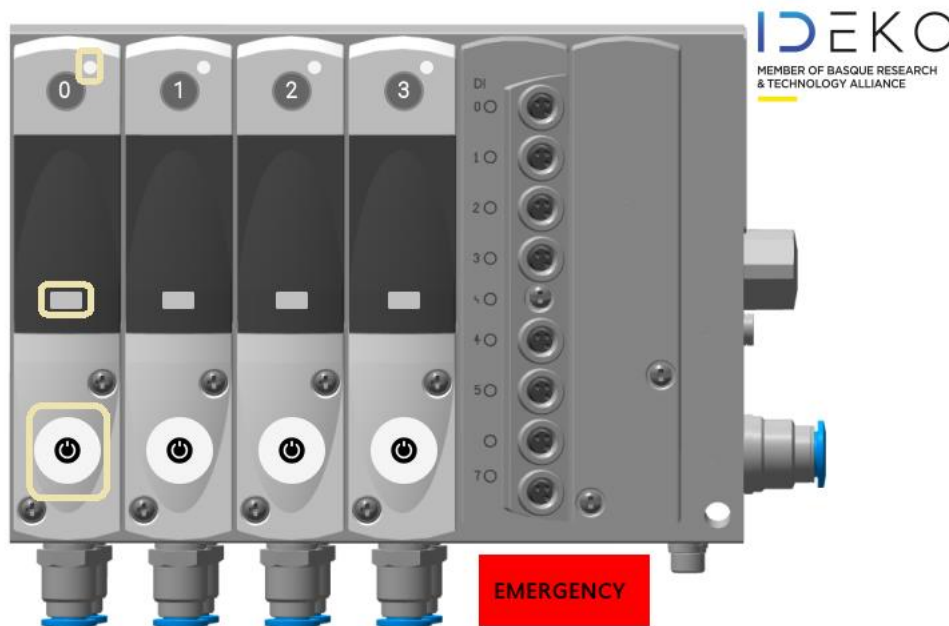


Figura 39. Pantalla principal de la GUI.

Siguiendo las líneas de diseño se presenta la pantalla de bienvenida de la HMI. En esta se quiere fijar la atención en algunos de los accionables que se muestran. Como en toda máquina, se tiene un botón de encendido/apagado que dará una orden al controlador para comprobar si es posible una puesta en movimiento del cilindro al que esté conectado cada una de las válvulas. Se quiere incidir en que este botón no pone en marcha inmediatamente los cilindros, sino que es necesario un chequeo de acción previo.

Este chequeo guarda una gran relación con el segundo elemento resaltado la imagen: el rectángulo que aparece de color gris. Este se trata de un indicador, a modo LED, que proporciona una información adicional al operador del banco. Dependiendo del estado del cilindro este tomará uno de los 3 colores que se muestran en la tabla siguiente:

Tabla 7. Modos de estado de las válvulas.

	El cilindro se encuentra operativo y en marcha.
	El cilindro se encuentra operativo, pero se ha de presionar el botón "Acknowledge".
	El cilindro no se encuentra operativo.

Una de las dudas que surgen cuando se detallan los modos de estado es el botón "Acknowledge". Debido al firmware que se instala en el VTEM todas las válvulas tienen unos comportamientos de seguridad intrínsecos. Siempre que se vuelvan a conectar la alimentación eléctrica a las válvulas, por defecto no podrán iniciar el movimiento si se pulsa el botón de encendido de la pantalla principal. Cuando este sea pulsado aparecerá en pantalla el mencionado botón "Acknowledge", el cual deberá ser pulsado por el operario si desea que se dé comienzo al movimiento en la válvula.

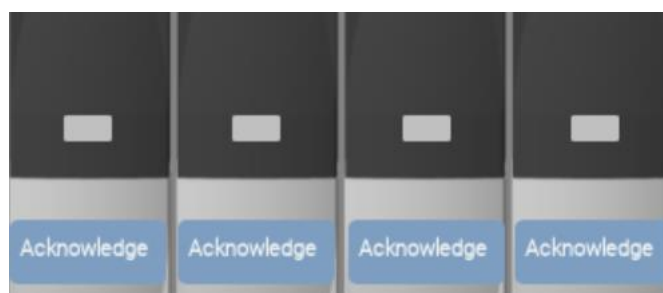


Figura 40. Ejemplo de aparición del botón Acknowledge.

Bien es cierto que el nombre puede resultar no muy representativo, pero lo cierto es que la definición del nombre viene dada por la propia empresa fabricante Festo.

Si se consulta en si manual de programación del VTEM [18] se observa cómo el anglicismo hace referencia a que el operario es consciente y se hace responsable de cualquier error que pueda conllevar el accionamiento de los cilindros.

Una vez aclarado el funcionamiento de los indicadores de las válvulas se puede dar paso a los dos botones restantes más llamativos de la página principal. En la esquina superior derecha al lado del indicador de válvula se encuentra un led rojo que indica si en la válvula asociada al cilindro se está llevando a cabo una grabación. Cuando este sea el caso, este indicador se iluminará y parpadeará de modo que llame la atención lo suficiente como para que el encargado del control del banco no detenga el cilindro.

Por último, se encuentra el botón más llamativo y grande de todos. Como su nombre indica, este es el botón que se ha de pulsar en caso de avería grave o en caso de tener que realizar algún cambio, ajuste o manipulación en el banco. Al pulsarlo, se cambiará el estado de todas las válvulas simultáneamente de modo que se liberen las dos compuertas de (compuertas 2 y 4) de la válvula, lo cual hará liberar toda la presión que quedaba almacenada en los cilindros. En este mismo momento se apagará también las válvulas automáticamente.

8.3.2.Pantalla de válvula

Aprovechando la utilidad táctil de la HMI se ha incorporado un botón en cada una de las válvulas de la pantalla principal. Siempre y cuando la entrada táctil se realice encima de las válvulas y no se efectúe en el botón de encendido/apagado de las mismas, se llegará a una de las pantallas como la que se muestra en la **Figura 42**. Pantalla de válvula..

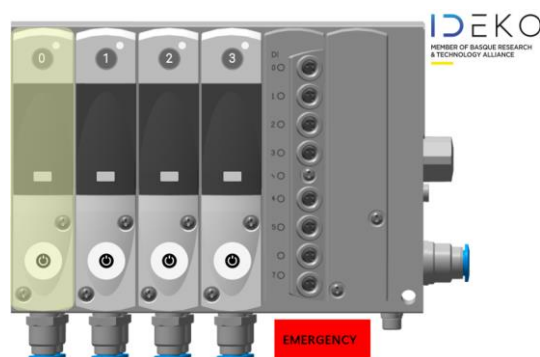


Figura 41. Zona táctil para acceder a las pantallas de válvula.

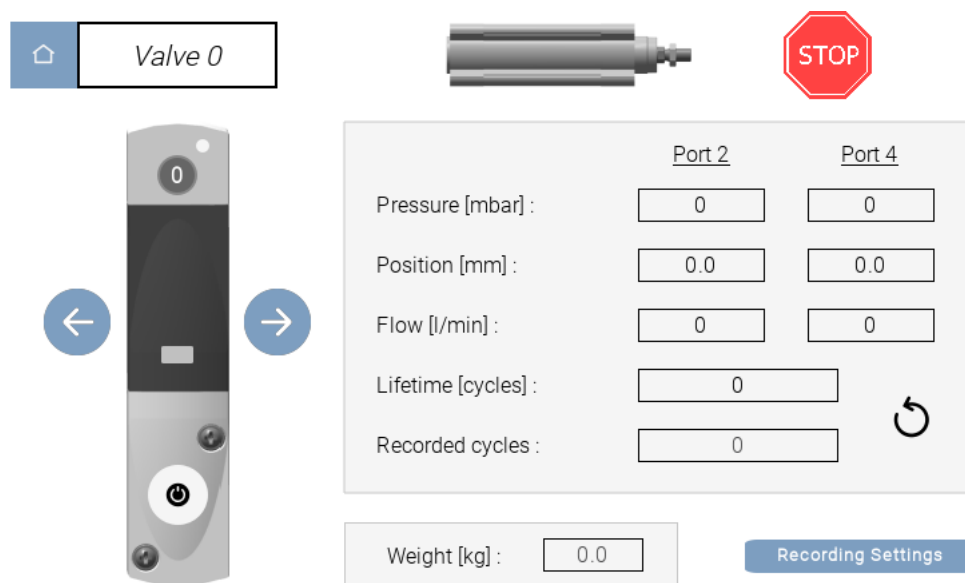


Figura 42. Pantalla de válvula.

Esta parte de la interfaz gráfica es donde se muestra la gran parte de la información referida a la operación de los cilindros. Por un lado, se guarda el tercio izquierdo de la pantalla para poder acceder a los mismos controles de válvula que los que se mostraban en la pantalla principal. Mediante esta zona se puede acceder tanto a los controles de encendido/apagado como a la información del estado de válvula que se mostraba en pantallas anteriores.

Adicionalmente a estos controles, se muestran: indicaciones del sentido del cilindro en la parte superior, botones de navegación que permiten posicionarse en la válvula deseada con facilidad y una visualización de las métricas que se tienen de la operación del cilindro como la presión, posición, caudal, el número de ciclos y el número de grabaciones.

Asimismo, se incluyen dos botones para acceder tanto a los ajustes de grabación de cada una de las válvulas como para realizar un reinicio de los contadores del número de ciclos y el número de grabaciones. Puesto que desde el inicio del proyecto se comprendió que los ensayos se realizarían hasta llevar a los cilindros a una ruptura total y que en caso de ruptura el elemento se vería sustituido por uno nuevo, se vio la necesidad de implementar prestaciones que ayudasen al operario realizar el cambio de cilindro en la HMI. Estas prestaciones se agrupan en lo que el autor ha denominado pantallas secundarias.

8.3.3. Pantallas secundarias

Dentro de cada una de las pantallas de válvula se puede acceder a funcionalidades adicionales como el ajuste de los parámetros de grabación. Estos ajustes que se muestran en la Figura permiten modificar con facilidad tanto la duración de la duración de las grabaciones como la cadencia de estas. Es decir, permiten modificar manualmente cada cuanto se realizan las grabaciones, parámetro esencial si se quiere aumentar o disminuir la continuidad con la que los datos son generados.

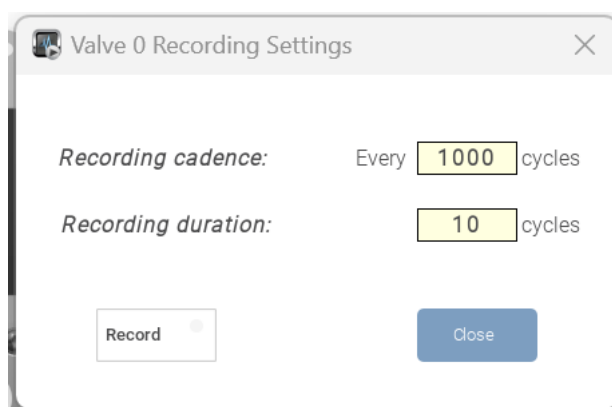


Figura 43. Ajustes de grabación de la válvula 0.

Los valores por defecto de grabación quedan fijados a los que se muestran en la Figura 43. Ajustes de grabación de la válvula 0. mediante su definición en el proyecto de CoDeSys. Esto significa que cada vez que se cargue en el PLC el proyecto, los ajustes de grabación se inicializarán a estos valores. Debido a las operaciones necesarias que se llevarán a cabo a lo largo del estudio (reaprovisionamiento de cilindros), se han implementado algunas instrucciones en la interfaz gráfica que permitan guiar al operario en sus tareas.

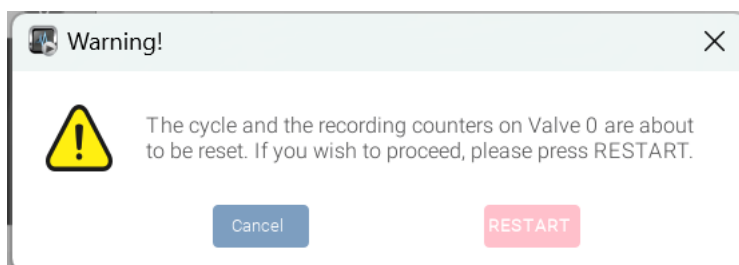


Figura 44. Ejemplo de mensaje de la GUI.

8.4. GUI del portal de acceso de Savvy

Una vez ha quedado definida la interfaz de usuario que ha sido desarrollada por el autor para el HMI del banco, se va a dar paso a presentar otra de las interfaces que se han tenido que desarrollar para el proyecto.

Empleando los portales de visualización que ofrece la empresa Savvy se ha decidido montar un servidor web que sea capaz de mostrar los datos más representativos de los que se ha tenido acceso mediante el protocolo Modbus de baja frecuencia. El resultado final se puede observar en la **Figura 45**. GUI de la plataforma Savvy, y se ha hecho el esfuerzo de intentar proveer al usuario de tanta información útil como sea posible mediante un rápido vistazo.

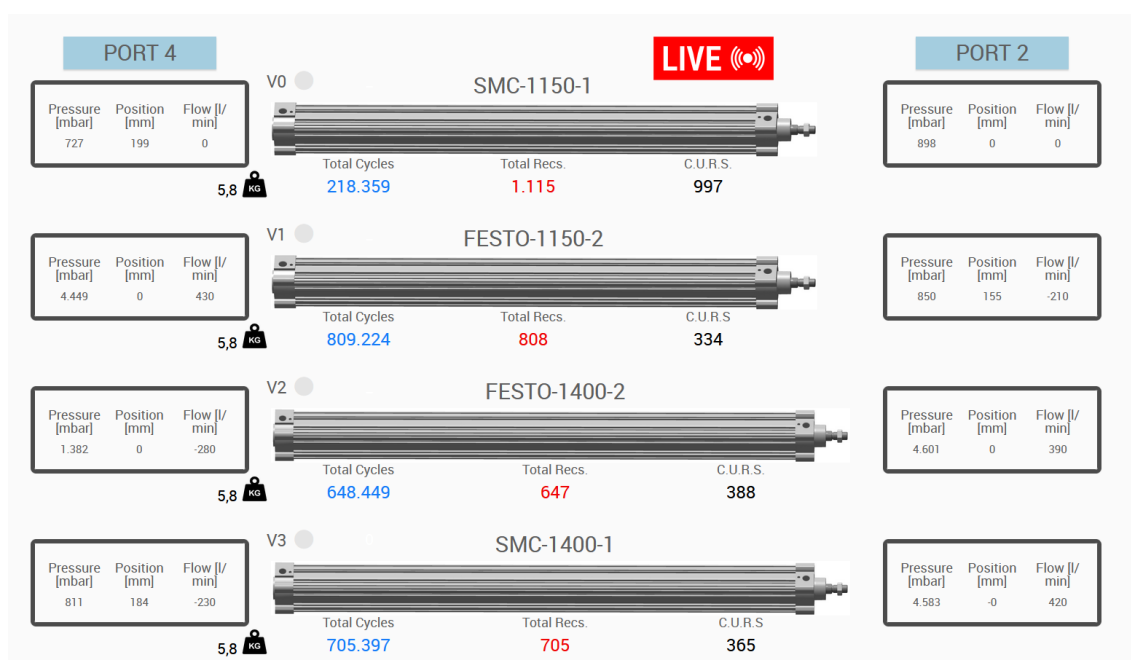


Figura 45. GUI de la plataforma Savvy.

El elemento central de estudio, los cilindros, se han colocado en la mitad de la pantalla. Se ha tenido en consideración la longitud de los cilindros instalados la cual queda reflejada en el tamaño de la imagen de la interfaz gráfica.

El segundo conjunto de objetos sobre los que se quiere comentar son los parámetros que se monitorizan en los rectángulos de color negro que se encuentra en cada uno de los laterales. En ellos se muestran los valores a tiempo real de la presión, posición y caudal de cada una de las válvulas. No obstante, en

comparación con los valores que se puedan obtener en las pantallas de las válvulas (como la de la **Figura 42**. Pantalla de válvula.), estos valores, pese a ser idénticos, no se actualizan con tanta frecuencia. Por tanto, su utilidad queda reducida en ayudar si se está dando algún tipo de movimiento en el cilindro.

Debajo de cada uno de los cilindros se encuentran los parámetros más significativos del banco de ensayos, los referidos al monitoreo del ciclo de vida del cilindro. En color azul se muestra la vida de los cilindros. Complementariamente, en color rojo se añadió un indicador del número de grabaciones que se han realizado sobre ese cilindro. Por último, se encuentran una cifra bajo las siglas C.U.R.S. El valor que se encuentra debajo de estas indica el número de ciclos restante hasta que se inicie la siguiente medición.

En conjunto, tanto la interfaz gráfica de la HMI como la que el usuario tiene acceso mediante el portal de Savvy proporcionan información suficiente para dar a conocer el estado general de los cilindros, pero no suficiente para observar si se está dando algún comportamiento anómalo. Para ello, se instaló una cámara IP de la marca axis permite conocer el estado de los cilindros. Para acceder a la imagen que proporciona esta cámara basta con pulsar el botón "Live".



Figura 46. Cámara IP instalada en el banco.



9. Descripción tareas, fases y procedimientos

Como ya ha sido varias veces mencionado, el proyecto en sí se compone por la compleción de 2 grandes hitos: siendo el primero la puesta en marcha de un banco de ensayos y el segundo la implementación de inteligencia artificial en un esquema de mantenimiento predictivo. Únicamente se quiere aclarar que el desglose de tareas, fases y procedimientos que se detallan a continuación hará referencia al primer bloque del proyecto, que es sobre el que se ha hablado a lo largo de este trabajo.

T1. Diseño conceptual del banco

ST 1.1. Objetivo y especificación de los requisitos del proyecto.

ST 1.1. Validación de automatismos con el responsable del equipo de automatización.

ST 1.1. Primeros prototipos del banco.

T2. Acopio de materiales

ST 2.1. Soporte para la colocación de los cilindros.

ST 2.2. 2 cilindros de marcas distintas con longitudes de 1400mm.

ST 2.3. 2 cilindros de marcas distintas con longitudes de 1150mm.

ST 2.4. 2 cilindros de marcas distintas con longitudes de 1400mm.

ST 2.5. Grupo de filtrado para el aire entrante al banco.

ST 2.6. Cerramiento del banco.

ST 2.7. 1 Savvy Smart Box.



T3. Automatización del banco

ST 3.1. Definición de modos de funcionamiento.

ST 3.2. Desarrollo de control mediante CoDeSys.

ST 3.3. Desarrollo de las interfaces del usuario en el PLC.

ST 3.4. Desarrollo de las interfaces del usuario en la plataforma de Savvy.

ST 3.5. Configuración de la Savvy Smart Box.

T4. Validación del banco de Ensayos

ST 4.1. Validación del movimiento del banco.

ST 4.2. Validación de los datos.

Fase 1: Diseño conceptual del banco:

En esta fase, mediante las reuniones de los encargados del grupo de investigación se realizaron las primeras aproximaciones del banco de ensayos. Se realizaron los primeros bocetos, así como se definieron la definición de los primeros criterios generales que el banco debería cumplir. Se dedicaron unas 120h a esta tarea.

Además de las tareas más materiales, también se vio involucrado una gran cantidad del tiempo a la investigación de los métodos de aceleración del tiempo de vida de los cilindros y hacia la previsión de los modos de fallo que se pudieran dar en estos elementos. La realización de estas tareas se hizo de manera contigua a la redacción de esta memoria. Asimismo, no se ha de olvidar que una gran parte de la investigación fue centrada en el entendimiento del marco teórico de las redes neuronales y del aprendizaje automático, esfuerzo que pese haber sido empleado en los resultados finales del proyecto, no puede definirse como perteneciente a este trabajo.

Fase 2: Acopio de materiales:

Una gran carga de del trabajo queda definida en el ámbito de la adquisición de todo el material que compone al banco, así como del seguimiento de los pedidos que se realizan. En esta fase se hace un despiece de los materiales más importantes para el montaje del banco de ensayos se le dedican dos meses (Unas 480h).

Fase 3: Automatización del banco:

En esta fase se definen tanto el software necesario para el control de los cilindros como los posibles modos en los que se podrían operar el banco. Es una fase con gran carga en programación a la que se le dedicarán unas 240h.

Fase 4: Validación del banco de ensayos:

Esta última fase se ve compuesta por la realización de la verificación de cumplimiento de las tareas definidas en la primera fase, además de verificar si los datos obtenidos del banco son de la calidad suficiente como para seguir adelante con la implementación de técnicas de inteligencia artificial.



10. Diagrama de Gantt

	Ene	Ene	Feb	Feb	Mar	Mar	Abr	Abr	May	May	Jun	Jun
Diseño del banco												
Objetivo y especificación de los requisitos		■										
Validación de automatismos			■									
Prototipos del BEC			■	■	■							
Acopio de materiales												
Soporte para la colocación de los cilindros					■	■						
2 cilindros de diámetro 32mm y longitud 1400mm			■	■	■	■						
2 cilindros de diámetro 32mm y longitud 1150mm												
Grupo de filtraje					■	■						
Cerramiento			■	■	■	■						
1 Savvy Smart Box			■	■	■	■						
Automatización del banco												
Definición de modos de funcionamiento				■	■							
Desarrollo de las interfaces del usuario en el PLC.							■	■	■			
Desarrollo de las interfaces del usuario en la plataforma de Savvy.							■	■	■			
Desarrollo del software de control mediante el empleo de CoDeSys como entorno de desarrollo integrado.							■	■	■			
Configuración de la Savvy Smart Box.									■	■		
Validación de puesta en marcha del banco												
Validación del movimiento de los cilindros											■	
Validación de los datos												■

11. Casos de aplicación de los modelos predictivos

El 27 de octubre de 2023, se realizó una inspección rutinaria en el banco de pruebas. Durante esta evaluación, se notó que uno de los cilindros emitía un sonido significativamente más fuerte. Se pudo identificar rápidamente que este aumento de nivel de ruido estaba vinculado a un mal funcionamiento, atribuido a la salida de la junta rascadora del vástago. Este mismo problema ya se había presentado en el cilindro adyacente, pero desafortunadamente en aquel momento la toma de datos no estaba activa y no se tienen registros de este fallo.



Figura 47. Salida de la junta rascadora.

El resultado de este fallo es el que se divisa en la **Figura 47**. Salida de la junta rascadora.. En esta, se comprueba que la junta está situada en el peso que se emplea como elemento de aceleración de la degradación en el elemento neumático.

Realmente, el problema de la salida de la junta se da en dos de los 4 cilindros que se habían puesto en marcha. Curiosamente, este mismo modo de fallo se dio únicamente en los cilindros de la marca Festo y lamentablemente el mismo modo de fallo no pudo ser reproducido para los nuevos elementos que sustituyeron a los originales. El problema que se achaca a este mal funcionamiento puntual es el mal montaje que pudo haberse realizado. Debido a la rareza en la longitud de carrera de los cilindros, se comprueba que tales elementos no suelen ser los más demandados y quizás hayan sufrido de una calidad de montaje inferior a los cilindros que provengan de tiradas más largas.

De todos modos, volviendo a centrar la mirada en el fallo producido, ante esta situación, una de las primeras respuestas fue verificar si se podía detectar algún cambio en cuanto a los datos se trataba. Afortunadamente, se pudo observar un cambio cualitativo en todos los parámetros monitorizados.

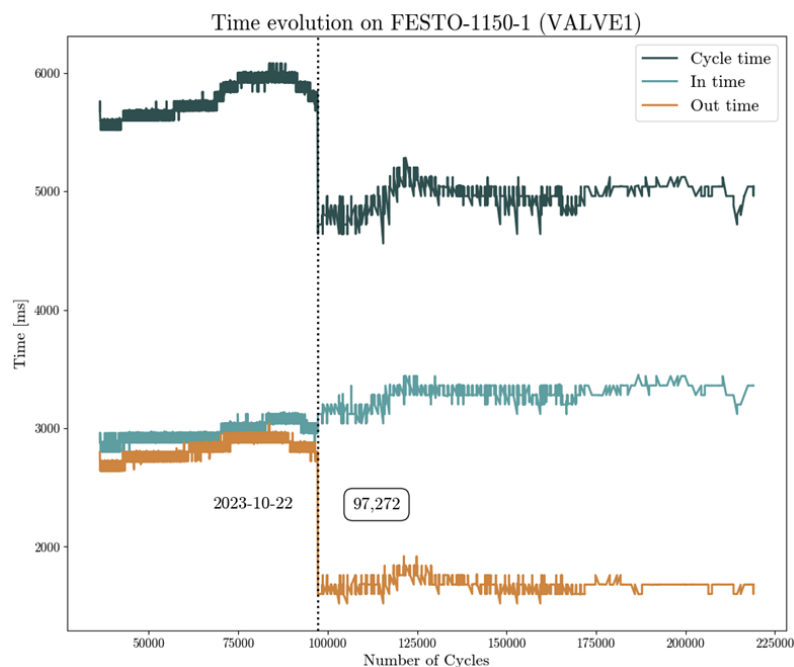


Figura 48. Evolución de parámetros temporales en el cilindro dañado.

Para la creación de la **Figura 48**. Evolución de parámetros temporales en el cilindro dañado. se realizaron los cálculos temporales de los tiempos de ida, de retorno y también del total del movimiento de los cilindros. El cambio de comportamiento es sobre todo notorio para el movimiento de salida, lo cual repercute en el tiempo total del ciclo del elemento.

11.1. Representación de los datos

Con la premisa de que se conocía la respuesta al fallo se decidió incidir en la implementación de algoritmos que permitieran la detección de comportamientos anómalos. De todas formas, antes de dar paso a los resultados obtenidos de los modelos implementados se quiere mostrar gráficamente los datos con los que se opera.

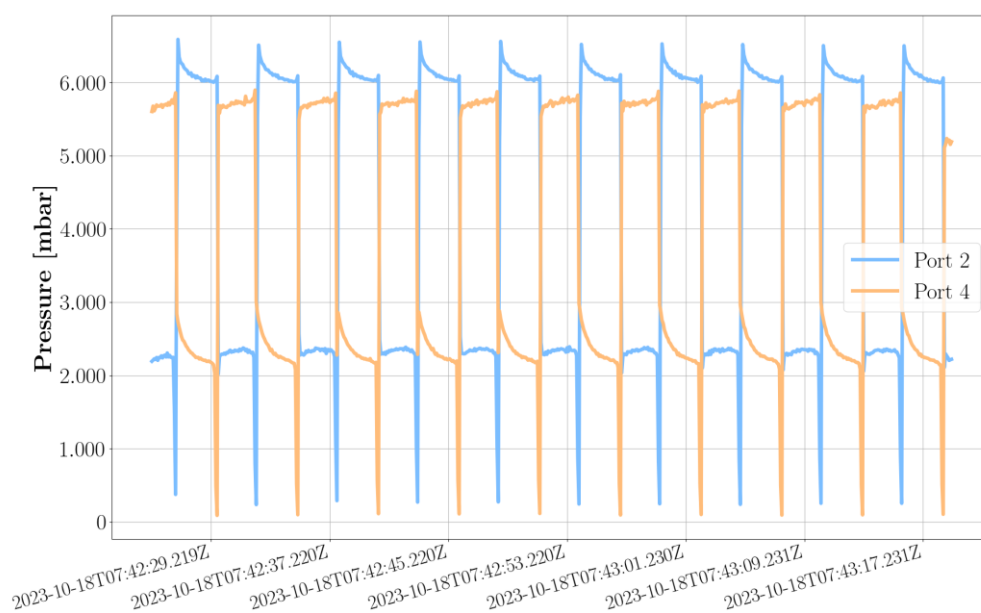


Figura 49. Datos en crudo de la presión.

En esta figura se completan los datos de la presión del puerto 2 con los del puerto 4. Realmente, pese a que la apreciación a simple vista no sea la técnica más rigurosa, se debe realizar un pequeño acto de fe con el autor y creer que la calidad de los datos de la presión es realmente elevada. De hecho, se optará por trabajar con los datos de presión siempre que sea posible, puesto que la información obtenida de los demás sensores no llegará a tener una mayor fidelidad.

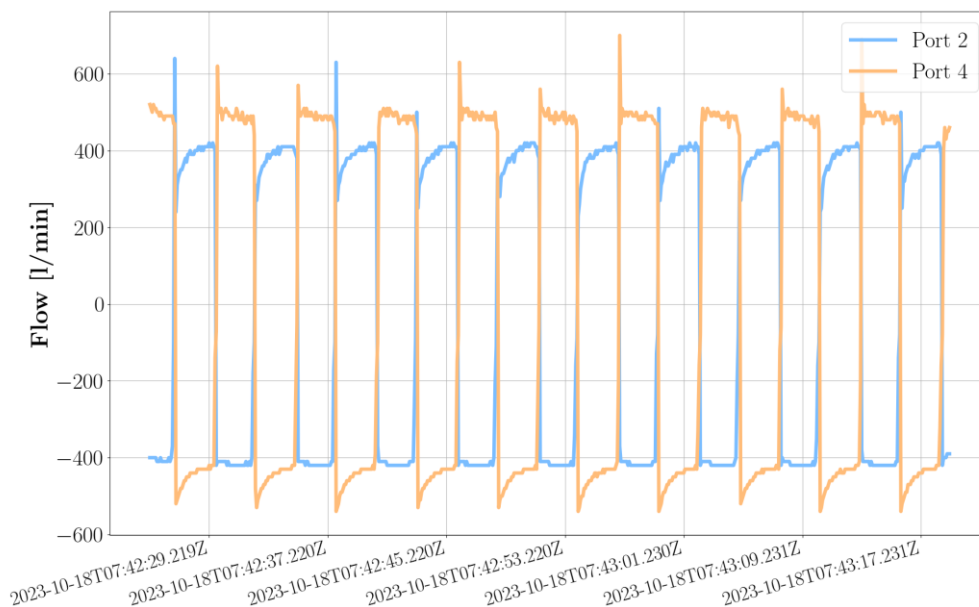


Figura 50. Datos en crudo del caudal.

La precisión obtenida en este caso es cuantitativamente peor. Por los manuales que se dispone de Festo se sabe que la resolución es de 10l/min y que, además, no siempre se tiene una disponibilidad de los datos. Es decir, debido a las tareas internas del controlador lógico, no se puede disponer en todo momento del dato del caudal. De hecho, se quiere guardar un apartado a la curiosa metodología que implementa el VTEM para conocer el valor del caudal.

11.1.1. El modo de transferencia

Para conocer el porqué del extraño comportamiento de los datos del caudal se tiene que indagar ligeramente en el software de VTEM. En el capítulo del control ya se informó de las funciones de control que se han desarrollado para las válvulas, así como también se han explicado en capítulos anteriores las MA disponibles. Sin embargo, todavía no se ha explicado una de las funcionalidades que esconden estas válvulas.

Muy a pesar del autor, si se emplea la MA número 1 en la función de control de la válvula, los valores que se pueden acceder desde esta función (`iActualValue1` e `iActualValue2`) son las presiones de los puertos 2 y 4 respectivamente, como se muestran en la **Figura 51**. Valores disponibles con la MA 1..

Actual Value 1 (Byte 2 ... 3): Measured pressure at port (2)

-1000 ... +32767 × 1 mbar

Actual Value 2 (Byte 4 ... 5): Measured pressure at port (4)

-1000 ... +32767 × 1 mbar

Figura 51. Valores disponibles con la MA 1.

Para poder acceder, parecía por tanto ser suficiente dar con aquella MA en la que se pudiera acceder a los valores del caudal. En otras palabras, se podría intentar observar si existe alguna MA en cuyos valores `iActualValue1` e `iActualValue2` se encuentren los valores de la presión. La respuesta a esta cuestión es afirmativa, la MA 10 da justo con lo que se está buscando. El problema es que no se tiene la licencia necesaria para poder operar con ella.

Asimismo, el problema es algo mayor. Pese a que se pudiera tener la licencia deseada, existe el problema de que, si se activa una válvula con una MA para conocer la presión, no se va a poder activar esa misma válvula para correr simultáneamente otra MA que permita acceder a la información del caudal. La solución a este problema es algo intrincado, puesto que se tienen que manejar ideas como las válvulas virtuales, los comandos acíclicos y el modo de transferencia.

Primero, se comienza por presentar el **transfer mode**. Este es un protocolo que permite acceder a un mayor nivel de información del que se dispone a primera instancia con las MA. Con sus diferentes canales se pueden tanto modificar ajustes de entrada, definir los periféricos que se acoplan a cada válvula, obtener información adicional y obtener códigos de error detallados. El acceso a este modo se realiza mediante la definición del *valve mode* de la función de control de la válvula al valor 63. Sin embargo, ya se ha detallado que será necesaria la definición del modo de funcionamiento en una válvula virtual. La definición de estas válvulas es la que se muestra en la **Figura 52**. Definición de una válvula virtual para lectura de parámetros..

```

(*****)
(* Control de valvula virtual para lectura de parametros *)
VirtualValve0_ValveMode := 63; (* Transfer Mode *)
VirtualValve0_AppControl := 2; (* Upload (read) *)
VirtualValve0_AppOption := 0;
VirtualValve0_SetpointValue1 := 0;
VirtualValve0_SetpointValue2 := 0;

VirtualValve0_FB_ValveControl(
  awBusDataFromVTEM := VirtualValve0_Inputs,
  awBusDataToVTEM := VirtualValve0_Outputs,
  xEnable := VirtualValve0_Enable,
  bySetValveMode := VirtualValve0_ValveMode,
  bySetAppControl := VirtualValve0_AppControl,
  bySetAppOption := VirtualValve0_AppOption,
  iSetpointValue1 := VirtualValve0_SetpointValue1,
  iSetpointValue2 := VirtualValve0_SetpointValue2,
  xManualAcknowledge := VirtualValve0_ManualAcknowledge,
  byActualValveMode => VirtualValve0_ActualValveMode,
  byActualValveState => VirtualValve0_ActualValveState,
  byActualAppState => VirtualValve0_ActualAppState,
  iActualValue1 => VirtualValve0_ActualValue1,
  iActualValue2 => VirtualValve0_ActualValue2,
  iResponseToValveModeSet => VirtualValve0_ResponseToValveModeSet,
  iErrorCode => VirtualValve0_ErrorCode);
  
```

Figura 52. Definición de una válvula virtual para lectura de parámetros.

Una vez expuesto uno de los conceptos que se encuentran detrás de la adquisición de los datos de caudal de las válvulas, se va a presentar el concepto del modo de transferencia. Buscando en los manuales de programación se puede encontrar el canal 25 y *addressed target* 25 en que se encuentra la información deseada.


Transfer Mode
Channel 25 (information)

Addressed target 14 (Flow rates)

Index	Meaning	Digit value
22	Flow rate at valve 0, air port (2)	0.1 l/min
24	Flow rate at valve 0, air port (4)	0.1 l/min
26	Flow rate at valve 0, air port (2) (low byte, signed) and (4) (high byte, signed)	10 l/min

Figura 53. Información de obtención del caudal mediante el modo de transferencia.



Con esta figura ya se puede comprender la resolución cuál es el origen de los 10l/min de resolución que quedaba por escrito en páginas anteriores. Con la cuestión de la resolución resulta, queda por explicar el porqué de las mediciones erróneas, y para ello se tiene que volver al código.

Primero, se tiene que definir en el código algunos parámetros que definan el tipo de datos que se quiere leer. Este es el que se encuentra en la **Figura 54**. Definición de parámetros para la lectura del caudal..

```
Valve0_UploadSingleParameter : BOOL := TRUE;  
VirtualValve0_FB_Upload_Single_Parameter : FestoVTEMdc.FB_Upload_Single_Parameter;  
VirtualValve0_FB_Upload_Single_Parameter_Execute : BOOL := FALSE;  
VirtualValve0_FB_Upload_Single_Parameter_Channel : INT := 25;  
VirtualValve0_FB_Upload_Single_Parameter_AddressedTarget : INT := 14;  
VirtualValve0_FB_Upload_Single_Parameter_Index : INT := 26;  
VirtualValve0_FB_Upload_Single_Parameter_Response : FestoVTEMdc.eTransferResponse;  
VirtualValve0_FB_Upload_Single_Parameter_UploadedValue : INT;
```

Figura 54. Definición de parámetros para la lectura del caudal.

Realmente, se quiere hacer ver cómo la información que se define en las líneas de código es la que se encuentra en la figura del manual (**Figura 53**. Información de obtención del caudal mediante el modo de transferencia.). El auténtico quid de la cuestión reside en el siguiente bloque código.

```
(* Activacion de las lecturas *)  
VirtualValve0_FB_Upload_Single_Parameter_Execute := TRUE;  
  
ELSE  
  
(* Lectura correcta *)  
IF VirtualValve0_FB_Upload_Single_Parameter_Execute AND VirtualValve0_FB_Upload_Single_Parameter_UploadDone AND  
VirtualValve0_FB_Upload_Single_Parameter_UploadError = FALSE  
THEN  
  
(* Memorizacion del valor leído *)  
  
valve0_port24_flow := INT_TO_WORD(VirtualValve0_FB_Upload_Single_Parameter_UploadedValue);  
  
(* Anulacion de la orden de lectura *)  
VirtualValve0_FB_Upload_Single_Parameter_Execute := FALSE;
```

En la figura se aprecia una condición muy importante. Y, es que, únicamente se hace un guardado de los valores cuando la válvula virtual decide que la lectura se ha realizado de manera satisfactoria. El funcionamiento detrás de la decisión se encuentra en el *firmware* de la válvula y es por ello desconocido para el autor. No obstante, la condición en texto estructurado es decisión del programador y resulta

en que cuando durante un periodo no se obtenga una lectura correcta del dato, este se mantenga constante desde el último valor correcto que se tuvo.

Siguiendo, con los tipos de datos disponibles a los que se tiene acceso, se presentan los datos obtenidos de los transmisores de posición. Como se puede apreciar en la forma de esta magnitud, los datos parecen proporcionar poca información. Esto se debe a que las mediciones se realizaron con sensores de márgenes de detección de movimiento de 160 mm y 200 mm para los cilindros de las marcas Festo y SMC, respectivamente. Justamente este es el resultado del empleo de los transmisores de posición que se mencionaba en el 487.

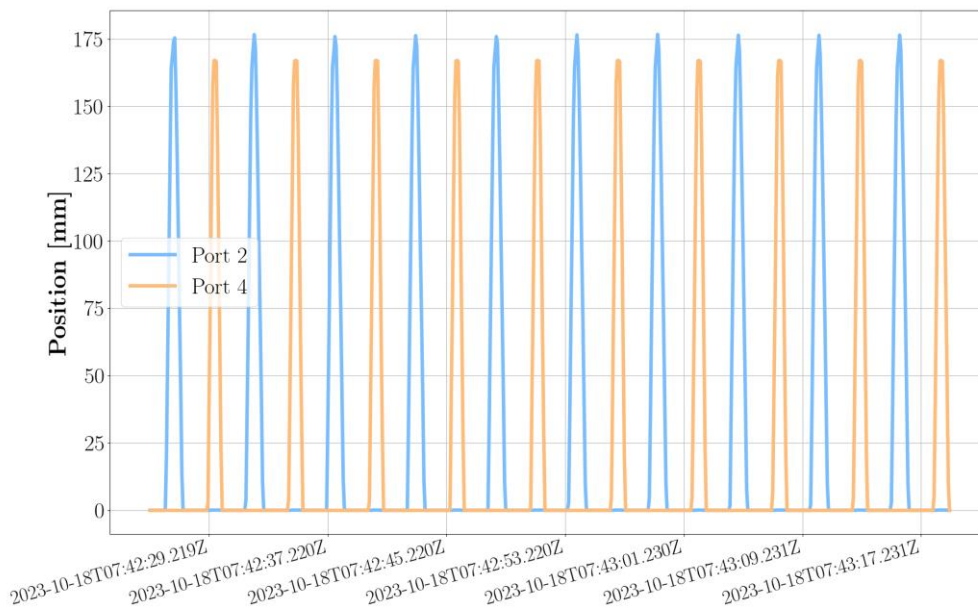


Figura 55. Datos sin tratar de la señal de posición.

La resolución proporcionada por ambos sensores de posición es de 0,5 mm [22] [23], la cual se verifica empíricamente como suficiente para las dinámicas de los cilindros. Sin embargo, como se puede inferir, es el rango de medición el que impide una aplicación adecuada de estos datos para el cálculo de indicadores que podrían resultar útiles. Algunos ejemplos de indicadores que no ha sido posible calcular incluyen el caudal desplazado en ambas cámaras, la velocidad y aceleración de los cilindros, y el trabajo termodinámico asociado a los movimientos de entrada y salida, entre otros.

12. Algoritmos desarrollados

En este apartado se recogen todos los algoritmos que han sido desarrollados para los resultados que se muestran a continuación. Para no solapar las competencias de los dos trabajos de fin de máster, se cree conveniente no entrar muy en detalle en la explicación del funcionamiento de estos, pero sí que se cree conveniente conocer la idea general de su funcionamiento. De todas maneras, se recuerda al lector que puede encontrar una explicación extensa cerca del funcionamiento de los algoritmos y del código de estos en el trabajo complementario que se guarda en las referencias [1].

12.1. **Autoencoder**

Los modelos autoencoder son un tipo de red neuronal que se utiliza para el aprendizaje no supervisado⁹. Está diseñado para codificar los datos de entrada en una representación comprimida y luego decodificarlos para reconstruir la entrada original con la mayor precisión posible. El objetivo principal de los codificadores automáticos es la reducción de dimensionalidad y el aprendizaje de patrones de los datos de entrada.

Su estructura se puede descomponer en tres partes:

- Codificador:

Esta parte de la red neuronal es la que se encarga de tomar los datos de entrada y asignarlos a una representación de dimensiones inferiores, a menudo denominada codificación o espacio latente. Esto resulta en una compresión de la información que se le alimenta.

- Espacio latente:

Son las neuronas que conforman la representación de menor dimensión de los datos. En esta capa es donde se retiene la información más importante.

⁹ El aprendizaje no supervisado lo conforman los problemas en las que no se tienen etiquetas. Es decir, no existe un criterio de qué tipo de comportamiento se está dando en el conjunto de aprendizaje.

- Decodificador:

Estas son las neuronas que intentan reconstruir los datos originales mediante el uso exclusivo de información que se encuentra en el espacio latente. Por norma general, la arquitectura de estas capas suele ser simétrica a la que se construye en el codificador.

Con la arquitectura de este tipo de modelos comprendida, se puede quizás entender de mejor manera el funcionamiento de este tipo de redes neuronales. A muy grandes rasgos, toman la información de entrada, se comprime mediante el uso del espacio latente y se reconstruye a través del decodificador.

Las técnicas de detección de anomalías que se desarrollan en el trabajo complementario analizan la diferencia entre la reconstrucción y los datos originales para dictaminar el grado de anomalía que se da en los datos de entrada. Existen varias métricas que se pueden aplicar para cuantificar el nivel de discrepancias de los datos originales y las reconstrucciones. No obstante, para este caso de estudio se ha empleado la métrica del valor máximo absoluto (MAE).

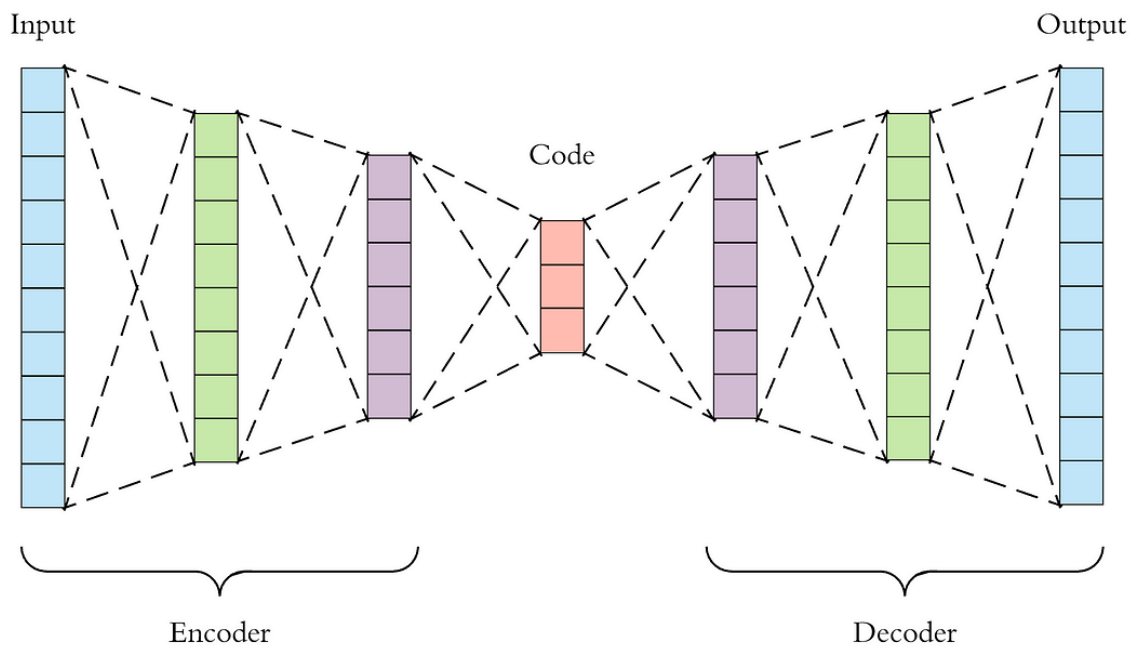


Figura 56. Esquema de arquitectura de AE.

12.2. Modelo VAE

El modelo VAE que ha sido desarrollado es una evolución o modificación del autoencoder que se acaba de exponer. Este tipo de arquitecturas se diseñan específicamente para aprender las representaciones latentes de los datos de entrada. Esto quiere decir que este tipo de modelos aprenden la distribución de los datos y son capaces de generar datos sintéticos, lo cual los convierte en modelos generativos de inteligencia artificial.

La arquitectura de este tipo de modelos varía únicamente en el espacio latente. Como se observa en la Figura 67. Predicción con datos anómalos (TCN). **Figura 57.** Arquitectura del VAE., en el espacio latente se definen neuronas que guardan los parámetros de media y varianza de la distribución normal a la que se ajustan los datos. Para generar datos sintéticos, basta con alimentar al decodificador con ruido gaussiano, el cual en la figura se define con la letra épsilon.

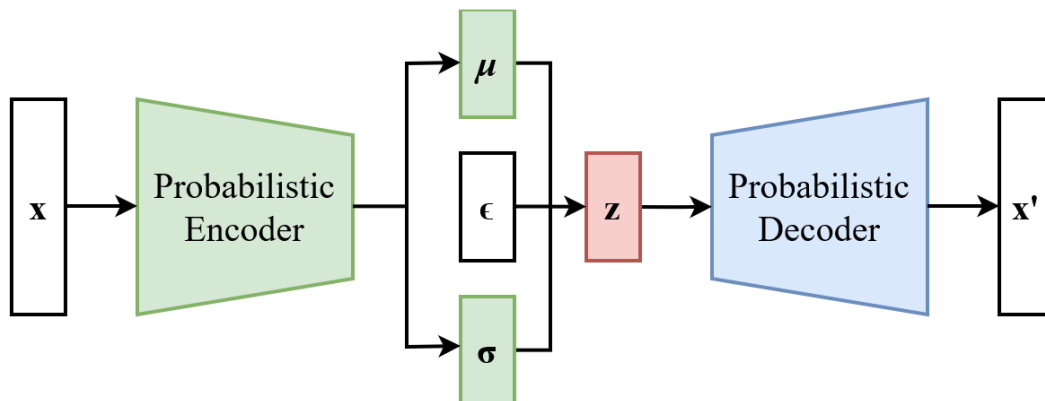


Figura 57. Arquitectura del VAE.

Sin embargo, pese a que la arquitectura pueda diferir, el método de detección de anomalías se sigue realizando de la misma manera. El modelo se sobreentrena con datos normales de funcionamiento y una vez entrenado, se realizan las reconstrucciones de los datos de entrada. Una vez más, si las reconstrucciones varían de gran manera con los datos de entrada, se define el comportamiento ligado a los datos de entrada como anómalo. El indicador empleado en este caso sigue siendo el mismo (MAE).

12.3. Modelo TCN

Este último modelo difiere en gran medida de los dos modelos que ya han sido mostrados. Para empezar, el modelo TCN no reconstruye los datos de entrada, sino que realiza predicciones basadas en la información definida en una ventana móvil. Tanto el número de predicciones a futuro como el tamaño de la ventana son parámetros modificables por el usuario. Sin embargo, las operaciones que se muestran en la **Figura 58**. Arquitectura de modelo TCN. no lo son.

Realmente, son dos las operaciones que se definen en este tipo de modelos: la operación convolución definida para una dimensión y la operación *max-pool*. A muy grandes rasgos, este operador unidimensional se emplea comúnmente en tareas como procesamiento de secuencias, análisis de series temporales y manejo de estructuras de datos unidimensionales como señales o texto. La idea detrás de esta operación es resaltar los patrones que pueda haber en la información de entrada.

A las operaciones de convolución le suelen seguir operaciones de *pooling*; las cuales se encargan de tomar valores máximos o mínimos (según sean *max-pooling* o *min-pooling*). El objetivo de estas es tomar los valores más relevantes de tramos de señal que la convolución esté estudiando. Por último, se quiere mencionar que en este modelo en concreto, la penúltima capa es una capa completamente conectada y la última capa el vector de valores predichos por el modelo.

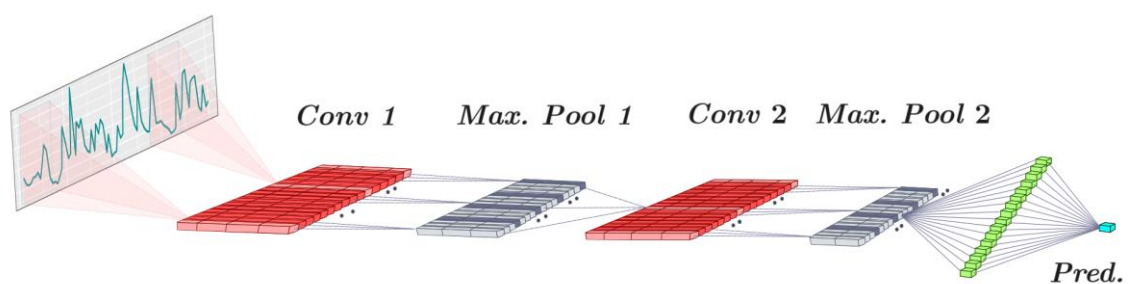


Figura 58. Arquitectura de modelo TCN.

Al igual que los modelos AE y VAE, se puede definir una métrica (por ejemplo, nuevamente el MAE) que permite identificar si los datos que está evaluando el modelo son anómalos o no.

13. Resultados

Como bien se ha mencionado en el apartado anterior, han sido 3 los modelos aplicados a los datos del BEC: un autoencoder profundo, un autoencoder variacional y una red neuronal de arquitectura TCN. Como ya ha sido vaticinado, no se va a entrar en la estructura de ellos, pero sí que se van a mostrar los resultados, los cuales ya se adelanta que han sido muy prometedores.

13.1. Autoencoder

El autoencoder profundo formado por 4 capas y una capa de salida es muy capaz de realizar las reconstrucciones fiables de datos. Esto es mismamente lo que se observa en la **Figura 59**. reconstrucción con datos normales (autoencoder)..

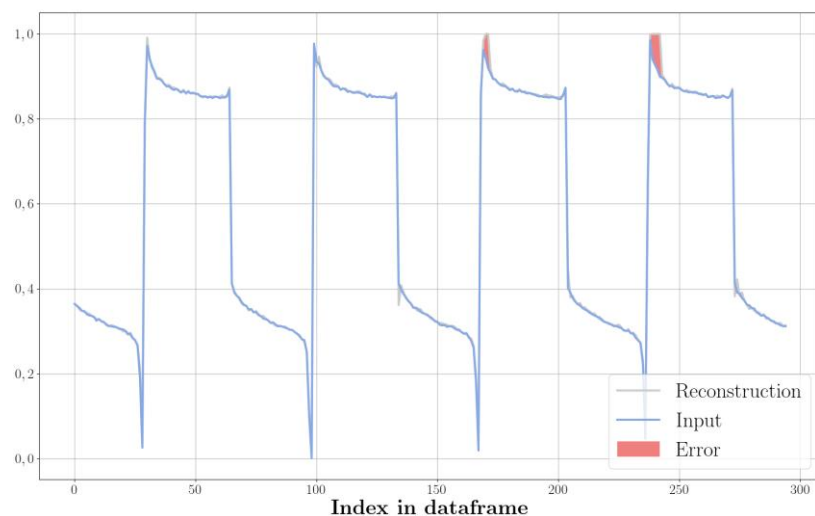


Figura 59. reconstrucción con datos normales (autoencoder).

Para el caso de que se le alimenten los datos correspondientes al movimiento del cilindro sin la junta, el error que se puede observar es realmente grande. De hecho, es tal la diferencia entre las dos gráficas que si se tuviera un algoritmo de clasificación este devolvería una clasificación perfecta. Esto es lo que se trata de ilustrar con la **Figura 61**. Histogramas de comportamientos normales y defectuosos..

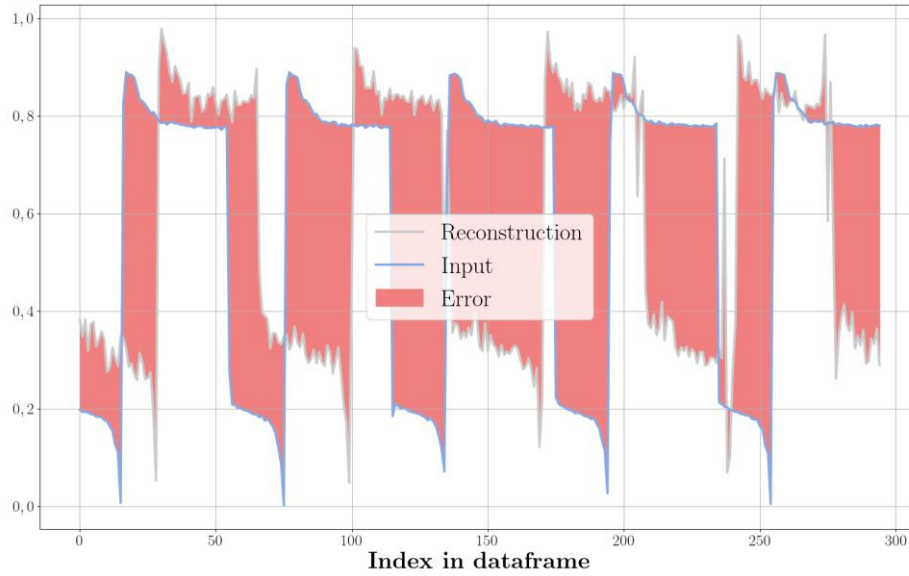


Figura 60. Reconstrucción con datos anómalos (autoencoder).

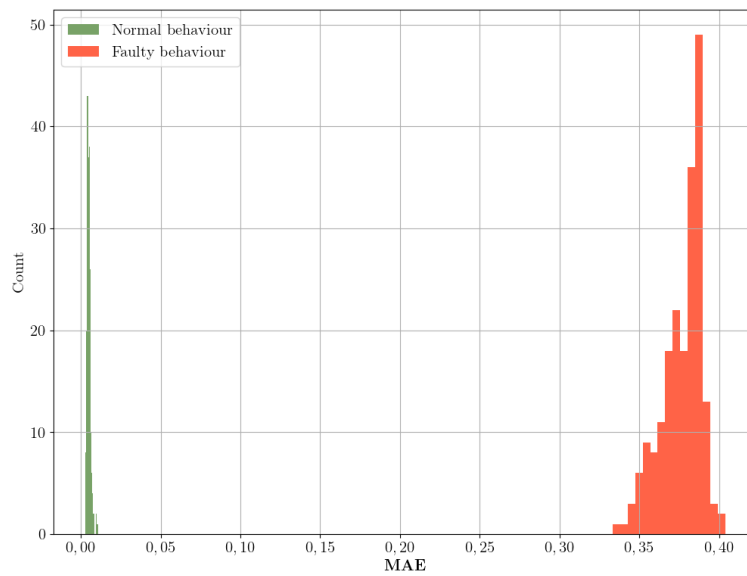


Figura 61. Histogramas de comportamientos normales y defectuosos.

13.2. Autoencoder variacional

Como su nombre indica este tipo de modelos guardan una gran similitud con los autoencoder normales, puesto que la tarea de estos modelos es también la de reconstruir los datos que se le introduzcan de manera que se minimice el error de reconstrucción. La distinción de este tipo de modelos es su potencial como modelos de inteligencia artificial generativa. Este tipo de modelos son capaces, mediante la definición de un espacio latente, de aprender de la distribución de los datos de entrada y generar así datos sintéticos.

De igual manera que ocurría en el caso del autoencoder profundo, el modelo es capaz de reconstruir con un error pequeño los datos que se asemejan al comportamiento normal de los cilindros. Esto es lo que se muestra en la **Figura 62**. Reconstrucción con datos normales (VAE).

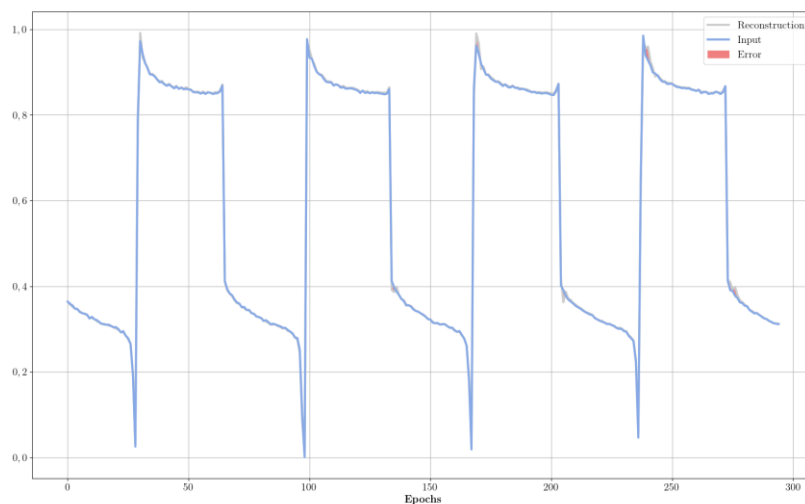


Figura 62. Reconstrucción con datos normales (VAE).

De manera análoga a lo que ocurre con el modelo anterior, el error de reconstrucción de los datos referidos al comportamiento del cilindro sin la junta da como resultado un error de reconstrucción gratamente notable (**Figura 63**. Reconstrucción con datos anómalos (VAE)).

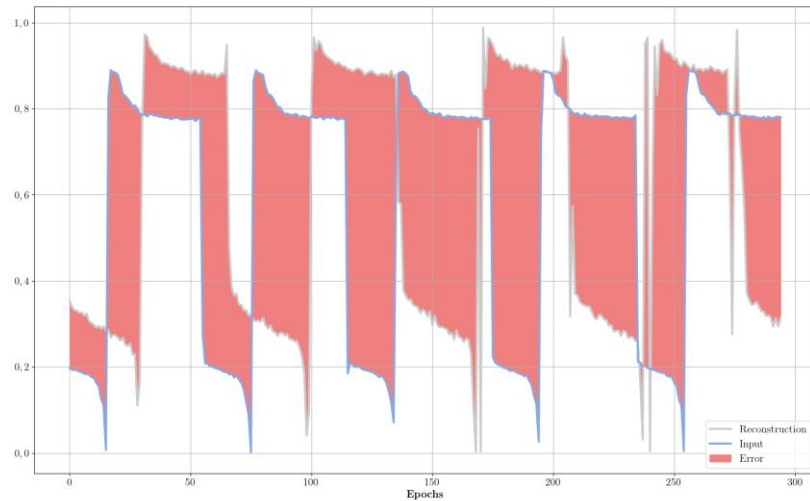


Figura 63. Reconstrucción con datos anómalos (VAE).

Como se expresaba al comienzo del capítulo, al tratarse de un modelo de inteligencia artificial generativa, este tipo de modelos son capaces de generar datos sintéticos. En la **Figura 64**. Datos sintéticamente generados por el VAE. Se muestran 3 ejemplos de generación de datos sintéticos. Se quiere recalcar que estos se tratan de señales producidas por la inteligencia, y que no son señales que se hayan dado en la realidad.

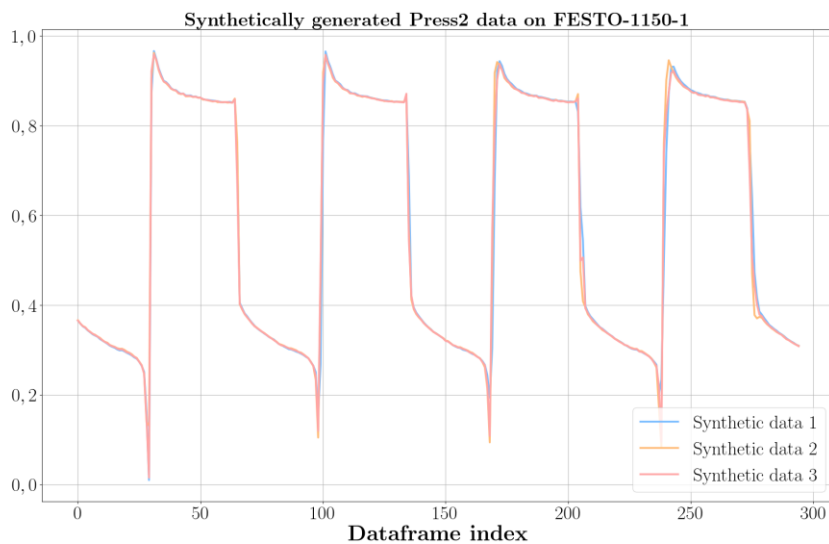


Figura 64. Datos sintéticamente generados por el VAE.

De una manera análoga a que sucedía con el modelo previamente mostrado, la posibilidad de la clasificación de este modelo es perfecta para todos los datos que han sido manipulados. Muestra de esto es la **Figura 65**. Histogramas de comportamientos normales y defectuosos (VAE).

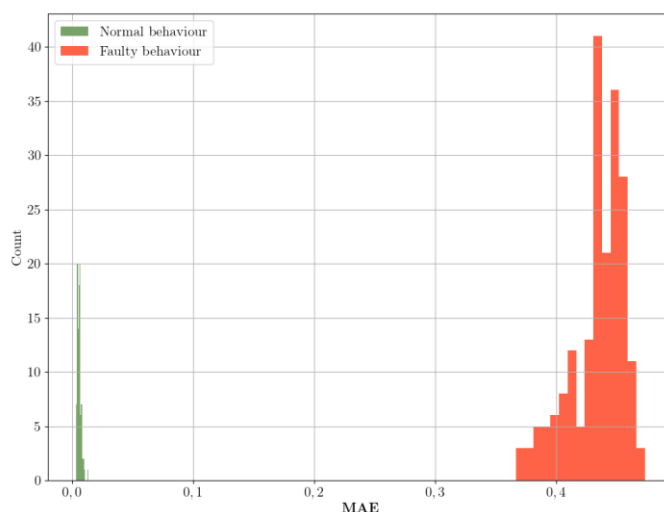


Figura 65. Histogramas de comportamientos normales y defectuosos (VAE)

13.3. TCN

Es el último modelo que se define para el estudio realizado. Debido a su naturaleza el modelo trata de predecir el siguiente valor de las series temporales y estos son los resultados que se obtienen con datos normales y anómalos (**Figura 66. Predicción con datos normales (TCN).** y **Figura 67. Predicción con datos anómalos (TCN).**).

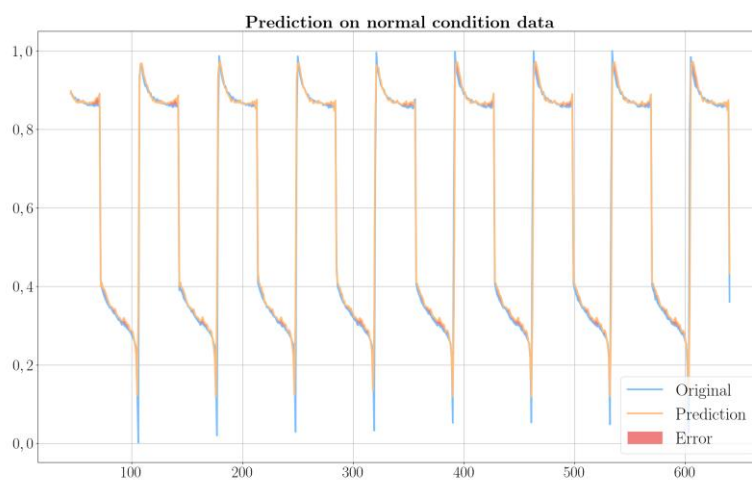


Figura 66. Predicción con datos normales (TCN).

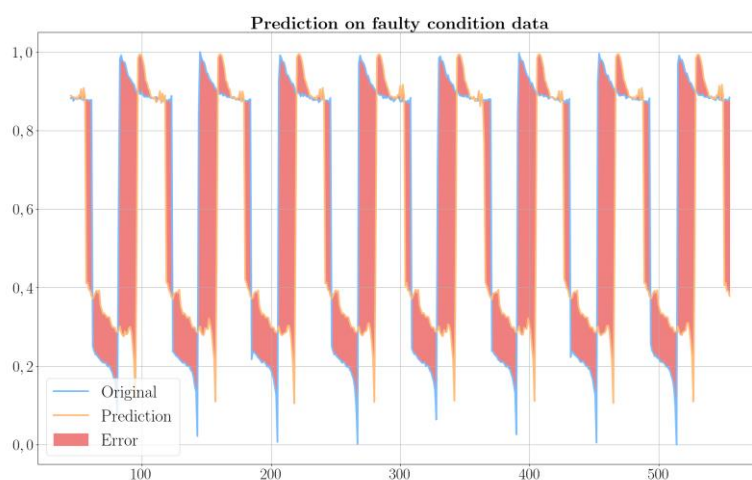


Figura 67. Predicción con datos anómalos (TCN).



14. Descripción del presupuesto

A lo largo de la vida del proyecto han sido efectuadas varias compras a distintos proveedores, de forma que cada compra cumplía con los requerimientos técnicos, y; sobre todo, logísticos. Se desea incidir en este último punto, puesto que los largos plazos de entrega de los componentes electrónicos (tales como los controladores) han supuesto tener que realizar unas planificaciones flexibles en la gestión de las tareas.

Tabla 8. Desglose de costes de los elementos de control.

Descripción		Precio Unitario	Cantidad	Valor Neto
Denominación	Código de referencia			
Motion Terminal VTEM	VTEM-S1-27-E1 -B1-Q10-U-Q8-4C-BMA -4SPL-DL 50E- F37GCQSIGWIGW-P	3.288,45 €/ud.	1	3.288,45 €
Unidad de indicación y control	CDPX-X-E1-W-7	1.172,74 €/ud.	1	1.172,74 €
Interfaz EtherCAT®	CPX-AP-I-EC-M12	259,96 €/ud.	1	259,96 €
Módulo de entrada/salida digital	CPX-AP-I-4DI4DO-M8-3P	103,99 €/ud.	1	103,99 €
Cable de conexión (I)	NEBC-D12G4-ES-5-S- D12G4-ET	26,98 €/ud.	1	26,98 €
Cable de conexión (II)	NEBC-D12G4-ES-5-S- R3G4-ET	27,68 €/ud.	1	27,68 €
Cable de conexión (III)	NEBC-D8G4-ES-0.5-N-S- D8G4-ET	17,33 €/ud.	1	17,33 €
Cable de conexión (IV)	NEBL-M8G4-E-5-N-LE4	11,70 €/ud.	1	11,70 €
Cable de conexión (V)	NEBL-M8G4-E-0.5-N- M8G4	8,66 €/ud.	1	8,66 €
Cilindro normalizado (I)	DSBC-63-1400-PPSA-N3	186,54 €/ud.	1	186,54 €
Cilindro normalizado (II)	DSBC-63-1150-PPSA-N3	170,62 €/ud.	1	170,62 €
Unidad de mantenimiento	MSB6-1/2:C4:H1:H4-WP	143,12 €/ud.	1	143,12 €
			Total Neto	5.417,77 €



Tabla 9. Desglose de costes de los cilindros y sus componentes.

Descripción		Precio Unitario	Cantidad	Valor Neto
Denominación	Código de referencia			
Cilindro normalizado (I)	CP96SDB63-1400C	173,35 €/ud.	2	274,7 €
Cilindro normalizado (II)	CP96SDB63-1400C	157,37 €/ud.	2	314,74 €
Escuadra	L5063	10,40 €/ud.	2	20,80 €
			Total Neto	610,24 €

En total, el presupuesto del proyecto asciende a los **6.028,01€**. Esto únicamente tiene en cuenta los elementos materiales del montaje del banco de cilindros. Por supuesto, detrás del proyecto se ha tenido el apoyo de un ingeniero junior (el autor) que ha estado inmiscuido en el proyecto durante 4 meses.

Además, también ha sido necesario la ayuda de un ingeniero senior para la ayuda con la programación del movimiento de los cilindros. El coste de este personal es de unos 8.860,8€ aproximadamente. En el caso del ingeniero junior, su coste puede aproximarse a los 4.800€. Por lo tanto, pese a ser estimaciones, se puede prever un presupuesto de alrededor de los casi **60.000€** para un proyecto como el que se ha realizado en este trabajo.

14.1. Análisis de rentabilidad

Aunque pueda parecer una suma de dinero desmesurada para la creación de unos algoritmos, no se ha de perder el enfoque en la utilidad de estos. Una detección de parada anticipada puede traer consigo una disminución en las pérdidas de los costes de producción.

Suponiendo que el coste de parada de una línea de producción sea de unos **20.300€/min** [25] y que el tiempo de cambio de un cilindro neumático sea de unos 15 minutos, un algoritmo que sea capaz de detectar un comportamiento anómalo puede ser rentable con una única parada que sea capaz de evitar.



Sin embargo, bien es cierto que las paradas no se darán ni mensualmente, ni quizás anualmente, por lo que quizás haya que esperar 2 o 3 años hasta que el algoritmo salvaguarde una parada.

Tabla 10. Análisis de retorno de inversión.

Coste aproximado total del proyecto	60.000 €
Pérdidas de parada de producción	20.300€/min
Tiempo aprox. de recambio de cilindro	15 min
Probabilidad diaria de fallo aprox.	$5 \cdot 10^{-6}$

$$\text{Retorno de Inversión} = \frac{60.000€}{(20.300€/min \cdot 15 \cdot 10^{-6})} = 2,82 \text{ años}$$

Conclusiones y líneas futuras

En cuanto al montaje del banco de ensayos el éxito ha sido rotundo. Pese a haber tenido grandes contratiempos en cuanto a la entrega del material se trata, se han podido solucionar el retraso de los plazos de una manera extraordinaria. Asimismo, el control de los cilindros montados ha cumplido los requisitos exigidos por parte del director del proyecto.

Por si fuera poco, la sensórica instalada ha sido suficiente para conocer la respuesta al fallo de los elementos neumáticos, y, pese a que exista una extensa línea de trabajo por descubrir en los posibles datos que se puedan obtener del banco, los sensores han resultado suficientes para implantar un novedoso sistema de detección de anomalías.

En el contexto de los resultados prometedores, es esencial destacar que, a pesar de las limitaciones de recursos, se ha logrado establecer una base sólida para la aplicación de la inteligencia artificial en el mantenimiento industrial. La falta de pruebas de validación adicional debido a la degradación de los cilindros ofrece una oportunidad para futuras investigaciones, con el potencial de profundizar en la eficacia y aplicabilidad de los modelos desarrollados.

Una perspectiva futura crucial implica la implementación de algoritmos de optimización en los modelos existentes. La sugerencia de utilizar algoritmos de PSO no solo se basa en su eficacia probada, sino también en su coherencia con el contenido del máster, proporcionando así una dirección clara para el desarrollo continuo.

En cuanto a los intentos de incorporar modelos generativos de inteligencia artificial, como el Generative Adversarial Network (GAN), los resultados insatisfactorios subrayan la complejidad y los desafíos involucrados. A pesar de la falta de dedicación de tiempo en este estudio, la mención de estos resultados abre la puerta a futuras investigaciones que podrían abordar y superar las limitaciones encontradas. En conclusión, este estudio ha explorado de manera cautelosa el vasto terreno de la implementación de la inteligencia artificial en el ámbito industrial. El autor se despide con la expectativa de seguir de cerca la evolución de la inteligencia artificial y su creciente impacto en la industria, manteniendo un interés continuo en el desarrollo y la innovación en este fascinante campo.

Referencias

- [1] J. C. d. Cid, «Desarrollo de métodos de detección de anomalías en componentes industriales basados en Inteligencia Artificial,» Bilbao, 2024.
- [2] DANOBTGROUP, «Corporativo,» 2023. [En línea]. Available: <https://www.danobatgroup.com/es/danobatgroup>. [Último acceso: 19 abril 2023].
- [3] SORALUCE, «SORALUCE. Producto,» 2023. [En línea]. Available: <https://www.soraluce.com/es/producto>. [Último acceso: 20 abril 2023].
- [4] DANOBTGROUP, «GOIMEK. Corporativo,» 2023. [En línea]. Available: <https://www.goimek.com/es/presentacion>. [Último acceso: 20 abril 2023].
- [5] DANOBTGROUP, «DANOBT. Rectificado,» 2023. [En línea]. Available: <https://www.danobatgroup.com/es/danobat>. [Último acceso: 20 abril 2023].
- [6] Festo, “Standards-based cylinders DSBC, ISO 15552,» mayo 2023. [Online]. Available: <https://www.festo.com/media/pim/132/D15000100122132.PDF>. [Accessed 20 junio 2023].
- [7] SMC, “smcpneumatics,» [Online]. Available: <https://www.smcpneumatics.com/pdfs/CP96.pdf>. [Accessed 19 mayo 2023].
- [8] Festo, “Motion Terminal,» febrero 2023. [Online]. Available: <https://www.festo.com/media/pim/013/D15000100150013.PDF>. [Accessed 23 abril 2023].

- [9] SAVVY DATA SYSTEMS, "Quiénes somos," 2023. [Online]. Available: <https://www.savvydatasystems.com/es/quienes-somos>. [Accessed 5 junio 2023].
- [10] J. H. S. Y. I. K. B. O. C. C. S. L. B. S. K. Mu Seong Chang, «Reliability Estimation of Pneumatic Cylinders using Performance Degradation Data,» *International Journal of Precision Engineering and Manufacturing*, nº 14, pp. 2081-2086, 2013.
- [11] S. Woo, "Estimating the Lifetime of the Pneumatic Cylinder in Machine Tool Subjected to Repetitive Pressure Loading," *Journal of US-China Public Administration*, vol. 15, 2018.
- [12] C.-S. L. B.-O. C. B.-S. K. Mu-Seong Chang, «Study on validation for accelerated life tests of pneumatic cylinders based on the test results of normal use conditions,» *Journal of Mechanical Science and Technology*, vol. 31, p. 2739–2745, 2017.
- [13] E. Z. J. L. Z. Z. a. C. B. J. Chen, "Accelerated Life Test for Reliability Evaluation of Pneumatic Cylinders," *IEEE Access*, vol. 6, pp. 75062-75075, 2018.
- [14] Y. I. K. B. S. K. Mu Seong Chang, "Design of reliability qualification test for pneumatic cylinders based on performance degradation data," *Journal of Mechanical Science and Technology*, vol. 28, p. 4939–4945, 2014.
- [15] Festo, "Motion Terminal VTEM," junio 2021. [Online]. Available: https://www.festo.com/net/en_en/SupportPortal/Files/730478/118488_Festo_VTEM-MotionTerminal_PSIplus_EN135847_202106_V01.pdf. [Accessed 23 julio 2023].



- [16] Festo, "Exploring The Benefits Of The Festo Motion Terminal VTEM," 2023. [Online]. Available: <https://festoblog.com/exploring-the-benefits-of-the-festo-motion-terminal-vtem/?cn-reloaded=1>. [Accessed 24 julio 2023].
- [17] Baumer, "GCA3-PM," Baumer, 2024. [Online]. Available: <https://www.baumer.com/se/en/product-overview/distance-measurement/cable-transducers/gca3/gca3-pm---analog-up-to-4-7-m/p/46688>. [Accessed 12 August 2023].
- [18] FESTO, "Motion Terminal Quick Reference PLC programming," febrero 2023. [Online]. Available: https://www.festo.com/net/SupportPortal/Files/684271/VTEM_PLC_Quick_Reference_V1-3.pdf. [Accessed 12 febrero 2024].



ANEXO I: CÓDIGO DE CONTROL DE MOVIMIENTO



(*****)

(* Ciclo inicial *)

IF firstCycle

THEN

valve0_minimum_pressure := MINIMUM_PRESSURE;

valve1_minimum_pressure := MINIMUM_PRESSURE;

valve2_minimum_pressure := MINIMUM_PRESSURE;

valve3_minimum_pressure := MINIMUM_PRESSURE;

valve0_recordingPendant := FALSE;

valve0_recordingActive := FALSE;

valve0_recordingLastStartCiclos := valve0_totalNumeroCiclos;

valve1_recordingPendant := FALSE;

valve1_recordingActive := FALSE;

valve1_recordingLastStartCiclos := valve1_totalNumeroCiclos;

valve2_recordingPendant := FALSE;

valve2_recordingActive := FALSE;

valve2_recordingLastStartCiclos := valve2_totalNumeroCiclos;

valve3_recordingPendant := FALSE;

valve3_recordingActive := FALSE;

valve3_recordingLastStartCiclos := valve3_totalNumeroCiclos;

firstCycle := FALSE;

END_IF



```

(*****)

(* Verificacion de emergencia para deshabilitacion de valvulas *)

IF EMERGENCY_OK = FALSE

THEN

    (* NOTA: Mantener Enable para liberar presiones *)

    Valve0_Enable := TRUE;

    Valve1_Enable := TRUE;

    Valve2_Enable := TRUE;

    Valve3_Enable := TRUE;

    Valve0_AppControl := 0;

    Valve1_AppControl := 0;

    Valve2_AppControl := 0;

    Valve3_AppControl := 0;

    IF previousEMERGENCY_OK = TRUE

    THEN

        IF      (valve0_recordingCadenciaCiclos      -      (valve0_totalNumeroCiclos      -
valve0_recordingLastStartCiclos)) < 10

        THEN

            valve0_recordingLastStartCiclos := valve0_recordingLastStartCiclos + 10;

        END_IF

        IF      (valve1_recordingCadenciaCiclos      -      (valve1_totalNumeroCiclos      -
valve1_recordingLastStartCiclos)) < 10

        THEN

            valve1_recordingLastStartCiclos := valve1_recordingLastStartCiclos + 10;

        END_IF

```




```
IF (valve2_recordingCadenciaCiclos - (valve2_totalNumeroCiclos -
valve2_recordingLastStartCiclos)) < 10

THEN

    valve2_recordingLastStartCiclos := valve2_recordingLastStartCiclos + 10;

END_IF

IF (valve3_recordingCadenciaCiclos - (valve3_totalNumeroCiclos -
valve3_recordingLastStartCiclos)) < 10

THEN

    valve3_recordingLastStartCiclos := valve3_recordingLastStartCiclos + 10;

END_IF

END_IF

ELSE

IF previousEMERGENCY_OK = FALSE

THEN

    valve0_enabled := FALSE;

    valve1_enabled := FALSE;

    valve2_enabled := FALSE;

    valve3_enabled := FALSE;

    Valve0_AppControl := Valve0_lastAppControl;

    Valve1_AppControl := Valve1_lastAppControl;

    Valve2_AppControl := Valve2_lastAppControl;

    Valve3_AppControl := Valve3_lastAppControl;

END_IF

Valve0_Enable := valve0_enabled AND (valve1_recordingActive = FALSE) AND
(valve2_recordingActive = FALSE) AND
(valve3_recordingActive = FALSE);

Valve1_Enable := valve1_enabled AND (valve0_recordingActive = FALSE) AND
```

```

      (valve3_recordingActive = FALSE);
      Valve2_Enable := valve2_enabled AND (valve0_recordingActive = FALSE) AND
      (valve3_recordingActive = FALSE);
      Valve3_Enable := valve3_enabled AND (valve0_recordingActive = FALSE) AND
      (valve2_recordingActive = FALSE);

      Valve0_lastAppControl := Valve0_AppControl;
      Valve1_lastAppControl := Valve1_AppControl;
      Valve2_lastAppControl := Valve2_AppControl;
      Valve3_lastAppControl := Valve3_AppControl;

    END_IF

    previousEMERGENCY_OK := EMERGENCY_OK;

    (*****)

    (* Control valvula 0 *)

    Valve0_FB_ValveControl(
      awBusDataFromVTEM := Valve0_Inputs,
      awBusDataToVTEM := Valve0_Outputs,
      xEnable := Valve0_Enable,
      bySetValveMode := Valve0_ValveMode,
      bySetAppControl := Valve0_AppControl,
      bySetAppOption := Valve0_AppOption,
      iSetpointValue1 := Valve0_SetpointValue1,
      iSetpointValue2 := Valve0_SetpointValue2,
      xManualAcknowledge := Valve0_ManualAcknowledge,
  
```



```
byActualValveMode => Valve0_ActualValveMode,  
byActualValveState => Valve0_ActualValveState,  
byActualAppState => Valve0_ActualAppState,  
iActualValue1 => Valve0_ActualValue1,  
iActualValue2 => Valve0_ActualValue2,  
iResponseToValveModeSet => Valve0_ResponseToValveModeSet,  
iErrorCode => Valve0_ErrorCode);
```

(* Verificacion del estado *)

```
valve0_running := (Valve0_ValveMode = Valve0_ActualValveMode);
```

```
valve0_acknowledgeNeeded := (Valve0_ResponseToValveModeSet =  
FestoVTEMDc.eResponseToValveModeSet.acknowledge_needed);
```

(* Led de status *)

```
Valve0_StatusLED := 0;
```

```
IF valve0_enabled
```

```
THEN
```

```
IF valve0_acknowledgeNeeded = FALSE
```

```
THEN
```

```
Valve0_StatusLED := 1;
```

```
ELSE
```

```
Valve0_StatusLED := 2;
```

```
END_IF
```

```
END_IF
```

(* Control de inversiones del desplazamiento *)

```
valve0_inicioCiclo := FALSE;
```

```
IF Valve0_StatusLED = 1 AND Valve0_AppControl = 1 AND Valve0_ActualValue2 < valve0_minimum_pressure
```

```
THEN
```

```
    valve0_control1CounterLimit := valve0_control1CounterLimit + 1;
```

```
IF valve0_control1CounterLimit > 2
```

```
THEN
```

```
    (* Inicio de un nuevo ciclo *)
```

```
    valve0_inicioCiclo := TRUE;
```

```
    (* Cambio de sentido *)
```

```
    Valve0_AppControl := 2;
```

```
    (* Incremento del numero de ciclos. Solamente se realiza en uno de los cambios. *)
```

```
    valve0_totalNumeroCiclos := valve0_totalNumeroCiclos + 1;
```

```
    (* Control de ejecucion de medidas *)
```

```
    IF valve0_record = TRUE
```

```
    THEN
```

```
        IF valve0_recordingActive = TRUE
```

```
        THEN
```

```
            valve0_recordingNumeroCiclos := valve0_recordingNumeroCiclos + 1;
```

```
            IF valve0_recordingNumeroCiclos >= valve0_recordingDuracionCiclos
```

```
            THEN
```

```
                valve0_recordingActive := FALSE;
```

```
                valve0_recordingNumeroCiclos := 0;
```

```
                valve0_totalNumeroMediciones := valve0_totalNumeroMediciones +
```

```
1;
```



```

                                END_IF
                            END_IF
                        END_IF
                    END_IF
                ELSE
                    valve0_control1CounterLimit := 0;
                END_IF

IF Valve0_StatusLED = 1 AND Valve0_AppControl = 2 AND Valve0_ActualValue1 < valve0_minimum_pressure
THEN
    valve0_control2CounterLimit := valve0_control2CounterLimit + 1;

    IF valve0_control2CounterLimit > 2
    THEN
        Valve0_AppControl := 1;
    END_IF
ELSE
    valve0_control2CounterLimit := 0;
END_IF

(* Control de ejecucion de medidas *)
IF valve0_record = TRUE
THEN
    IF valve0_recordingActive = FALSE AND
        valve0_recordingPendant = FALSE AND
        (valve0_totalNumeroCiclos - valve0_recordingLastStartCiclos)
        valve0_recordingCadenciaCiclos >=
```

```
THEN

    valve0_recordingPendant := TRUE;

    IF nextPendantRecordingIndex <= 10

        THEN

            nextPendantRecording[nextPendantRecordingIndex] := 0;

            nextPendantRecordingIndex := nextPendantRecordingIndex + 1;

        END_IF

    END_IF

ELSE

    valve0_recordingActive := FALSE;

    valve0_recordingNumeroCiclos := 0;

END_IF

IF (EMERGENCY_OK = FALSE) OR (Valve0_Enable = FALSE)

THEN

    valve0_recordingActive := FALSE;

    valve0_recordingNumeroCiclos := 0;

END_IF

(*****)

(* Control de valvula virtual para lectura de parametros *)

VirtualValve0_ValveMode := 63; (* Transfer Mode *)

VirtualValve0_AppControl := 2; (* Upload (read) *)

VirtualValve0_AppOption := 0;

VirtualValve0_SetpointValue1 := 0;

VirtualValve0_SetpointValue2 := 0;
```



```
VirtualValve0_FB_ValveControl(  
  
    awBusDataFromVTEM := VirtualValve0_Inputs,  
  
    awBusDataToVTEM := VirtualValve0_Outputs,  
  
    xEnable := VirtualValve0_Enable,  
  
    bySetValveMode := VirtualValve0_ValveMode,  
  
    bySetAppControl := VirtualValve0_AppControl,  
  
    bySetAppOption := VirtualValve0_AppOption,  
  
    iSetpointValue1 := VirtualValve0_SetpointValue1,  
  
    iSetpointValue2 := VirtualValve0_SetpointValue2,  
  
    xManualAcknowledge := VirtualValve0_ManualAcknowledge,  
  
    byActualValveMode => VirtualValve0_ActualValveMode,  
  
    byActualValveState => VirtualValve0_ActualValveState,  
  
    byActualAppState => VirtualValve0_ActualAppState,  
  
    iActualValue1 => VirtualValve0_ActualValue1,  
  
    iActualValue2 => VirtualValve0_ActualValue2,  
  
    iResponseToValveModeSet => VirtualValve0_ResponseToValveModeSet,  
  
    iErrorCode => VirtualValve0_ErrorCode);
```

(*****)

(* Lectura de un parametro simple *)

IF Valve0_UploadSingleParameter

THEN

```
VirtualValve0_FB_Upload_Single_Parameter(  
  
    awBusDataFromVTEM := VirtualValve0_Inputs,  
  
    awBusDataToVTEM := VirtualValve0_Outputs,  
  
    xExecute := VirtualValve0_FB_Upload_Single_Parameter_Execute, (* Rising edge *)
```

```
iChannel := VirtualValve0_FB_Upload_Single_Parameter_Channel,  
iAddressedTarget := VirtualValve0_FB_Upload_Single_Parameter_AddressedTarget,  
iIndex := VirtualValve0_FB_Upload_Single_Parameter_Index,  
iResponse => VirtualValve0_FB_Upload_Single_Parameter_Response,  
iUploadedValue => VirtualValve0_FB_Upload_Single_Parameter_UploadedValue,  
xUploadDone => VirtualValve0_FB_Upload_Single_Parameter_UploadDone,  
xUploadError => VirtualValve0_FB_Upload_Single_Parameter_UploadError);
```

```
valve0_previousExecute := VirtualValve0_FB_Upload_Single_Parameter_Execute;  
valve0_counterCycles := valve0_counterCycles + 1;
```

```
IF VirtualValve0_FB_Upload_Single_Parameter_Execute = FALSE
```

```
THEN
```

```
valve0_numberCycles := valve0_counterCycles;
```

```
(* Activacion de las lecturas *)
```

```
VirtualValve0_FB_Upload_Single_Parameter_Execute := TRUE;
```

```
ELSE
```

```
(* Lectura correcta *)
```

```
IF VirtualValve0_FB_Upload_Single_Parameter_Execute AND  
VirtualValve0_FB_Upload_Single_Parameter_UploadDone AND
```

```
VirtualValve0_FB_Upload_Single_Parameter_UploadError = FALSE
```

```
THEN
```


(* Memorizacion del valor leído *)

```
valve0_port24_flow :=  
INT_TO_WORD(VirtualValve0_FB_Upload_Single_Parameter_UploadedValue);
```

(* Anulacion de la orden de lectura *)

```
VirtualValve0_FB_Upload_Single_Parameter_Execute := FALSE;
```

END_IF

(* Lectura incorrecta *)

```
IF VirtualValve0_FB_Upload_Single_Parameter_Execute AND  
VirtualValve0_FB_Upload_Single_Parameter_UploadError
```

THEN

(* Memorizacion del status (?) *)

(*

```
VirtualValve0_FB_Upload_Single_Parameter_Response:
```

```
FestoVTEMdc.eTransferResponse.aborting;
```

```
FestoVTEMdc.eTransferResponse.access_denied_DataInUse;
```

```
FestoVTEMdc.eTransferResponse.access_denied_saving;
```

```
FestoVTEMdc.eTransferResponse.access_denied_WebConfig;
```

```
FestoVTEMdc.eTransferResponse.communication_failed;
```

```
FestoVTEMdc.eTransferResponse.FB_needs_rising_edge_xExecute;
```

```
FestoVTEMdc.eTransferResponse.invalid_AddressedTarget;
```

```
FestoVTEMdc.eTransferResponse.invalid_Channel;
```

```
FestoVTEMdc.eTransferResponse.invalid_Combination;
```

```
FestoVTEMdc.eTransferResponse.invalid_firmware_version;
```

```
FestoVTEMdc.eTransferResponse.invalid_Index;  
  
FestoVTEMdc.eTransferResponse.invalid_module_number;  
  
FestoVTEMdc.eTransferResponse.invalid_ParameterSet;  
  
FestoVTEMdc.eTransferResponse.invalid_TransferControl;  
  
FestoVTEMdc.eTransferResponse.invalid_TransferValue;  
  
FestoVTEMdc.eTransferResponse.no_data_empty_slot;  
  
FestoVTEMdc.eTransferResponse.no_valve_detected;  
  
FestoVTEMdc.eTransferResponse.preparing_transfer;  
  
FestoVTEMdc.eTransferResponse.reading_transfermode;  
  
FestoVTEMdc.eTransferResponse.ready_for_new_transfer;  
  
FestoVTEMdc.eTransferResponse.reserved;  
  
FestoVTEMdc.eTransferResponse.resetting;  
  
FestoVTEMdc.eTransferResponse.transfer_successful;  
  
FestoVTEMdc.eTransferResponse.transferring;  
  
FestoVTEMdc.eTransferResponse.Valve_Mode_is_not_61_inactive;  
  
*)
```

(* Anulacion de la orden de lectura *)

```
VirtualValve0_FB_Upload_Single_Parameter_Execute := FALSE;
```

```
END_IF
```

```
END_IF
```

```
IF valve0_previousExecute = FALSE AND VirtualValve0_FB_Upload_Single_Parameter_Execute = TRUE
```

```
THEN
```

```
valve0_counterCycles := 0;
```

END_IF

END_IF

(*****)

(* Control valvula 1 *)

Valve1_FB_ValveControl(

awBusDataFromVTEM := Valve1_Inputs,

awBusDataToVTEM := Valve1_Outputs,

xEnable := Valve1_Enable,

bySetValveMode := Valve1_ValveMode,

bySetAppControl := Valve1_AppControl,

bySetAppOption := Valve1_AppOption,

iSetpointValue1 := Valve1_SetpointValue1,

iSetpointValue2 := Valve1_SetpointValue2,

xManualAcknowledge := Valve1_ManualAcknowledge,

byActualValveMode => Valve1_ActualValveMode,

byActualValveState => Valve1_ActualValveState,

byActualAppState => Valve1_ActualAppState,

iActualValue1 => Valve1_ActualValue1,

iActualValue2 => Valve1_ActualValue2,

iResponseToValveModeSet => Valve1_ResponseToValveModeSet,

iErrorCode => Valve1_ErrorCode);

(* Verificacion del estado *)

valve1_running := (Valve1_ValveMode = Valve1_ActualValveMode);



```
valve1_acknowledgeNeeded := (Valve1_ResponseToValveModeSet = FestoVTEMdc.eResponseToValveModeSet.acknowledge_needed);
```

```
(* Led de status *)
```

```
Valve1_StatusLED := 0;
```

```
IF valve1_enabled
```

```
THEN
```

```
    IF valve1_acknowledgeNeeded = FALSE
```

```
    THEN
```

```
        Valve1_StatusLED := 1;
```

```
    ELSE
```

```
        Valve1_StatusLED := 2;
```

```
    END_IF
```

```
END_IF
```

```
(* Control de inversiones del desplazamiento *)
```

```
valve1_inicioCiclo := FALSE;
```

```
IF Valve1_StatusLED = 1 AND Valve1_AppControl = 1 AND Valve1_ActualValue2 < valve1_minimum_pressure
```

```
THEN
```

```
    valve1_control1CounterLimit := valve1_control1CounterLimit + 1;
```

```
    IF valve1_control1CounterLimit > 2
```

```
    THEN
```

```
        (* Inicio de un nuevo ciclo *)
```

```
        valve1_inicioCiclo := TRUE;
```

```
        (* Cambio de sentido *)
```

```
Valve1_AppControl := 2;
```

```
(* Incremento del numero de ciclos. Solamente se realiza en uno de los cambios. *)
```

```
valve1_totalNumeroCiclos := valve1_totalNumeroCiclos + 1;
```

```
(* Control de ejecucion de medidas *)
```

```
IF valve1_record = TRUE
```

```
THEN
```

```
    IF valve1_recordingActive = TRUE
```

```
    THEN
```

```
        valve1_recordingNumeroCiclos := valve1_recordingNumeroCiclos + 1;
```

```
        IF valve1_recordingNumeroCiclos >= valve1_recordingDuracionCiclos
```

```
        THEN
```

```
            valve1_recordingActive := FALSE;
```

```
            valve1_recordingNumeroCiclos := 0;
```

```
            valve1_totalNumeroMediciones := valve1_totalNumeroMediciones +
```

```
1;
```

```
        END_IF
```

```
    END_IF
```

```
END_IF
```

```
END_IF
```

```
ELSE
```

```
    valve1_control1CounterLimit := 0;
```

```
END_IF
```

```
IF Valve1_StatusLED = 1 AND Valve1_AppControl = 2 AND Valve1_ActualValue1 < valve1_minimum_pressure
```

```
THEN
```



```
valve1_control2CounterLimit := valve1_control2CounterLimit + 1;

IF valve1_control2CounterLimit > 2
THEN
    Valve1_AppControl := 1;
END_IF

ELSE
    valve1_control2CounterLimit := 0;
END_IF

(* Control de ejecucion de medidas *)

IF valve1_record = TRUE
THEN
    IF valve1_recordingActive = FALSE AND
        valve1_recordingPendant = FALSE AND
        (valve1_totalNumeroCiclos - valve1_recordingLastStartCiclos)
        valve1_recordingCadenciaCiclos >=
    THEN
        valve1_recordingPendant := TRUE;
        IF nextPendantRecordingIndex <= 10
        THEN
            nextPendantRecording[nextPendantRecordingIndex] := 1;
            nextPendantRecordingIndex := nextPendantRecordingIndex + 1;
        END_IF
    END_IF
ELSE
    valve1_recordingActive := FALSE;
```

```
        valve1_recordingNumeroCiclos := 0;

    END_IF

IF (EMERGENCY_OK = FALSE) OR (Valve1_Enable = FALSE)

THEN

    valve1_recordingActive := FALSE;

    valve1_recordingNumeroCiclos := 0;

END_IF

(*****)

(* Control de valvula virtual para lectura de parametros *)

VirtualValve1_ValveMode := 63; (* Transfer Mode *)

VirtualValve1_AppControl := 2; (* Upload (read) *)

VirtualValve1_AppOption := 0;

VirtualValve1_SetpointValue1 := 0;

VirtualValve1_SetpointValue2 := 0;

VirtualValve1_FB_ValveControl(

    awBusDataFromVTEM := VirtualValve1_Inputs,

    awBusDataToVTEM := VirtualValve1_Outputs,

    xEnable := VirtualValve1_Enable,

    bySetValveMode := VirtualValve1_ValveMode,

    bySetAppControl := VirtualValve1_AppControl,

    bySetAppOption := VirtualValve1_AppOption,

    iSetpointValue1 := VirtualValve1_SetpointValue1,

    iSetpointValue2 := VirtualValve1_SetpointValue2,

    xManualAcknowledge := VirtualValve1_ManualAcknowledge,
```



```
byActualValveMode => VirtualValve1_ActualValveMode,  
byActualValveState => VirtualValve1_ActualValveState,  
byActualAppState => VirtualValve1_ActualAppState,  
iActualValue1 => VirtualValve1_ActualValue1,  
iActualValue2 => VirtualValve1_ActualValue2,  
iResponseToValveModeSet => VirtualValve1_ResponseToValveModeSet,  
iErrorCode => VirtualValve1_ErrorCode);
```

```
(*****)
```

```
(* Lectura de un parametro simple *)
```

```
IF Valve1_UploadSingleParameter
```

```
THEN
```

```
VirtualValve1_FB_Upload_Single_Parameter(  
    awBusDataFromVTEM := VirtualValve1_Inputs,  
    awBusDataToVTEM := VirtualValve1_Outputs,  
    xExecute := VirtualValve1_FB_Upload_Single_Parameter_Execute, (* Rising edge *)  
    iChannel := VirtualValve1_FB_Upload_Single_Parameter_Channel,  
    iAddressedTarget := VirtualValve1_FB_Upload_Single_Parameter_AddressedTarget,  
    iIndex := VirtualValve1_FB_Upload_Single_Parameter_Index,  
    iResponse => VirtualValve1_FB_Upload_Single_Parameter_Response,  
    iUploadedValue => VirtualValve1_FB_Upload_Single_Parameter_UploadedValue,  
    xUploadDone => VirtualValve1_FB_Upload_Single_Parameter_UploadDone,  
    xUploadError => VirtualValve1_FB_Upload_Single_Parameter_UploadError);  
  
valve1_previousExecute := VirtualValve1_FB_Upload_Single_Parameter_Execute;  
valve1_counterCycles := valve1_counterCycles + 1;
```



```
IF VirtualValve1_FB_Upload_Single_Parameter_Execute = FALSE
THEN

    valve1_numberCycles := valve1_counterCycles;

    (* Activacion de las lecturas *)

    VirtualValve1_FB_Upload_Single_Parameter_Execute := TRUE;

ELSE

    (* Lectura correcta *)

    IF          VirtualValve1_FB_Upload_Single_Parameter_Execute          AND
VirtualValve1_FB_Upload_Single_Parameter_UploadDone AND

        VirtualValve1_FB_Upload_Single_Parameter_UploadError = FALSE

    THEN

        (* Memorizacion del valor leído *)

        valve1_port24_flow :=
INT_TO_WORD(VirtualValve1_FB_Upload_Single_Parameter_UploadedValue);

        (* Anulacion de la orden de lectura *)

        VirtualValve1_FB_Upload_Single_Parameter_Execute := FALSE;

    END_IF

    (* Lectura incorrecta *)
```



```
IF VirtualValve1_FB_Upload_Single_Parameter_Execute AND  
VirtualValve1_FB_Upload_Single_Parameter_UploadError
```

```
THEN
```

```
(* Anulacion de la orden de lectura *)
```

```
VirtualValve1_FB_Upload_Single_Parameter_Execute := FALSE;
```

```
END_IF
```

```
END_IF
```

```
IF valve1_previousExecute = FALSE AND VirtualValve1_FB_Upload_Single_Parameter_Execute = TRUE
```

```
THEN
```

```
valve1_counterCycles := 0;
```

```
END_IF
```

```
END_IF
```

```
(*****)
```

```
(* Control valvula 2 *)
```

```
Valve2_FB_ValveControl(
```

```
awBusDataFromVTEM := Valve2_Inputs,
```

```
awBusDataToVTEM := Valve2_Outputs,
```

```
xEnable := Valve2_Enable,
```

```
bySetValveMode := Valve2_ValveMode,
```

```
bySetAppControl := Valve2_AppControl,
```

```
bySetAppOption := Valve2_AppOption,
```



```
iSetpointValue1 := Valve2_SetpointValue1,  
  
iSetpointValue2 := Valve2_SetpointValue2,  
  
xManualAcknowledge := Valve2_ManualAcknowledge,  
  
byActualValveMode => Valve2_ActualValveMode,  
  
byActualValveState => Valve2_ActualValveState,  
  
byActualAppState => Valve2_ActualAppState,  
  
iActualValue1 => Valve2_ActualValue1,  
  
iActualValue2 => Valve2_ActualValue2,  
  
iResponseToValveModeSet => Valve2_ResponseToValveModeSet,  
  
iErrorCode => Valve2_ErrorCode);
```

(* Verificacion del estado *)

```
valve2_running := (Valve2_ValveMode = Valve2_ActualValveMode);
```

```
valve2_acknowledgeNeeded := (Valve2_ResponseToValveModeSet =  
FestoVTEMdc.eResponseToValveModeSet.acknowledge_needed);
```

(* Led de status *)

```
Valve2_StatusLED := 0;
```

```
IF Valve2_enabled
```

```
THEN
```

```
IF valve2_acknowledgeNeeded = FALSE
```

```
THEN
```

```
Valve2_StatusLED := 1;
```

```
ELSE
```

```
Valve2_StatusLED := 2;
```

```
END_IF
```

```
END_IF
```

(* Control de inversiones del desplazamiento *)

valve2_inicioCiclo := FALSE;

IF Valve2_StatusLED = 1 AND Valve2_AppControl = 1 AND Valve2_ActualValue2 < valve2_minimum_pressure

THEN

valve2_control1CounterLimit := valve2_control1CounterLimit + 1;

IF valve2_control1CounterLimit > 2

THEN

(* Inicio de un nuevo ciclo *)

valve2_inicioCiclo := TRUE;

(* Cambio de sentido *)

Valve2_AppControl := 2;

(* Incremento del numero de ciclos. Solamente se realiza en uno de los cambios. *)

valve2_totalNumeroCiclos := valve2_totalNumeroCiclos + 1;

(* Control de ejecucion de medidas *)

IF valve2_record = TRUE

THEN

IF valve2_recordingActive = TRUE

THEN

valve2_recordingNumeroCiclos := valve2_recordingNumeroCiclos + 1;

IF valve2_recordingNumeroCiclos >= valve2_recordingDuracionCiclos

THEN

valve2_recordingActive := FALSE;



```
valve2_recordingNumeroCiclos := 0;

valve2_totalNumeroMediciones := valve2_totalNumeroMediciones +
1;

END_IF

END_IF

END_IF

END_IF

ELSE

valve2_control1CounterLimit := 0;

END_IF

IF Valve2_StatusLED = 1 AND Valve2_AppControl = 2 AND Valve2_ActualValue1 < valve2_minimum_pressure

THEN

valve2_control2CounterLimit := valve2_control2CounterLimit + 1;

IF valve2_control2CounterLimit > 2

THEN

Valve2_AppControl := 1;

END_IF

ELSE

valve2_control2CounterLimit := 0;

END_IF

(* Control de ejecucion de medidas *)

IF valve2_record = TRUE

THEN

IF valve2_recordingActive = FALSE AND
```



```
        valve2_recordingPendant = FALSE AND  
  
        (valve2_totalNumeroCiclos - valve2_recordingLastStartCiclos) >=  
valve2_recordingCadenciaCiclos  
  
    THEN  
  
        valve2_recordingPendant := TRUE;  
  
        IF nextPendantRecordingIndex <= 10  
  
            THEN  
  
                nextPendantRecording[nextPendantRecordingIndex] := 2;  
  
                nextPendantRecordingIndex := nextPendantRecordingIndex + 1;  
  
            END_IF  
  
        END_IF  
  
    ELSE  
  
        valve2_recordingActive := FALSE;  
  
        valve2_recordingNumeroCiclos := 0;  
  
    END_IF  
  
IF (EMERGENCY_OK = FALSE) OR (Valve2_Enable = FALSE)  
  
THEN  
  
    valve2_recordingActive := FALSE;  
  
    valve2_recordingNumeroCiclos := 0;  
  
END_IF  
  
(*  
(* Control de valvula virtual para lectura de parametros *)  
  
VirtualValve2_ValveMode := 63; (* Transfer Mode *)  
  
VirtualValve2_AppControl := 2; (* Upload (read) *)  
  
VirtualValve2_AppOption := 0;
```



VirtualValve2_SetpointValue1 := 0;

VirtualValve2_SetpointValue2 := 0;

VirtualValve2_FB_ValveControl(

```
awBusDataFromVTEM := VirtualValve2_Inputs,
awBusDataToVTEM := VirtualValve2_Outputs,
xEnable := VirtualValve2_Enable,
bySetValveMode := VirtualValve2_ValveMode,
bySetAppControl := VirtualValve2_AppControl,
bySetAppOption := VirtualValve2_AppOption,
iSetpointValue1 := VirtualValve2_SetpointValue1,
iSetpointValue2 := VirtualValve2_SetpointValue2,
xManualAcknowledge := VirtualValve2_ManualAcknowledge,
byActualValveMode => VirtualValve2_ActualValveMode,
byActualValveState => VirtualValve2_ActualValveState,
byActualAppState => VirtualValve2_ActualAppState,
iActualValue1 => VirtualValve2_ActualValue1,
iActualValue2 => VirtualValve2_ActualValue2,
iResponseToValveModeSet => VirtualValve2_ResponseToValveModeSet,
iErrorCode => VirtualValve2_ErrorCode);
```

(*****)

(* Lectura de un parametro simple *)

IF Valve2_UploadSingleParameter

THEN

```
VirtualValve2_FB_Upload_Single_Parameter(
```

```

awBusDataFromVTEM := VirtualValve2_Inputs,

awBusDataToVTEM := VirtualValve2_Outputs,

xExecute := VirtualValve2_FB_Upload_Single_Parameter_Execute, (* Rising edge *)

iChannel := VirtualValve2_FB_Upload_Single_Parameter_Channel,

iAddressedTarget := VirtualValve2_FB_Upload_Single_Parameter_AddressedTarget,

iIndex := VirtualValve2_FB_Upload_Single_Parameter_Index,

iResponse => VirtualValve2_FB_Upload_Single_Parameter_Response,

iUploadedValue => VirtualValve2_FB_Upload_Single_Parameter_UploadedValue,

xUploadDone => VirtualValve2_FB_Upload_Single_Parameter_UploadDone,

xUploadError => VirtualValve2_FB_Upload_Single_Parameter_UploadError);
  
```

```

valve2_previousExecute := VirtualValve2_FB_Upload_Single_Parameter_Execute;
  
```

```

valve2_counterCycles := valve2_counterCycles + 1;
  
```

```

IF VirtualValve2_FB_Upload_Single_Parameter_Execute = FALSE
  
```

```

  THEN
  
```

```

    valve2_numberCycles := valve2_counterCycles;
  
```

```

    (* Activacion de las lecturas *)
  
```

```

    VirtualValve2_FB_Upload_Single_Parameter_Execute := TRUE;
  
```

```

  ELSE
  
```

```

    (* Lectura correcta *)
  
```

```

    IF
      VirtualValve2_FB_Upload_Single_Parameter_Execute
      AND
      VirtualValve2_FB_Upload_Single_Parameter_UploadDone
  
```



```
VirtualValve2_FB_Upload_Single_Parameter_UploadError = FALSE

THEN

(* Memorizacion del valor leído *)

valve2_port24_flow :=
INT_TO_WORD(VirtualValve2_FB_Upload_Single_Parameter_UploadedValue);

(* Anulacion de la orden de lectura *)

VirtualValve2_FB_Upload_Single_Parameter_Execute := FALSE;

END_IF

(* Lectura incorrecta *)

IF VirtualValve2_FB_Upload_Single_Parameter_Execute AND
VirtualValve2_FB_Upload_Single_Parameter_UploadError

THEN

(* Anulacion de la orden de lectura *)

VirtualValve2_FB_Upload_Single_Parameter_Execute := FALSE;

END_IF

END_IF

IF valve2_previousExecute = FALSE AND VirtualValve2_FB_Upload_Single_Parameter_Execute = TRUE

THEN

valve2_counterCycles := 0;
```



END_IF

END_IF

(*****)

(* Control valvula 3 *)

Valve3_FB_ValveControl(

awBusDataFromVTEM := Valve3_Inputs,

awBusDataToVTEM := Valve3_Outputs,

xEnable := Valve3_Enable,

bySetValveMode := Valve3_ValveMode,

bySetAppControl := Valve3_AppControl,

bySetAppOption := Valve3_AppOption,

iSetpointValue1 := Valve3_SetpointValue1,

iSetpointValue2 := Valve3_SetpointValue2,

xManualAcknowledge := Valve3_ManualAcknowledge,

byActualValveMode => Valve3_ActualValveMode,

byActualValveState => Valve3_ActualValveState,

byActualAppState => Valve3_ActualAppState,

iActualValue1 => Valve3_ActualValue1,

iActualValue2 => Valve3_ActualValue2,

iResponseToValveModeSet => Valve3_ResponseToValveModeSet,

iErrorCode => Valve3_ErrorCode);

(* Verificacion del estado *)

valve3_running := (Valve3_ValveMode = Valve3_ActualValveMode);



```
valve3_acknowledgeNeeded := (Valve3_ResponseToValveModeSet =  
FestoVTEMdc.eResponseToValveModeSet.acknowledge_needed);
```

```
(* Led de status *)
```

```
Valve3_StatusLED := 0;
```

```
IF Valve3_enabled
```

```
THEN
```

```
    IF valve3_acknowledgeNeeded = FALSE
```

```
    THEN
```

```
        Valve3_StatusLED := 1;
```

```
    ELSE
```

```
        Valve3_StatusLED := 2;
```

```
    END_IF
```

```
END_IF
```

```
(* Control de inversiones del desplazamiento *)
```

```
valve3_inicioCiclo := FALSE;
```

```
IF Valve3_StatusLED = 1 AND Valve3_AppControl = 1 AND Valve3_ActualValue2 < valve3_minimum_pressure
```

```
THEN
```

```
    valve3_control1CounterLimit := valve3_control1CounterLimit + 1;
```

```
    IF valve3_control1CounterLimit > 2
```

```
    THEN
```

```
        (* Inicio de un nuevo ciclo *)
```

```
        valve3_inicioCiclo := TRUE;
```

```
        (* Cambio de sentido *)
```

```
Valve3_AppControl := 2;
```

```
(* Incremento del numero de ciclos. Solamente se realiza en uno de los cambios. *)
```

```
valve3_totalNumeroCiclos := valve3_totalNumeroCiclos + 1;
```

```
(* Control de ejecucion de medidas *)
```

```
IF valve3_record = TRUE
```

```
THEN
```

```
    IF valve3_recordingActive = TRUE
```

```
    THEN
```

```
        valve3_recordingNumeroCiclos := valve3_recordingNumeroCiclos + 1;
```

```
        IF valve3_recordingNumeroCiclos >= valve3_recordingDuracionCiclos
```

```
        THEN
```

```
            valve3_recordingActive := FALSE;
```

```
            valve3_recordingNumeroCiclos := 0;
```

```
            valve3_totalNumeroMediciones := valve3_totalNumeroMediciones +
```

```
1;
```

```
        END_IF
```

```
    END_IF
```

```
END_IF
```

```
END_IF
```

```
ELSE
```

```
    valve3_control1CounterLimit := 0;
```

```
END_IF
```

```
IF Valve3_StatusLED = 1 AND Valve3_AppControl = 2 AND Valve3_ActualValue1 < valve3_minimum_pressure
```

```
THEN
```



```
valve3_control2CounterLimit := valve3_control2CounterLimit + 1;

IF valve3_control2CounterLimit > 2
THEN
    Valve3_AppControl := 1;
END_IF

ELSE
    valve3_control2CounterLimit := 0;
END_IF

(* Control de ejecucion de medidas *)

IF valve3_record = TRUE
THEN
    IF valve3_recordingActive = FALSE AND
        valve3_recordingPendant = FALSE AND
        (valve3_totalNumeroCiclos - valve3_recordingLastStartCiclos) /
        valve3_recordingCadenciaCiclos >=
    THEN
        valve3_recordingPendant := TRUE;
        IF nextPendantRecordingIndex <= 10
        THEN
            nextPendantRecording[nextPendantRecordingIndex] := 3;
            nextPendantRecordingIndex := nextPendantRecordingIndex + 1;
        END_IF
    END_IF
ELSE
    valve3_recordingActive := FALSE;
```



```
valve3_recordingNumeroCiclos := 0;

END_IF

IF (EMERGENCY_OK = FALSE) OR (Valve3_Enable = FALSE)

THEN

    valve3_recordingActive := FALSE;

    valve3_recordingNumeroCiclos := 0;

END_IF

(*****)

(* Control de valvula virtual para lectura de parametros *)

VirtualValve3_ValveMode := 63; (* Transfer Mode *)

VirtualValve3_AppControl := 2; (* Upload (read) *)

VirtualValve3_AppOption := 0;

VirtualValve3_SetpointValue1 := 0;

VirtualValve3_SetpointValue2 := 0;

VirtualValve3_FB_ValveControl(

    awBusDataFromVTEM := VirtualValve3_Inputs,

    awBusDataToVTEM := VirtualValve3_Outputs,

    xEnable := VirtualValve3_Enable,

    bySetValveMode := VirtualValve3_ValveMode,

    bySetAppControl := VirtualValve3_AppControl,

    bySetAppOption := VirtualValve3_AppOption,

    iSetpointValue1 := VirtualValve3_SetpointValue1,

    iSetpointValue2 := VirtualValve3_SetpointValue2,

    xManualAcknowledge := VirtualValve3_ManualAcknowledge,
```



```
byActualValveMode => VirtualValve3_ActualValveMode,  
byActualValveState => VirtualValve3_ActualValveState,  
byActualAppState => VirtualValve3_ActualAppState,  
iActualValue1 => VirtualValve3_ActualValue1,  
iActualValue2 => VirtualValve3_ActualValue2,  
iResponseToValveModeSet => VirtualValve3_ResponseToValveModeSet,  
iErrorCode => VirtualValve3_ErrorCode);
```

```
(*****)
```

```
(* Lectura de un parametro simple *)
```

```
IF Valve3_UploadSingleParameter
```

```
THEN
```

```
VirtualValve3_FB_Upload_Single_Parameter(  
    awBusDataFromVTEM := VirtualValve3_Inputs,  
    awBusDataToVTEM := VirtualValve3_Outputs,  
    xExecute := VirtualValve3_FB_Upload_Single_Parameter_Execute, (* Rising edge *)  
    iChannel := VirtualValve3_FB_Upload_Single_Parameter_Channel,  
    iAddressedTarget := VirtualValve3_FB_Upload_Single_Parameter_AddressTarget,  
    iIndex := VirtualValve3_FB_Upload_Single_Parameter_Index,  
    iResponse => VirtualValve3_FB_Upload_Single_Parameter_Response,  
    iUploadedValue => VirtualValve3_FB_Upload_Single_Parameter_UploadedValue,  
    xUploadDone => VirtualValve3_FB_Upload_Single_Parameter_UploadDone,  
    xUploadError => VirtualValve3_FB_Upload_Single_Parameter_UploadError);  
  
valve3_previousExecute := VirtualValve3_FB_Upload_Single_Parameter_Execute;  
valve3_counterCycles := valve3_counterCycles + 1;
```

```
IF VirtualValve3_FB_Upload_Single_Parameter_Execute = FALSE
THEN

    valve3_numberCycles := valve3_counterCycles;

    (* Activacion de las lecturas *)

    VirtualValve3_FB_Upload_Single_Parameter_Execute := TRUE;

ELSE

    (* Lectura correcta *)

    IF          VirtualValve3_FB_Upload_Single_Parameter_Execute          AND
VirtualValve3_FB_Upload_Single_Parameter_UploadDone AND

        VirtualValve3_FB_Upload_Single_Parameter_UploadError = FALSE

    THEN

        (* Memorizacion del valor leído *)

        valve3_port24_flow          :=
INT_TO_WORD(VirtualValve3_FB_Upload_Single_Parameter_UploadedValue);

        (* Anulacion de la orden de lectura *)

        VirtualValve3_FB_Upload_Single_Parameter_Execute := FALSE;

    END_IF

    (* Lectura incorrecta *)
```



```
IF VirtualValve3_FB_Upload_Single_Parameter_Execute AND  
VirtualValve3_FB_Upload_Single_Parameter_UploadError
```

```
THEN
```

```
(* Anulacion de la orden de lectura *)
```

```
VirtualValve3_FB_Upload_Single_Parameter_Execute := FALSE;
```

```
END_IF
```

```
END_IF
```

```
IF valve3_previousExecute = FALSE AND VirtualValve3_FB_Upload_Single_Parameter_Execute = TRUE
```

```
THEN
```

```
valve3_counterCycles := 0;
```

```
END_IF
```

```
END_IF
```

```
(*****)
```

```
(* Asignación de valores de Modbus *)
```

```
(* Valve 0 *)
```

```
valve0_fast_modbus_port2_pressure := Valve0_ActualValue1;
```

```
valve0_fast_modbus_port2_position := CalculatePosition(VALUE0_P2_POSITION_MAX,  
VALUE0_P2_POSITION_RANGE, valve0_port2_position_input);
```

```
valve0_fast_modbus_port4_pressure := Valve0_ActualValue2;
```



```
valve0_fast_modbus_port4_position := CalculatePosition(VALUE0_P4_POSITION_MAX,  
VALVE0_P4_POSITION_RANGE, valve0_port4_position_input);
```

```
valve0_port24_flow_low := ADR(valve0_port24_flow);
```

```
valve0_fast_modbus_port2_flow := 10 * SINT_TO_REAL(valve0_port24_flow_low^);
```

```
valve0_port24_flow_high := valve0_port24_flow_low + 1;
```

```
valve0_fast_modbus_port4_flow := 10 * SINT_TO_REAL(valve0_port24_flow_high^);
```

```
valve0_fast_modbus_trigger := BOOL_TO_REAL(valve0_recordingActive);
```

```
valve0_fast_modbus_numeroTotalCiclos := UDINT_TO_REAL(valve0_totalNumeroCiclos);
```

```
valve0_fast_modbus_numeroTotalMediciones := UDINT_TO_REAL(valve0_totalNumeroMediciones);
```

```
valve0_slow_modbus_port2_pressure := valve0_fast_modbus_port2_pressure;
```

```
valve0_slow_modbus_port2_position := valve0_fast_modbus_port2_position;
```

```
valve0_slow_modbus_port2_flow := valve0_fast_modbus_port2_flow;
```

```
valve0_slow_modbus_port4_pressure := valve0_fast_modbus_port4_pressure;
```

```
valve0_slow_modbus_port4_position := valve0_fast_modbus_port4_position;
```

```
valve0_slow_modbus_port4_flow := valve0_fast_modbus_port4_flow;
```

```
valve0_slow_modbus_enabled := BOOL_TO_REAL(valve0_enabled);
```

```
valve0_slow_modbus_statusLED := INT_TO_REAL(Valve0_StatusLED);
```

```
valve0_slow_modbus_numeroTotalCiclos := UDINT_TO_REAL(valve0_totalNumeroCiclos);
```

```
valve0_slow_modbus_record := BOOL_TO_REAL(valve0_record);
```

```
valve0_slow_modbus_cadenciaCiclos := UDINT_TO_REAL(valve0_recordingCadenciaCiclos);
```

```
valve0_slow_modbus_ciclosMedidos := UDINT_TO_REAL(valve0_recordingDuracionCiclos);
```

```
valve0_slow_modbus_ciclosParaCaptura := UDINT_TO_REAL(valve0_recordingLastStartCiclos +  
valve0_recordingCadenciaCiclos - valve0_totalNumeroCiclos);
```

```
valve0_slow_modbus_recordingPendant := BOOL_TO_REAL(valve0_recordingPendant);
```



```
valve0_slow_modbus_recordingActive := BOOL_TO_REAL(valve0_recordingActive);

valve0_slow_modbus_weight := valve0_weight;

valve0_slow_modbus_numeroTotalMediciones := UDINT_TO_REAL(valve0_totalNumeroMediciones);

(* Valve 1 *)

valve1_fast_modbus_port2_pressure := Valve1_ActualValue1;

valve1_fast_modbus_port2_position := CalculatePosition(VALUE1_P2_POSITION_MAX,
VALUE1_P2_POSITION_RANGE, valve1_port2_position_input);

valve1_fast_modbus_port4_pressure := Valve1_ActualValue2;

valve1_fast_modbus_port4_position := CalculatePosition(VALUE1_P4_POSITION_MAX,
VALUE1_P4_POSITION_RANGE, valve1_port4_position_input);

valve1_port24_flow_low := ADR(valve1_port24_flow);

valve1_fast_modbus_port2_flow := 10 * SINT_TO_REAL(valve1_port24_flow_low^);

valve1_port24_flow_high := valve1_port24_flow_low + 1;

valve1_fast_modbus_port4_flow := 10 * SINT_TO_REAL(valve1_port24_flow_high^);

valve1_fast_modbus_trigger := BOOL_TO_REAL(valve1_recordingActive);

valve1_fast_modbus_numeroTotalCiclos := UDINT_TO_REAL(valve1_totalNumeroCiclos);

valve1_fast_modbus_numeroTotalMediciones := UDINT_TO_REAL(valve1_totalNumeroMediciones);

valve1_slow_modbus_port2_pressure := valve1_fast_modbus_port2_pressure;

valve1_slow_modbus_port2_position := valve1_fast_modbus_port2_position;

valve1_slow_modbus_port2_flow := valve1_fast_modbus_port2_flow;

valve1_slow_modbus_port4_pressure := valve1_fast_modbus_port4_pressure;

valve1_slow_modbus_port4_position := valve1_fast_modbus_port4_position;
```



```
valve1_slow_modbus_port4_flow := valve1_fast_modbus_port4_flow;

valve1_slow_modbus_enabled := BOOL_TO_REAL(valve1_enabled);

valve1_slow_modbus_statusLED := INT_TO_REAL(Valve1_StatusLED);

valve1_slow_modbus_numeroTotalCiclos := UDINT_TO_REAL(valve1_totalNumeroCiclos);

valve1_slow_modbus_record := BOOL_TO_REAL(valve1_record);

valve1_slow_modbus_cadenciaCiclos := UDINT_TO_REAL(valve1_recordingCadenciaCiclos);

valve1_slow_modbus_ciclosMedidos := UDINT_TO_REAL(valve1_recordingDuracionCiclos);

valve1_slow_modbus_ciclosParaCaptura := UDINT_TO_REAL(valve1_recordingLastStartCiclos +
valve1_recordingCadenciaCiclos - valve1_totalNumeroCiclos);

valve1_slow_modbus_recordingPendant := BOOL_TO_REAL(valve1_recordingPendant);

valve1_slow_modbus_recordingActive := BOOL_TO_REAL(valve1_recordingActive);

valve1_slow_modbus_weight := valve1_weight;

valve1_slow_modbus_numeroTotalMediciones := UDINT_TO_REAL(valve1_totalNumeroMediciones);

(* Valve 2 *)

valve2_fast_modbus_port2_pressure := Valve2_ActualValue1;

valve2_fast_modbus_port2_position := CalculatePosition(VALUE2_P2_POSITION_MAX,
VALUE2_P2_POSITION_RANGE, valve2_port2_position_input);

valve2_fast_modbus_port4_pressure := Valve2_ActualValue2;

valve2_fast_modbus_port4_position := CalculatePosition(VALUE2_P4_POSITION_MAX,
VALUE2_P4_POSITION_RANGE, valve2_port4_position_input);

valve2_port24_flow_low := ADR(valve2_port24_flow);

valve2_fast_modbus_port2_flow := 10 * SINT_TO_REAL(valve2_port24_flow_low^);

valve2_port24_flow_high := valve2_port24_flow_low + 1;

valve2_fast_modbus_port4_flow := 10 * SINT_TO_REAL(valve2_port24_flow_high^);
```



```
valve2_fast_modbus_trigger := BOOL_TO_REAL(valve2_recordingActive);

valve2_fast_modbus_numeroTotalCiclos := UDINT_TO_REAL(valve2_totalNumeroCiclos);

valve2_fast_modbus_numeroTotalMediciones := UDINT_TO_REAL(valve2_totalNumeroMediciones);

valve2_slow_modbus_port2_pressure := valve2_fast_modbus_port2_pressure;

valve2_slow_modbus_port2_position := valve2_fast_modbus_port2_position;

valve2_slow_modbus_port2_flow := valve2_fast_modbus_port2_flow;

valve2_slow_modbus_port4_pressure := valve2_fast_modbus_port4_pressure;

valve2_slow_modbus_port4_position := valve2_fast_modbus_port4_position;

valve2_slow_modbus_port4_flow := valve2_fast_modbus_port4_flow;

valve2_slow_modbus_enabled := BOOL_TO_REAL(valve2_enabled);

valve2_slow_modbus_statusLED := INT_TO_REAL(Valve2_StatusLED);

valve2_slow_modbus_numeroTotalCiclos := UDINT_TO_REAL(valve2_totalNumeroCiclos);

valve2_slow_modbus_record := BOOL_TO_REAL(valve2_record);

valve2_slow_modbus_cadenciaCiclos := UDINT_TO_REAL(valve2_recordingCadenciaCiclos);

valve2_slow_modbus_ciclosMedidos := UDINT_TO_REAL(valve2_recordingDuracionCiclos);

valve2_slow_modbus_ciclosParaCaptura := UDINT_TO_REAL(valve2_recordingLastStartCiclos +
valve2_recordingCadenciaCiclos - valve2_totalNumeroCiclos);

valve2_slow_modbus_recordingPendant := BOOL_TO_REAL(valve2_recordingPendant);

valve2_slow_modbus_recordingActive := BOOL_TO_REAL(valve2_recordingActive);

valve2_slow_modbus_weight := valve2_weight;

valve2_slow_modbus_numeroTotalMediciones := UDINT_TO_REAL(valve2_totalNumeroMediciones);

(* Valve 3 *)

valve3_fast_modbus_port2_pressure := Valve3_ActualValue1;

valve3_fast_modbus_port2_position := CalculatePosition(VALUE3_P2_POSITION_MAX,
VALUE3_P2_POSITION_RANGE, valve3_port2_position_input);
```



```
valve3_fast_modbus_port4_pressure := Valve3_ActualValue2;

valve3_fast_modbus_port4_position      :=      CalculatePosition(VALUE3_P4_POSITION_MAX,
VALVE3_P4_POSITION_RANGE, valve3_port4_position_input);

valve3_port24_flow_low := ADR(valve3_port24_flow);

valve3_fast_modbus_port2_flow := 10 * SINT_TO_REAL(valve3_port24_flow_low^);

valve3_port24_flow_high := valve3_port24_flow_low + 1;

valve3_fast_modbus_port4_flow := 10 * SINT_TO_REAL(valve3_port24_flow_high^);

valve3_fast_modbus_trigger := BOOL_TO_REAL(valve3_recordingActive);

valve3_fast_modbus_numeroTotalCiclos := UDINT_TO_REAL(valve3_totalNumeroCiclos);

valve3_fast_modbus_numeroTotalMediciones := UDINT_TO_REAL(valve3_totalNumeroMediciones);

valve3_slow_modbus_port2_pressure := valve3_fast_modbus_port2_pressure;

valve3_slow_modbus_port2_position := valve3_fast_modbus_port2_position;

valve3_slow_modbus_port2_flow := valve3_fast_modbus_port2_flow;

valve3_slow_modbus_port4_pressure := valve3_fast_modbus_port4_pressure;

valve3_slow_modbus_port4_position := valve3_fast_modbus_port4_position;

valve3_slow_modbus_port4_flow := valve3_fast_modbus_port4_flow;

valve3_slow_modbus_enabled := BOOL_TO_REAL(valve3_enabled);

valve3_slow_modbus_statusLED := INT_TO_REAL(Valve3_StatusLED);

valve3_slow_modbus_numeroTotalCiclos := UDINT_TO_REAL(valve3_totalNumeroCiclos);

valve3_slow_modbus_record := BOOL_TO_REAL(valve3_record);

valve3_slow_modbus_cadenciaCiclos := udINT_TO_REAL(valve3_recordingCadenciaCiclos);

valve3_slow_modbus_ciclosMedidos := udINT_TO_REAL(valve3_recordingDuracionCiclos);
```



```
valve3_slow_modbus_ciclosParaCaptura := UDINT_TO_REAL(valve3_recordingLastStartCiclos +  
valve3_recordingCadenciaCiclos - valve3_totalNumeroCiclos);
```

```
valve3_slow_modbus_recordingPendant := BOOL_TO_REAL(valve3_recordingPendant);
```

```
valve3_slow_modbus_recordingActive := BOOL_TO_REAL(valve3_recordingActive);
```

```
valve3_slow_modbus_weight := valve3_weight;
```

```
valve3_slow_modbus_numeroTotalMediciones := UDINT_TO_REAL(valve3_totalNumeroMediciones);
```

```
(*****)
```

```
(* Control de recordings *)
```

```
IF EMERGENCY_OK
```

```
THEN
```

```
    (* Recording valve 0 *)
```

```
    IF (valve0_recordingActive = FALSE) AND (valve1_recordingActive = FALSE) AND (valve2_recordingActive =  
FALSE) AND (valve3_recordingActive = FALSE) AND (* Verificacion de todos los recordings inactivos *)
```

```
        (nextPendantRecording[0] = 0) AND (nextPendantRecordingIndex > 0) AND  
valve0_inicioCiclo
```

```
    THEN
```

```
        valve0_recordingActive := TRUE;
```

```
        valve0_recordingLastStartCiclos := valve0_totalNumeroCiclos;
```

```
        FOR counterPendantRecordings := 1 TO nextPendantRecordingIndex DO
```

```
            nextPendantRecording[counterPendantRecordings - 1] :=  
nextPendantRecording[counterPendantRecordings];
```

```
        END_FOR
```

```
        nextPendantRecordingIndex := nextPendantRecordingIndex - 1;
```

```
        valve0_recordingPendant := FALSE;
```

```
    END_IF
```



(* Recording valve 1 *)

IF (valve0_recordingActive = FALSE) AND (valve1_recordingActive = FALSE) AND (valve2_recordingActive = FALSE) AND (valve3_recordingActive = FALSE) AND (* Verificacion de todos los recordings inactivos *)

valve1_inicioCiclo (nextPendantRecording[0] = 1) AND (nextPendantRecordingIndex > 0) AND

THEN

valve1_recordingActive := TRUE;

valve1_recordingLastStartCiclos := valve1_totalNumeroCiclos;

FOR counterPendantRecordings := 1 TO nextPendantRecordingIndex DO

nextPendantRecording[counterPendantRecordings - 1] := nextPendantRecording[counterPendantRecordings];

END_FOR

nextPendantRecordingIndex := nextPendantRecordingIndex - 1;

valve1_recordingPendant := FALSE;

END_IF

(* Recording valve 2 *)

IF (valve0_recordingActive = FALSE) AND (valve1_recordingActive = FALSE) AND (valve2_recordingActive = FALSE) AND (valve3_recordingActive = FALSE) AND (* Verificacion de todos los recordings inactivos *)

valve2_inicioCiclo (nextPendantRecording[0] = 2) AND (nextPendantRecordingIndex > 0) AND

THEN

valve2_recordingActive := TRUE;

valve2_recordingLastStartCiclos := valve2_totalNumeroCiclos;

FOR counterPendantRecordings := 1 TO nextPendantRecordingIndex DO

nextPendantRecording[counterPendantRecordings - 1] := nextPendantRecording[counterPendantRecordings];

END_FOR

nextPendantRecordingIndex := nextPendantRecordingIndex - 1;

valve2_recordingPendant := FALSE;



END_IF

(* Recording valve 3 *)

IF (valve0_recordingActive = FALSE) AND (valve1_recordingActive = FALSE) AND (valve2_recordingActive = FALSE) AND (valve3_recordingActive = FALSE) AND (* Verificacion de todos los recordings inactivos *)

valve3_inicioCiclo (nextPendantRecording[0] = 3) AND (nextPendantRecordingIndex > 0) AND

THEN

valve3_recordingActive := TRUE;

valve3_recordingLastStartCiclos := valve3_totalNumeroCiclos;

FOR counterPendantRecordings := 1 TO nextPendantRecordingIndex DO

nextPendantRecording[counterPendantRecordings - 1] := nextPendantRecording[counterPendantRecordings];

END_FOR

nextPendantRecordingIndex := nextPendantRecordingIndex - 1;

valve3_recordingPendant := FALSE;

END_IF

END_IF



ANEXO II: RESUMEN DE OPERACIONES DE MECANIZADO DE LAS RECTIFICADORAS DE DANOBAT

Tabla 11. Rectificadoras de exteriores.

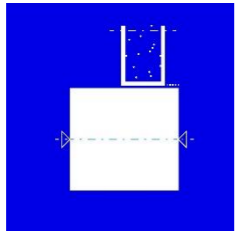
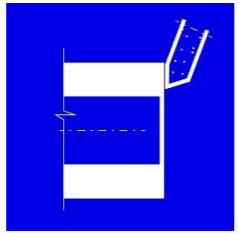
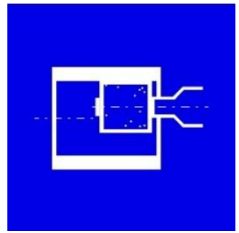
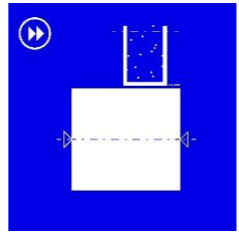
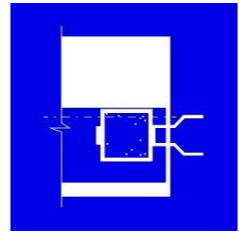
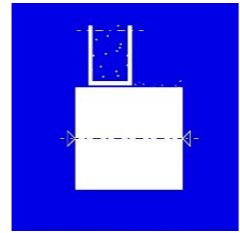
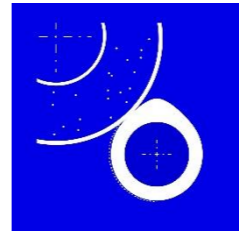
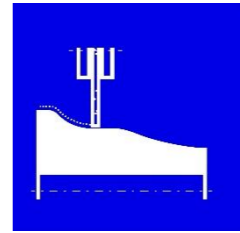
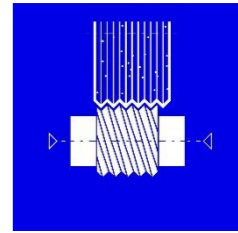
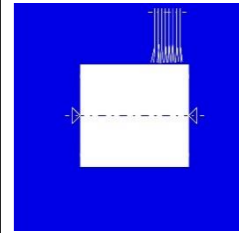
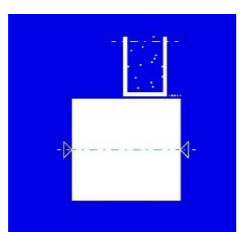
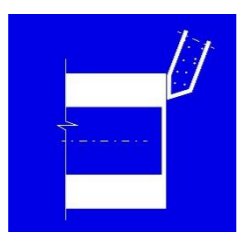
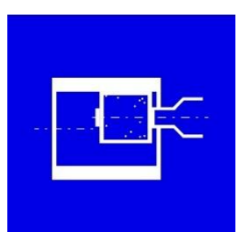
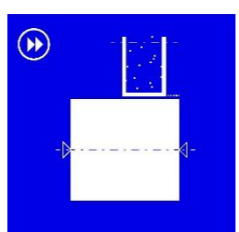
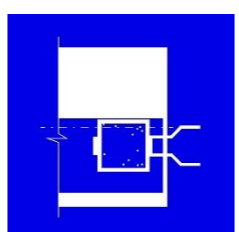
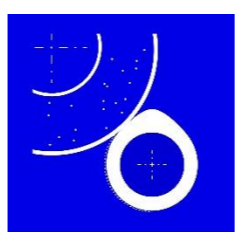
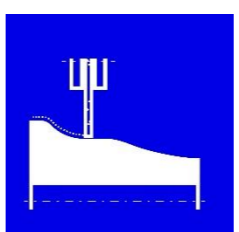
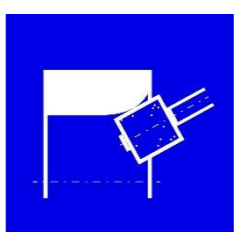
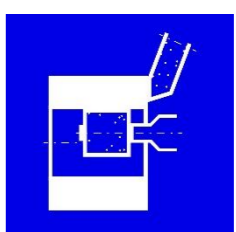
Máquina	 Rectificado cilíndrico exterior	 Rectificado tangencial de caras	 Rectificado cilíndrico interior	 Rectificado de alta velocidad	 Rectificado excéntrico	 Superacabado	 Rectificado de formas	 Rectificado por contorneado	 Rectificado de rosca	 Rebabado
CG/PG	✓	✓	✓							
LG	✓	✓	✓	✓	✓		✓	✓		
HG	✓		✓		✓	✓	✓	✓	✓	✓
WT	✓		✓		✓	✓	✓	✓	✓	✓
mBTG-DANTIP ¹⁰	✓									

Tabla 12. Rectificadoras de interiores.

Máquina	 Rectificado cilíndrico exterior	 Rectificado tangencial de caras	 Rectificado cilíndrico interior	 Rectificado de alta velocidad	 Rectificado excéntrico	 Rectificado de formas	 Rectificado por contorneado	 Rectificado de radios	 Rectificado simultáneo
ID	✓	✓	✓	✓	✓	✓	✓		
IRD	✓	✓	✓	✓	✓	✓	✓	✓	
ILD	✓	✓	✓	✓	✓	✓	✓		
IED	✓	✓	✓	✓	✓		✓		✓

¹⁰ Rectificado y medición de puntas de álabes para el sector aeroespacial

Tabla 13. Rectificadoras sin centros.

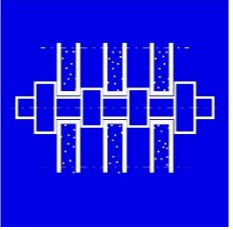
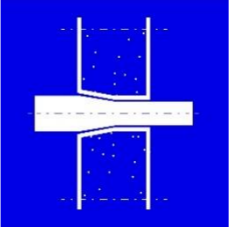
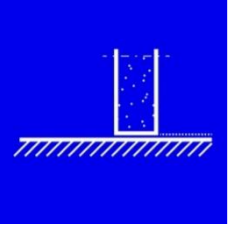
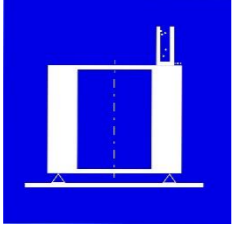
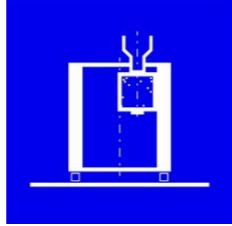
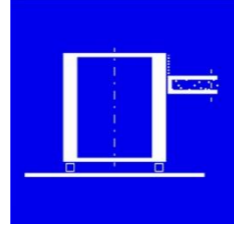
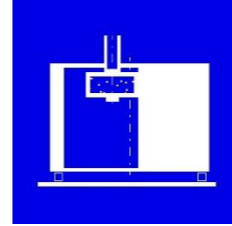
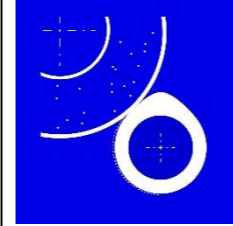
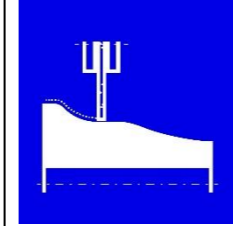
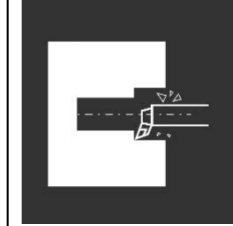
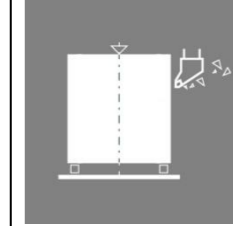
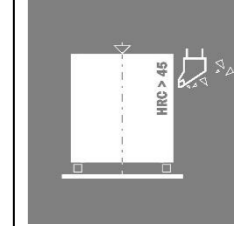
Máquina		
	Rectificado a plongeé	Rectificado en pasante
ESTARTA-175 / ESTARTA-250	✓	✓
ESTARTA-400 / ESTARTA-650	✓	✓

Tabla 14. Rectificadoras verticales.

Máquina										
	Rectificado de planos	Rectificado tangencial de caras	Rectificado vertical de interiores	Rectificado vertical de exteriores	Rectificado excéntrico	Rectificado de formas	Rectificado por contorno	Mandrinado	Torneado vertical	Torneado en duro
VG	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
DVG	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓