

## MeiA • 4.0 PARA ABORDAR LOS RETOS ACTUALES DE FORMACIÓN EN AUTOMATIZACIÓN

Arantzasu Burgos

Departamento de Ingeniería de Sistemas y Automática, Escuela de Ingeniería de Bilbao, UPV/EHU  
[arantzasu.burgos@ehu.es](mailto:arantzasu.burgos@ehu.es)

Nagore Iriondo, Maria Luz Álvarez, Isabel Sarachaga

Departamento de Ingeniería de Sistemas y Automática, Escuela de Ingeniería de Bilbao, UPV/EHU  
[nagore.iriondo@ehu.es](mailto:nagore.iriondo@ehu.es), [marialuz.alvarez@ehu.es](mailto:marialuz.alvarez@ehu.es), [isabel.sarachaga@ehu.es](mailto:isabel.sarachaga@ehu.es)

### Resumen

*El software de control de sistemas de producción automatizados está experimentando un rápido aumento en cantidad y complejidad que lo convierte es un factor clave en la transición de una visión tradicional a una más flexible capaz de hacer frente a los nuevos retos de la Industria 4.0. La industria requiere de personal capacitado para afrontar estos nuevos retos, con formación tanto en tecnologías para la transformación digital como en metodologías de desarrollo de software de control necesarias para poder conectar la información del proceso productivo con procesos de gestión bajo este nuevo paradigma. En este artículo se presenta un ciclo de desarrollo de software de control en base a una actualización de la metodología MeiA• que incluye los estándares propuestos por las arquitecturas de referencia para las empresas del futuro. Además, se realiza una propuesta de cómo introducir estos nuevos conceptos en la formación en enseñanzas de ingeniería.*

**Palabras clave:** Sistemas de producción automatizados; Software de control; Metodología MeiA•; Formación en automatización

### 1 INTRODUCCIÓN

En el ámbito de los sistemas de automatización industrial, el crecimiento de la cantidad de software de control, así como su complejidad, lo convierte en un aspecto cada vez más relevante. La cuarta revolución industrial o el paradigma Industry 4.0 ha promovido, además, la adopción de nuevas arquitecturas de referencia como RAMI 4.0 (Reference Architectural Model for Industrie 4.0), IIRA (Industrial Internet Reference Architecture), Smart Manufacturing Ecosystem o IMSA (Intelligent Manufacturing System Framework). Asimismo, define nuevos conceptos como son los componentes Io4, CPPS (CyberPhysical Production Systems) o IoT (Internet of Things). También propone una serie de

estándares destacando IEC 62264-1 [10], IEC 61512-1 [9] e IEC 61131-3 para el desarrollo de software de control, y OPC-UA (Open Platform Communications - Unified Architecture) y MQTT (Message Queuing Telemetry Transport) para comunicaciones [12], [7]. Todo ello tiene una implicación directa y plantea nuevos retos en el ciclo de desarrollo de software de control (en adelante referido como SW) [6].

En los últimos años son muchos los trabajos relacionados con nuevos enfoques de desarrollo de SW, y de ensayos sistemáticos de los mismos. El estudio realizado en [16] refleja los desarrollos y avances más importantes relacionados con la ingeniería de software de automatización industrial, considerados como una base de referencia para salvar la distancia entre los mundos de la automatización y la ingeniería del software.

Otro estudio interdisciplinario presentado en [14] expone los retos asociados a la evolución de los sistemas de producción automatizados (aPS – automated Production Systems) en las distintas fases de desarrollo y en varios ámbitos transversales, y realiza una revisión de los enfoques existentes para hacer frente a dichos retos.

El estudio realizado a 61 empresas de fabricación y presentado en [15] proporciona resultados cuantitativos y cualitativos interesantes acerca de la ingeniería del software para aPS, como, por ejemplo, que la cantidad de software requiere un esfuerzo de entre el 35% y 50% del personal de desarrollo. También identifican obstáculos que se deben abordar en futuras investigaciones, además de contramedidas industriales en aspectos relacionados con la cualificación del personal y las cuestiones organizativas para aplicar nuevos paradigmas. Otro resultado destacable es que entre el 50% y el 75% del código de control industrial es código extrafuncional, es decir, tareas transversales que se realizan en todas las unidades de producción, como la gestión de fallos, la recopilación de datos de funcionamiento o el cambio de modos de operación. Este código

extrafuncional no suele estar representado en el diseño, lo que lo hace más difícil de entender que las partes de control de operación.

Este trabajo presenta, por un lado, el ciclo de desarrollo de SW basado en una versión actualizada de la metodología **MeiA**, en la que se integran los estándares de la Industry 4.0 y los conceptos de la ingeniería del software. Por otro lado, plantea una propuesta de cómo introducir estas nuevas necesidades de formación en enseñanzas de ingeniería, para proporcionar a las personas que se van a incorporar a la industria los principios de ingeniería del software y el conocimiento de los estándares necesarios en el desarrollo de este tipo de software.

La estructura del artículo es la siguiente: la motivación y evolución de la metodología **MeiA**, se aborda en el apartado de antecedentes. El apartado 3 presenta el ciclo de desarrollo del SW **MeiA 4.0**, y el apartado 4 ilustra cómo se desarrollan las competencias a adquirir en las titulaciones de Grado y de Máster de la Escuela de Ingeniería de Bilbao. El último apartado presenta las conclusiones del trabajo.

## 2 ANTECEDENTES

La integración de las disciplinas de la ingeniería del software con los métodos y estándares extendidos en el ámbito de los aPS fue la motivación de una propuesta de formación, estructurada en tres niveles: inicial, de diseño y metodológico [5]. La formación está basada en la metodología **MeiA** de análisis y diseño de SW, que combina la guía GEMMA, los diagramas de casos de uso UML y el lenguaje de modelado GRAFCET, e incluye, implícitamente, conceptos fundamentales de la ingeniería del software relacionados con la flexibilidad, modularidad y extensibilidad de los diseños que pueden ser reutilizados en otros sistemas.

**MeiA** guía el análisis y el diseño a través de seis fases, cada una de ellas relacionada con un modo de operación (Modo automático, Modo Manual, Modo de Pruebas, Fallos, Paro de Emergencia y Producción Normal), utilizando la terminología del dominio, lo que permite conseguir una mejora en la comunicación entre todos los agentes involucrados en el desarrollo, puesta a punto y operación del sistema [1]. La metodología ofrece guías de diseño y plantillas para cada una de las fases, incluyendo aspectos estructurales, así como los aspectos de la ingeniería del software mencionados. La utilización del lenguaje de modelado GRAFCET permite generar unidades de organización de diseño (DOUs – Design Organization Units) de tres tipos: de decisión, de producción y auxiliares.

El trabajo presentado en [3] plantea la extensión de la metodología **MeiA**, mostrando cómo incorporar el estándar IEC 61512-1 [10]. Así, siguiendo la metodología, se obtienen los modelos propuestos en el estándar IEC 61512 y la descripción de los modos de operación acordes al mismo.

En este punto, cabe destacar dos aspectos relevantes de la metodología **MeiA**. Por un lado, el diseño por fases contempla todos los estados de la Guía GEMMA, es decir, identifica, ya desde la fase de análisis, todas las funcionalidades transversales o código extrafuncional identificado en el estudio en [15] como código que en un alto porcentaje no suele representarse en el diseño.

Por otro lado, resaltar la importancia que tiene la utilización del color asignado a cada una de las fases de la metodología (Figura 1). Estos colores se utilizan en los componentes que conforman los modelos de análisis y ayudan a interpretar y conectar los mismos conceptos en diferentes modelos. A modo de ejemplo, la figura incluye un diagrama en el cual se pueden distinguir los casos de uso asociados a cada fase por medio del color.

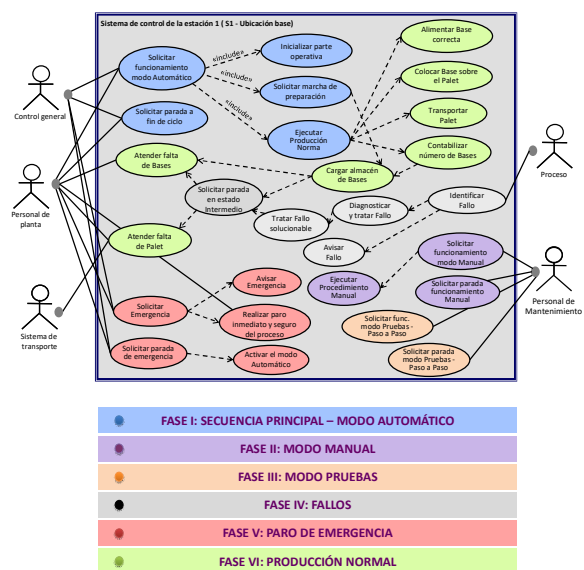


Figura 1: Colores asociados a las fases de **MeiA**.

El trabajo en [4] presenta la metodología de transformación de los diseños (DOUs) obtenidos con **MeiA** a código IEC 61131-3, concretamente a código en texto estructurado (ST) y un prototipo de herramienta que realiza dicha transformación. La elección de este lenguaje pretende que el código resultante tenga una estructura clara y familiar tanto para el personal de ingeniería como para el personal de planta, y permite, además, identificar fácilmente los diseños a partir de los que fue generado.

Por último, la plataforma presentada en [2] proporciona los beneficios de la sinergia entre la aplicación de la metodología **MeiA** para el análisis, diseño e implementación, con el uso de tecnologías MDE para la definición de ontologías de dominio y modelos de transformación, con objeto de generar automáticamente código PLCopen XML del aPS.

### 3 CICLO DE DESARROLLO DE SW MeiA 4.0

MeiA. 4.0 adopta el modelo en V [12] como ciclo de desarrollo para construir el software de control de los aPS (Figura 2).

El punto de partida son los **Requisitos de Usuario/a**, que deben contemplar la descripción de los siguientes aspectos: 1) el funcionamiento y las operaciones del

la tarea de análisis a través de una secuencia ordenada de pasos.

El resultado del análisis se recoge en los siguientes modelos que propone el estándar IEC 61512: el Modelo de Proceso, en el que se representan las operaciones a realizar sobre el aPS; el Modelo de Procedimientos con los procedimientos (DOUs) que realizarán las operaciones identificadas; el Modelo Físico en el que se encuentran agrupados los equipos físicos en función de las operaciones; y el Modelo Entidad Equipo donde se establece la relación entre los procedimientos que realizan las operaciones y los equipos físicos en los que se ejecutan dichos procedimientos (Figura 3). Las operaciones y procedimientos identificados, además de las operaciones de producción, incluyen las operaciones de mando que organizan el arranque y la parada de los distintos modos de funcionamiento, y coordinan todos los posibles estados del aPS, así como, las

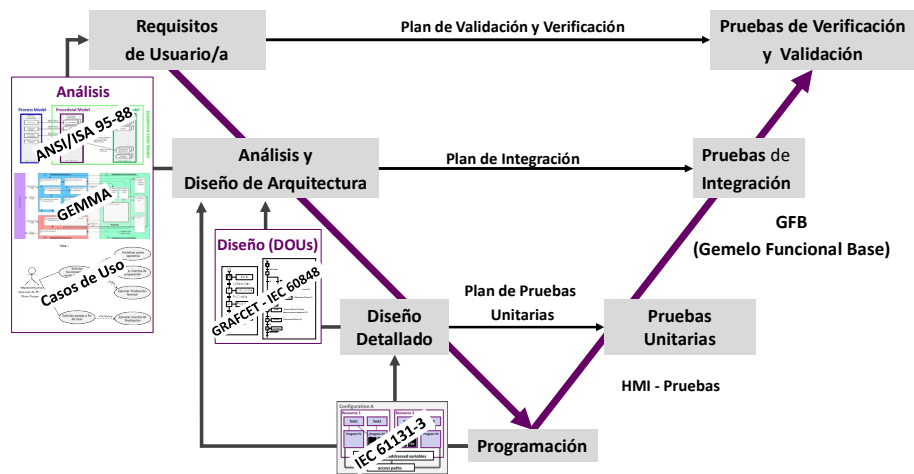


Figura 2: Ciclo de desarrollo de SW MeiA 4.0

aPS, que deben incluir el objetivo (qué hace), las operaciones que permiten alcanzar el objetivo (de qué forma) y el orden en que se deben realizar las operaciones, considerando tanto los requisitos funcionales como los no funcionales (restricciones de tiempos, prioridades, etc.); 2) los modos de operación automático, manual, paro de emergencia, paro a fin de ciclo, etc.; 3) las condiciones de arranque; 4) la información que procesará el sistema de supervisión referente al aPS, a la trazabilidad del aPS y a los controles a realizar sobre el mismo; 5) las señales, tanto las señales asociadas a los sensores y a los actuadores del aPS, como las señales de control del sistema de supervisión.

Partiendo de los requisitos de usuario/a, se realiza el **Análisis de Requisitos del software y Diseño de Arquitectura**, identificando los modos de operación que determinan las perspectivas a analizar en las distintas fases de MeiA 4.0. En cada fase se realiza

operaciones auxiliares de inicialización, preparación, paros, avisos, etc.

El Modelo Físico se construye en la Fase VI (Producción Normal) de MeiA, donde se identifican

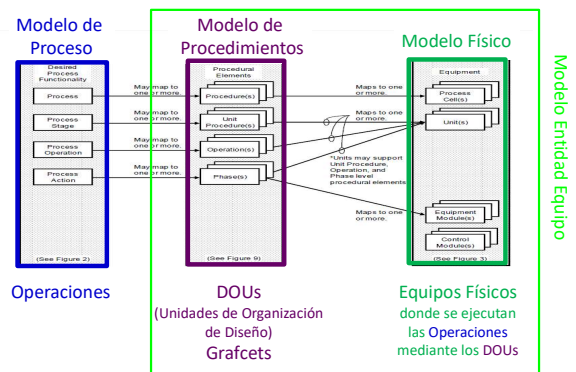


Figura 3: Relación entre los modelos IEC 61512

las operaciones de producción, los procedimientos y los equipos donde se realizan, así como su orden de ejecución. Para su realización se consideran los principios de subdivisión aportados por el estándar IEC 61512, ya que es en esta fase donde las subdivisiones de equipos inapropiadas pueden comprometer la efectividad del enfoque modular. El Modelo Entidad Equipo es el último en realizarse, ya que se construye conectando el Modelo Físico con el Modelo de Procedimientos.

La representación del análisis de los distintos estados del aPS se realiza con el Modelo de Estados del IEC 61512 adaptado a las necesidades de los aPS discretos [3]. No obstante, también se mantiene la representación con la guía GEMMA, ya que la semántica de IEC 61512 se centra en las máquinas automatizadas, mientras que GEMMA utiliza una semántica muy próxima al personal de planta.

También se sigue utilizando el Diagrama de Casos de Uso para describir las funcionalidades (operaciones) con los actores que intervienen en las mismas. Las solicitudes de los actores externos al sistema de control van a determinar las pre-condiciones de ejecución de los casos de uso y las condiciones de evolución entre los estados del aPS. Los casos de uso

permiten identificar los requisitos de control y supervisión, así como la configuración del panel de operación o HMI y demás paneles auxiliares necesarios.

En la Figura 4 se presentan las plantillas de la Fase I que deben ser particularizadas durante el análisis del sistema de control de un aPS. Además de las plantillas GEMMA y del diagrama de casos de uso, se han incorporado las correspondientes a los modelos de proceso, de procedimientos y de estados del IEC 61512.

El siguiente paso será realizar el **Diseño detallado de los procedimientos (DOUs)**, identificados en el análisis, que realizarán las operaciones. Para ello, se utiliza el lenguaje de modelado GRAFCET y, de cada una de las fases, se obtienen las plantillas correspondientes a los Graficets de mando, junto con las señales de control para la coordinación con los Graficets auxiliares y las señales que se deberán considerar en los Graficets de producción. Estas plantillas se particularizan en función del análisis realizado a través de los pasos.

El **diseño del proyecto de automatización** se estructura siguiendo el Modelo de Software

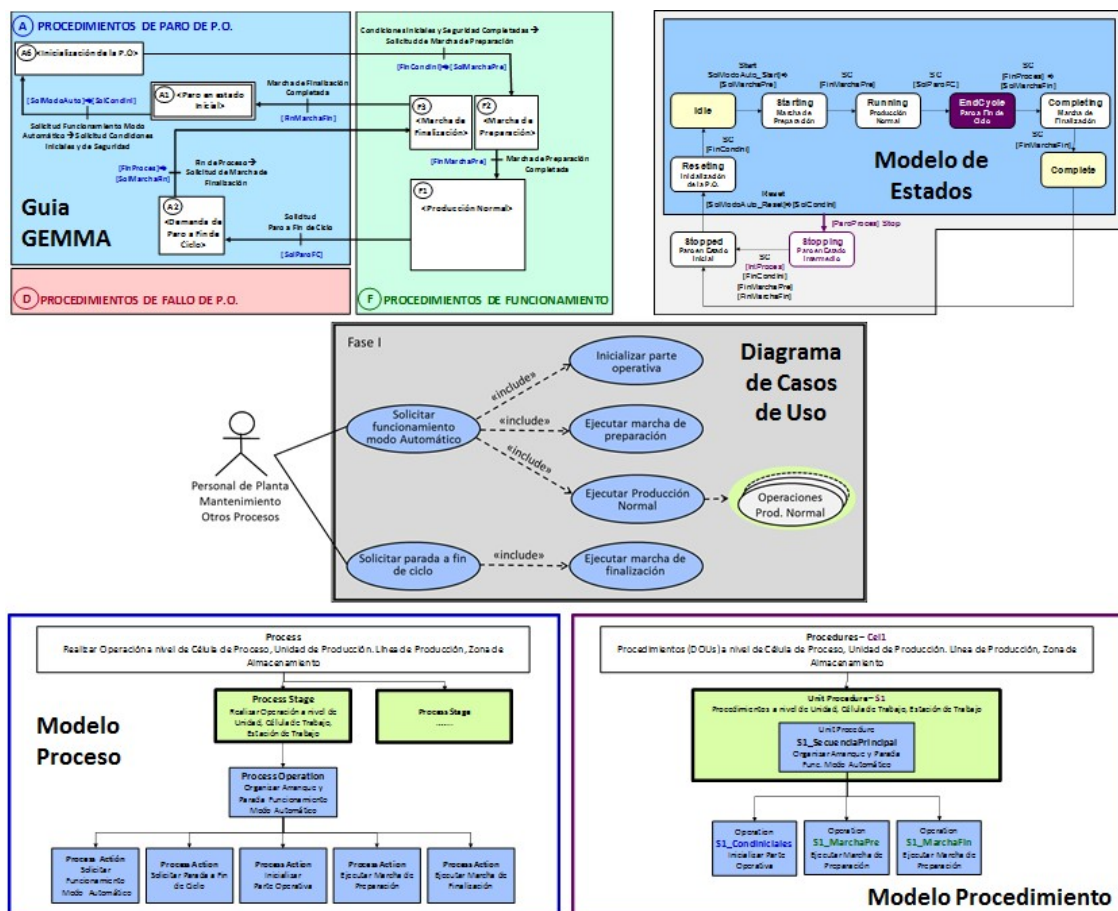


Figura 4: Plantillas de análisis para la Fase I

propuesto en el estándar IEC-61131-3 [8], y la **programación** se realiza utilizando los lenguajes que propone dicho estándar. La programación debe ser un reflejo exacto del diseño, así para cada DOU realizado en el diseño se genera un POU de tipo Bloque de Función (FB – Functional Block), donde se desarrolla la parte secuencial del Grafcet; en un POU de tipo PROGRAM se monta el sistema de control y se generan las acciones diseñadas. En [4] se describe con mayor detalle el proceso de transformación del diseño a código en texto estructurado (ST).

Las pruebas del software son un proceso fundamental dentro del ciclo de desarrollo. En la rama derecha del modelo en V de la Figura 2 se establecen tres fases de pruebas: pruebas unitarias, pruebas de integración, y pruebas de validación y verificación. Estas fases están relacionadas con las fases de la rama izquierda al mismo nivel en las que se definen las pruebas a realizar. Por ejemplo, en la fase de diseño detallado también se realiza el detalle de las pruebas unitarias que se ejecutarán en la fase de pruebas unitarias (al mismo nivel).

Las **pruebas unitarias** permitirán comprobar por separado el correcto funcionamiento de cada uno de los DOUs programados. A cada uno de ellos se le realizan dos tipos de pruebas: estructurales y funcionales. En las pruebas estructurales de la parte secuencial, se testea que la activación y desactivación de cada una de las etapas es correcta, probándose todos los posibles caminos de evolución y cerciorándose de que se recorren adecuadamente. En las pruebas funcionales de la parte combinatorial se testea que las acciones asociadas a las etapas se realizan correctamente al estar éstas activadas.

Como herramienta para realizar las pruebas unitarias se proponen los sistemas de simulación que ofrecen los IDE de programación de PLCs, que permiten la activación de las entradas al SW (salidas del aPS) y la visualización del resto de señales. Otra opción es emplear el HMI o pantalla de explotación como herramienta de pruebas que permite activar las entradas al SW a través de pulsadores que representan los sensores y los controles de mando. Para las pruebas unitarias estas dos opciones pueden ser suficiente, no obstante, requiere tener un conocimiento preciso del comportamiento del aPS.

Las **pruebas de integración** garantizan la correcta coordinación y sincronización entre los DOUs programados. Para ello, por una parte, se realizan pruebas de sincronización que permiten testear las señales de coordinación entre los distintos DOUs y, por otra parte, pruebas de operaciones, en las que se deberá poner el aPS en todos los estados identificados y validar todas las operaciones

involucradas en cada uno de ellos (arranque, ejecución y parada correcta de cada operación).

Como herramienta para realizar las pruebas de integración se propone el uso de un componente software (Gemelo Funcional Base – GBF) cuya función es simular el comportamiento del aPS, creado como una entidad por sí sola y vinculada con el sistema físico en cuestión. Este componente software responde a las órdenes del SW (entradas al aPS) con la activación de las correspondientes entradas (salidas del aPS).

Finalmente, en las **pruebas de verificación y validación**, se realiza un proceso de revisión que verifica que el SW producido cumple con los requisitos funcionales y no funcionales especificados y cumple con las expectativas del cliente. Como herramienta para realizar estas pruebas también se propone el uso de un GFB para la simulación de fallos, sobre todo en aPS que ejecutan operaciones en paralelo.

Como alternativa al GFB se podría utilizar un gemelo digital (DT - Digital Twin) a modo de modelo virtual del aPS. En [13] se describe el desarrollo de un DT de una planta para realizar la puesta en marcha virtual que permite probar el sistema de control en simulación.

En la Figura 5 se distinguen las tres partes en las que se divide un sistema automatizado: la parte de mando o de control (en lila) que precisa el desarrollo metodológico del software de control; la parte operativa o de potencia (en verde) que contempla el aPS y los paneles de operación y/o HMI; y la parte de instrumentación y comunicaciones (en gris) que permite conectar las anteriores. Dicha figura también ilustra los modelos asociados a cada una de las partes, así como los modelos que sirven como nexo permitiendo establecer un vocabulario de unión entre las partes. Así, el Modelo de Proceso identifica las operaciones a realizar en el aPS, las cuales requerirán el desarrollo del SW con el consiguiente Modelo de Procedimientos. El Modelo Entidad Equipo conecta el Modelo Físico que identifica los equipos físicos del aPS con el Modelo de Procedimientos que engloba los procedimientos que las realizan las operaciones.

## 4 FORMACIÓN BASADA EN MeiA. 4.0

La metodología MeiA. está siendo aplicada, desde hace más de una década, en la formación en enseñanzas de ingeniería en automatización en la Escuela de Ingeniería de Bilbao. Las competencias a adquirir por el alumnado se desarrollan integrándolas

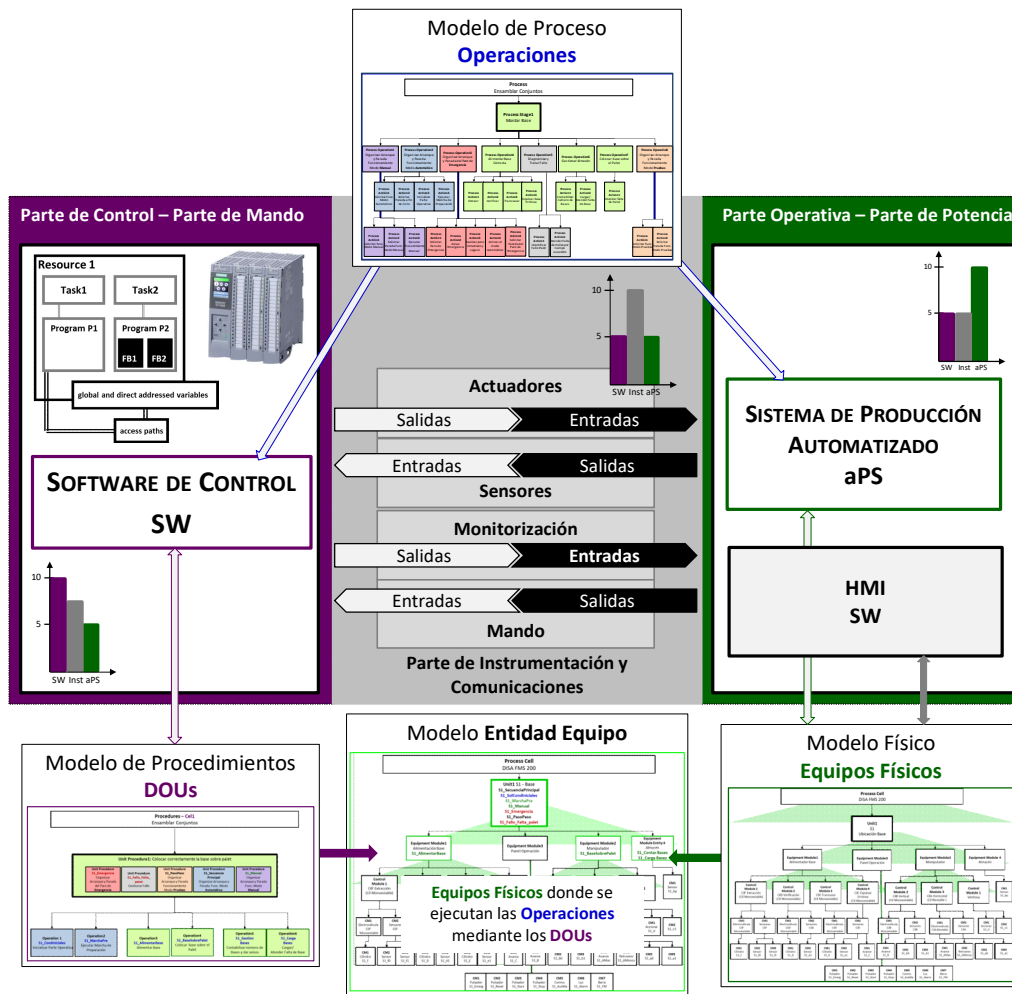


Figura 5: Modelos del estándar IEC 61512 relacionados con las partes de un sistema automatizado

de forma incremental en tres niveles [5]. En un nivel inicial en el que se consolidan conceptos sobre sistemas de control, se realiza el diseño mediante GRAFCET y la implementación conforme al estándar IEC 61131-3. En un segundo nivel o nivel de diseño, se integran operaciones en paralelo y tareas concurrentes que las controlan, resolviendo problemas de coordinación y sincronización, y se establecen, además, requisitos de información para monitorizar el sistema, los elementos de control e indicadores para los paneles de operación, HMI, sistemas de seguridad, etc. En el tercer nivel o nivel metodológico, se trabajan conceptos de jerarquía y se introducen aspectos fundamentales sobre las fases del ciclo de desarrollo de software, arquitecturas software de control conformes a los estándares del sector y nuevos conceptos del dominio de la ingeniería del software aplicados al área. Los casos de estudio o ejemplos aumentan en complejidad a medida que se avanza en los niveles.

En este trabajo, estos niveles se han adecuado con el objetivo de introducir la formación de los nuevos estándares y conceptos aportados por la Industria 4.0.

En el nivel inicial se adapta el vocabulario para emplear los términos utilizados en los modelos del IEC 61512 y se utilizan dichos modelos para explicar los problemas presentados en el aula. La presentación del análisis resultante se acompaña con las pautas para su correcta interpretación.

El nivel de diseño se ha dividido en dos subniveles. El primero coincide con el nivel 2 referenciado en el que se trabaja la identificación de los elementos que integran los distintos modelos del IEC 61512. En el segundo subnivel se trabaja con el ciclo completo de desarrollo de software, realizando el análisis de los problemas trabajados en el primer subnivel mediante los modelos del IEC 61512. También se presenta el estándar IEC-61131-3 identificando los elementos del modelo de software de las implementaciones realizadas, reforzando las fases de pruebas con el GFB, introduciendo conceptos de reutilización desde el análisis, pasando por el diseño detallado hasta los módulos de SW y aportando conceptos de gestión de proyectos de SW.

En el nivel metodológico también se trabaja con el ciclo completo, pero en este nivel se desarrollan

		Nivel Inicial	Nivel Diseño	Nivel Metodológico
<b>Grado de Ingeniería Electrónica Industrial y Automática</b>				
2º Curso 2º Cuat.	Automatismos y Control Oblí. (6cr/3cr)			
3º Curso 1º Cuat.		Informática Industrial Oblí. (6cr/3cr)		
3º Curso 2º Cuat.			Automatización Industrial Oblí. (6cr/6cr)	
<b>Máster en Ingeniería de Control, Automatización y Robótica</b>				
2º Curso				Diseño de Sistemas de Automatización Industrial Opt. (3cr/3cr)
<b>Grado en Ingeniería Informática de Gestión y Sistemas de Información</b>				
3º Curso 1º Cuat.		Programación Práctica de PLCs Opt. (6cr/6cr)		
3º Curso 2º Cuat.			Automatización Industrial Opt. (6cr/6cr)	
<b>Grado de Ingeniería Mecánica / Grado de Ingeniería Eléctrica</b>				
2º Curso 2º Cuat.	Automatismos y Control Oblí. (6cr/3cr)			

Figura 6: Relación de los niveles con las asignaturas de las Titulaciones.

todos los modelos descritos en el apartado 3. Además, se presenta el modelo funcional del estándar IEC 62264 para contextualizar el control de la producción, y se introducen las recetas como herramienta para definir de manera única los requisitos para la fabricación de un producto específico. Se trabaja con los distintos tipos de recetas (general, sitio, maestra y control), contenedores de información según los tipos, procedimientos, etc. y su relación con los modelos del IEC 61512 trabajados en el análisis. También se introduce el modelo de estados del IEC 61512 conectándolo con GEMMA e identificando los estados no contemplados para los aPS discretos. Por último, se diseña e implementa el control de célula conectándolo con los controles directos del aPS, el planificador, supervisor y gestor de recetas.

En la Figura 6 se muestran las asignaturas que permiten obtener las competencias en cada uno de los niveles mencionados en las titulaciones de grado y máster de la Escuela de Ingeniería de Bilbao. Para cada una de las titulaciones se incluye un diagrama de barras que indica el grado de conocimiento requerido por la titulación en relación con las tres partes constituyentes de un aPS identificadas en la Figura 5 (software de control en lila, instrumentación y comunicaciones en gris, y aPS en verde). El mismo tipo de diagrama de barras se utiliza en la figura 5 para resaltar las necesidades de formación en cada una de las partes en relación a las restantes.

## 6 CONCLUSIONES

La versión actualizada de la metodología **MeiA•** permite desarrollar metodológicamente software de control de aPS de acuerdo a un modelo de desarrollo

en **V. MeiA• 4.0** contempla, ya desde la fase de análisis, no sólo las funcionalidades de producción sino todas las funcionalidades transversales o código extrafuncional. Asimismo, **MeiA• 4.0** pone especial atención en las fases de pruebas y en las herramientas que realizan cada tipo de pruebas (sistemas de simulación de los IDE de programación de PLCs y/o HMI o pantalla de explotación para pruebas unitarias; componente software GBF para pruebas de integración, verificación y validación).

El SW desarrollado está preparado para integrarlo dentro de las necesidades de digitalización del mundo físico, ofreciendo sus servicios de operación y de información (operaciones definidas en el Modelo de Proceso, información de sensores y actuadores definidos en el Modelo Físico, información del estado del aPS definido en el Modelo de Estados, etc.)

La necesidad de trabajar con diferentes modelos en los que se involucran conceptos similares desde diferentes vistas de dominio supone una dificultad, que se puede minimizar en parte mediante el uso del color asignado a cada una de las fases de la metodología. Los colores representados en los modelos facilitan la comprensión del análisis realizado y la conexión de conceptos en los distintos modelos resultantes.

La incorporación de **MeiA• 4.0** en las enseñanzas de ingeniería permite abordar las necesidades de formación en estándares de la Industry 4.0 y en ingeniería del software que precisan las personas que se van a incorporar a la industria.

### Agradecimientos

Este trabajo está financiado por MCIU/AEI/FEDER, UE (proyecto RTI2018-096116-B-I00) y GV/EJ (proyecto IT1324-19).

### English summary

## **MeiA• 4.0 TO ADDRESS CURRENT CHALLENGES IN AUTOMATION TRAINING**

### Abstract

Control software for automated production systems is experiencing a rapid increase in quantity and complexity that makes it a key factor in the transition from a traditional vision to a more flexible one capable of meeting the new challenges of Industry

4.0. The industry requires trained personnel to meet these new challenges, with training in both technologies for digital transformation and control software development methodologies necessary to connect the information of the production process with management processes under this new paradigm. This article presents a control software development cycle based on an update of the MeiA methodology that includes the standards proposed by the reference architectures for the companies of the future. In addition, a proposal is made on how to introduce these concepts in engineering education.

**Keywords:** Automated Production Systems; Control software; MeiA. Methodology; Training in automation.

## Referencias

- [1] Álvarez, M.L., Burgos, A., Sainz de Murieta, J. y Sarachaga, M., *MeiA. Metodología para Ingeniería de Automatización. Nivel de Diseño*, Curso OCW - EHU/UPV, Bilbao, España (2017).
- [2] Álvarez, M.L., Sarachaga, I., Burgos, A., Estévez, E., Marcos, M., (2018) “A Methodological Approach to Model-Driven Design and Development of Automation Systems”, *International Journal of Advanced Manufacturing Technology*, 68, pp. 711-725.
- [3] Burgos, A., Álvarez, M.L., Sarachaga, I., Marcos, M., (2016) “Metodología para el desarrollo de procesos de automatización según la norma ANSI/ISA-88”, *XXXVII Jornadas de Automática*, 7-9 Sep., Madrid, España. pp. 987-994.
- [4] Burgos, Álvarez, M.L., A., Iriondo, N., Sarachaga, I., (2020) “Metodología para la transformación de diseños en GRAFCET a código IEC 61131-3”, *Información Tecnológica*, 31(6), [dx.doi.org/10.4067/S0718-07642020000600133](https://doi.org/10.4067/S0718-07642020000600133).
- [5] Burgos, A., Sarachaga, I., Álvarez, M.L., Estévez, E., Marcos, M., (2014) “A Training Proposal base don MeiA to face Automation Challenges”, *International Journal of Engineering Education*, 30(5), pp. 1254-1270.
- [6] DIN/DKE - Deutsches Institut für Normung/ Deutsche Kommission Elektrotechnik, (2016) “German Standardization Roadmap”. Industry 4.0. [www.din.de/go/roadmapindustrie40-en](http://www.din.de/go/roadmapindustrie40-en)
- [7] Fraile, F., Sanchis, R., Poler, R. y Ortiz, A., (2019) “Reference Models for Digital Manufacturing Platforms”, *Applied Sciences*, 9, pp. 4433-4357.
- [8] IEC 61131-3: Int. Electrotechnical Commission, (2003) *IEC 61131-3, Programmable Controllers, Part 3: Programming Languages*.
- [9] IEC 61512-1, (2001) “Batch Control-Part 1: Models and terminology”.
- [10] IEC 62264, (2013) “Enterprise-control system Integration-Part 1: Models and terminology”.
- [11] IEC 12207, (2017) “Systems and software engineering – Software life cycle processes”.
- [12] Lu, Y., Morris, K. C., Frechette, S., (2016) “Current Standards Landscape for Smart Manufacturing Systems”. *National Institute of Standards and Technology (NISTR 8107)*, <http://dx.doi.org/10.6028/NIST.IR.8107>
- [13] Orive, D., Iriondo, N., Burgos, A., Sarachaga, I., Álvarez, M.L., Marcos, M., (2019) “Fault injection in Digital Twin as a means to test the response to process faults at virtual commissioning”, *24th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, 10-13 Sep. Zaragoza, Spain.
- [14] Vogel-Heuser, B., Fay, A., Schaefer, I., Tichy, M., (2015) “Evolution of software in automated production systems: Challenges and research directions”, *The Journal of Systems and Software*, 110, pp. 54-84.
- [15] Vogel-Heuser, B., Fischer, J., Neumann, E. M., Kreiner, M., (2021) “Evolution of software in automated production systems: Challenges and research directions”, *The International Journal of Advanced Manufacturing Technology (Preprint)*, [http://dx.doi.org/10.21203/rs.3.rs-168613/v1](https://doi.org/10.21203/rs.3.rs-168613/v1)
- [16] Vyatkin, V., (2013) “Software Engineering in Factory and Energy Automation: State of the Art Review”, *IEEE Transactions on Industrial Informatics*, 9(3), pp. 1234-1249.



© 2021 by the authors. Submitted for possible open access publication under the terms and conditions of the Creative Commons Attribution CC BY-NC-SA 4.0 license (<https://creativecommons.org/licenses/by-ncsa>)