



Universidad de Valladolid



ESCUELA DE INGENIERÍAS
INDUSTRIALES

Máster en Informática Industrial

MASTER EN INFORMÁTICA INDUSTRIAL

ESCUELA DE INGENIERÍAS INDUSTRIALES
UNIVERSIDAD DE VALLADOLID

TRABAJO FIN DE MÁSTER

SIMULADOR DEL SISTEMA DE PILOTAJE DE RENAULT

Autor: D. Javier Laíz del Agua

Tutor: D. Eusebio de la Fuente López

Valladolid, Mayo, 2016



Resumen.

Con los entornos industriales cada día más cambiantes la simulación ha pasado de ser una necesidad a ser una obligación. La complejidad de las instalaciones es cada vez mayor, y muchas veces no se presenta al inicio de la vida del proyecto, sino que va surgiendo con el tiempo. La única forma que tenemos de anticiparnos al cambio es simulándolo.

Además, el sector del automóvil es especialmente crítico, la fiabilidad que se exige a las instalaciones es cada vez mayor, cercana al 100% y las pérdidas de producción relacionadas con los errores de programación han de eliminarse para no perder competitividad.

Todo esto contribuye a la necesidad de introducir la simulación en todas las fases de los proyectos de automatización: diseño eléctrico, diseño mecánico e ingeniería del software si no queremos quedarnos fuera del mercado.

Abstract.

In the continuous evolution of contemporary industrial environments, the simulation has evolved of being a need to be an obligation. Modern factories present a growing complexity that often does not appear at the beginning of the project but instead it arises over time. The only way to anticipate the change, is simulating the hypothetical scenarios.

In addition, the automotive sector is especially critical due to the high reliability required, reliability that is growing at almost 100 %. The production losses, related to programming errors, have to be eliminated to avoid losing competitiveness.

All this contributes to the need for simulation in all phases of automation projects: electrical design, mechanical design and engineering software in order to compete in the current market.

Keywords: Simulación, PLCs, Renault, PSFV.



INDICE

1	INTRODUCCIÓN.....	1
1.1	El ciclo de vida en los proyectos de automatización.....	1
1.2	Antecedentes.....	4
1.3	Justificación del sistema.....	5
1.4	Planteamiento del problema.....	8
1.5	Objetivos del sistema.....	15
2	ESTADO DEL ARTE.....	17
2.1	Simulación eléctrica.....	17
2.2	Simulación mecánica.....	18
2.3	Simulación robótica.....	21
2.4	Simulación PLCs.....	22
2.5	Simulación de sistemas.....	23
3	ANÁLISIS Y DISEÑO DEL SISTEMA.....	26
3.1	Análisis.....	26
3.1.1	Requisitos no funcionales.....	26
3.1.2	Requisitos funcionales.....	27
3.1.3	Casos de uso del sistema.....	28
3.1.4	Casos de uso del administrador del sistema.....	29
3.1.5	Casos de uso del usuario del sistema.....	32
3.1.6	Diagramas de actividades.....	33
3.1.7	Diagrama de clases.....	37
3.2	Diseño.....	37
3.2.1	Diseño del fichero de configuración.....	37
3.2.2	Diseño de datos de los vehículos.....	39
3.2.3	Diseño de la interface de usuario.....	40
4	RESULTADOS.....	42
4.1	Manual de instalación.....	42
4.2	Manual de configuración.....	43
4.2.1	Definición de los diálogos y sus tablas.....	43
4.2.2	Parámetros generales del diálogo.....	45
4.2.3	Configuración de la tabla de diálogo.....	50
4.2.4	Creación de los datos de los vehículos a transmitir al PLC.....	54



4.3	Manual de usuario.....	54
4.3.1	Proceso de diálogo.	54
4.3.2	PANTALLA DE LA APLICACION.	56
5	CONCLUSIONES.....	59
5.1	Conclusiones del TFM.....	59
5.2	Uso en una aplicación real.	61
5.2.1	Descripción de las modificaciones.	61
5.2.2	Justificación de la simulación.	62
5.2.3	Conclusiones a priori.	63
5.3	Ampliaciones y posibles mejoras.	64
6	REFERENCIAS.	65
A.	ANEXOS.....	66
A.1	Presupuesto.....	66
A.2	Tecnología OPC.....	67
A.2.1	Introducción OPC.....	67
A.2.2	Historia del OPC.....	68
A.3	Proceso de diálogo con PLCs bajo norma Renault EB03.64.210/B.	71



Lista de Imágenes.

Imagen 1. Ciclo de vida en cascada.....	2
Imagen 2. Simulación robótica con Robcad.	3
Imagen 3. Pantalla del simulador de Siemens PLCSim	4
Imagen 4. Funciones de PSFV en las factorías de Renault.	6
Imagen 5. Stock de reconstrucción del film.	11
Imagen 6. Tabla de animación del PLC para el forzado de valores.	14
Imagen 7. Software de diseño See-Electrical.....	18
Imagen 8. Software de diseño CAD SolidEdge.....	19
Imagen 9. Software de diseño CAM Catia.....	20
Imagen 10. Software de diseño CAE Presys.....	21
Imagen 11. Software de diseño Robcad.....	22
Imagen 12. Software de simulación RSLogix.	23
Imagen 13. Software de simulación Simulink.	24
Imagen 14. Software de simulación Arena simulation.....	25
Imagen 15. Casos de uso del sistema.....	28
Imagen 16. Diagrama de actividad arrancar el simulador	33
Imagen 17. Diagrama de actividad detener el simulador.....	34
Imagen 18. Diagrama de actividad configurar el simulador	35
Imagen 19. Diagrama de actividad dialogar con el simulador	36
Imagen 20. Diagrama de clases del sistema.....	37
Imagen 21. Boceto en papel de la interface de usuario.....	41
Imagen 22. Plantilla Excel de configuración del diálogo.....	44
Imagen 23. Parámetros generales del diálogo configuración del canal físico de comunicación para el servidor OPC de Schneider.....	46
Imagen 24. Configuración tabla de diálogo.	52
Imagen 25. Configuración campo ASCII en tabla de diálogo.	53
Imagen 26. Fichero de vehículos.....	54
Imagen 27. Pantalla principal el simulador PSFV.....	56
Imagen 28. Detalle pantalla principal el simulador PSFV.....	57
Imagen 29. Detalle diálogo pantalla principal el simulador PSFV.....	57
Imagen 30. Configuración de respuesta del simulador PSFV.	58
Imagen 31. Configuración de respuesta del simulador PSFV.	58
Imagen 32. Integración de datos en OPC.	67
Imagen 33. Arquitectura cliente – Servidor OPC.....	68



Lista de tablas.

Tabla 1: Comparativo ciclo de vida Ingeniería del Software-Proyectos de automatización.....	2
Tabla 2: Información que intercambia PSFV con las instalaciones de Pintura.....	8
Tabla 3: Secuencia teórica de fabricación.....	12
Tabla 4: Secuencia real de fabricación.....	13
Tabla 5. Parámetros generales del diálogo.....	45
Tabla 6. Configuración del punto de paso “EMON”. En los dos ficheros Excel del simulador.....	47
Tabla 7. Campos de control.....	51
Tabla 8. Desglose de horas del TFM.....	66



Abreviaturas.

PSFV	Pilotage et Suivi des Flux Véhicules
PLC	Programmable Logic Controller.
CAD	Computer aided design.
CAM	Computer aided manufacturing.
CAE	Computer aided engineering.
EMON	Encadenamiento Montaje.
FF	Film Ferme
SSAR	Indice de fabricación del vehículo.
PJI	Identificador del vehículo.
CdC	Cuaderno de Cargas.
TFM	Trabajo fin de máster



1 INTRODUCCIÓN.

1.1 El ciclo de vida en los proyectos de automatización.

Los proyectos de automatización son una parte de los proyectos de ingeniería. Al igual que en el resto de proyectos de ingeniería la tendencia es utilizar modelos de ciclo de vida de desarrollo software clásicos, como puede ser el ciclo de vida en cascada [1].

La ejecución del proyecto es secuencial y hay poca retroalimentación entre las distintas etapas. Esto es uno de los motivos que contribuyen a la dificultad de gestionar el cambio. En un proyecto clásico de ingeniería a lo largo del proyecto se van construyendo equipos y por tanto es difícil retroceder y hacer cambios en el diseño, conllevarán cambios en los medios construidos y un sobre coste derivado de los errores de diseño iniciales.

La modificación del software por errores de diseño en las primeras etapas se percibe como más fácil de modificar, por eso el ciclo de vida en cascada no se adapta del todo bien esta disciplina.

La versión original del desarrollo en cascada fue propuesta por Winston W. Royce en 1970.

El término cascada sugiere la metáfora de la fuerza de la gravedad y el esfuerzo necesario para realizar cambios en las fases tardías del proyecto. Detectar un error en la fase de pruebas, que conlleva un rediseño o una nueva definición de requisitos supone un esfuerzo importante. Hay que volver a realizar de nuevo el análisis, hacer un rediseño, recodificar el software, modificar el hardware construido, y volver a probarlo.

A pesar de esto, es uno de los métodos que más se utilizan.

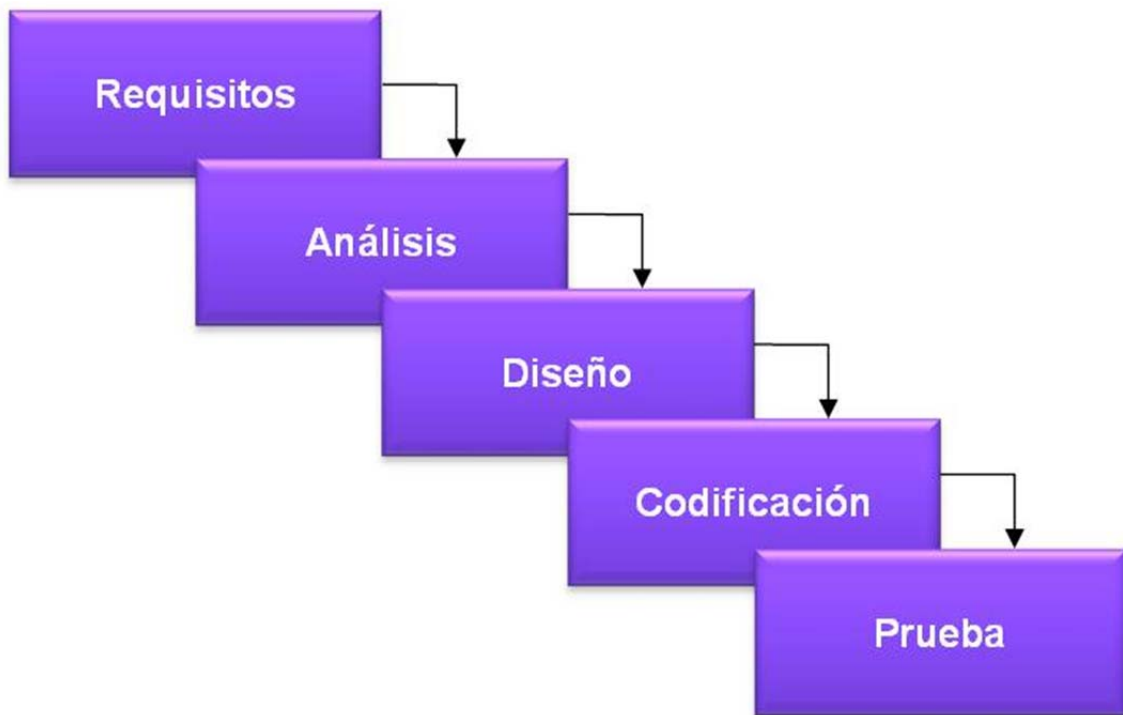


Imagen 1. Ciclo de vida en cascada.

Si comparamos estas etapas con un proyecto de automatización tendremos las siguientes equivalencias.

Ciclo de vida en cascada	Proyecto de automatización
Requisitos	Diseños iniciales. Análisis funcional.
Análisis	Cálculos eléctricos y mecánicos. Proyecto eléctrico básico y mecánico. Estudios de factibilidad.
Diseño	Elección de materiales y compra de equipos.
Codificación	Programación del PLC, pantallas y otros equipos. Fabricación de armarios eléctricos y elementos mecánicos
Prueba	Puesta en producción. Ajustes in-situ.

Tabla 1: Comparativo ciclo de vida Ingeniería del Software-Proyectos de automatización.

El cambio y la necesidad de la simulación.

A lo largo de los años la ingeniería del software ha evolucionado y han surgido otros métodos de desarrollo más eficientes, menos rígidos y más flexibles al cambio. Porque el cambio es inherente a un proyecto de desarrollo software: cambian los requisitos y las necesidades del cliente, la tecnología evoluciona, hay cambios de normativa y aspectos legales, etc. [2][3][4]

Cuanto más tarde nos demos cuenta de los errores, más críticos serán y más rediseño tendremos que hacer, como consecuencia, más caro será corregirlo. Por eso es vital ir introduciendo las pruebas a lo largo del ciclo de vida. Estas pruebas se conseguirán mediante la simulación fuera del entorno final de producción. Llegaremos a las fases finales del proyecto con el menor número de fallos.

Los proyectos de automatización industrial incluyen varias disciplinas: diseño eléctrico, diseño mecánico, desarrollo de software, etc. En las tareas “más clásicas” del proyecto, como son las eléctricas y mecánicas el propio software de diseño, por ejemplo See-Electrical o Eplan en la parte eléctrica, o Catia y Solid Works en la parte mecánica nos permitirán validar los diseños en busca de errores. Estos componentes están sometidos a leyes físicas, por eso es fácil verificar que cumplen estas reglas. El software no se comporta así.

Siguiendo una metodología tendente a reducir los errores diseñaremos y fabricaremos los componentes eléctricos, y una vez finalizada la construcción, los someteremos a ensayos estáticos para asegurarnos de su buen funcionamiento.

Algo similar se hará también con la parte mecánica, donde haremos ensayos y someteremos a test a los elementos que construyamos.

En otras tareas, como es el diseño de células robotizadas, sería indispensable construir la célula sin hacer una simulación previa a la construcción, el coste de la reingeniería superaría el presupuesto del proyecto.

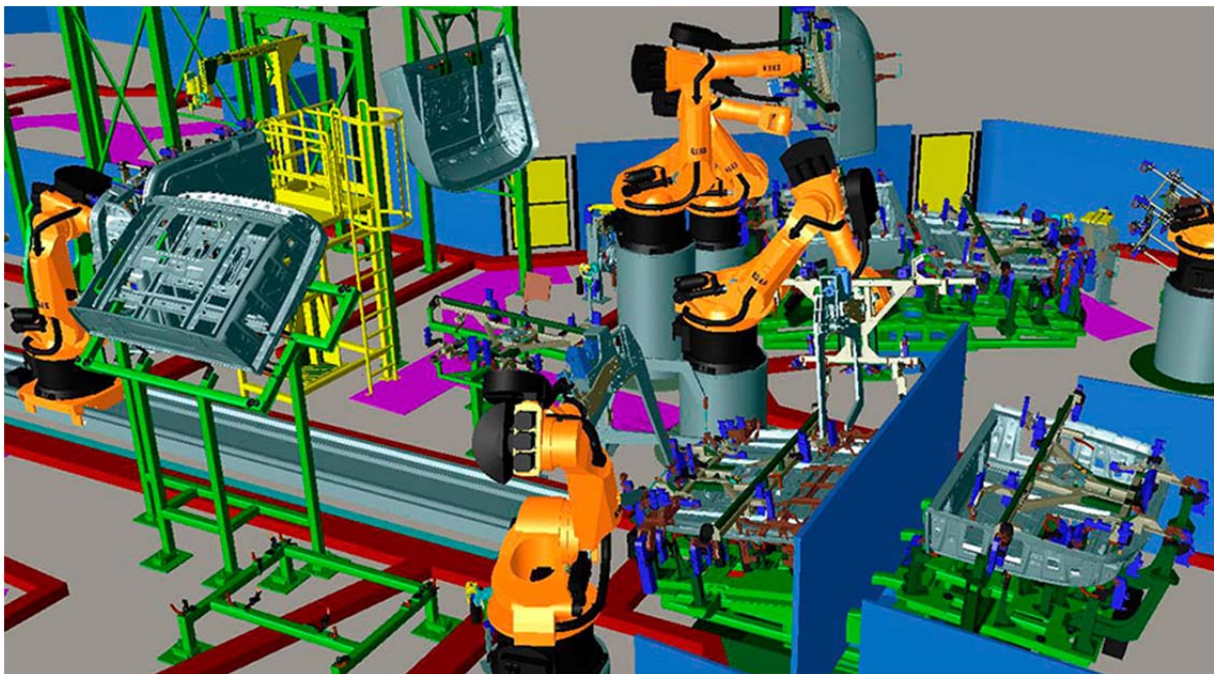


Imagen 2. Simulación robótica con Robcad.

Sin embargo esto no se termina de aplicar al desarrollo de software industrial, y en gran parte es debido a la dificultad para probarlo, por la ausencia de simuladores que reproduzcan el comportamiento del sistema final, y por la multitud de variables y casos que deberíamos probar.

1.2 Antecedentes.

La simulación es uno de los puntos flacos en la programación de PLCs. Algunos fabricantes como Siemens y Omron, incorporan un simulador junto con el software de programación de su autómatas, otros como Schneider Electric, ni siquiera eso.

La solución es transferir nuestro programa en la fase de pruebas a una maqueta física que reproduce el hardware autómatas final, pero sólo el hardware del autómatas, no simulamos el entorno en el que irá.

El simulador virtual o físico se limita a reproducir el hardware del autómatas en la pantalla de nuestro ordenador y realizar tareas como arrancar el programa, pararlo y poder simular las entradas y salidas del sistema de forma virtual.

Realizamos cambios en las variables de forma manual e introducimos los datos en la memoria del autómatas mediante una tabla de variables.

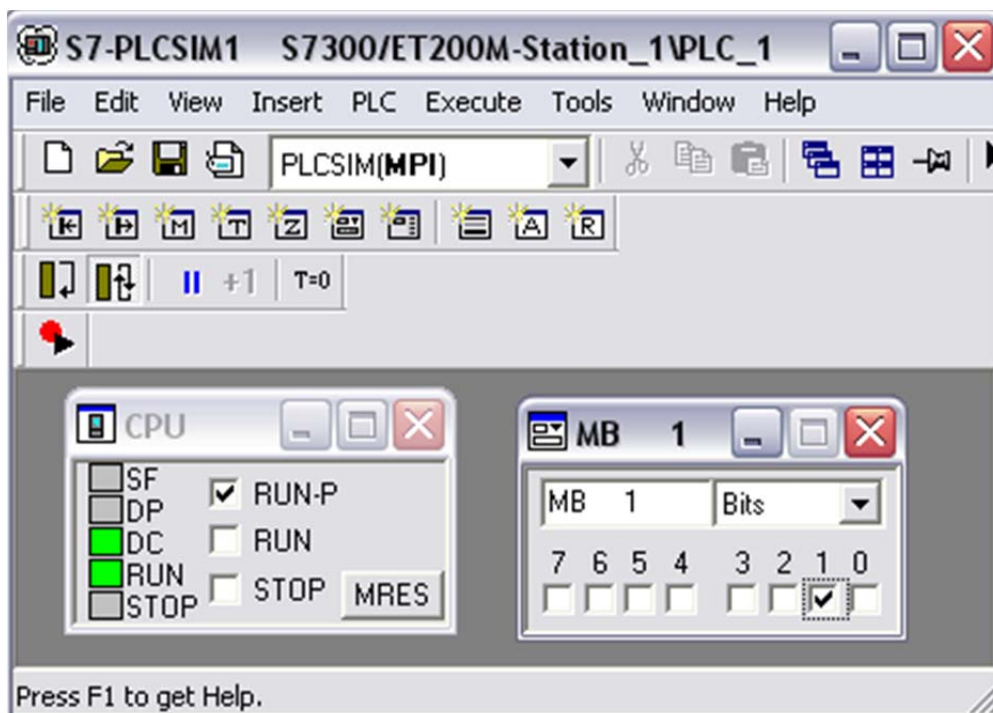


Imagen 3. Pantalla del simulador de Siemens PLCSim

Por eso, cuando disponemos de él, es el propio programador quien debe ir “moviendo los hilos” del simulador para reproducir el comportamiento del sistema, forzando las entradas físicas.

Un primer paso en la simulación es encontrar aquellas señales que para un correcto funcionamiento deben estar a uno o a cero y forzar su valor. Un ejemplo típico son las señales de seguridad, sin el forzado de las cuales, seríamos incapaces de simular el comportamiento.



Después iremos variando el estado de algunas entradas de forma manual. De esta forma es difícil simular los tiempos de respuesta o los funcionamientos complejos, estamos intentando simular un sistema que reacciona en función del tiempo ante eventos (entradas) activando las entradas manualmente sin poder reproducir el funcionamiento temporal del sistema.

Cuando lo que queremos simular es la conexión con un sistema superior, el trabajo se complica. El volumen de información que se intercambia es grande y dinámico y los tiempos de respuesta deben ser pequeños. También hay que simular las condiciones de error: ausencias de respuestas, sobrecarga de la red, respuestas incorrectas, retrasos, etc.

La falta de este software en las etapas de simulación hace que nos enfrentemos al problema real durante la puesta en producción, y que a partir de ese momento, empecemos la depuración de errores. Algo que se traduce en retrasos, sobrecostos y que no nos podrá garantizar entregar un software libre de errores, o al menos con la mayor parte de los escenarios posibles probados.

La detección tardía de errores se traduce en rediseños, aumento de plazos y desviaciones en el presupuesto.

En la etapa de puesta en producción el tiempo es un factor clave, tenemos un medio de producción que debería estar fabricando parado, y por eso no disponemos de mucho tiempo, porque la falta de producción significa una pérdida de ingresos. Deberíamos llegar a esta etapa con el programa lo más probado posible, para así, ser capaces de detectar los últimos errores.

Por eso es necesario simular todo lo posible antes de esta fase, y más en aquellos proyectos donde la comunicación con los sistemas de nivel 2 es clave y hay mucha casuística.

1.3 Justificación del sistema.

El objetivo de nuestro simulador es muy concreto. Simular el intercambio de información entre el PLC que gobierna la instalación y el sistema de gestión de fabricación de Renault, conocido como PSFV (*Pilotage et Suivi des Flux Véhicules*) que le debe proporcionar la información necesaria, para después, con esos datos, poder simular el comportamiento de nuestro programa ante ellos y realizar sus tareas de forma correcta y eficiente.

Este intercambio viene regulado por las norma Renault: “**GE03.FP.035 /A Echanges Automate maître Calculateur**”. [Ref2] y “**Eb0364210_B_Fr Norme d'interface avec l'informatique de Pilotage**”, donde se describen de forma exhaustiva cómo es del proceso de comunicación, el tipo de información que se transmite y el tratamiento de errores. Además, incorpora un bloque función (DFB) o librería para incluir en el autómatas, una especie de caja negra, que dentro tiene toda la programación necesaria para el intercambio.[5]

El proceso de comunicación en sí no es difícil, y más cuando podemos ayudarnos en algo ya programado y probado. Lo complicado es qué hacer con los datos y cómo reaccionará el sistema a lo largo del tiempo recibiendo un sinfín de datos heterogéneos.

Nivel 2 en Renault. PSFV.

PSFV es uno de los sistemas de gestión informático de Renault. Maneja el flujo de vehículos y proporciona a las instalaciones automatizadas la información necesaria para trabajar. Es un sistema informático global presente en todas las fábricas de Renault encargado de gestionar la información completa del vehículo con más de 30 funciones a su cargo. Ningún taller sería capaz de fabricar sus vehículos sin la ayuda de PSFV.

La información que nos proporciona PSFV es cada vez es más necesaria y en muchas ocasiones obligada. Las fábricas de hoy en día producen varios modelos, muy diferentes entre sí. Además, cada vez existen más diversidades, más opciones que podemos elegir y eso se traduce en una mayor cantidad de información que deben trasladar los sistemas de gestión a las máquinas para fabricar los vehículos.

Este intercambio de información es bidireccional. Lo habitual es que los autómatas pregunten a PSFV sobre las características del vehículo a trabajar (modelo, tipo, color, etc.) o de los procesos a realizar (tipos de apriete y pares, soldadura a realizar, sellante a usar, etc.) aunque hay ocasiones en los que son los autómatas los que trasladan información a PSFV, como por ejemplo la hora a la que ha pasado el vehículo, los posibles defectos en el proceso que ha sufrido, si se ha desviado de su ruta original y hacia donde se le ha enviado o informarle de los resultados de la fabricación para poder gestionar la trazabilidad de los vehículos.[6]

PSFV dans le contexte Usine de Fabrication

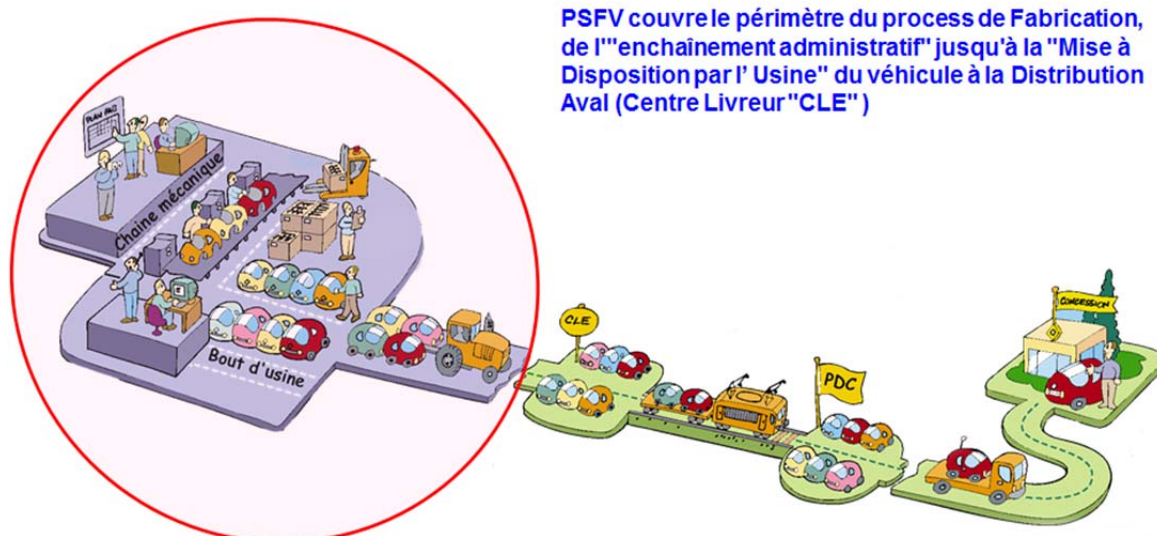


Imagen 4. Funciones de PSFV en las factorías de Renault.



Podemos hacernos una idea de la información necesaria para fabricar un vehículo en un taller sencillo como es la nave de Pintura, donde no se añaden piezas al vehículo y sólo se le aplican los tratamientos de pintura viendo la tabla que intercambian estas instalaciones con PSFV.

Nº de Palabra	Tipo Dato	Formato	Valores	Descripción
Palabra 0	Indicador escritura API (VALAPI)	Binario / 5	1, 0	Indica el inicio del intercambio
Palabra 1	Portador N	Binario / 4	0-9999	Información utilizada para pasar de trineo a balanceta.
Palabra 2	Portador N+1	Binario / 4	0-9999	Información utilizada para pasar de trineo a balanceta.
Palabra 3	Programa (PJI)	ASCII / 2		Primera palabra del PJI
Palabra 4	Día (PJI)	ASCII / 1		Segunda palabra del PJI
Palabra 5	Identificación (PJI)	ASCII / 2		Tercera palabra del PJI.
Palabra 6	Identificación (PJI)	/		Cuarta palabra del PJI
Palabra 7	Vacío	---	---	---
Palabra 8	Orientación ZONA	Binario / Var	(1)	Indica zona destino de una caja con defecto.
Palabra 9	Vacío	---	---	---
Palabra 10	Familia (criterio)	ASCII / 2	95 / FE /FB	Criterio en claro
Palabra 11	Modelo (criterio)	ASCII / 1	B, D, K, H, V, J, E, S, F, G, N	Criterio en claro
Palabra 12	Vacío	---	---	---
Palabra 13	Motorización (criterio)	ASCII / 2	(2)	Criterio en claro
Palabra 14	Equipamiento (criterio)	ASCII / 2	SA, EO, AM, EA, E3, E5, E2	Criterio en claro
Palabra 15	Diversidad techo (criterio)	ASCII / 2	CA, TO, TN, TV, HA	Criterio en claro

Palabra 16	Diversidad dirección (criterio)	ASCII / 2	DG, DD	Criterio en claro
Palabra 17	Tipo caja (criterio)	Binario / 5	(3)	Libelle methode modope
Palabra 18	Color 1 (criterio)	ASCII / 2	(4)	Criterio en claro
Palabra 19	Color 2 (criterio)	ASCII / 2	(4)	Criterio en claro
Palabra 20	Color 3 (criterio)	ASCII / 2	(4)	Criterio en claro
Palabra 21	Nº Secuencia	Binario / 5		
Palabra 22	Nº Secuencia	Binario / 4		
Palabra 23	Nº Secuencia	Binario / 4		
Palabra 24	Vacío			
Palabra 25	Vacío			
Palabra 26	Vacío			
Palabra 27	Vacío			
Pal 28-36			
Palabra 37	Raqueta			
Palabra 38	Indicador Lectura Calculador (VALCAL)	Binario / 5		
Palabra 39	Indicador Marche Locale	Binario / 5		

Tabla 2: Información que intercambia PSFV con las instalaciones de Pintura.

Características del simulador a desarrollar.

- Debe ser fácil de configurar.
- Debe permitirnos simular diferentes escenarios, y tener varios procesos de comunicación, porque a menudo, una instalación tiene diferentes diálogos con PSFV.
- Debe ser rápido para poder probar el comportamiento temporal de nuestro programa de PLC.
- Ha de permitirnos probar los errores del sistema que veremos más tarde.
- La información que nos proporcionará es muy heterogénea, y debemos poder variarla de forma sencilla, para probar todos los escenarios posibles.
- El interface debe proporcionar “feedback” al usuario.
- Debe permitirnos comunicar con las diferentes variantes que existen: por PJI o trineo, devolver los datos de un vehículo concreto o el siguiente, etc.

1.4 Planteamiento del problema.

En los puntos anteriores hemos visto que la comunicación en sí con PSFV no presenta una gran complejidad, disponemos de una librería o bloque función DFB ya probado para realizar esta comunicación. La dificultad estriba en tratar los datos que recibimos, gestionarlos y hacer que nuestra instalación se comporte de la forma más eficiente.



Cada vehículo dispone de la siguiente información importante para nuestra instalación, y que recibiremos a través de PSFV:

- PJI: Número de 7 dígitos que identifica de forma unívoca el vehículo.
- Modelo: Tipo de coche a fabricar (Mégane 5 puertas, Mégane Ranchera, Kadjar, etc.).
- Número de secuencia: Número correlativo que define la secuencia de fabricación de los vehículos. Este número evoluciona de forma creciente, y para respetarlo, el coche con número de secuencia 1020 debe fabricarse antes que el número 1021 y después del coche con número de secuencia 1019.

El PJI.

En PSFV se guarda información de cada coche a trabajar asociada a su número de bastidor (VIN).

Cómo el número VIN es un número complicado para trabajar, de forma interna, se utiliza el identificador llamado PJI (Programme/Jour/Identité) que aunque se repite año a año, es único a lo largo del año.

El PJI está compuesto por siete caracteres numéricos:

- P: 2 caracteres. Representa la semana de fabricación. No puede valer cero, aunque en algunos casos extraordinarios puede llegar a valer 90 (cajas de pruebas).
- J: 1 carácter. Día de la semana en el que se fabrica el vehículo.
- I: 4 caracteres. Número secuencial.

A su vez los coches van montados sobre un bastidor apropiado para cada fase de producción. El bastidor, que a partir de ahora llamaremos trineo, es distinto para el proceso de soldadura, pintura, y a su vez distinto al de montaje.

Todos los trineos tienen un número que los identifica, y son únicos. No habrá nunca dos trineos del mismo tipo (soldadura, pintura, montaje) con el mismo número.

PSFV siempre sabrá sobre que trineo va montado cada coche. Este dato, se denomina identificador secundario, porque de forma indirecta, identifica al coche que soporta.

Podemos consultar los datos de un vehículo a través de su VIN, PJI o número de trineo sobre el que va montado.

Información en PSFV.

En PSFV se almacena toda la información necesaria para fabricar el vehículo:

- Tipo de modelo, familia, color y resto de características físicas del modelo.
- Piezas que lo componen en función de su nivel de equipamiento.
- Información de los procesos a aplicar: tipos de soldadura, aplicación de sellado, programas de atornillado, etc.

A su vez, se van guardando los resultados de la propia fabricación:



- Hora de paso por cada una de las instalaciones.
- Resultados de las operaciones efectuadas.
- Defectos encontrados: persona que lo identifica, descripción del defecto, y hora de su subsanación.

Existe una normativa Renault que define como traspasar estas cantidades de información entre el sistema de pilotaje y los autómatas. GE03.FP.035

En esta norma se especifica el tipo de comunicación entre el sistema y el autómata y cómo se gestiona este intercambio.

En la memoria del autómata existe una zona dedicada (máximo 100 palabras) donde por turnos escribe cada uno de ellos. A menudo con este tamaño es suficiente, aunque el sistema debe estar abierto a que puedan existir otras tablas para aumentar la información, o a realizar varios diálogos consecutivos para completar la información que necesita la instalación.

Un caso típico:

Para entender el problema vamos a exponer un caso típico de aplicación de PSFV. Nuestra instalación es un stock o almacén que debe ordenar las carrocerías que van entrando, para sacarlas hacía la siguiente instalación de forma ordenada.

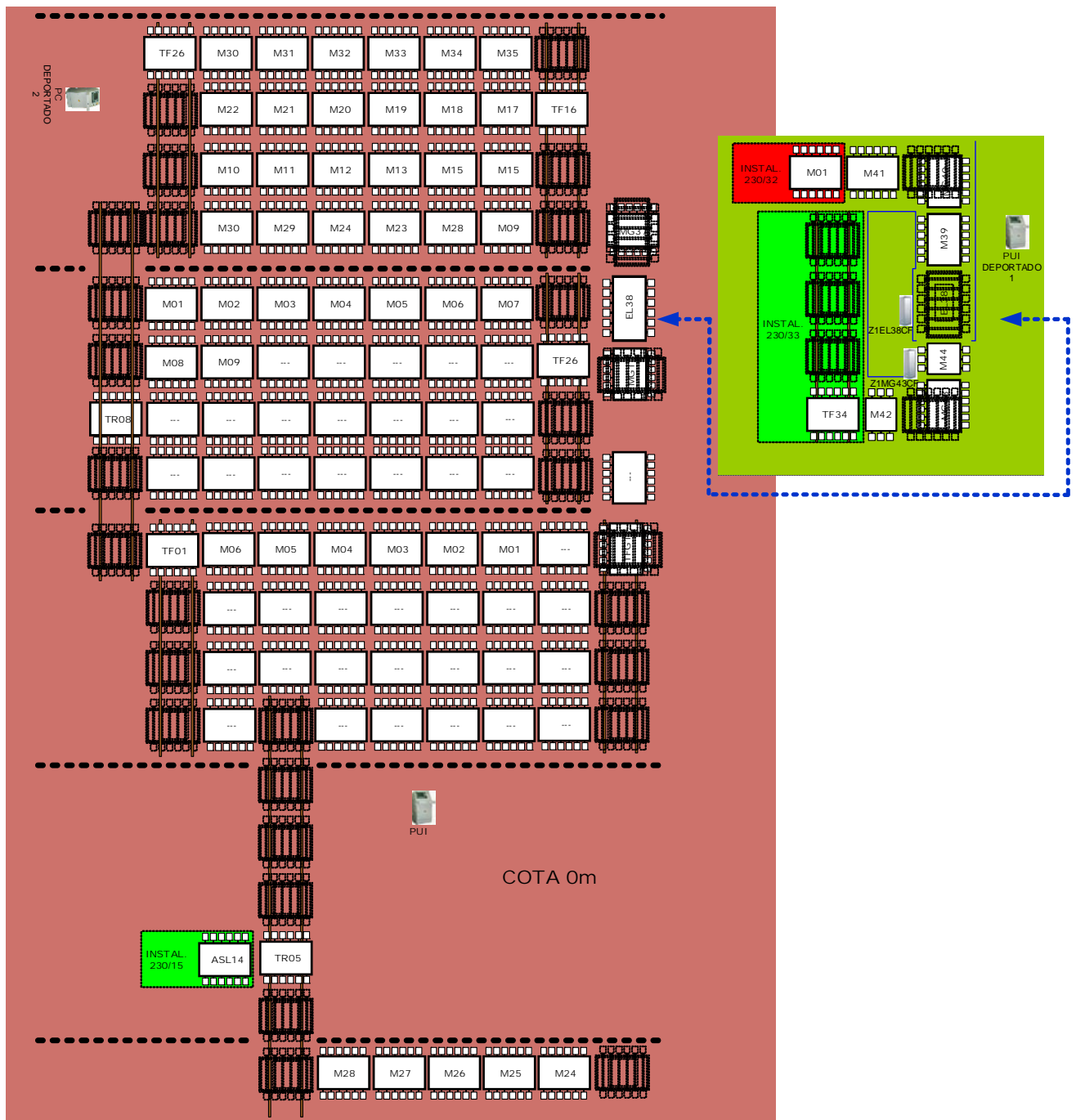


Imagen 5. Stock de reconstrucción del film.

Los coches a fabricar deben hacerlo según una secuencia de fabricación. Deben salir de la cadena ordenados por la fecha teórica de entrega al cliente, aunque conforme van pasando por las diferentes etapas de fabricación (soldadura, pintura, montaje) se pueden ir retrasando o adelantando si han tenido problemas. Los stocks de reconstrucción de film, son los encargados de volver a reordenarlos. Esta es la función del caso que vamos a exponer para entender la utilidad de nuestro simulador.

En las siguientes tablas podemos ver un caso real. Primero la secuencia teórica de producción, y después, la secuencia de producción en la nave de Montaje, que son las últimas estaciones del proceso de fabricación. En Montaje ya nos llegarán los coches algo desordenados.

PJI	Orden de fabricación	Modelo
2820480	2879989	HFE
2330194	2879990	K95
2130812	2879991	HFE
2330132	2879992	B95
2150232	2879993	HFE
2640261	2879994	B95
2910589	2879995	HFE
2320207	2879996	B95
2130766	2879997	HFE
2330465	2879998	K95
2320461	2879999	HFE
2230156	2880000	K95
2150216	2880001	HFE
2330523	2880002	B95
2120577	2880003	HFE

Tabla 3: Secuencia teórica de fabricación.

En la secuencia teórica de fabricación todos los coches van ordenados. En la realidad esto es muy difícil que suceda y nos encontraremos un caso como el siguiente:

PJI	Orden de fabricación	Modelo
2820480	2879989	HFE
2150232	2879993	HFE
2330194	2879990	K95
2130812	2879991	HFE
2910589	2879995	HFE

2130766	2879997	HFE
2330465	2879998	K95
2320207	2879996	B95
2320461	2879999	HFE
2230156	2880000	K95
2150216	2880001	HFE
2330132	2879992	B95
2640261	2879994	B95
2330523	2880002	B95
2120577	2880003	HFE

Tabla 4: Secuencia real de fabricación.

La segunda tabla refleja la secuencia real de fabricación, donde los vehículos amarillos se han adelantado, y el PLC deberá retrasarlos. Los vehículos marcados de color naranja van retrasados, y el algoritmo del PLC deberá adelantarlos para sacarlos en su orden.

Los stocks de reconstrucción deberán intentar que el orden de los coches que llegan a montaje se parezca lo más posible a la secuencia teórica de fabricación.

Por eso deberá ir ordenando los coches según le van entrando (desordenados) para poder sacarlos hacia la instalación siguiente lo más ordenados posibles.

La eficiencia del algoritmo de ordenación, que es lo que queremos verificar mediante la simulación, dependerá de múltiples factores:

- El estado de ocupación del stock. Si está lleno, no será capaz de ordenar los coches correctamente, su funcionamiento será FIFO.
- También nos influirá la salida del stock. Si la instalación siguiente no nos demanda coches, empezaremos a saturarnos, y nuestra capacidad de reordenación será menor.
- Cómo de ordenados o desordenados llegan los coches. Si llegan relativamente ordenados su tarea será más sencilla. En cambio, si llegan muy desordenados la salida no será tan perfecta.
- La programación del propio algoritmo. El programa del PLC incorporará algún tipo de algoritmo de ordenación (burbuja, inserción, quick sort, etc.)

Simulación manual (Tradicional).

Ahora pensemos como podemos simular esto sin la ayuda de un simulador.

Lo primero que deberíamos hacer es utilizar un simulador de PLC simple, como PLCSim, que nos permite simular el hardware, o cargar el programa autómatas en un PLC como el que iremos a utilizar en la instalación real.

Iremos cargando datos en la memoria de nuestro autómatas para partir de una configuración inicial.

Pasamos después a simular la llegada de un vehículo. Utilizando una tabla de variables, forzaremos la respuesta de PSFV escribiendo de forma manual los datos de un vehículo. Estos datos pueden ser más o menos complejos, los que tienen una representación decimal serán más fáciles de forzar que los que tienen formato ASCII y en la memoria del autómatas se guardan en formato hexadecimal.

Al escribirlos manualmente, tampoco podremos reproducir el tiempo de respuesta de PSFV, que en términos generales no debería sobrepasar los 2000 milisegundos.

Variable	Símbolo	Valor actual	Naturaleza	Tipo	
%Mw15000	M01tip				MESA M01:Dato etiqueta:Tipo Caja (Inicio Tabla 15 Palabras)
%Mw15001	M01mod				MESA M01:Dato etiqueta:Modelo Caja
%Mw15002	M01clc				MESA M01:Dato etiqueta:Color (Millares,Centenas)
%Mw15003	M01colu				MESA M01:Dato etiqueta:Color (Decenas,Unidades)
%Mw15004	M01div				MESA M01:Dato etiqueta:Diversidad
%Mw15005	M01pji_s				MESA M01:Dato etiqueta:PJI Semana
%Mw15006	M01pji_d				MESA M01:Dato etiqueta:PJI(Dia)
%Mw15007	M01pji_m_c				MESA M01:Dato etiqueta:PJI(Mil,Cent)
%Mw15008	M01pji_d_u				MESA M01:Dato etiqueta:PJI(Dec,Uni)
%Mw15009	M01des				MESA M01:Dato etiqueta:Destino
%Mw15010	M01status				MESA M01:Dato etiqueta:Status
%Mw15011	M01despg				MESA M01:Dato etiqueta:Destino(SOLO GESTION INTERNA)

Imagen 6. Tabla de animación del PLC para el forzado de valores.

Este es un procedimiento tedioso, no falto de errores, y que debemos repetir mecánicamente durante bastante tiempo, para probar todas las variables de nuestro algoritmo. Esta complejidad a la hora de simular será uno de las principales causas por la que el programa no llegará lo suficientemente probado a la fase de puesta en marcha. Es un proceso farragoso y costoso.

Si además tenemos varios puntos de conexión con PSFV, para recibir datos de dos fuentes distintas la “simulación manual” se complica.

Utilización de un simulador.

Necesitamos disponer de una máquina, que una vez iniciada nos proporcione datos de forma automática igual que lo haría PSFV, pero con la posibilidad de poder variar esos datos a nuestro antojo, para probar diferentes combinaciones.

Además, los datos de los vehículos se deben poder variar de forma sencilla, sin complicados conocimientos sobre bases de datos. El simulador ha de ser capaz de comunicar con el autómatas de diferentes maneras. Supongamos que nuestra instalación tiene dos zonas de entrada. El simulador debe disponer de dos procesos de comunicación para responder a las peticiones por cada punto.

Una forma sencilla de trabajar es guardar en un fichero Excel la configuración de nuestro simulador, y en otro fichero Excel, los datos de los vehículos que tenemos en PSFV.

Si queremos variar el orden de los vehículos, bastará con variar el orden de las filas del fichero Excel.

De la misma forma, si queremos cambiar la configuración de nuestro diálogo PLC-PSFV, con cambiar el fichero Excel de configuración y reiniciar el simulador será suficiente.



El simulador debe representar el comportamiento temporal de PSFV y también los posibles errores de comunicación, que se resumen en:

- PSFV no responde. El programa del PLC debería generar un defecto e informar al usuario.
- PSFV responde que no conoce el vehículo por el que preguntamos. El programa del PLC debería generar un defecto e informar al usuario.
- El coche por el que estamos preguntando a PSFV es el último de su lista. No conoce más y nos propone esperar a que entre otro vehículo en el sistema. El programa del PLC informa al usuario y reintenta el diálogo pasado un tiempo prudencial.
- PSFV responde con normalidad.

1.5 Objetivos del sistema.

El objetivo principal del sistema a crear es diseñar y construir un sistema informático que nos permita simular el comportamiento del sistema de pilotaje de fabricación de Renault.

Para ello este sistema ha de cumplir una serie de objetivos o requisitos que a analizaremos más exhaustivamente en la fase de análisis de requisitos antes de comenzar con el desarrollo.

Podemos clasificar los objetivos en dos tipos:

No funcionales.

Determinan el comportamiento del sistema. No se deben confundir con objetivos generales o buenas ideas. Son tan necesarios como los objetivos funcionales, porque de nada vale que nuestro sistema realice sus tareas si no las hace de forma eficiente, o presenta una complejidad alta de uso.

- Facilidad de configuración.
- Rapidez de respuesta.
- Consumo de pocos recursos.
- Independencia de drivers de comunicación.

Funcionales.

Determinan la funcionalidad del sistema.

- La aplicación ha de poder ser iniciada y parada. Al hacerlo se bloquearán o liberarán de forma automática los recursos de comunicación del sistema.
- La aplicación debe funcionar de forma autónoma sin intervención del usuario, a no ser que este, quiera cambiar algo.
- La aplicación debe dar soporte para comunicar hasta con seis autómatas distintos, o un mismo autómata con seis puntos de diálogo distintos.



- La tabla de comunicación con el PLC puede ser de hasta 100 palabras.
- El fichero de coches que maneja la aplicación para enviar al autómeta puede contener varias hojas, con los diferentes trayectos que realizar el coche, uno por ejemplo con el orden en soldadura, pintura, etc.
- El usuario ha de poder provocar desde la aplicación los distintos errores presentes en este tipo de comunicación (falta de respuesta, respuesta incorrecta, etc.)
- La aplicación debe tener una interface visible que transmita al usuario en tiempo real el funcionamiento del sistema.

-
-
-



2 ESTADO DEL ARTE.

La simulación en los proyectos industriales abarca tanto como disciplinas incluyen este tipo de proyectos. A continuación haremos un breve repaso sobre los sistemas existentes y las bondades y carencias que tienen.

Lo primero que debemos hacer al estudiar un campo tan complejo es hacer una clasificación previa, aunque dado lo extenso del tema, corremos el riesgo de dejarnos olvidado algún sistema.

- Simulación eléctrica.
- Simulación mecánica.
- Simulación robótica
- Simulación PLCs.
- Simulación informática industrial.
- Simulación Completa.

2.1 Simulación eléctrica.

Dentro de este campo hay aplicaciones específicas para el diseño de componentes electrónicos y circuitos integrados aunque nos quedaremos sólo con el estudio de las utilizadas en el diseño de sistemas eléctricos que son las que utilizaremos más habitualmente en nuestros proyectos industriales. Podemos destacar programas como Eplan, See-Electrical, X-Elec, que proporcionan al usuario un entorno de diseño donde “dibujar” los esquemas eléctricos.

Además de servirnos para dibujar nuestros planos, estas aplicaciones también nos permitirán validar eléctricamente nuestro esquema e informarnos de errores como puede ser juntar dos hilos de distinto potencial, no conectar algún cable, etc.

También nos generarán de forma automática el listado de materiales eléctricos, el etiquetado de los cables, el diseño de borneros, etc. Sin la ayuda de este tipo de software sería muy complicado hacer un buen diseño eléctrico, más si tenemos en cuenta, que un proyecto industrial de tamaño medio puede tener en torno a 700 planos eléctricos.

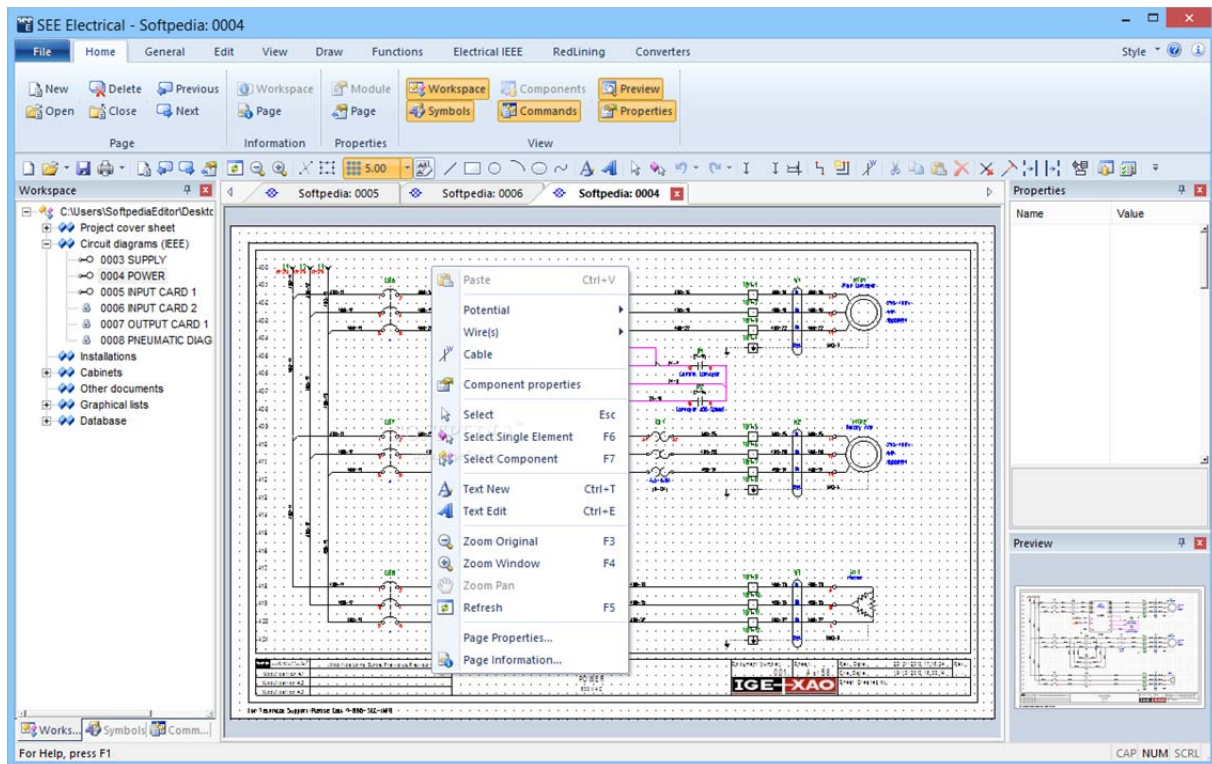


Imagen 7. Software de diseño See-Electrical.

La aparición de este software ha contribuido a reducir el coste del proyecto eléctrico. Por un lado al aumentar la eficacia del trabajo de los diseñadores y por otro al reducir los errores en las fases de construcción.

2.2 Simulación mecánica.

El diseño mecánico al igual que el eléctrico obedece a leyes físicas. Con las herramientas informáticas actuales, no sólo podemos diseñar nuestras máquinas, sino que también podemos probar su comportamiento y realizar ensayos antes de construirlas creando modelos que simulen la realidad.

También podemos hacer ensayos sobre ergonomía, materiales a utilizar, dinámica de los componentes, durabilidad, etc.

Son también conocidas como herramientas CAD (Diseño Asistido por Computadora), CAM (Fabricación Asistida por Computadora) y CAE (Ingeniería Asistida por Computadora).

Las herramientas CAD son las más antiguas de los tres tipos vistos (CAD, CAM y CAE) y han servido para dibujar en 2D y modelar en 3D los sistemas, aquí encontramos software como AutoCAD o Catia.

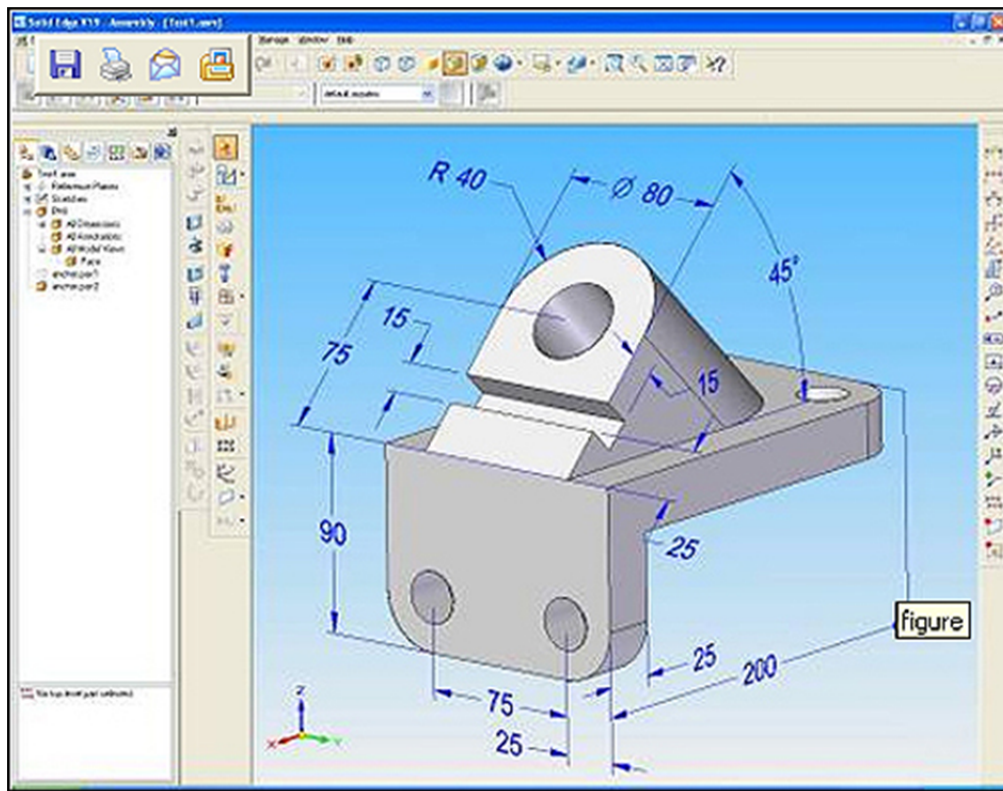


Imagen 8. Software de diseño CAD SolidEdge.

Las herramientas CAM aparecieron después de las CAD para asistir a los diseñadores mecánicos y facilitar la construcción de sus diseños. Puede haber un traspaso directo de los diseños del ordenador a la maquinaria de mecanizado (CNC) encargada de fabricar las piezas. El ejemplo más conocido de este tipo de herramientas es CATIA.

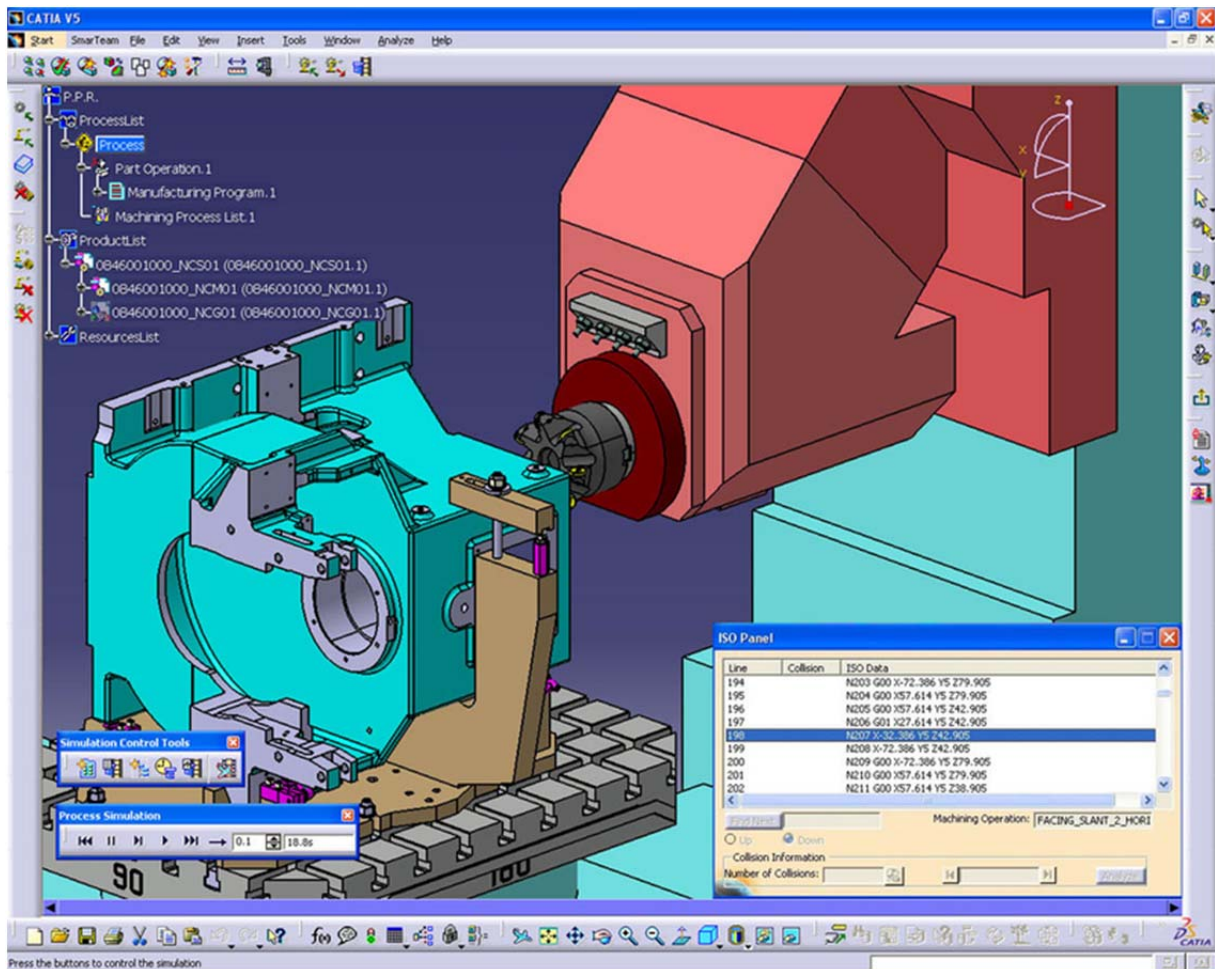


Imagen 9. Software de diseño CAM Catia.

Las herramientas CAE [7] (Ingeniería Asistida por Computadora) nos permiten analizar, diseñar y simular nuestros diseños, para valorar sus características, propiedades, viabilidad, y rentabilidad. Con el fin es optimizar su desarrollo y consecuentes costos de fabricación, y reducir al máximo las pruebas para la obtención del producto deseado, incorporando para ello:

- Análisis cinemático.
- Análisis por el método de elementos finitos (FEM, Finite Elements Method).
- Maquinado por control numérico CNC (Computered Numeric Control).
- Exportación de ficheros "Stl" (Estereolitografía) para máquinas de prototipado rápido.
- CAD

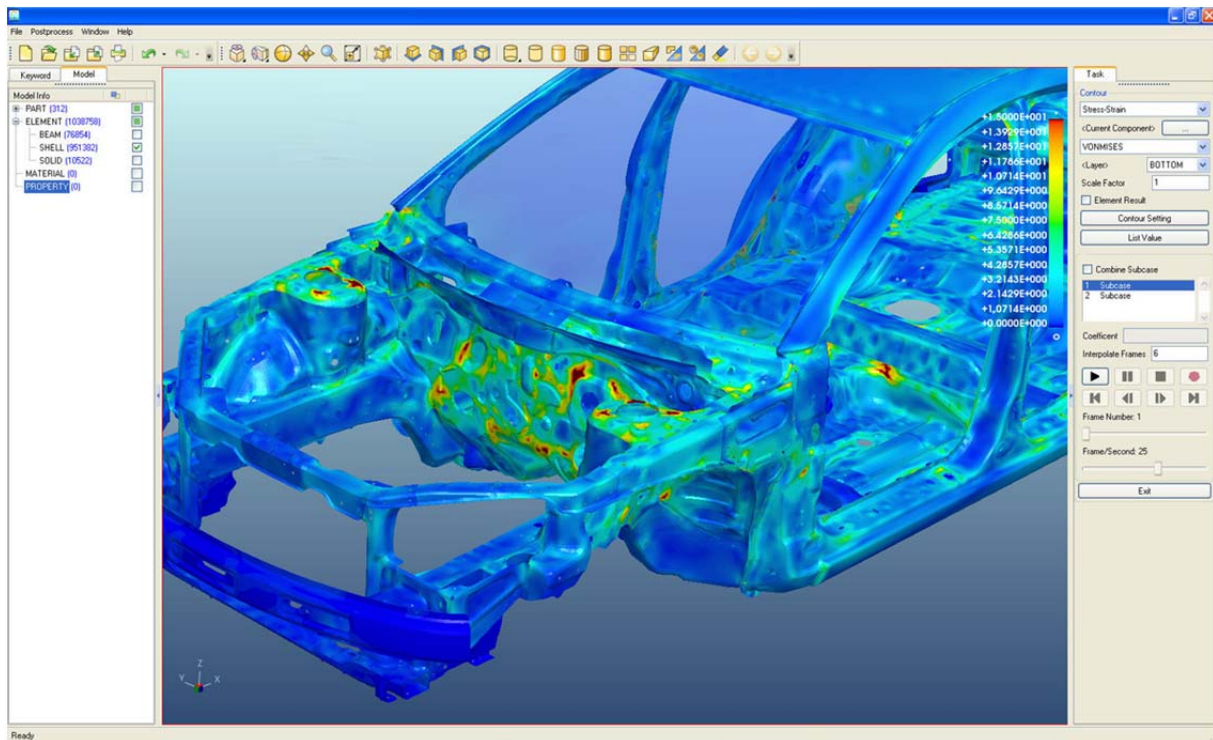


Imagen 10. Software de diseño CAE Presys.

2.3 Simulación robótica.

La simulación robótica a menudo se engloba dentro de la simulación mecánica porque los mecanismos y manipuladores robóticos son elementos mecánicos o con la simulación de sistemas de automatización porque son gobernados por un controlador industrial. Aunque este tipo de simulación tiene la bastante entidad e importancia como para estudiarla por sí sola.

En el diseño de un sistema robotizado complejo es obligado realizar una simulación robótica previa, para determinar la viabilidad del sistema, el alcance del manipulador, las interferencias con otros elementos mecánicos y elegir los elementos a utilizar antes de comenzar el trabajo.

Con la simulación robótica de una estación de soldadura podemos determinar si el robot elegido es el correcto porque alcanza todas las posiciones que necesitamos, si la pinza de soldadura nos permitirá soldar los puntos necesarios y si el utillaje a utilizar no tiene interferencias con el robot y su pinza.

Además podemos simular la cinemática del robot, trayectorias, trabajo combinado con otros robots, tiempos de ciclo e incluso programar las trayectorias fuera de línea y cargárselas al robot final para realizar un último ajuste.

Cada fabricante de robots tiene un software para hacer estas tareas (KUKA, ABB, FANUC, etc.) aunque hay también software genérico que es independiente del robot, como Robcad donde podemos combinar diferentes tecnologías robóticas.

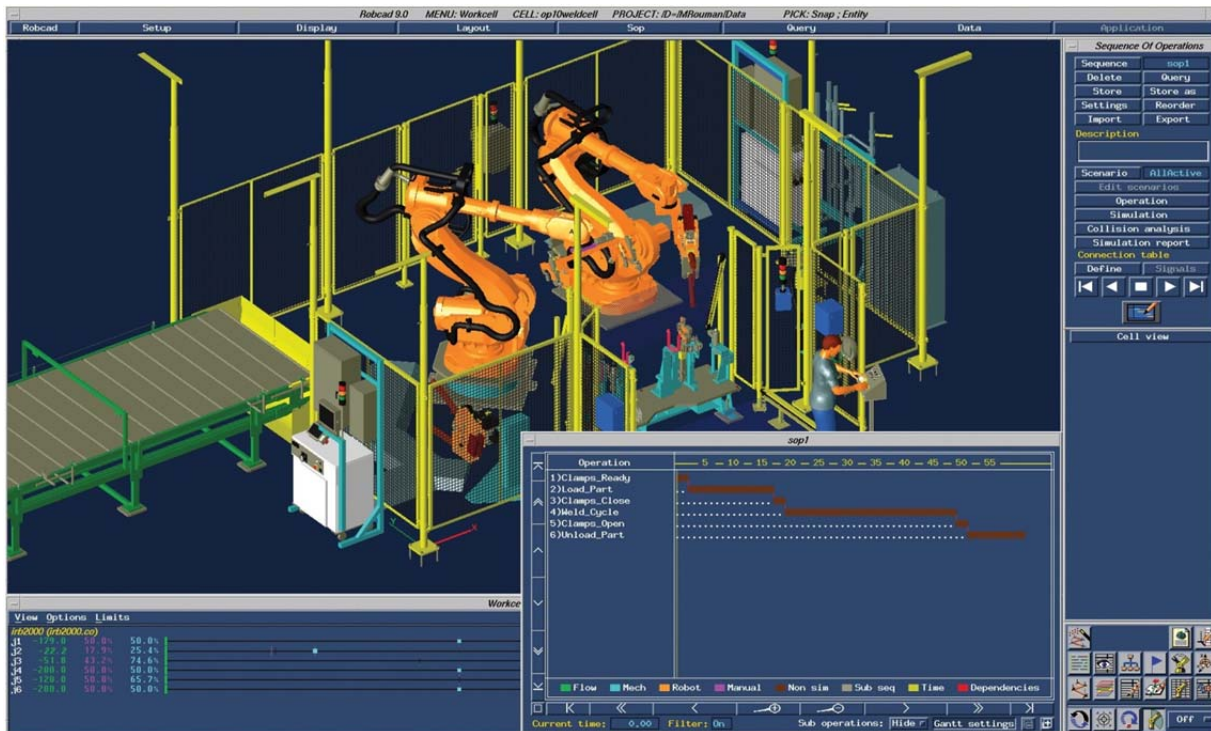


Imagen 11. Software de diseño Robcad.

2.4 Simulación PLCs.

En el capítulo introducción de la memoria de este TFM, para poder abordar el problema, hicimos una panorámica de los simuladores de PLCs. La mayoría son proporcionados por el fabricante de hardware y con una funcionalidad limitada, en algunos casos, como Schneider Electric, ni siquiera podíamos contar con un simulador virtual y tendremos que probar nuestro programa en una maqueta física del hardware final.

La realidad es que nosotros no queremos simular nuestro hardware, ni el programa en sí, sino cómo reacciona nuestro programa dentro de un sistema real. Si queremos simular esto, estamos obligados a crear una maqueta del sistema real contra la que probar nuestro programa PLC.

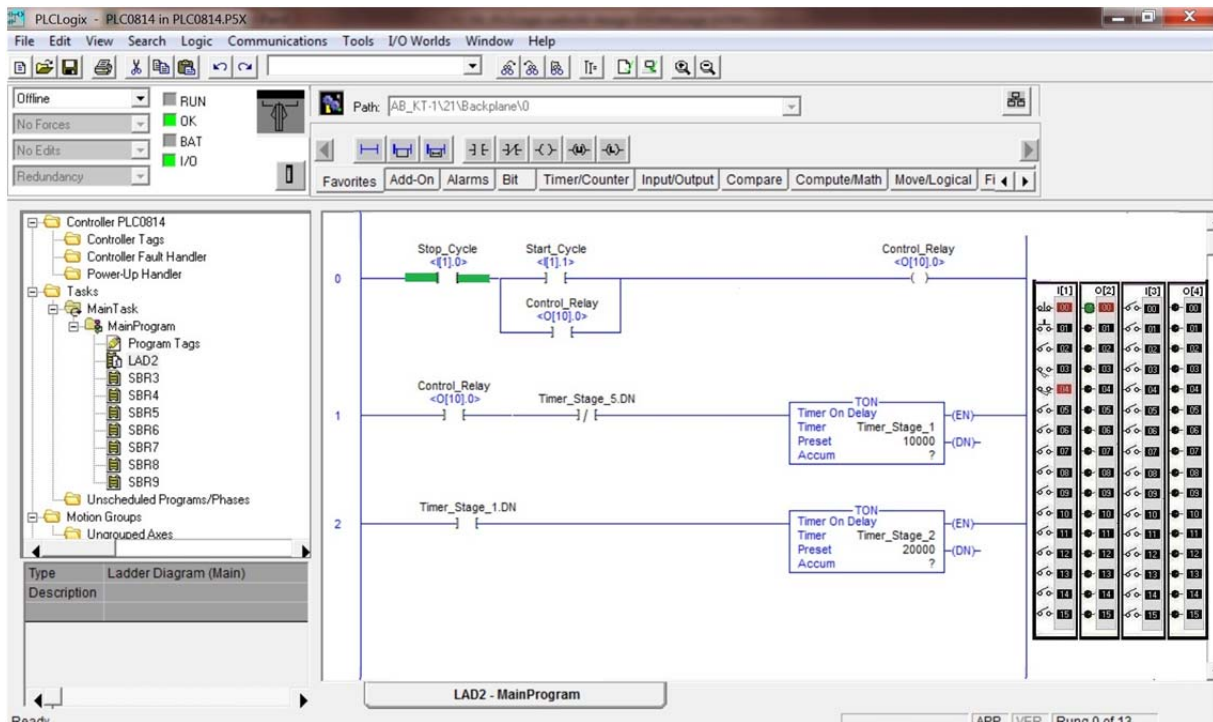


Imagen 12. Software de simulación RSLogix.

Si el sistema es sencillo, construirla mediante programación informática también lo será. Pero si el sistema es complejo y tiene muchas variables, como pasa con los sistemas que estamos analizando en este TFM, donde además hay interrelaciones con sistemas informáticos de nivel superior, el trabajo se complica y adquiere una mayor envergadura que construir el propio programa del PLC.

Algunos fabricantes de automóviles y similares, como Renault y Michelin, que además tienen arquitecturas de programación estandarizadas, disponen de simuladores propietarios para probar sus nuevos programas. Aunque sigue siendo algo muy costoso y poco empleado.

2.5 Simulación de sistemas.

Durante el máster que da origen a este TFM hemos estudiado y analizado software de simulación de sistemas, que mediante la creación de un modelo, matemático o lógico, representa el sistema y nos permite testarlo ante diferentes entradas.

Dos ejemplos de ello son Simulink y Arena Simulation.

En simulink disponemos Toolbox ya creados para modelar nuestro sistema utilizando componentes de cualquier tipo (eléctricos, electrónicos, mecánicos, químicos, etc.) y simular la respuesta de nuestro sistema ante entradas del mundo real. La parte compleja vuelve a ser modelar el sistema. Si el sistema es físico y no demasiado complejo podemos llegar a modelarlo lo suficientemente bien como para poder simularlo.

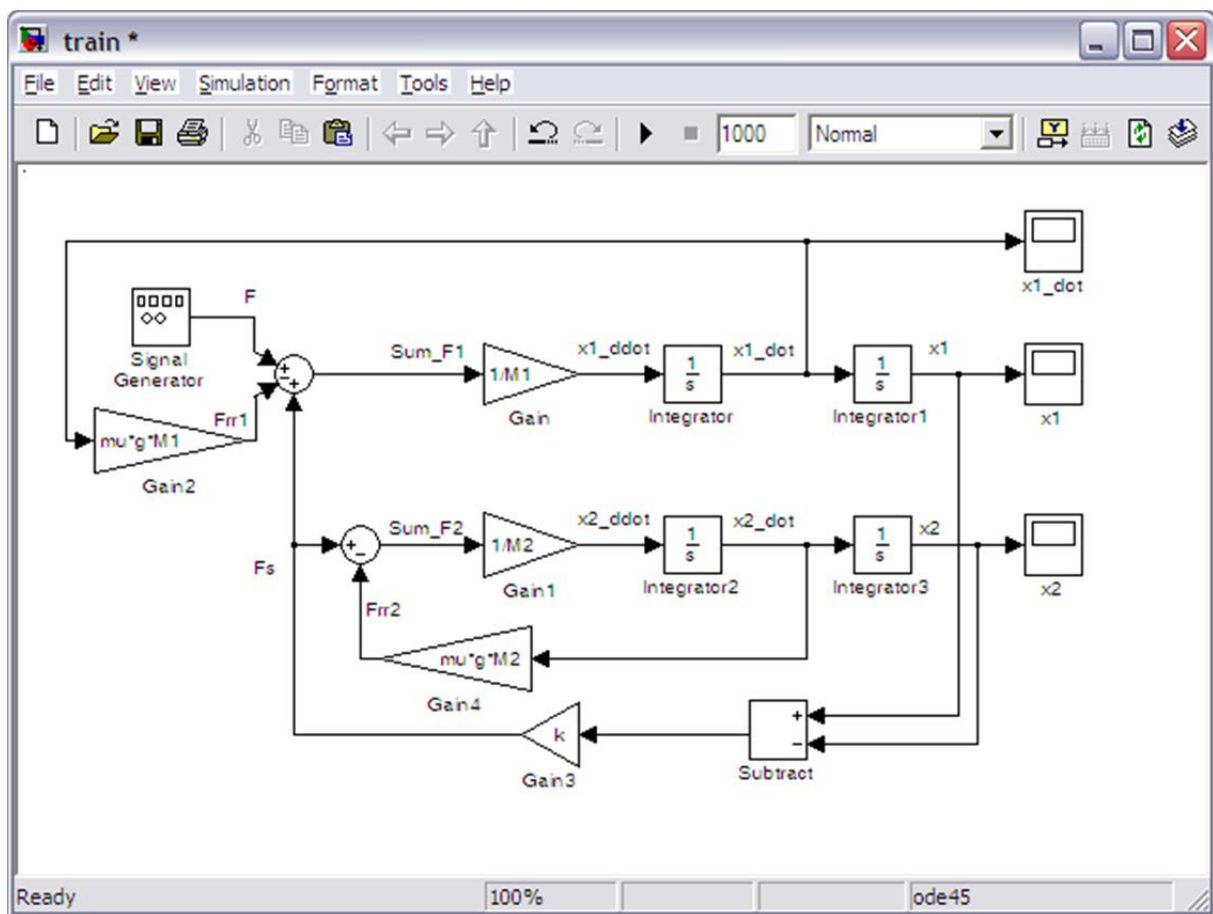


Imagen 13. Software de simulación Simulink.

Con Arena Simulation podremos también simular nuestros sistemas construyéndolos como elementos lógicos. Es muy útil para simular procesos de fabricación, procesos logísticos, sistemas donde existen colas de procesamiento, etc.

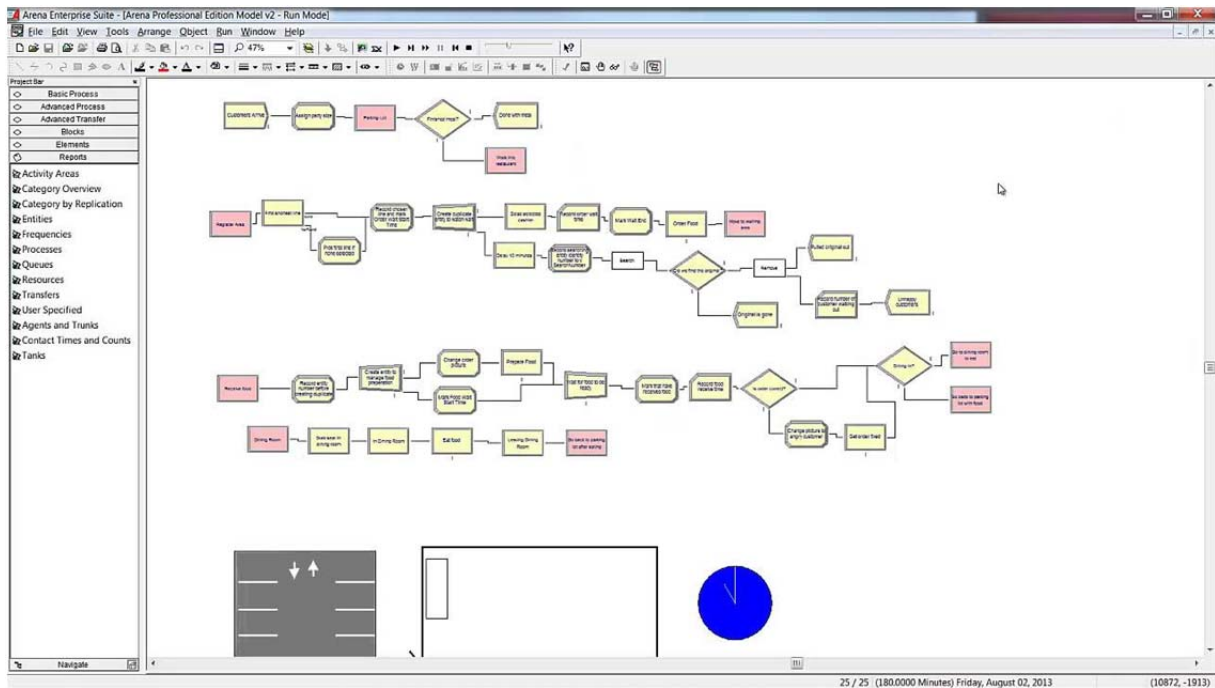


Imagen 14. Software de simulación Arena simulation.

Si se trata de un sistema digital complejo, como el que pretendemos simular con ayuda de nuestro simulador, con un orden de miles de entradas físicas digitales e interconexiones con varios sistemas, no podremos utilizar una herramienta única, sino que deberemos simularlo por partes, y para cada una de esas partes, utilizar el simulador adecuado.



3 ANALISIS Y DISEÑO DEL SISTEMA.

3.1 Análisis.

El simulador debe representar el comportamiento temporal de PSFV y también los posibles errores de comunicación, que se resumen en:

3.1.1 Requisitos no funcionales.

El simulador de PSFV tiene que cumplir los siguientes requisitos no funcionales:

- Facilidad de configuración.
- Rapidez de respuesta.
- Consumo de pocos recursos.
- Independencia de drivers de comunicación.

Facilidad de configuración.

El sistema ha de poder ser configurado por personal no experto en informática. La edición de los ficheros de configuración ha de ser sencilla. Para ello, se ha optado por almacenar la configuración y los ficheros de datos en formato EXCEL, y no utilizar una arquitectura de BB.DD SQL, que aunque es más potente, haría algo más complejo el proceso de configuración.

Existirán dos ficheros Excel:

- Fichero de configuración del simulador.
- Fichero con los datos de trabajo.

Si decidimos cambiar alguno de los datos, lo haremos directamente con la aplicación Excel y no desde la aplicación.

Deberemos reiniciar la aplicación para que coja en el arranque la configuración modificada.

Rapidez de respuesta.

El simulador nos servirá para probar programas de autómatas, donde el tiempo es un factor clave.

La comunicación se controla desde el autómata, así como los tiempos de respuesta. Por eso nuestro simulador ha de contestar pronto, para intentar simular también esta parte.



Consumo de pocos recursos.

El simulador ha de poder ejecutarse en un PC con pocos recursos, las únicas necesidades son tener el software EXCEL instalado y un servidor OPC para comunicar con el autómeta.

Lo normal es tener el simulador ejecutándose en el mismo PC con el que estamos probando y modificando el programa de autómeta. Nuestro simulador no debe ralentizar el funcionamiento de este.

Independencia de drivers de comunicación.

Con la independencia de drivers de comunicación conseguimos una mayor portabilidad del software.

La mejor forma de lograr esto es con el uso de servidores OPC. Nuestro simulador debe ser cliente OPC. De esta manera, si queremos cambiar el protocolo de comunicación sólo deberemos configurar el servidor OPC y no será necesario modificar la aplicación.

3.1.2 Requisitos funcionales.

En el punto de objetivos del sistema ya definimos los objetivos de nuestra aplicación:

- La aplicación ha de poder se iniciada y parada. Al hacerlo se bloquearán o liberarán de forma automática los recursos de comunicación del sistema.
- La aplicación debe funcionar de forma autónoma sin intervención del usuario, a no ser que este, quiera cambiar algo.
- La aplicación debe dar soporte para comunicar hasta con seis autómetas distintos, o un mismo autómeta con seis puntos de diálogo distintos.
- La tabla de comunicación con el PLC puede ser de hasta 100 palabras.
- El fichero de coches que maneja la aplicación para enviar al autómeta puede contener varias hojas, con los diferentes trayectos que realizar el coche, uno por ejemplo con el orden en soldadura, pintura, etc.
- El usuario ha de poder provocar desde la aplicación los distintos errores presentes en este tipo de comunicación (falta de respuesta, respuesta incorrecta, etc.)
- La aplicación debe tener una interface visible que transmita al usuario en tiempo real el funcionamiento del sistema.

Para lograr estos objetivos el simulador de PSFV tendrá los siguientes usuarios, aunque no tiene por qué haber una diferenciación clara de roles.

- Administrador del sistema.
- Usuario del simulador.

Cada uno de ellos tendrá diferentes requisitos funcionales que derivarán en casos de uso distintos. [8][9][10]

Para el usuario administrador tendremos los siguientes casos de uso, que detallaremos en el siguiente punto:

- Configurar el simulador.
- Arrancar el simulador.
- Parar el simulador.

Para el usuario utilizador del simulador hay un único caso de uso, que es el propio uso del simulador:

- Dialogar con el simulador. Este caso de uso englobará el 100% del uso del sistema, ya que este en la práctica actúa como un servicio.

3.1.3 Casos de uso del sistema.

Estos requisitos dan lugar a los siguientes casos de uso:

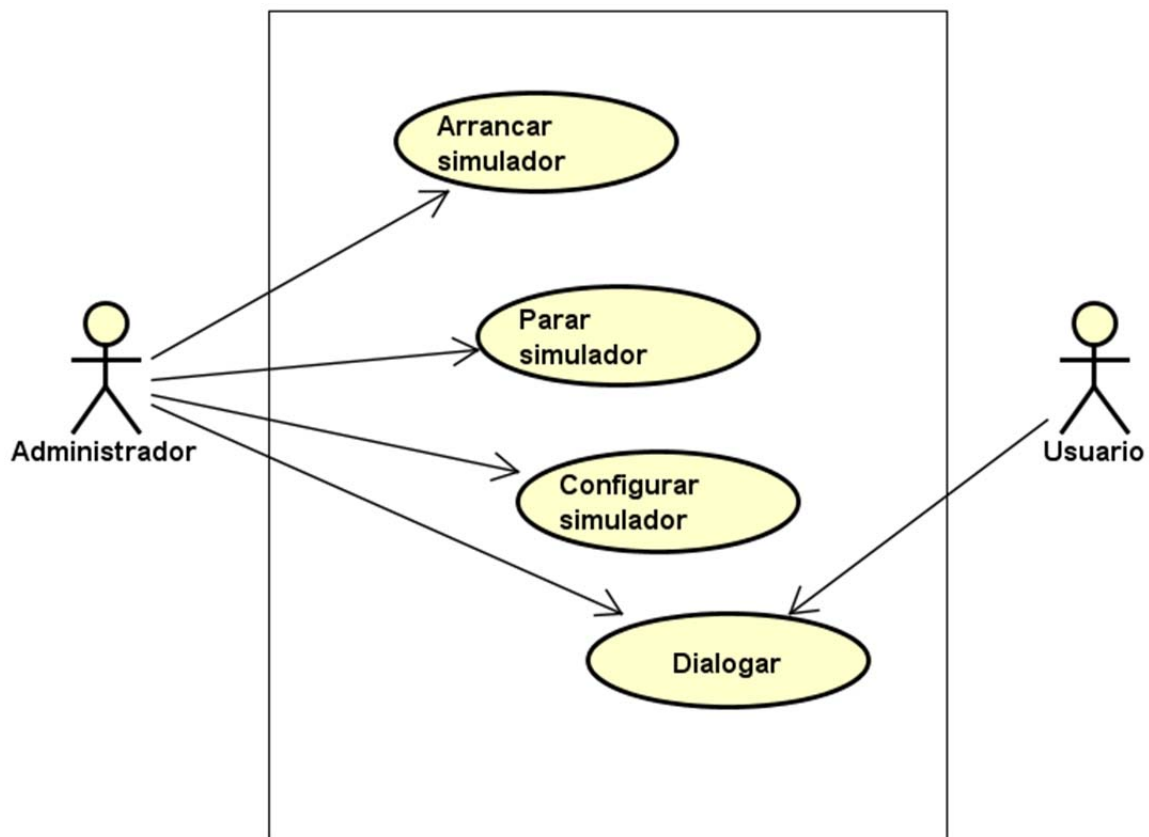


Imagen 15. Casos de uso del sistema.



3.1.4 Casos de uso del administrador del sistema.

Configurar el simulador.

Autores	Javier Laíz	
Descripción	El administrador podrá configurar el sistema para su posterior uso	
Precondición	El sistema existe y no está arrancado	
Secuencia Normal	Paso	Acción
	1	El administrador configura las comunicaciones del simulador
	2	El administrador configura cada uno de los puntos de diálogo
Postcondición	Sistema configurado y listo para su uso	
Excepciones	Paso	Acción
	1	No existe el fichero de configuración. Deberá generar uno nuevo partiendo de una plantilla
Rendimiento	Paso	Cota de tiempo
	1	2 minutos
	2	10 minutos
Frecuencia esperada	1 vez a la semana	
Importancia	Vital	
Urgencia	Inmediatamente	
Comentarios	Esta acción no se realiza desde la aplicación software	



Arrancar el simulador.

Autores	Javier Laíz	
Descripción	El administrador podrá iniciar el simulador	
Precondición	El sistema existe, está configurado y no está arrancado	
Secuencia Normal	Paso	Acción
	1	El administrador inicia el simulador
	2	El simulador lee el fichero de configuración
	3	El simulador abre el fichero de coches para realizar la simulación
	4	El simulador abre las comunicaciones con el PLC
	5	El simulador queda a la espera de la petición del usuario para comunicar
Postcondición	Simulador iniciado y listo para su uso	
Excepciones	Paso	Acción
	2/3	No existen los ficheros. No se puede abrir el simulador.
	4	No se consigue iniciar la comunicación. Se informa al usuario
Rendimiento	Paso	Cota de tiempo
	1 al 4	60 segundos. Debe haber feedback del proceso para el usuario.
Frecuencia esperada	1 vez cada 4 horas	
Importancia	Vital	
Urgencia	Inmediatamente	
Comentarios		



Parar el simulador.

Autores	Javier Laíz	
Descripción	El administrador podrá parar el simulador	
Precondición	El sistema existe, está configurado y arrancado	
Secuencia	Paso	Acción
Normal	1	El administrador para el simulador
	2	El simulador cierra el fichero de configuración
	3	El simulador cierra el fichero de coches.
	4	El simulador detiene las comunicaciones con el PLC
Postcondición	Simulador detenido, ficheros cerrados y comunicaciones liberadas	
Excepciones	Paso	Acción
Rendimiento	Paso	Cota de tiempo
	1 al 4	60 segundos. Debe haber feedback del proceso para el usuario.
Frecuencia esperada	1 vez cada 4 horas	
Importancia	Vital	
Urgencia	Inmediatamente	
Comentarios		



3.1.5 Casos de uso del usuario del sistema.

El usuario del sistema sólo tiene un caso de uso, que es la propia utilización de la herramienta.

Dialogar con el simulador

Autores	Javier Laíz	
Descripción	El usuario dialoga con el simulador	
Precondición	El sistema existe, está configurado y arrancado	
Secuencia Normal	Paso	Acción
	1	El usuario inicia el diálogo con el simulador
	2	El simulador analiza la petición del usuario
	3	El simulador responde al usuario en función de su configuración
Postcondición	Se produce un diálogo entre el usuario y el simulador	
Excepciones	Paso	Acción
Rendimiento	Paso	Cota de tiempo
	1 al 3	5 segundos.
Frecuencia esperada	1 vez cada minuto	
Importancia	Vital	
Urgencia	Inmediatamente	
Comentarios		

3.1.6 Diagramas de actividades.

Detallamos para cada caso de uso su diagrama de actividades, donde vemos de forma secuencial la ejecución de actividades, las transiciones y los resultados esperados.

Arrancar el simulador.

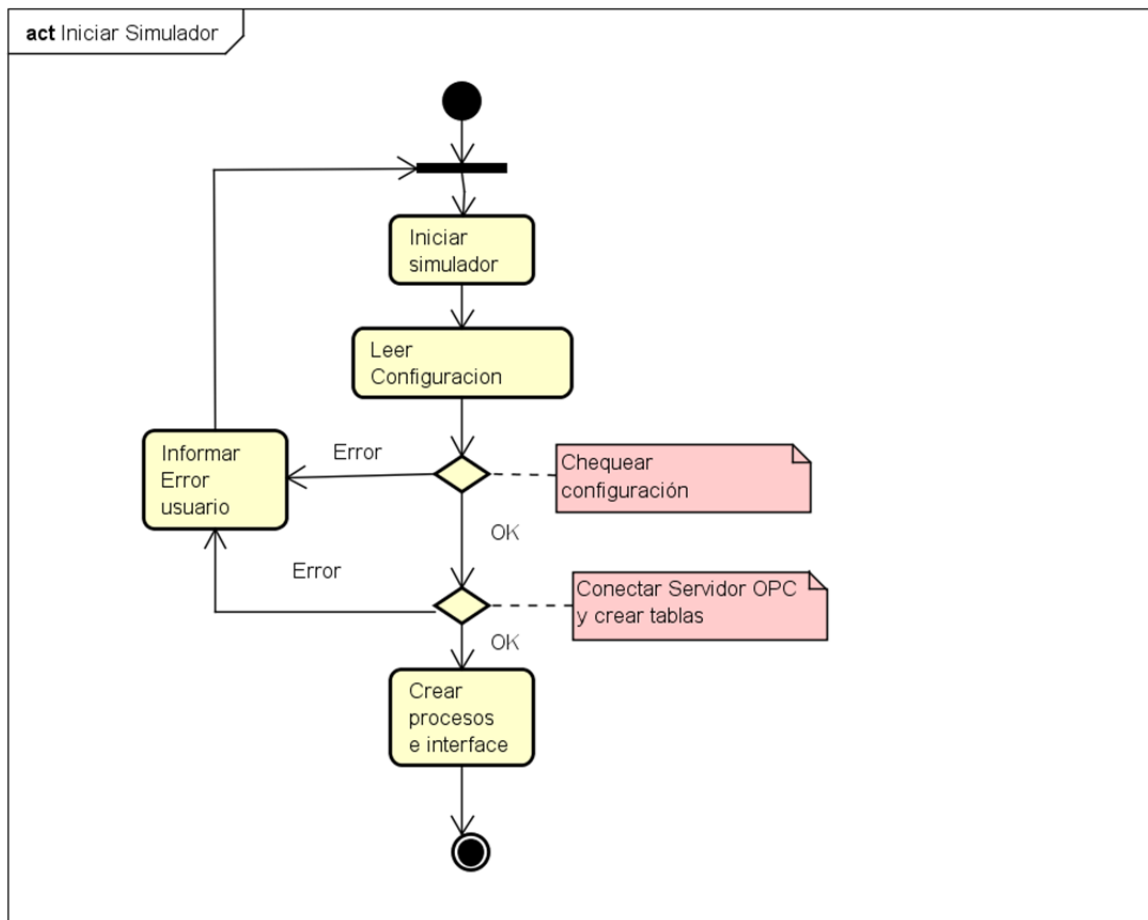


Imagen 16. Diagrama de actividad arrancar el simulador

Parar el simulador.

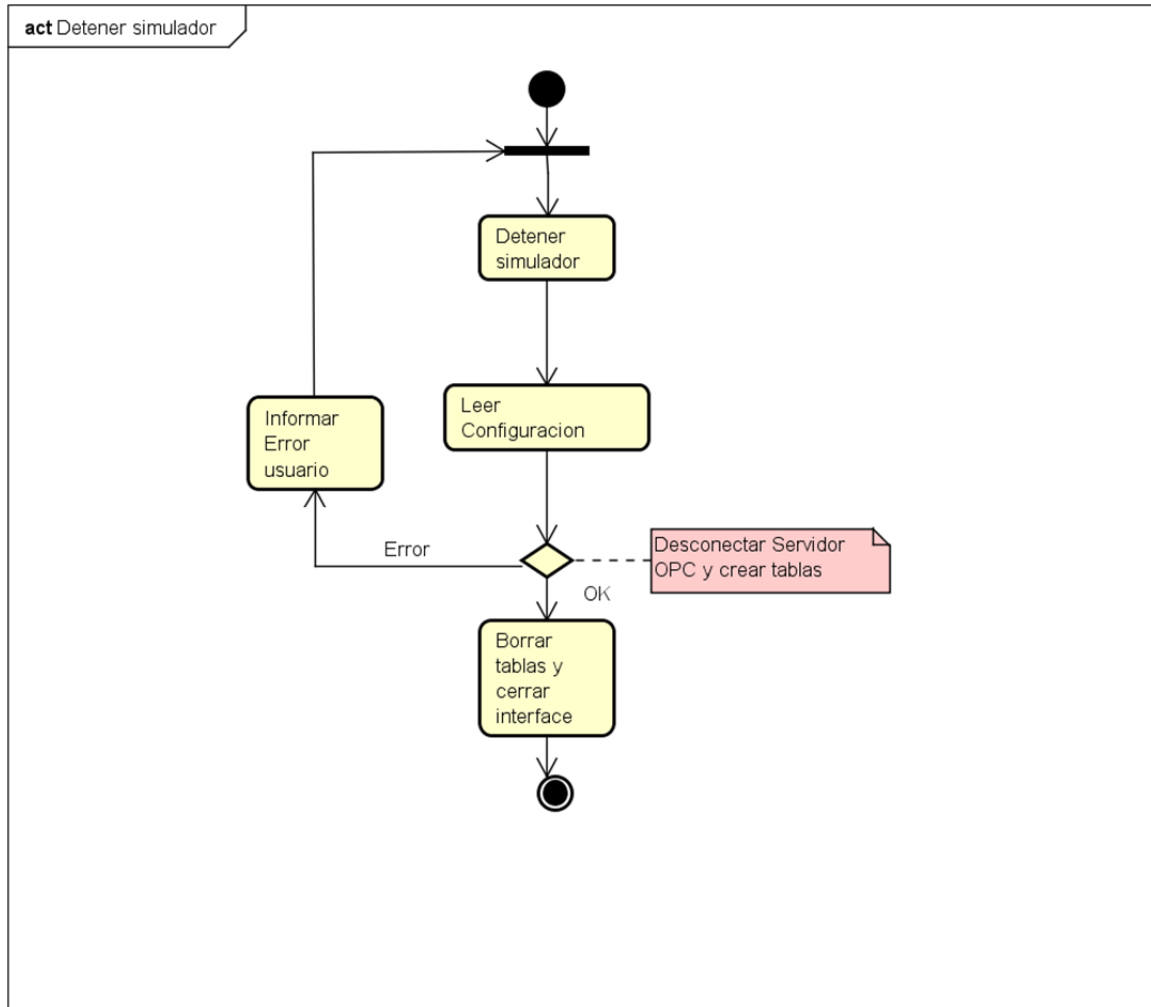


Imagen 17. Diagrama de actividad detener el simulador

Configurar el simulador

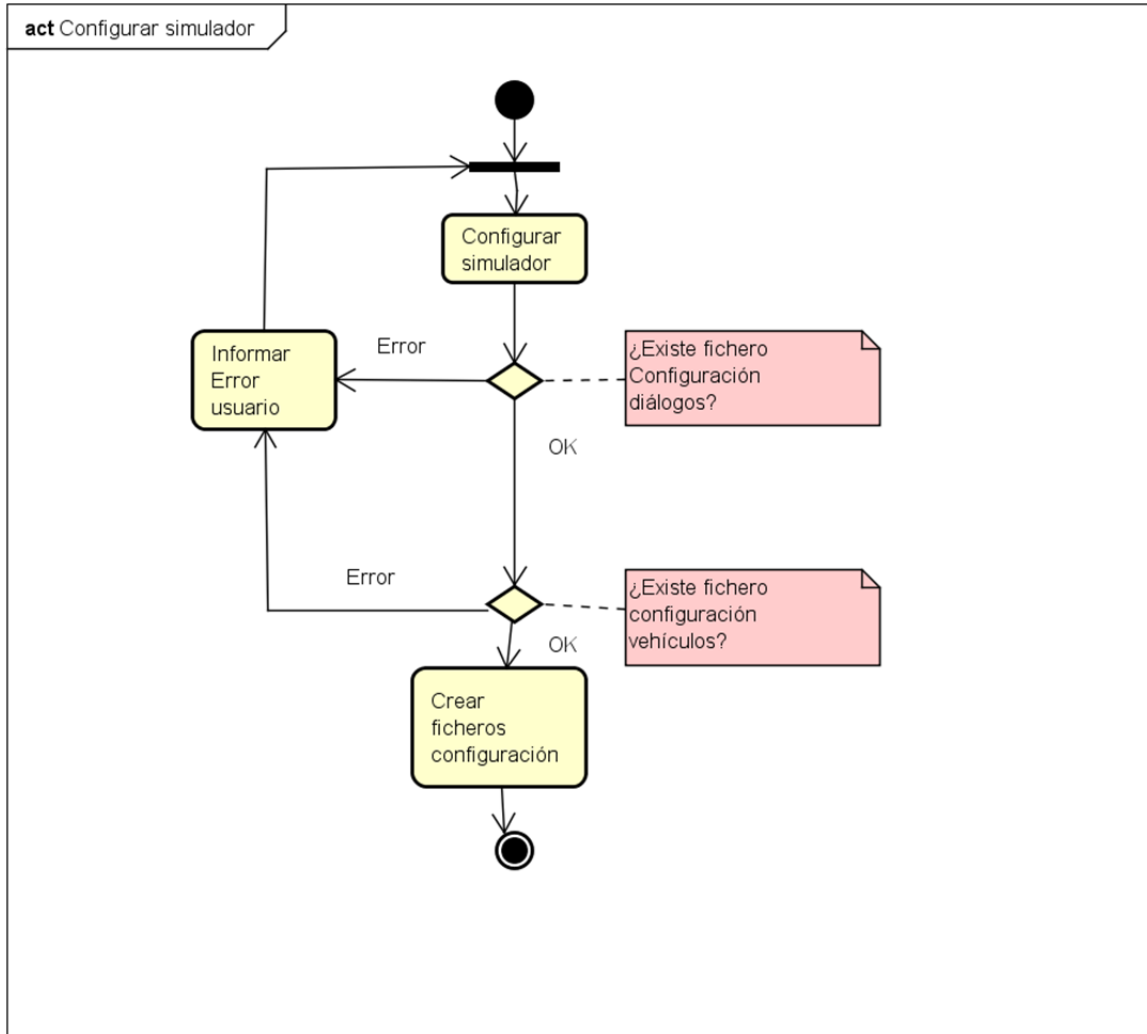


Imagen 18. Diagrama de actividad configurar el simulador

Dialogar.

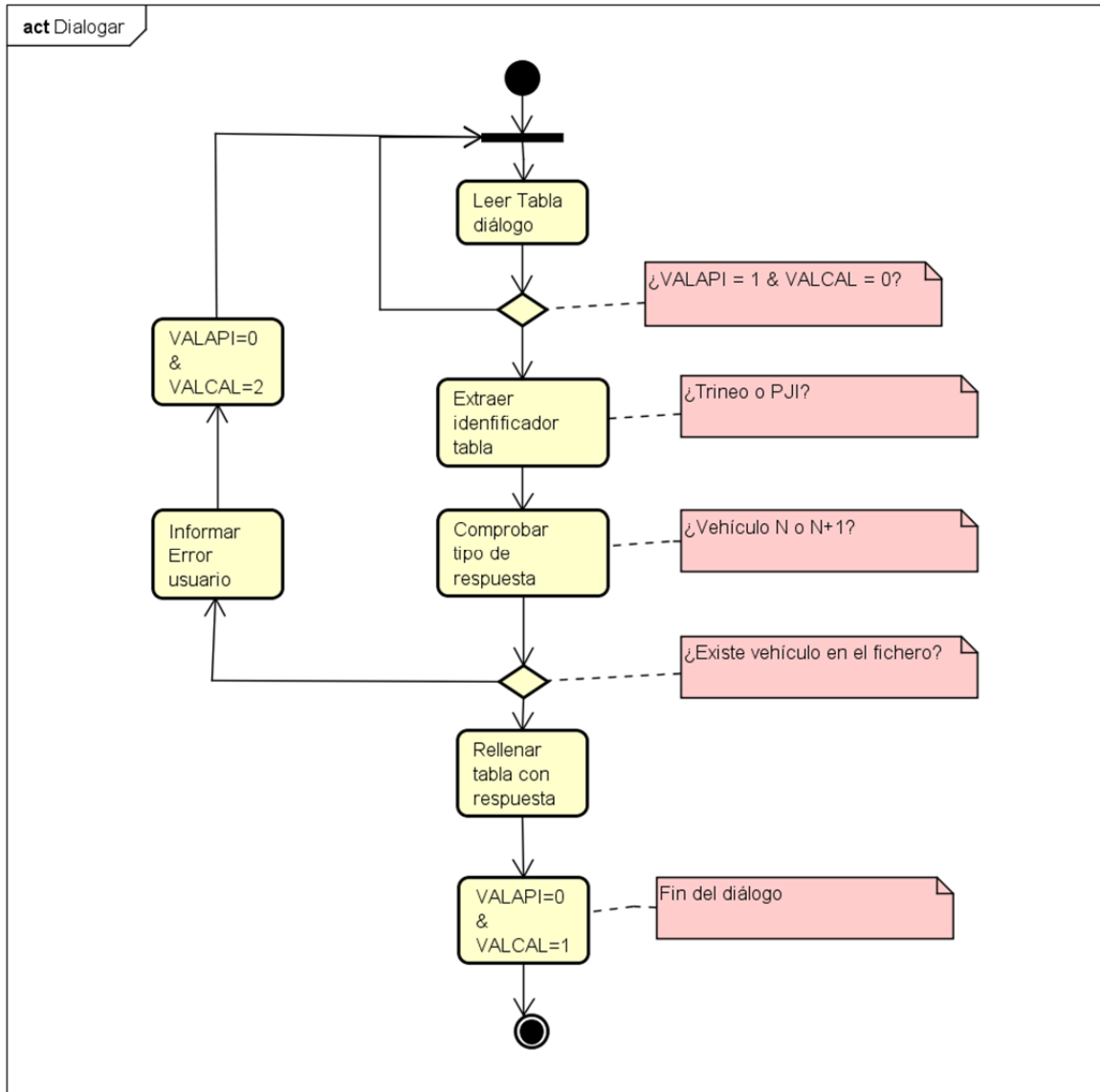


Imagen 19. Diagrama de actividad dialogar con el simulador

3.1.7 Diagrama de clases.

El sistema necesitará tres clases para poder realizar sus tareas:

- Tabla diálogo.
- Datos del vehículo, que se apoyará en otra subclase que guarda los datos particulares de cada vehículo, y pueden ser de tipo variable.

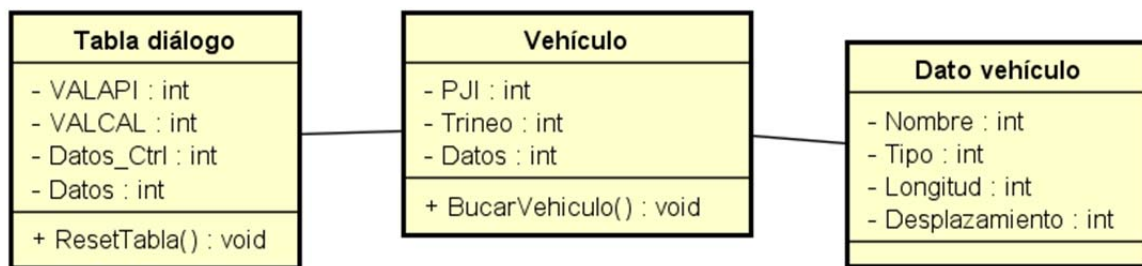


Imagen 20. Diagrama de clases del sistema

3.2 Diseño.

Una vez analizado nuestro sistema ya sabemos qué ha de hacer, que usuarios tendrá, cómo funcionará y cuál será su rendimiento esperado. Queda diseñarlo para poder construirlo.

Debemos salir de la etapa de diseño con los conceptos claros para comenzar a fabricar el software, como si se tratará de otro proceso más de ingeniería.

Debemos diseñar:

- La estructura y parámetros del fichero de configuración de nuestro simulador.
- El fichero de datos de vehículos.
- La estructura de la aplicación, que programaremos después en C++.
- La arquitectura del sistema a nivel de comunicaciones.
- La interface de usuario.

3.2.1 Diseño del fichero de configuración.

Para realizar nuestras tareas hemos de ser capaz de configurar el funcionamiento de nuestro simulador de forma sencilla, pero también amplia, para que se adapte a todos los tipos de comunicación que nos podemos encontrar.

La configuración tiene dos partes, una primera donde parametrizamos la parte de comunicaciones y una segunda donde definimos como será el intercambio con el PLC.



Los parámetros que nos podemos encontrar en la parte de comunicaciones, y que harán que la aplicación sea flexible son:

- **Nombre del servidor OPC.** Texto que nos permitirá conectarnos con uno de los servidores OPC registrados en el equipo y a través del cual accederemos a la memoria del autómatas. En este campo de texto plano debemos introducir el ProgID, que es la cadena de caracteres con la que el servidor OPC se ha registrado en el sistema. Si tenemos dudas de cuál es el ProgID podemos consultarlo en la documentación del propio servidor y con la ayuda de un cliente OPC. [11]
- **Nombre del alias OPC.** El servidor OPC nos sirve para comunicar con diferentes equipos de forma simultánea, al definir un alias dentro del servidor, definimos cual será nuestro equipo, configurando por ejemplo su dirección IP y le damos un nombre a esta conexión (alias OPC). Si en un futuro decidimos cambiar la dirección IP del PLC, o directamente comunicar con otro protocolo que no sea Ethernet, solo deberemos configurar de nuevo el alias, pero para nuestro simulador será transparente, porque el solo conoce el alias OPC, y utiliza el servidor OPC para abstraerse de las comunicaciones.
- **Punto de paso.** Cadena de caracteres que definirá que hoja del fichero Excel de vehículos utilizamos para obtener los datos que nos interesan. Es el equivalente al puntero a una tabla de una base de datos donde tenemos los datos que queremos consultar. Si decidimos que no nos interesa los datos de esa hoja, y queremos otro punto de paso, bastará con cambiarlo aquí y reiniciar el simulador.
- **Tipo de diálogo.** Número que define el tipo de diálogo a utilizar. Puede tomar dos valores, 1 o 2. Si vale 1 cuando preguntemos por el coche A, devolveremos los datos del coche A. Si vale 2, cuando preguntemos por el coche A (qué ha sido el último que hemos fabricado) devolveremos los datos del vehículo que le sigue. La comunicación de tipo 2, llamada de vehículo N+1, porque dando los datos del vehículo N devolvemos los datos del vehículo N+1 se utiliza en aquellos procesos de fabricación donde no hay posibilidad de que los coches varíen su orden, y basta con conocer el último fabricado para obtener la información del siguiente a fabricar.
- **Tipo de identificador del diálogo.** Número que define el identificador a utilizar en el diálogo. Puede tomar dos valores posibles, 1 o 2. Si vale 1 dialogamos utilizando el PJI, que es un número de 7 dígitos que define de forma única el vehículo, como su DNI. Si este campo vale 2, utilizamos como identificador del diálogo el número de trineo sobre el que va montado el coche. Cuando preguntamos a PSFV por un vehículo, le preguntamos ¿qué vehículo va sobre el trineo 300?
- **Dirección de inicio de la tabla.** Número que define la primera dirección de la tabla en la memoria del autómatas.
- **Tamaño de la tabla.** Número que define el tamaño de la tabla de comunicación en la memoria del PLC. Su valor máximo es de 100 palabras de memoria del autómatas. [12]



Una vez terminado de diseñar la parte de comunicaciones, debemos diseñar la tabla de diálogo con el PLC.

Tendremos tantas palabras a documentar como tamaño de la tabla hayamos definido en el parámetro tamaño de la tabla.

Para cada palabra podremos documentar:

- **Dirección.** Valor fijo para ayudarnos a rellenar la tabla. No modificar.
- **Contenido.** Nombre del campo que vamos a utilizar. Existirán campos con un valor predefinido como VALAPI, VALCAL, PJI que llamaremos campos de control y otros que podrán tener un valor variable y que serán el índice para buscar en la tabla de vehículos.
- **Tipo.** Formato del campo. Dos valores posibles: ASCII o DECIMAL.
- **Longitud.** Longitud del campo. Tiene mayor sentido cuando definimos que el campo es ASCII, porque con este parámetro decimos cuantos caracteres contendrá.
- **Desplazamiento.** Si utilizamos el formato ASCII y estamos construyendo un campo compuesto por varias palabras, con este campo definimos la posición relativa de esta palabra con respecto al resto.

3.2.2 Diseño de datos de los vehículos.

El fichero de vehículos es el otro conjunto de datos que utilizaremos en nuestro simulador.

Tendrá tantas pestañas como puntos de paso hayamos definido. Aunque el contenido de todas ellas puede ser el mismo.

Al menos contendrá las siguientes columnas, que almacenan los datos básicos del vehículo.

- **PJI.** Identificador numérico principal del vehículo. Lo utilizaremos para buscarlo dentro del fichero y devolver los datos al PLC.
- **Trineo.** Identificador número secundario del vehículo. Lo utilizaremos cuando el PLC no nos pregunta por un PJI, sino por el vehículo que va montado encima de un determinado trineo.
- **Modelo.** Modelo del vehículo. Campo de texto.

Puede haber más campos que definiremos a nuestro antojo, bastará con crear una nueva columna Excel y darle como nombre de la columna el identificador del campo.

Por ejemplo, si en la tabla de diálogo queremos proporcionar al PLC el par de apriete del tornillo V20, configuraremos en la tabla de diálogo una palabra con el siguiente contenido:

- Dirección. La que hayamos elegido.
- Contenido. V20_PAR.
- Tipo. DECIMAL.
- Long. VACIO



- Desplazamiento. VACIO

Y en el fichero Excel de datos de vehículos añadiremos una nueva columna llamada V20_PAR, que después rellenaremos fila a fila con los datos que nos interesan para cada vehículo, o en blanco si no procede.

3.2.3 Diseño de la interface de usuario.

El simulador no es una herramienta del tipo tradicional donde el usuario deba estar interactuando de forma constante con ella. Más bien se trata de un proceso de comunicaciones para traspasar datos a la memoria de nuestro PLC desde unos ficheros de datos.

Sin embargo no podemos obviar ciertos requerimientos de diseño en la interface de usuario:

- **Feedback.** El usuario debe saber en todo momento que la aplicación está funcionando, que lo hace correctamente y que responde a tiempo al PLC, también que es reactiva. Esto se conseguirá mostrando en tiempo real el contenido de la tabla de diálogo (tantas tablas como hayamos definido) y en la barra de la aplicación la hora del sistema, que la propia aplicación, deberá ir actualizando de forma periódica y que nos servirá para saber que todo funciona bien y rápido.
- **Control del usuario.** El usuario ha de poder cerrar la aplicación cuando necesite, sin esperas.
- **Remarcar lo importante.** En la comunicación hay dos palabras más importantes que el resto, VALAPI y VALCAL, deberán quedar remarcadas en a interface.
- **Ayuda a la visualización.** Hay campos que configuramos como ASCII, pero que en la memoria del PLC se ven en decimal o hexadecimal. Por ejemplo la cadena ASCII B9 (que corresponde con el modelo del Megane 5 puertas) en decimal se verá como 16952 (16#4239). La aplicación debe incorporar algún mecanismo para cambiar la visualización y ayudar al programador a ver los datos ASCII en su representación.
- **Utilizar elementos conocidos.** La representación de la tabla de diálogo debe ser lo más similar a las representaciones utilizadas en este tipo de procesos, de esa forma, será más fácil aprender a utilizar la aplicación.
- **Tratamiento de errores.** La aplicación informará al usuario de los posibles errores, el más frecuente, la pérdida de comunicación con el PLC.

Boceto de la interface.

En esta fase de diseño, haremos un boceto de la interface que nos ayudará a construirla en la fase de implementación.

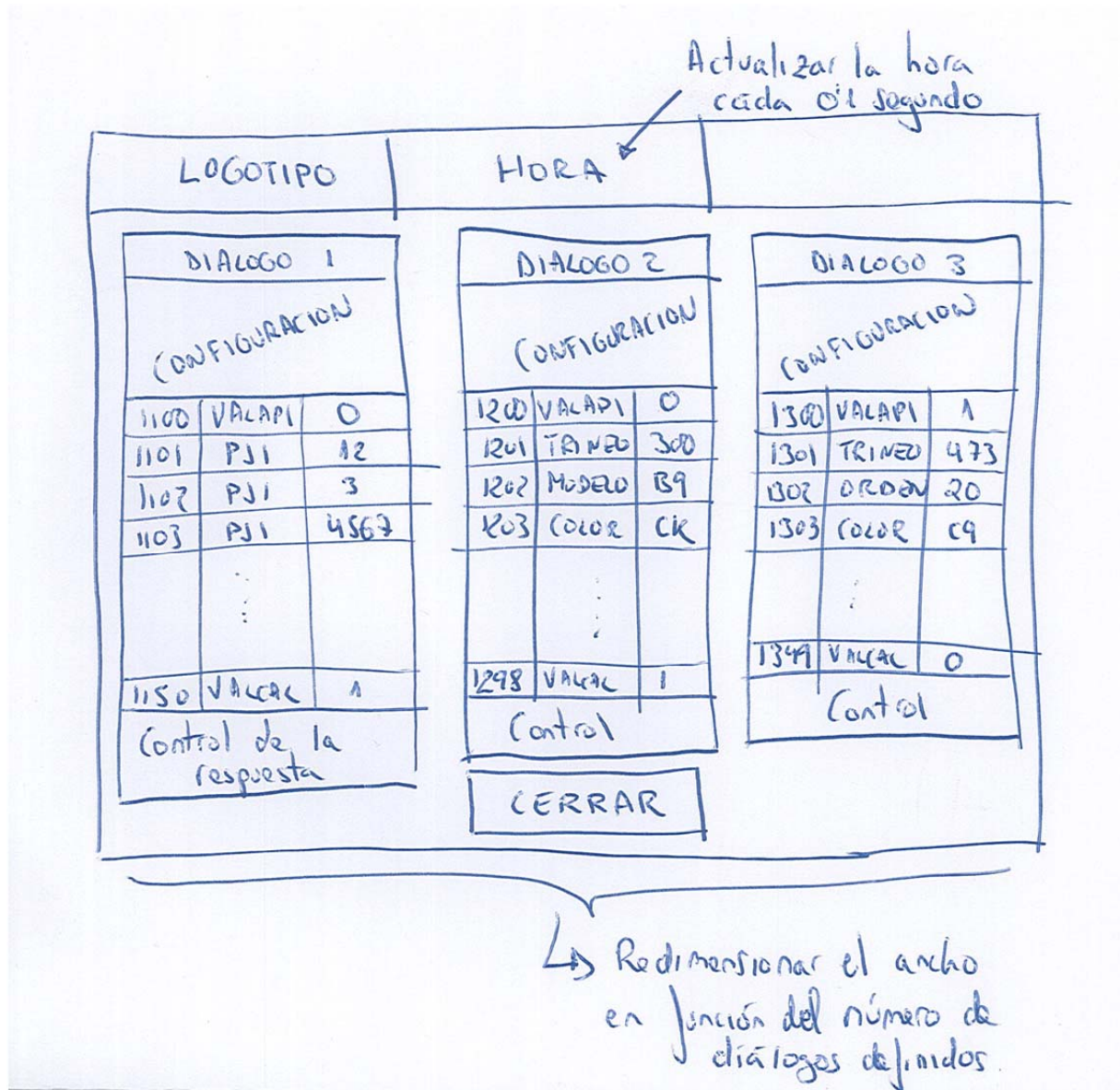


Imagen 21. Boceto en papel de la interface de usuario.



4 RESULTADOS.

El objetivo del simulador de PSFV es la prueba fuera de línea de los diálogos automática con PSFV, la prueba del programa y el tratamiento de datos recibidos y poder corregir errores y realizar las modificaciones necesarias antes de la puesta en marcha en el cliente.

El simulador debe dar soporte para comunicar hasta con 6 puntos de diálogo distintos, donde puede variar la tabla de diálogo, su contenido, la forma de dialogar (por PJI o trineo) el dato a obtener (coche actual o N+1) y el punto de paso del vehículo (EMON, FF, PINTURA, etc.)

La aplicación creada debe ser sencilla de configurar y robusta. El fichero de configuración es un fichero Excel que podremos editar rápidamente. Los datos de los vehículos también se almacenan en un fichero Excel.

Para su funcionamiento no necesita instalación, tan sólo registrar una DLL en Windows que nos dará acceso a los servidores OPC del instalados.

El manual tiene tres partes:

- Manual de instalación. Explica como instalar la aplicación y las carpetas y ficheros que contiene.
- Manual de configuración. Explica como configurar el servidor y manejar los ficheros de datos para realizar las pruebas que necesitemos.
- Manual de usuario. Explica el uso de la propia aplicación.

4.1 Manual de instalación.

La carpeta de la aplicación contiene los siguientes directorios:

- **Configuración.** Contiene el fichero Conf_Diálogos.xls que almacena la configuración de los diálogos con los autómatas (una hoja Excel por cada diálogo) y el fichero vehículos.xls que guarda los datos de los coches que puede enviar el simulador.
- **Ejecutable.** Contiene la aplicación de simulación (Simulador_PSFV.exe) y librerías que necesita para su ejecución.
- **Gráficos.** Almacena ficheros gráficos como el logo de la aplicación.
- **Incidencias.** Guarda el registro de funcionamiento y errores del programa.

Los pasos para instalar la aplicación son:

- Iniciar la sesión en Windows como ADMINISTRADOR del equipo.



- Copiar la carpeta en nuestro disco duro. Debemos asegurarnos de que contiene todas las carpetas vistas anteriormente.
- Desde la ventana de comandos de Windows, y situados en la carpeta EJECUTABLE, ejecutar el siguiente comando: “regsvr32 SAOFSGRPS.DLL”. Esto registrará en Windows una DLL necesaria para iniciar la aplicación. En caso de no hacerlo, al iniciar la aplicación nos saldrá un mensaje de error “CLASE NO REGISTRADA” y no podremos trabajar.

4.2 Manual de configuración.

Antes de empezar a configurar el simulador hemos de asegurarnos de haber instalado correctamente el simulador (Ver punto anterior).

La configuración tiene dos fases:

- Definición de los diálogos a utilizar y la estructura de la tabla.
- Creación de los datos de los vehículos a transmitir al PLC.

4.2.1 Definición de los diálogos y sus tablas.

Creamos las tablas de diálogo en el fichero Conf_dialogos.xls que encontraremos en la carpeta Configuración.

Cada diálogo a utilizar, hasta un máximo de seis, irá en una hoja Excel independiente. El nombre de estas hojas debe seguir una regla, para poder ser entendido por la aplicación: PSFV_N donde N es el número del diálogo. Si tenemos 3 diálogos activos tendremos 3 hojas Excel en nuestro fichero:

- PSFV_1
- PSFV_2
- PSFV_3

No se pueden dejar huecos en la numeración ni tener más de seis diálogos, ya que esta es una limitación de la aplicación.

Todas las hojas Excel tendrán la misma estructura. Una primera parte donde configuramos las características básicas del diálogo, y la propia tabla del diálogo.

La aplicación va a buscar la configuración del diálogo a unas celdas Excel concretas, así que tendremos que respetar el orden y no cambiar ni insertar celdas.

Sí que podremos hacer cambios sobre la tabla de diálogo, que es variable hasta un máximo de 100 palabras.

Estructura de la hoja PSFV_N.

La hoja de configuración tiene dos partes diferenciadas:

- Configuración del funcionamiento del diálogo y el tipo. (En la imagen el cuadrado de color rojo)
- Configuración de la tabla de diálogo palabra por palabra. (En la imagen el cuadrado de color azul)

Nombre del servidor OPC	Schneider-Aut.OFS
Alias OPC	PSF
Punto de paso	EMOH
Tipo dialogo	2
Tipo identificador dialogo	1
Populacion	BFB+HFE
Dirección inicio tabla	1100
Tamaño de la tabla	100

1->devolvemos los datos del vehiculo por el que pregunta
1-> dialogo por PJI, 2-> Dialogo por Trineo
2->devolvemos el N-1

Dirección	Contenido	TIPO	LONG	DESPLAZAMIENTO
1100	VALAPI	DECIMAL		
1101	TRINEO	DECIMAL		
1102				
1103				
1104				
1105	PJI_P	DECIMAL	2	0
1106	PJI_J	DECIMAL	1	2
1107	PJI_I	DECIMAL	4	3
1108				
1109	PJI_P_N1	DECIMAL	2	0
1110	PJI_J_N1	DECIMAL	1	2
1111	PJI_I_N1	DECIMAL	4	3
1112	ENCADENAMIENTO	DECIMAL		
1113	OBJ_536	ASCII	2	2
1114	MODELO	ASCII	2	0
1115	MOFF_1	DECIMAL		
1116	MOFF_2	DECIMAL		
1117				
1118				
1119				
1120				
1121	PPUM_74	ASCII	2	0
1122	PPUM_74	ASCII	2	2
1123	PPUM_74	ASCII	2	4
1124	PPUM_74	ASCII	2	6
1125	PPUM_74	ASCII	2	8
1126	COEF_74	ASCII	2	0
1127	COEF_74	ASCII	2	2
1128	PPUM_222	ASCII	2	0
1129	PPUM_222	ASCII	2	2
1130	PPUM_222	ASCII	2	4

Imagen 22. Plantilla Excel de configuración del diálogo.

4.2.2 Parámetros generales del diálogo.

Tenemos ocho campos posibles para configurar nuestro diálogo:

Nombre del servidor OPC	Schneider-Aut.OFS
Alias OPC	PSF
Punto de paso	EMON
Tipo dialogo	2
Tipo identificador dialogo	1
Populacion	BFB+HFE
Dirección inicio tabla	1100
Tamaño de la tabla	100

Tabla 5. Parámetros generales del diálogo.

Nombre del servidor OPC.

Este campo es de texto libre., aquí anotaremos el servidor OPC que utilizaremos para conectar con nuestro autómeta. El nombre que figurará en este campo es el ProgID con el que el servidor OPC se habrá registrado en nuestro sistema.

Alias OPC.

Nombre del alias que habremos creado en nuestro servidor OPC como punto de conexión con nuestro PLC. Gracias a la utilización del servidor OPC y sus alias conseguimos independencia del hardware.

Si decidimos evolucionar el tipo de conexión física entre el ordenador en el que se ejecuta el simulador y nuestro autómeta, deberemos cambiarlo dentro del servidor OPC, por ejemplo, cambiando la configuración del alias de RS232 a Ethernet.

Los cambios serán transparentes para la aplicaciones que como cliente utilizan el servidor OPC (nuestro simulador).

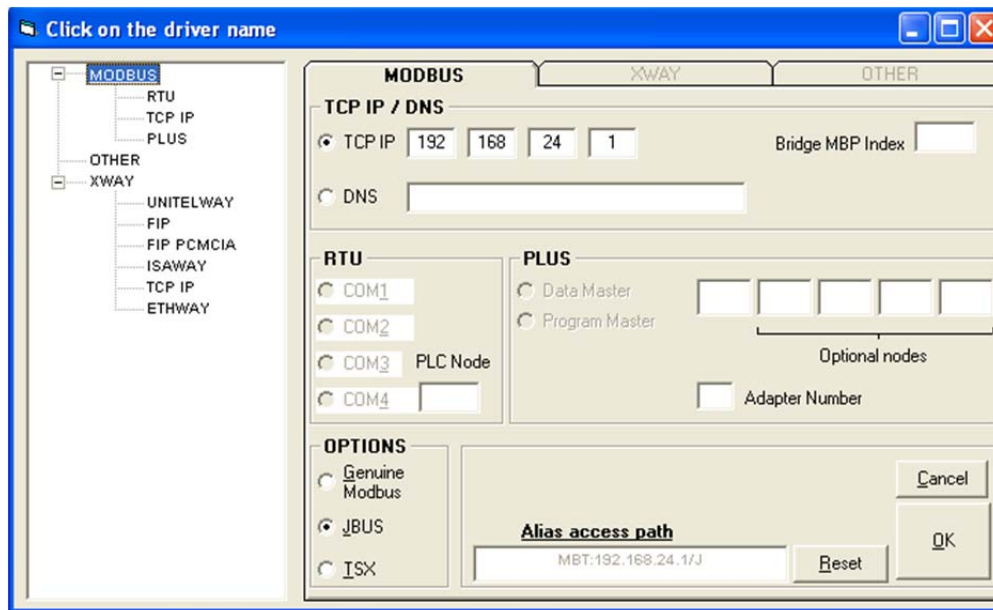


Imagen 23. Parámetros generales del diálogo configuración del canal físico de comunicación para el servidor OPC de Schneider.

Punto de paso.

Este campo es de texto libre, pero ha de coincidir con el nombre de una de las hojas del fichero de vehículos de pruebas (vehiculos.xls).

En función de la instalación que estamos pasando, nos interesará que PSFV nos de unos datos u otros del vehículo, un orden relativo o la posición con respecto a otros coches. Esto dependerá del punto sobre el que consultemos.

Si necesitamos la información del coche en pintura, en el fichero de vehículos (vehiculos.xls) deberemos crear una hoja Excel con nombre de punto de paso que hayamos decidido. Cuando el sistema este dialogando con el autómatas, el simulador irá a buscar los datos del coche de esa hoja y no otra, aunque esté presente en todas ellas.

Nombre del servidor OPC	Schneider-Aut.OFS
Alias OPC	PSF
Punto de paso	EMON
Tipo diálogo	2
Tipo identificador dialogo	1
Populación	BFB+HFE
Dirección inicio tabla	1110
Tamaño de la tabla	300

	A	B	C	D	E	F	G	H	I
1	PJI	ENCADENAMIENTO	MOFF_1	MOFF_2	MODELO	OBJ_536	PPUM_74	COEF_74	PPUM_222
2	1211067	6855	0308	2778	BFB	CSRGAC	969129752R	1	363158841R
3	1112236	6859	0308	2782	BFB	CSRGAC	969129752R	1	363158841R
4	1211686	6863	0308	2786	BFB	CSRGAC	969129752R	1	363158841R
5	1221000	6868	0308	2790	BFB	CSRGAC	969129752R	1	363158841R
6	1112301	6870	0308	2792	BFB	CSRGAC	969121990R	1	
7	1112346	6866	0308	2788	BFB	CSRGAC	969129752R	1	363158841R
8	921168	6872	0308	2797	BFB	CSRGAC	969121990R	1	
9	830287	6874	0308	2799	BFB	CSRGAC	969129752R	1	363158841R
10	1111005	6876	0308	2801	BFB	CSRGAC	969121990R	1	
11	422167	6878	0308	2803	BFB	CSRGAC	969121990R	1	
12	631184	6880	0308	2805	BFB	CSRGAC	969121990R	1	
13	1110950	6882	0308	2808	BFB	CSRGAC	969129752R	1	363158841R
14	1121403	6884	0308	2812	BFB	CSRGAC	969129752R	1	363158841R
15	830450	6886	0308	2814	BFB	CSRGAC	969129752R	1	363158841R
16	1131118	6888	0308	2816	BFB	CSRGAC	969129752R	1	363158841R
17	1220895	6890	0308	2818	BFB	CSRGAC	969129752R	1	363158841R
18	1211874	6893	0308	2821	BFB	CSRGAC	969129752R	1	363158841R
19	910897	6895	0308	2823	BFB	CSRGAC	969121990R	1	
20	931053	6897	0308	2825	BFB	CSRGAC	969129752R	1	363158841R
21	1211629	6899	0308	2297	BFB	CSRGAC	969129752R	1	363158841R
22	1110930	6901	0308	2828	BFB	CSRGAC	969121990R	1	
23	1211681	6903	0308	2830	BFB	CSRGAC	969129752R	1	363158841R
24	921200	6905	0308	2832	BFB	CSRGAC	969121990R	1	
25	1220766	6907	0308	2835	BFB	CSRGAC	969129752R	1	363158841R
26	1110954	6909	0308	2838	BFB	CSRGAC	969129752R	1	363158841R
27	1130842	6912	0308	2841	BFB	CSRGAC	969129752R	1	363158841R
28	1112040	6914	0308	2843	BFB	CSRGAC	969129752R	1	363158841R
29	1520230	6916	0308	2845	BFB	CSRGAC	969129752R	1	363158841R
30	1150022	6918	0308	2847	BFB	CSRGAC	969129752R	1	363158841R
31	1221072	6920	0308	2850	BFB	CSRGAC	969121990R	1	
32	921213	6922	0308	2852	BFB	CSRGAC	969121990R	1	

Tabla 6. Configuración del punto de paso "EMON". En los dos ficheros Excel del simulador.

Con esto conseguiremos que el diálogo se comporte como nosotros queremos, por ejemplo:

- El autómatas configurado como PSFV_1 nos pide dialogar para obtener los datos del coche A. Su punto de paso es EMON.
- El simulador detecta que este autómatas quiere comunicar y que necesita los datos del coche A. Revisa su configuración y ve que tiene que obtener los datos del vehículo dentro de la hoja Excel EMON del fichero vehículos.xls. Estos datos no tienen por qué ser iguales a los del resto de hojas Excel del fichero de vehículos.
- Rellena la tabla del autómatas y le responde que ya ha terminado.
- El autómatas recibe los datos y los procesa.

Tipo de diálogo.

Las instalaciones automatizadas en Renault pueden tener dos tipos de diálogo:

Al paso del vehículo. Preguntamos a PSFV por información del vehículo que tenemos delante de nosotros.

Al vehículo siguiente. Preguntamos a PSFV los datos del vehículo siguiente al que termina de pasar por delante de nosotros.

El primer tipo de diálogo, al paso del vehículo, se utiliza en aquellas instalaciones donde la secuencia de fabricación puede variar y no nos podemos fiar de un orden previo. Primero debemos identificar la carrocería de alguna manera, leyendo el trineo sobre el que va montada y alguna etiqueta de código de barras con su PJI (identificador numérico) para después consultar a PSFV los datos de esa caja concreta que terminamos de identificar. Este es el caso de las instalaciones de pintura y soldadura, donde el flujo de cajas puede variar.

El segundo tipo de diálogo, al vehículo siguiente, se utiliza en aquellos talleres donde existe un orden preestablecido e inmutable, como es el caso de Montaje, donde hay un flujo tenso, y no se pueden adelantar ni retrasar los vehículos. En ese caso, con conocer el vehículo que terminamos de fabricar, podemos preguntar a PSFV cuál es el siguiente de la lista. Una vez fabricado, repetimos la consulta. Esto simplifica bastante la instalación ya que elimina el paso previo de identificar el vehículo como si ocurre en el otro tipo de diálogo.

Los valores que puede tomar este campo en nuestro fichero Excel de configuración son:

Diálogo al paso del vehículo = 1.

Tipo dialogo	1
--------------	---

Diálogo al vehículo siguiente = 2.

Tipo dialogo	2
--------------	---

Tipo de identificador del diálogo.

A la hora de consultar los datos de un vehículo en PSFV tenemos dos opciones, preguntar por el identificador primario (PJI) o por el identificador secundario (número de trineo sobre el que va montado).

Las dos consultas nos tienen que devolver el mismo dato.



En las instalaciones de pintura lo habitual es preguntar utilizando el número de trineo, mientras que en Montaje se suele preguntar por un PJI. El simulador PSFV puede trabajar en un modo otro en función del valor de este parámetro.

Diálogo por PJI = 1

Tipo identificador diálogo	1
----------------------------	---

Diálogo por Número de trineo = 2

Tipo identificador diálogo	2
----------------------------	---

Lógicamente, en la configuración de la tabla de diálogo hemos de incluir estos campos para realizar la consulta.

Populación.

Este campo queda como reserva para una futura ampliación. Su función es filtrar un determinado conjunto de vehículos para que cuando preguntamos a PSFV por un coche, nos devuelva el más cercano que cumpla este filtro, a utilizar en aquellas instalaciones que sólo tratan un determinado modelo de coche y no pasa por ellas el 100% de la producción.

Dirección de inicio de la tabla de diálogo.

Palabra de la memoria del PLC a partir de la cual comienza la tabla de diálogo.

Es un valor numérico.

Si tenemos varios diálogos a la vez sobre el mismo autómatas no podemos solapar estas tablas.

Tamaño de la tabla de diálogo.

Tamaño de la tabla de diálogo en la memoria del PLC. La tabla comenzará en la dirección que indica el parámetro anterior.

Es un valor numérico.

Su valor máximo es de 100.



4.2.3 Configuración de la tabla de diálogo.

Una vez configurada la parte de comunicación del diálogo queda por configurar la propia tabla de diálogo.

Tendremos hasta N posibles palabras a documentar, tantas como hayamos declarado en la variable tamaño de la tabla de diálogo (configurada en el punto anterior).

Para cada palabra de la tabla de diálogo tenemos cinco posibles datos, aunque sólo podemos documentar 4 de ellos:

- **Dirección.** Valor fijo. No es posible cambiarlo. Nos servirá de ayuda para ver qué campo estamos documentando.
- **Contenido.** Tipo de dato a utilizar. Tenemos dos posibilidades, dato de control e información del propio vehículo.
- **Tipo.** Formato del campo. Dos posibilidades: DECIMAL | ASCII. (En mayúsculas)
- **Long.** Número de caracteres que alberga el campo.
- **Desplazamiento.** Cuando estemos concatenando una cadena de caracteres, posición de este campo dentro de toda la cadena.

Dirección.

Este campo se rellena de forma automática para ayudarnos a documentar la tabla de diálogo. Su valor dependerá de la posición dentro de la tabla y del valor que hemos marcado como inicio de la tabla.

Si no vamos a utilizar la palabra, dejaremos el resto de campos vacíos.

Contenido.

Este campo indica la función de la palabra y para qué va a ser utilizada en el diálogo.

El valor de este campo puede ser de dos tipos. Un campo de control o de datos.

Si el campo es de control puede tomar unos valores determinados, los posibles valores son:

Campo	Función
VALAPI	Semáforo de la comunicación. Parte autómata
VALCAL	Semáforo de la comunicación. Parte PSFV
PJI_P	Primeros dos caracteres del PJI (P) por el que estamos preguntando
PJI_J	Tercer carácter del PJI (J) por el que estamos preguntando
PJI_I	Cuarto al séptimo carácter del PJI (I) por el que estamos preguntando
PJI_I1	Campo a utilizar cuando especificamos el cuarto y quinto carácter del PJI de forma separada.
PJI_I2	Campo a utilizar cuando especificamos el sexto y séptimo carácter del PJI de forma separada.
TRINEO	Número del trineo por el que preguntamos
TRINEO_N1	Número del trineo del vehículo posterior al que consultamos
PJI_P_N1	Letra P del PJI del coche posterior al que consultamos
PJI_J_N1	Letra J del PJI del coche posterior al que consultamos
PJI_I_N1	Letra I del PJI del coche posterior al que consultamos
PJI_I1_N1	Primer y segundo carácter de la letra I del PJI del coche posterior al que consultamos
PJI_I2_N1	Tercer y cuarto carácter de la letra I del PJI del coche posterior al que consultamos

Tabla 7. Campos de control.

Si el campo contenido alberga un campo de datos, el nombre del campo hará referencia a una columna del fichero Excel de vehículos.

Por ejemplo, cuando consultemos el ENCADENAMIENTO de un vehículo, el simulador irá al fichero de vehículos, seleccionará la columna llamada ENCADENAMIENTO y dentro de esa columna, el valor correspondiente al vehículo consultado.

Dirección	Contenido	TIPO	LONG	DESPLAZAMIENTO
1100	VALAPI	DECIMAL		
1101	TRINEO	DECIMAL		
1102				
1103				
1104				
1105	PJI_P	DECIMAL	2	0
1106	PJI_J	DECIMAL	1	2
1107	PJI_I	DECIMAL	4	3
1108				
1109	PJI_P_N1	DECIMAL	2	0
1110	PJI_J_N1	DECIMAL	1	2
1111	PJI_I_N1	DECIMAL	4	3
1112	ENCADENAMIENTO	DECIMAL		
1113	OBJ_536	ASCII	?	?

	A	B	C	D	E	F
1	PJI	ENCADENAMIENTO	MOFF_1	MOFF_2	MODELO	OBJ_536
2	1211067	6855	0308	2778	BFB	CSRGAC
3	1112236	6859	0308	2782	BFB	CSRGAC
4	1211686	6863	0308	2785	BFB	CSRGAC
5	1221000	6868	0308	2790	BFB	CSRGAC
6	1112301	6870	0308	2792	BFB	CSRGAC
7	1112346	6866	0308	2788	BFB	CSRGAC
8	921168	6872	0308	2797	BFB	CSRGAC
9	830287	6874	0308	2799	BFB	CSRGAC
10	1111005	6876	0308	2801	BFB	CSRGAC
11	422167	6878	0308	2803	BFB	CSRGAC
12	631184	6880	0308	2805	BFB	CSRGAC
13	1110950	6872	0308	2808	BFB	CSRGAC
14	1121403	6884	0308	2812	BFB	CSRGAC
15	830450	6886	0308	2814	BFB	CSRGAC
16	1131118	6888	0308	2816	BFB	CSRGAC
17	1220895	6890	0308	2818	BFB	CSRGAC
18	1211874	6893	0308	2821	BFB	CSRGAC

Imagen 24. Configuración tabla de diálogo.

En el ejemplo devolverá el valor 6884 en decimal para el vehículo 1121403.

Nota: Si utilizamos PJI_I no podemos utilizar los valores PJI_I1 y PJI_I2, son excluyentes. A su vez si pedimos PJI_I1 deberemos pedir también PJI_I2 para completar el PJI.

Tipo.

Indica el formato del campo, puede tener dos valores posibles: ASCII | DECIMAL.

Long.

Este campo tiene sentido cuando transmitimos caracteres ASCII y queremos enviar parte de una cadena más amplia. Por ejemplo, tenemos un campo llamado MODELO en nuestra base de datos que puede tomar los siguientes valores B95, KFB, HFE y queremos enviar sólo un carácter (el primero). Configuraremos que la longitud de nuestra palabra es 1.

Desplazamiento.

Cuando creamos un campo combinado por varias palabras, como puede ser el PJI, debemos ir configurando la posición de cada palabra dentro del campo global.

Es el caso típico al que nos enfrentamos al transmitir el PJI.

Un PJI son 7 caracteres:

- Carácter P: Dos caracteres.
- Carácter J: 1 carácter.
- Carácter I: 4 caracteres.

Si queremos transmitir el PJI en ASCII, por cada palabra de autómeta como mucho podremos transferir 2 caracteres ASCII (las palabras son de 16 bits). Por tanto necesitaremos 4 palabras.

Con una configuración como la siguiente lo podremos hacer:

Dirección	Contenido	TIPO	LONG	DESPLAZAMIENTO
4400	VALAPI	DECIMAL		
4401	TRINEO	DECIMAL		
4402				
4403	PJI_P_N1	ASCII	2	0
4404	PJI_J_N1	ASCII	1	2
4405	PJI_I1_N1	ASCII	2	3
4406	PJI_I2_N1	ASCII	2	5

Imagen 25. Configuración campo ASCII en tabla de diálogo.

Aunque también lo podemos utilizar cuando queremos transmitir un número en decimal que está compuesto por dos palabras de 16 bits, pero no queremos ver el número final descompuesto, sino unido.

4.2.4 Creación de los datos de los vehículos a transmitir al PLC.

Esta es la parte más fácil de todo el proceso de configuración.

Deberemos tener en la carpeta de Configuración el fichero Excel vehiculos.xls donde almacenaremos los datos de los posibles vehículos a recibir por el autómata.

Existirá una hoja Excel por cada punto de paso que hayamos definido. En todas las hojas Excel puede haber los mismo datos para un vehículo, aunque no es obligatorio, lo que cambiará es la posición relativa de los vehículos con el resto.

En la hoja Excel de Pintura los vehículos no llevarán el mismo orden que en la hoja Excel de montaje. En taller de pintura los vehículos van ordenados por color y mientras que en el taller de montaje van ordenados por la secuencia de fabricación.

El simulador buscará en estas hojas los campos que hayamos definido en la tabla de diálogo (Ver punto anterior) en el nombre de las columnas. Una vez localizado el campo, buscará el vehículo que le corresponde y escribirá el contenido en la memoria del PLC. Con excepción de los campos calificados como campos de control.

	A	B	C	D	E	F	G	H	I	J
1	PJI	ENCADENAMIENTO	MOFF_1	MOFF_2	MODELO	OBJ_536	PPUM_74	COEF_74	PPUM_222	COEF_222
2	1211067	1235	308	2778	BFB	CSRGAC	969129752R	1	363158841R	1
3	1112236	1239	308	2782	BFB	CSRGAC	969129752R	1	363158841R	1
4	1211686	1242	308	2785	BFB	CSRGAC	969129752R	1	363158841R	1
5	1112346	1245	308	2788	BFB	CSRGAC	969129752R	1	363158841R	1

Imagen 26. Fichero de vehículos.

Si codificamos un campo en la tabla de diálogo que después no está presente como columna Excel en la hoja de vehículos, el simulador se limitará a dejar la palabra de la tabla de diálogo vacía, pero sin generar un error de aplicación.

4.3 Manual de usuario.

4.3.1 Proceso de diálogo.

El diálogo entre la aplicación y el autómata funciona bajo la estructura VALAPI / VALCAL según la norma Renault correspondiente.

Cuando el PLC quiere dialogar ha de:



- Escribir el coche para el que necesita conocer los datos (PJI o trineo).
- VALAPI = 1
- VALCAL = 0

El simulador verá que ha basculado el VALAPI y el VALCAL y:

- Leerá la tabla de diálogo.
- Obtendrá el identificador del vehículo.
- Buscará ese identificador en el fichero de vehículos según la configuración del diálogo.
- Rellenará la tabla con los datos obtenidos.
- VALAPI = 0
- VALCAL = 1

4.3.2 PANTALLA DE LA APLICACION.

En la pantalla del programa veremos en tiempo real el estado de la tabla de diálogo, y podremos interactuar con el diálogo forzando algunos casos.

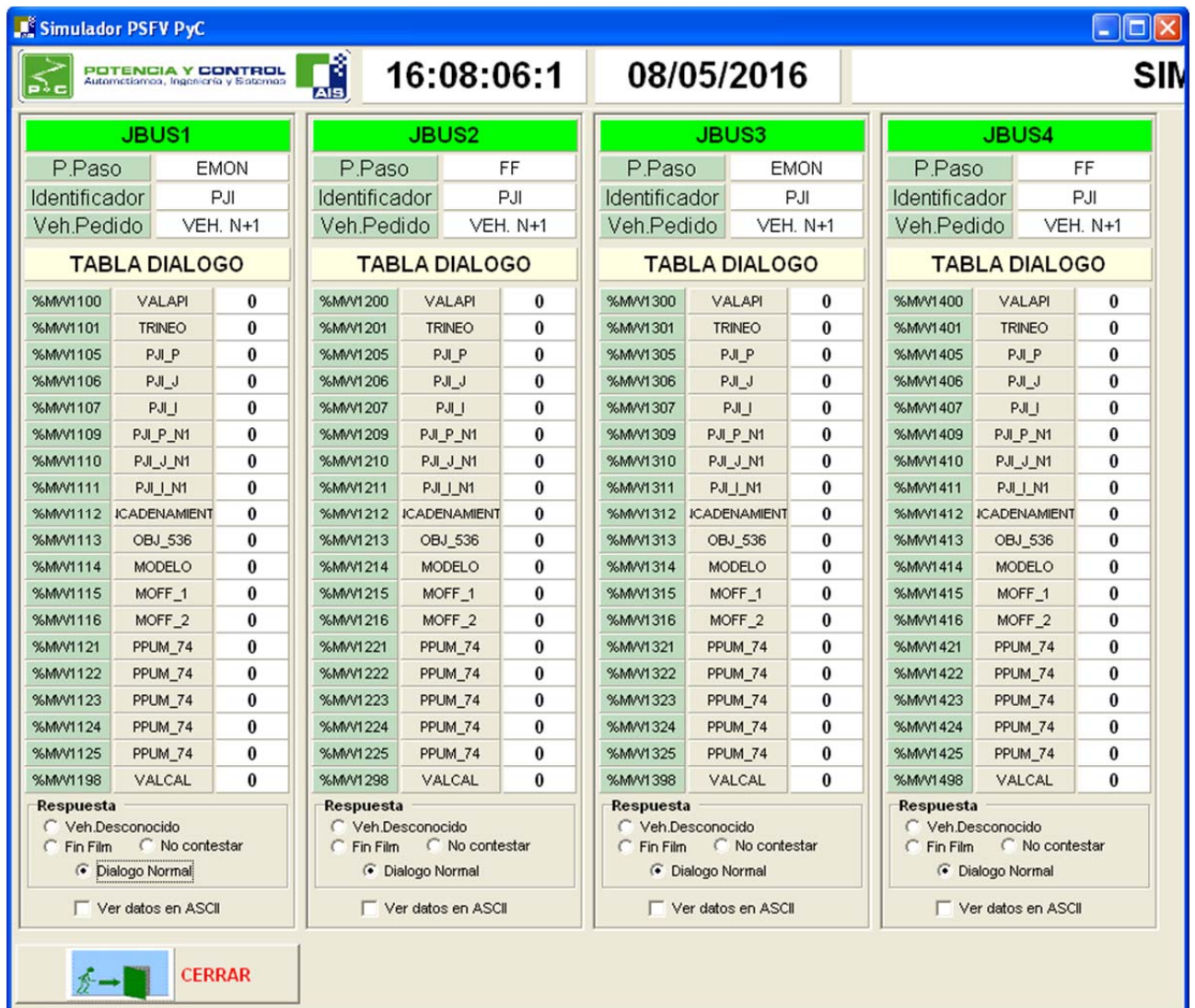


Imagen 27. Pantalla principal el simulador PSFV.

En la pantalla del simulador veremos cada una de las tablas de diálogo definidas, y para cada una de ellas, tenemos tres zonas: configuración, tabla de diálogo y selección de respuesta.

Configuración.

Vemos los datos básicos de la configuración del diálogo:

- P.Paso: Hoja Excel donde se buscarán los datos para el vehículo consultado.

- Identificador: identificador que utilizamos para dialogar, puede ser PJI o trineo.
- Veh. Pedido: datos del vehículo con el que el simulador contestará, hay dos opciones, vehículo actual, contesta con los datos del vehículo por el que preguntamos, o vehículo N+1, cuando contesta con la información del vehículo siguiente al que hemos indicado en el diálogo.

JBUS1	
P.Paso	EMON
Identificador	PJI
Veh.Pedido	VEH. N+1

Imagen 28. Detalle pantalla principal el simulador PSFV.

Tabla de diálogo.

Visualizamos el contenido de la tabla en tiempo real. También su estructura.

En esta tabla hay dos palabras, que hacen el control de la comunicación y que deberán ser visibles siempre, el VALAPI y el VALCAL.

TABLA DIALOGO		
%MW1100	VALAPI	0
%MW1101	TRINEO	0
%MW1105	PJI_P	0
%MW1106	PJI_J	0
%MW1107	PJI_I	0
%MW1109	PJI_P_N1	0
%MW1110	PJI_J_N1	0
%MW1111	PJI_I_N1	0
%MW1112	ICADENAMIENT	0
%MW1113	OBJ_536	0
%MW1114	MODELO	0
%MW1115	MOFF_1	0
%MW1116	MOFF_2	0
%MW1121	PPUM_74	0
%MW1122	PPUM_74	0
%MW1123	PPUM_74	0
%MW1124	PPUM_74	0
%MW1125	PPUM_74	0
%MW1198	VALCAL	0

Imagen 29. Detalle diálogo pantalla principal el simulador PSFV.

En la tabla habrá campos configurados en formato ASCII, aunque la visualización por defecto será decimal. Con el control Ver datos en ASCII podremos cambiar esta representación.

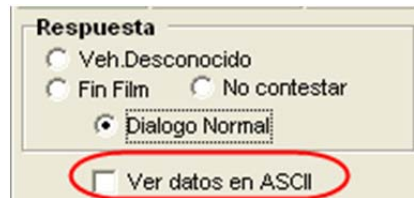


Imagen 30. Configuración de respuesta del simulador PSFV.

Selección de respuesta.

Podemos adecuar la respuesta del simulador a nuestras necesidades sin necesidad de manipular la aplicación o el fichero Excel de vehículos.

Tenemos disponibles las siguientes opciones:

- Diálogo normal. Respuesta automática, el simulador se comporta de forma normal.
- Veh. Desconocido. El simulador contesta como si no conociera el vehículo por el que preguntamos, aunque si este presente en el fichero Excel de vehículos.
- Fin de Film. El simulador contesta que no tiene más datos disponibles, ha llegado al final del fichero.
- No contestar. El simulador no contesta. Se utiliza este caso para probar el timeout de la comunicación.



Imagen 31. Configuración de respuesta del simulador PSFV.



5 CONCLUSIONES.

5.1 Conclusiones del TFM.

La elaboración de este TFM nos ha permitido ver que dos campos distintos, como son la ingeniería del software y la automatización de procesos tienen aspectos comunes. Qué no sólo es posible realizar ingeniería del software en los proyectos de informática industrial, sino que es recomendable, y nos ayudará a llevar el proyecto a buen puerto.

La simulación cada vez es más importante, y probar sistemas complejos cada vez lleva más tiempo, con este TFM, hemos conseguido crear un sistema que simula el comportamiento de PSFV y que nos servirá de apoyo en nuestras simulaciones para reducir el número de errores del software en las primeras fases del proyecto.

No sólo podremos simular el comportamiento y tratamiento de errores de nuestro programa autómatas, sino que también podremos simular su comportamiento ante un flujo de datos cada vez más heterogéneo, y probarlo ante condiciones y casos, que en circunstancias normales no se dan, pero que se presentarán a lo largo de la vida de la instalación. Sin la simulación, estos “casos raros” quedarían sin probar. Y no podríamos garantizar la entrega un programa autómatas libre de errores.

Pero ¿cómo podemos cuantificar el beneficio de la simulación previa? De tres formas, por la reducción de horas de desarrollo, por la disminución de fallos en la puesta en producción de nuestra instalación, al entregar un software más probado, y por el aumento de la eficiencia de nuestra aplicación, ya que entregaremos una instalación probada bajo todo tipo de condicionantes y flujos y con el algoritmo de control lo más probado y eficiente posible.

A nivel académico, con el desarrollo de este proyecto se han puesto en práctica numerosos conocimientos adquiridos a lo largo del Máster de Informática Industrial, por enumerar algunos:

- Comunicaciones industriales PLC vs PC: OPC.
- Redes industriales.
- Programación de PLCs.
- Ingeniería del software industrial, análisis y diseño de aplicaciones informáticas industriales.
- Diseño de interfaces.

Además, al tratarse de un proyecto en el campo de automatización industrial y del control, ha permitido conocer otros campos en los que se integra la informática.

Ahorro económico en un caso de análisis: reforma de stock de colores de pintura. Renault Palencia.

Si analizamos cada uno de estos tres puntos para un caso concreto, como puede ser el expuesto en la memoria del TFM hallaremos una justificación económica al desarrollo del simulador y a la propia etapa de simulación.



El ahorro económico se concentra en:

- La disminución de las horas de programación en las fases de desarrollo. Simular el programa a la vez que se construye nos permitirá una reducción de 4.500 €.
- La corrección de fallos a lo largo de la vida de la instalación. Corregir el programa una vez puesto en producción es complicado, la modificación no debe interferir en el funcionamiento de la instalación y además, siempre hay la urgencia de corregir algo que no funciona bien. En una instalación como la descrita, que tendrá una vida útil de al menos 20 años nos podemos encontrar del orden de 15 errores graves de funcionamiento, la mayor parte concentrados en los primeros años donde el software todavía contiene errores. Cada uno de estos fallos conlleva un coste por sí mismo, por la eventual pérdida de producción. Si la parada es de 30 minutos, el coste de cada defecto será de 78.750 €. A ese coste habría que sumarle la posible pérdida de eficiencia de la instalación que es difícil de cuantificar sin conocer el error concreto, y el coste de la reparación del error.
- El aumento de la eficiencia. La eficiencia de por sí de la instalación debe ser alta, este stock tiene que enviar los vehículos a la siguiente instalación ordenados por color para su pintado. Cuanto mejor vayan ordenados por color, menos tiempos muertos tendrá la instalación siguiente, y menos purgas de pintura tendrá que realizar. Estas instalaciones fabrican a su máxima capacidad 300.000 vehículos año, lo que a una trancha media de 5 vehículos del mismo color, supone que encadenan 60.000 tranchas de colores. El coste de una purga de disolvente, sin contar el tiempo que se pierde, es de 0,5 € de producto. Con el algoritmo funcionando al 100% de eficacia, haremos 60.000 purgas con un coste de 30.000 € de disolvente. Si la eficiencia se reduce un 1% comenzaremos a realizar más purgas ya que las tranchas de colores serán peores y tendremos un sobrecoste de 400 € de disolvente. Además, la siguiente instalación tendrá su sobre-tiempo de purgado que no la permitirá fabricar al 100%, bajando su capacidad de 300.000 vehículos al año a tan sólo 299.800 vehículos, una pérdida aproximada de un vehículo por cada día de fabricación. La pérdida económica es de 450.000 € al año en vehículos no fabricados.

El ahorro es considerable, y amortizable a lo largo de toda la vida útil de la instalación.

Este ahorro es sólo para un caso concreto, el simulador de PSFV tiene aplicación en todas aquellas instalaciones complejas que necesiten datos de PSFV para trabajar.



5.2 Uso en una aplicación real.

5.2.1 Descripción de las modificaciones.

Una de las primeras aplicaciones de uso de este simulador va a ser la verificación de los algoritmos creados para el nuevo stock de tranchas de la nave de Pintura, en la factoría de Renault Palencia.

Durante el mes de Agosto se va a reformar esta instalación automatizada, en el marco de la estrategia de hipercompetitividad del grupo Renault.

Objetivos del proyecto.

Los objetivos de este proyecto van incluidos en el **“CdC Renault Modification du stock rafaleur & stock reconstruction pour passage à des rafales à 8 ald 2”**.

Y describe sus objetivos de la siguiente forma:

“Cómo se ha definido en el CDC general “CdC Modification du stock rafaleur & stock reconstruction pour passage à des rafales à 8 ald 2” se trata de acondicionar las instalaciones 230/12 (Manut. Aprestos), 230/15 (Stock de ráfagas), 230/31 (manut), 230/32 (manut. Bases) y 230/21 (Stock final) para garantizar la entrega de una trancha mínima de color a la Línea de Bases Hidro sin degradar el Film de Fabricación. Y, mediante el aumento de líneas de encadenamiento y la modificación del programa de gestión en el autómata, dotar al taller de pintura de capacidad para encadenar ráfagas de 8 sin perder versatilidad, aumentando el número de calles del stock, consultando las cajas que están en el encurso de aprestos, las que se encuentran en proceso de pintado y de la ocupación real en el stock final de pintura.

- *Validando constructivamente el sistema para las diferentes instalaciones a modificar.*
- *Armonizando la programación de los sistemas con el estándar RENAULT.*
- *Implantando en cada una de las instalaciones las modificaciones, sistemas y sus elementos auxiliares para el correcto tratamiento de las cajas.*
- *Pasando los jalones del proyecto sin retraso imputable a los automatismos con el nivel de calidad requerido.*
- *Obteniendo las prestaciones de automatismos contractuales*

La instalación gestiona las cajas según tres criterios:

- *Tranchas de Color, coloca las cajas casadas por color.*
- *Ordenamiento SSAR, se ordenan las cajas por orden de fabricación.*
- *FIFO, primera en entrar primera en salir.*

La selección del modo de Funcionamiento la realiza el sistema de Gestión con el objetivo de garantizar una trancha de colores mínima.

Se calcula la trancha de colores teórica de las cajas en curso hacia la instalación 230/15 desde la instalación 230/12 ordenadas por film de fabricación, las cajas en



curso por la instalación 230/12 se reciben por un punto de dialogo PSFV O3, si dicha trancha es mayor que la mínima el sistema conmuta al modo 'Ordenamiento SSAR' si es inferior selecciona el modo 'Trancha de Color'.

La gestión se compone de dos partes:

- Gestión de Entrada del Stock, colocar las cajas en las líneas del stock según modo de funcionamiento.
- Gestión de Salida del Stock, envío de calas a la líneas de pintura seleccionando según modo de funcionamiento

El Sistema de Supervisión se encarga de seleccionar el modo de funcionamiento SSAR/Trancha y calcular la cadencia por colores de la producción, eligiendo los 5 de mayor cadencia que se asignará a las líneas 1, 2, 3, 4, 5 y 6. Para conseguirlo utiliza el encurso de cajas entre la Instalación 230/12 y la 230/15, y las cajas presentes dentro del Stock.

Para optimizar el funcionamiento el cambio del modo de funcionamiento SSAR/FIFO de la salida de la instalación no es inmediato se realiza una vez agotadas las cajas que entraron con el funcionamiento anterior.

El fallo en el sistema de supervisión o su desconexión temporal no bloquea el funcionamiento de la instalación, siendo posible asignar la cadencia manualmente.

También se realiza la gestión de:

- Destinos prohibidos a las líneas por modelos y colores.
- Extracción de cajas por la línea de extracción.
- Bloqueo de destinos.
- Envío de cajas provenientes de la cabina de retoques solamente entre dos ráfagas de colores diferentes, sin afectar a la eficiencia de salida del stock.

Funcionalidades extra.

- Temporizador ajustable en minutos para la supresión de órdenes de fabricación que estén con retraso.
- El cálculo de la antigüedad con el "último bueno" se realiza con una tolerancia configurable.
- Priorización de la salida de una caja configurable.
- Cinco configuraciones de tramos de "valor máximo" configurables.

5.2.2 Justificación de la simulación.

Para entender la necesidad de la simulación en este proyecto, hemos añadido la justificación del personal de Renault al solicitarla. En concreto de **D. Juan Carlos Vallecillo, responsable de Mantenimiento Pintura.**

"En un entorno industrial como puede ser una fábrica de automóviles, las paradas de las líneas de producción suelen ser muy limitadas, de en torno a 3 semanas como



máximo, por lo que las ventanas de intervención para realizar modificaciones/mejoras o nuevas integraciones están muy ajustadas y delimitadas. Este es uno de los motivos principales por los que las simulaciones cobran una importancia muy grande, ya que sino ciertos proyectos no serían viables y supondría un estancamiento de la Performance de la fábrica.

El otro motivo que justifica las simulaciones es el económico, cuando la fábrica comienza de nuevo la fabricación, todos los modos de funcionamiento y modos de fallo deben estar bien probados para evitar pérdidas de producción. Estos modos de funcionamiento / fallo solo pueden ser probados a través de la simulación ya que en un escenario real haría falta una cantidad enorme de coches y tiempo para poder reproducir todas las casuísticas. Por hacernos una idea del impacto económico, en una fábrica como la de Renault Palencia, donde se fabrica un coche por cada 55 segundos aproximadamente y trabajando 24 horas al día durante 5 días a la semana y en ocasiones también los fines de semana, por cada minuto de parada, la fábrica de Palencia pierde aproximadamente 100 Euros.

En resumen, sin las simulaciones, el sector del automóvil estaría tecnológicamente atrasado y podría someterse a pérdidas de muchos miles de Euros que no puede permitirse dado la competencia existente hoy día.

Otro de los objetivos de una simulación de un stock complejo, como este caso, es la mejora continua, ya que durante una simulación, se pueden ver ciertas situaciones de mejora no contempladas en fase de proyecto y que contribuya a la mayor eficiencia de los algoritmos programados.

Por todo esto, Renault considera imprescindible la contratación de la fase de simulación en proyectos complejos y los resultados hasta la fecha han sido tremendamente satisfactorios.”

5.2.3 Conclusiones a priori.

En el CdC Renault se pide realizar la simulación fuera de línea de los algoritmos de ordenación del stock de tranchas, para verificar que proporcionan los resultados correctos. El objetivo es llegar a la instalación de campo con los algoritmos chequeados y validados, dado que el período de modificación e instalación de equipos es muy pequeño y no habría tiempo suficiente para realizar la validación in-situ.

Esta prevista una simulación informática durante el mes de Julio de 2016, previa a la instalación de verano, de una semana de duración. Para ello se partirá de la secuencia de fabricación de la semana previa a la simulación y se compararán los resultados obtenidos con el nuevo algoritmo con los resultados reales.

Si no se dispusiera de la herramienta de simulación de PSFV esta fase llevaría al menos tres semanas, y no se podrían probar todos los casos posibles.

Una vez verificado que el algoritmo da los resultados esperados, se probará el resto de los casos, que aun no dándose en el día día, pueden presentarse más adelante.



5.3 Ampliaciones y posibles mejoras.

Algunas de las posibles ampliaciones de este proyecto surgirán con la explotación del sistema final por parte del usuario, ante la aparición de nuevas necesidades.

Una de las posibles mejoras es la inclusión de PLCs de Siemens. El simulador está diseñado para trabajar con PLCs Schneider basándose en la norma CNOMO correspondiente. Se ha partido de este tipo de autómatas por ser mayoritarios en Renault, cerca del 90% del parque instalado en las naves de montaje.

Otra de las posibles mejoras a realizar, sería migrar el proceso de configuración de ficheros Excel a un sistema de base de datos, esto se descartó inicialmente porque suponía una mayor complejidad y una curva de aprendizaje mayor, no todos los usuarios del sistema tienen porque conocer el manejo de base de datos, pero si todos ellos son usuarios de Excel. Esta mejora dotaría al sistema de una mayor complejidad, pero a su vez, de una mayor potencia.

Por último, y no como aplicación de este proyecto, pero si como consecuencia de él, podría ser la construcción de una herramienta informática para hacer trazabilidad del funcionamiento del programa automático. Hemos creado un simulador para probar nuestro programa automático, pero no tenemos la herramienta para realizar una trazabilidad de su funcionamiento para analizar su comportamiento a posteriori.



6 REFERENCIAS.

[1] Ciclo de vida desarrollo en cascada. [Fecha de consulta Mayo 2016]. Disponible en https://es.wikipedia.org/wiki/Desarrollo_en_cascada

[2] La importancia de la simulación en la industria. Néstor Olivas Tecero. [Fecha de consulta Mayo 2016]. Disponible en <http://uniionestor.files.wordpress.com/2011/03/la-importancia-de-la-simulacion-de-procesos-en-la-industria-tendencias-tecologicas-en-la-produccion-industrial1.pdf>

[3] Ponencia sobre simulación de procesos. Aplicación real. II Jornada reingeniería procesos logísticos. Junio 2012.

[4] Guía práctica para la simulación de procesos industriales. CETEM. [Fecha de consulta Mayo 2016]. Disponible en www.cetem.es

[5] Norma Renault diálogo PSFV. [Fecha de consulta Mayo 2016]. Disponible en www.cnomo.org

[6] Manual de funcionamiento PSFV V810. Renault.

[7] Sistemas CAE. [Fecha de consulta Mayo 2016]. Disponible en https://es.wikipedia.org/wiki/Ingenier%C3%ADa_asistida_por_computadora

[8] Ingeniería del software. Un enfoque práctico. Roger S. Pressman. Séptima edición. McGraw-Hill.

[9] Ingeniería del software. Ian Sommerville. Novena edición. Ed. Pearson.

[10] Lenguaje unificado de modelado (UML). [Fecha de consulta Mayo 2016]. Disponible en www.uml.org

[11] Especificación protocolo OPC. [Fecha de consulta Mayo 2016] Disponible en <https://opcfoundation.org>

[12] Manual de programación PL7pro. Schneider Electric.



A. ANEXOS.

A.1 Presupuesto.

Elaboramos un presupuesto desglosado por tareas y horas, en base a la planificación previa.

Tarea	Horas
Análisis	
Análisis inicial, planificación, requisitos	16
Diagramas casos de uso y otros	32
Otros diagramas de análisis	16
Total análisis	64
Diseño	
Diseño aplicación e interface	24
Diseño ficheros de configuración	16
Total diseño	40
Programación y pruebas	
Programación aplicación	150
Pruebas de desarrollo	50
Total programación	200
Documentación	
Documentación de proyecto	44
Presentaciones y otros documentos	16
Total documentación	60
TOTAL PROYECTO	364

Tabla 8. Desglose de horas del TFM.

El coste total del proyecto es de **18.000 €**

A.2 Tecnología OPC.

A.2.1 Introducción OPC.

El OLE para control de procesos (OPC) es un estándar abierto para compartir datos entre dispositivos de campo y aplicaciones de ordenador basado en OLE de Microsoft. Permite a las aplicaciones leer y escribir valores de proceso y que los datos sean compartidos fácilmente en una red de ordenadores.

El estándar, gobernado por la Fundación OPC, es de dominio público y disponible para cualquiera que quiera usarlo. Tradicionalmente, los fabricantes de software para acceso de datos de proceso tenían que desarrollar drivers específicos para cada tipo de hardware al que querían acceder. Cada software requería un driver distinto para cada hardware, implicando un esfuerzo enorme, al que hay que añadir el de las actualizaciones continuas.

Con OPC, los fabricantes de hardware sólo tendrán que preparar un conjunto de componentes de software para que los clientes los utilicen en sus aplicaciones. Los desarrolladores de software no tendrán que reescribir los drivers debido a nuevas versiones de hardware.

Los usuarios finales tendrán muchas más alternativas de integrar distintos sistemas.

Una aplicación OPC, como cualquier otra aplicación OLE (o DDE), constará de servidores y clientes OPC. Cada cliente, es decir, cada aplicación de usuario, SCADA, módulo histórico, o aplicación de usuario en C++ o VB interroga al servidor que contiene los datos que necesita



Imagen 32. Integración de datos en OPC.

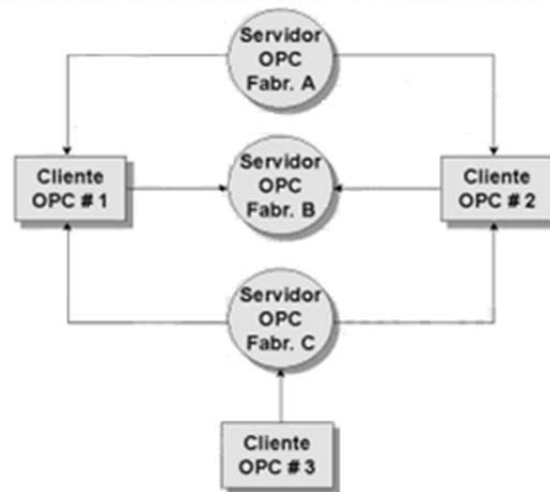
DIAGRAMA CLIENTE - SERVIDOR OPC

Imagen 33. Arquitectura cliente – Servidor OPC.

Los servidores están organizados en grupos y cada grupo puede contener distintos ítems.

Las diferentes partes de la aplicación (displays de operador, informes, etc.) pueden usar distintos grupos, los cuales pueden tener distinta frecuencia de refresco y pueden ser de acceso secuencial o basado en excepciones (eventos). Los ítems representan conexiones a fuentes de datos dentro del servidor (variables de proceso). A cada ítem se asocia un valor (valor de la variable de proceso), un cualificador (estado de la variable, OK, bajo rango, etc.) y una marca de tiempo

A.2.2 Historia del OPC

La historia del interface OPC es larga:

- 1982 DOS: Desde los “viejos” tiempos del DOS se ha intentado conectar los equipos de campo a ordenadores personales. En aquella época oscura, los PC sólo podían hacer una cosa a la vez y se necesitaban drivers específicos para cada dispositivo externo que se quería conectar.
- 1990 Windows 3.0 y Dynamic Data Exchange (DDE Intercambio Dinámico de Datos): Con él apareció la posibilidad (a bajo precio) de ejecutar varias tareas simultáneamente en un ordenador y un mecanismo estándar para que se intercambiaran información entre ellas. Se podía ya, por ejemplo, ver datos de proceso de instrumentos de campo en una hoja de Excel. Desafortunadamente, DDE tenía limitaciones: no era excesivamente robusto,



el ancho de banda era limitado (la información se transfiere en formato de caracteres) y no se podía enviar información a través de redes.

- 1992 Windows 3.1 & Object Linking & Embedding 1.0 (OLE): En cuanto apareció, ya se intuía que OLE iba a desplazar al DDE para intercambio de datos, ya que era más flexible, robusto y usaba mecanismos de transporte más eficientes.
- 1993 OLE 2.0 Comité WinSEM: Este grupo de desarrolladores de software técnico se reunía regularmente en las instalaciones de Microsoft y estaban interesados en aplicar el PC a control industrial y adquisición de datos. Se centraron en técnicas OLE para intercambiar datos entre aplicaciones en (cuasi) real-time. En particular, los fabricantes de SCADA se mostraron muy interesados en estandarizar la interface entre el “núcleo” del SCADA y los drivers de los dispositivos de campo. Eso podía beneficiar tanto al fabricante del SCADA como al del dispositivo de campo.

A pesar de los esfuerzos realizados no se acabó de definir un estándar. Varios de los participantes incluido Microsoft) decidieron crear un grupo más pequeño y más manejable que pudiera producir un estándar en poco tiempo, este grupo sería el OPC Task force.

- 1994 OLE 2.1 (32 bit).
- 1995 OPC Task Force: La OPC Task Force se hace pública en la presentación de ISA de 1995, donde se anuncian los objetivos del grupo. Los primeros miembros son: Fisher-Rosemount, Intellution, Intuitive Technology, OPTO 22, Rockwell Software y Microsoft. El primer borrador de la especificación aparece en diciembre.
- 1995 Windows 95: Hacia el final de 1995 aparece Windows 95, se generaliza el Windows (a secas) como sistema operativo.
- 1996 Especificación OPC V 1.0 (A): En agosto 1996 aparece la versión 1.0.
- 1996 Windows NT 4.0: Aparece en 1996, combina la fiabilidad del NT 3.5 con el interface de usuario del W 95. Incluye soporte de DCOM (Distributed Common Object Model), que permite a las aplicaciones crear y trabajar con objetos residentes en otros ordenadores a través de la red.
- 1996 Fundación OPC: En agosto de 1996 se crea la fundación como organización sin interés económico con la tarea de gestionar el estándar OPC. Su “misión” es: Desarrollar un estándar abierto, basado en los requerimientos funcionales de la tecnología OLE/COM y DCOM, que fomente mayor interoperabilidad entre aplicaciones de control/automatización, sistemas/dispositivos de campo y aplicaciones de gestión. La primera presentación importante fue en el ISA Show de 1996 en Chicago.



- 1997 Comités Técnicos OPC: Para poder gestionar el amplio rango de asuntos relacionados con los datos del control de los procesos la fundación OPC forma una serie de “comités” para investigar y diseñar mejoras a la especificación original.
- 1998 Especificación OPC V1.01.
- 1998 Windows 98: Se espera para la segunda mitad de 1998, W98 es una mejora incremental sobre W95 y su importancia para la industria del control de procesos por ser un sistema operativo multitarea con soporte para DCOM integrado y por lo tanto para tecnología de servidor de OPC barato y probado.
- 1998 Windows NT 5.0: También aparecerá en la segunda mitad de 1998, ofrece mejoras significativas sobre NT 4.0.



A.3 Proceso de diálogo con PLCs bajo norma Renault EB03.64.210/B.

Esta norma Renault define la forma de dialogar con PSFV para obtener la información necesaria en los autómatas.

Se apoya en otro conjunto de normas, en concreto la GE03.FP.035/C que documenta como utilizar los bloques función estándar para programar estos intercambios en los autómatas Schneider.

La norma está escrita sólo francés, aunque si estuviese documentada en otro idioma y hubiera diferencias de interpretación entre ellas, la que prevalecería siempre sería la versión en francés.

En este anexo incluiremos algunos puntos significativos de la norma que explicarán el funcionamiento de nuestro simulador:



Norme d'interface avec l'informatique de pilotage	EB03.64.210	/B
	Norme	
	<i>Statut</i> Exécutoire	

Objet Spécifier les principes d'échanges d'Informations entre les automates et les calculateurs de pilotage de type SiPxxx et PSFV.
Ce document ne spécifie pas le contenu des données échangées.

Champ d'application Machines et Installations Industrielles pour les ateliers de carrosserie-montage.

Emetteur 65940 - Ingénierie Automatismes et Robotique

Confidentialité Non confidentiel

<i>Approuvé par</i>	<i>Fonction</i>	<i>Signature</i>	<i>Date d'application</i>
A. Jouaire	Chef du service 65940		12/2009



EB03.64.210 /B

Historique des versions

Version	Mise à jour	Objet des principales modifications	Rédacteur
A	02/2000	Création (avec VAL CAL =4 sur points de lancement)	M.AUBRY(1)
B	12/2009	Ajout point Y (chapitre 9) Les modifications sont repérés par un trait dans la marge	M.SUCHAREAU

Remplace EB03.64.210 / A du 02/2000

Mise à disposition En Interne Renault, sur Intranet : http://odxpeql_ava.tor.renault.fr
En externe Renault, sur Internet : www.cnomo.com
E-mail : norminfo.moyens@renault.com

Documents cités Réglementation :
International :
Européen :
Français :
CNOMO :
Renault :
Autres doc Internes :
Autres doc externes :

Codification ICS : 25.040.01

Classe E03

Mots-clés Automate programmable, informatique, pilotage, traitement de l'information, programmable controller, Information processing data processing, control.

Langue Français

(1) Ont collaboré à la rédaction du document

Site	Service	Nom	Site	Service	Nom
DDIV/DPSI	65940	G.CLEMENT-GRANDCOURT			
DDIV/DPSI	65940	R.MARCHAIS			
DOII	12060	LOISEAU			



Sommaire

	Page
1	Principes généraux d'un bloc d'échange 4
1.1	Echange standard simple trame (PSFV, SIPTOL, SIPTK) 4
1.2	Echange double trames (PSFV) 5
2	Définition simplifiée des typologies de points 6
2.1	Points de passage 6
2.2	Points de lancement ou de recalage 6
3	Définition des types de gestion 7
3.1	Gestion "marche normale" 7
3.2	Gestion "attente" 7
3.3	Gestion "marche dégradée ou forcée" 7
3.4	Gestion "anomalie" 8
3.5	Gestion "double dialogue" 8
4	Tableaux synthétiques des différents cas de figure 9
4.1	Analyse du mot de validation API 9
4.2	Analyse du mot de validation CALCULATEUR 10
4.3	Utilisation du MOT de VALIDATION par TYPE DE POINTS 11
5	Principes des échanges pour un point de passage 12
5.1	Gestion "normale" 12
5.2	Gestion "attente" 12
5.3	Gestion "dégradée ou forcée" 13
5.4	Gestion "anomalie" 20
5.5	Gestion "double dialogue" 24
6	Principes des échanges pour un point d'autorisation de passage 25
6.1	Gestion "normale" 25
6.2	Gestion "attente" 26
6.3	Gestion "dégradée ou forcée" 27
6.4	Gestion "anomalie" 30
6.5	Gestion "double dialogue" 34
7	Principe des échanges pour un point de lancement 35
7.1	Gestion "normale" 35
7.2	Gestion "attente" 36
7.3	Gestion "dégradée ou forcée" 36
7.4	Gestion "anomalie" 39
7.5	Gestion "double dialogue" 41
8	Principes des échanges pour un point de recalage (SIPTOL) 42
8.1	Gestion "normale" 42
8.2	Gestion "attente" 43
8.3	Gestion "dégradée ou forcée" 44
8.4	Gestion "anomalie" 48
8.5	Gestion "double dialogue" 48
9	Principes des échanges pour un point Y (regroupement de flux) (SIPTOL) 49
9.1	Gestion "normale" 49
9.2	Gestion "attente" 50
9.3	Gestion "dégradée ou forcée" 51
9.4	Gestion "anomalie" 55
9.5	Gestion "double dialogue" 55

Avant-propos

Cette norme, qui remplace le document E530 53449, reprend la totalité des principes spécifiés dans ce document. Il existe une compatibilité ascendante entre cette norme et l'ancien document.

Cette norme spécifie, en plus :

- la gestion « marche forcée » pour un point de lancement,
- la gestion « anomalies » pour un point de lancement.

1 Principes généraux d'un bloc d'échange

1.1 Echange standard simple trame (PSFV, SIPTOL, SIPTK)

Un bloc d'échange se compose d'un ensemble de données encadrées par 2 mots de validation.

Ces 2 mots de validation sont :

- Validation automate appelé VAL API situé avant la zone des données
- Validation calculateur appelé VAL CAL situé après la zone des données

Pour des raisons techniques d'échanges, il est également créé :

- 1 mot juste avant le mot VAL CAL. Ce mot (numéro de requête) est réservé pour le calculateur. Il sert au contrôle écriture par le calculateur.
- 1 mot juste après VAL CAL. Ce mot (Mode Local) est uniquement écrit par l'automate. Il définit (mise à 1) le passage en Mode Local de l'automate.

VAL API
Informations de la pièce (ou caisse) qui est présente ou attendue et informations d'états ou de flux
Informations de la pièce (ou caisse) suivante (éventuellement)
Définition du véhicule (éventuellement)
mot réservé au calculateur (n° requête)
VAL CAL
Mode Local

La structure VAL API + Données + VAL CAL permet de faire des écritures en une seule fois, des consultations en une seule lecture en s'affranchissant de tous problèmes d'aléas (en particulier ceux liés à la technologie des automates programmables).

Néanmoins, il est impératif que le mot VAL API soit le dernier écrit par le programme automate :

- en fonctionnement normal
- en cas de forçage des données par un technicien de maintenance.

1.2 Echange double trames (PSFV)

Les 2 blocs d'échanges se composent d'un ensemble de données

Le premier bloc est encadré par 2 mots de validation.

Ces 2 mots de validation sont :

- Validation automate appelé VAL API situé avant la première zone de données
- Validation calculateur appelé VAL CAL situé après la première zone de données

Pour des raisons techniques d'échanges, il est également créé :

- 1 mot juste après VAL CAL. Ce mot (Mode Local) est uniquement écrit par l'automate. Il définit (mise à 1) le passage en Mode Local de l'automate.

Première TRAME

VAL API
Informations de la pièce (ou caisse)
PJl
mot réservé au calculateur
Enchaînement
Seconde Trame
Informations de la pièce (ou caisse) suivante (éventuellement)
VAL CAL
Mode Local

Deuxième TRAME

Informations de la pièce (ou caisse)
N° d'enchaînement

Les données VAL API et VAL CAL sont écrites dans l'adressage de la première trame après que les 2 trames aient été écrites. Donc les données sont stables lorsque les VAL API et VAL CAL basculent.

Néanmoins, il est impératif que le mot VAL API soit le dernier écrit par le programme automate :

- en fonctionnement normal
- en cas de forçage des données par un technicien de maintenance.

Le deuxième bloc contient que des données associées au véhicule, le dernier mot (N° d'enchaînement) doit être testé par l'API pour contrôler l'association des 2 trames.

ESSENTIEL :

C'est le changement d'état des mots de validation qui conditionnent et entraînent l'évolution du cycle d'automatisme et pas d'évolution quelconque d'une valeur dans la zone de données.

Les mots VAL API et VAL CAL renseignent sur l'état du dialogue automate/calculateur.

La visualisation de ces 2 mots indique rapidement à toute personne de maintenance l'état du dialogue, son type et son enchaînement.

Cette information étant essentielle pour le diagnostic de toute anomalie dans l'échange.



2 Définition simplifiée des typologies de points

Les définitions présentées sont des définitions simplifiées. Des documents spécifiques propres à chaque métier définissent plus précisément les principes de ces points de gestion flux.

2.1 Points de passage

Dans ce type de point, c'est l'automate qui est maître de l'échange.

Point de passage simple

Point permettant au calculateur de connaître les pièces ou les caisses qui arrivent à ce niveau de fabrication.

Point de passage orientation

Point permettant au calculateur de connaître les pièces ou les caisses qui arrivent à ce niveau de fabrication et de les orienter dans le flux principal (ou l'un des flux principaux) ou dans un flux secondaire en fonction de la définition de la caisse (film 1 ou film 2) de sa destination (CKD, contrôle, état de la caisse, ...)

Point d'autorisation de passage

Point autorisant la pièce ou la caisse à passer quand il existe au moins une place pour elle en aval. Ce point est surtout utilisé pour la gestion des brins communs.

2.2 Points de lancement ou de recalage

Dans ce type de point, c'est le calculateur qui est maître de l'échange.

Point de lancement

Point permettant de lancer la fabrication de pièces diversifiées en suivant l'ordre donné par le film de fabrication ou par un point de passage dans le flux (dans le cas d'un lancement par encyclage).

Points de recalage par diversité (ou numéro d'ordre) sans mariage

Point de comparaison entre le flux de pièce (ou caisse) diversifié (ou un numéro d'ordre) et l'ordre donné par le film de fabrication ou un point de passage en amont permettant :

- de vérifier le bon déroulement du flux de pièce (ou de caisse)
- de déclencher et/ou valider des entrées/sorties de pièces.

Note : Fonction de définition

Pour les points de passage, lancement, recalage, il est possible d'y adjoindre la fonction de définition. C'est-à-dire que le calculateur fournit à l'automate des informations complémentaires de définition (en général des informations de diversités produits et/ou process, des informations d'identification comme le PJI, ...) pour que celui-ci les écrive dans une étiquette dynamique.



3 Définition des types de gestion

3.1 Gestion "marche normale"

Information fournie par l'automate ou le calculateur de la fin et du bon déroulement de son travail.

3.2 Gestion "attentes"

Information fournie par le calculateur indiquant qu'il fonctionne correctement, mais qu'il n'a rien à proposer à l'automate.

On trouve cette information :

- sur un point de lancement
 - ⇒ le calculateur n'a plus de véhicule à lancer dans son film de référence
 - c'est le cas en tôlerie (film tôlerie)
 - ⇒ ou le pré-enchaînement de l'unité à lancer n'est pas effectué
 - c'est le cas en tôlerie (units pré-enchaînés à un passage caisse par exemple)
 - c'est le cas au montage (atelier de préparation pré-enchaîné au passage de la caisse en sortie tristöck)
- sur un point d'autorisation de passage
 - ⇒ le calculateur gère un calcul d'encours strict
- sur un point de recalage
 - ⇒ le calculateur n'attend plus de pièce (encours amont vide)

3.3 Gestion marche "dégradée ou forcée"

Marche dégradée et marche forcée

Ces 2 cas indiquent que l'automate peut faire évoluer les caisses sans acquit du calculateur

Marche dégradée

L'automate fait partir la caisse au bout d'un certain temps.

La gestion est assurée par l'automate. Celui-ci n'assurant pas de gestion supplémentaire.

Marche forcée

Un opérateur positionne un commutateur sur la position "marche forcée"

Le départ de la caisse est sous la responsabilité de l'opérateur :

- que ce soit pour déclarer sur un pupitre des informations process (code teinte, ...),
- que ce soit pour une gestion des encours.

L'automate n'assurant pas, en automatique, de gestion supplémentaire.

Fonctions à remplir par le protocole d'échange

Le calculateur doit être informé :

- qu'au moins 1 caisse est passée sans acquit calculateur,
- que le commutateur est en position "marche forcée",
- Le calculateur doit entreprendre au bon moment des procédures de reprise,
- Pas de désynchronisation (mots de VALIDATIONS incohérents).

IMPORTANT :

La gestion des informations de "marches dégradées ou forcées" avec utilisation d'une information "Mode Local" et du mot "VAL API = 3" ne doit pas être systématisée. Elle ne doit être utilisée que lorsque le calculateur a réellement besoin de savoir que des caisses sont passées en son absence.



3.4 Gestion "anomalie"

3.4.1 Anomalie détectée par le calculateur

Si on veut corriger l'anomalie

Information fournie par le calculateur qui constate que l'information d'identification du véhicule (ou son support, ou le numéro d'ordre, ou ...) est incorrecte (hors limites, inconnues par le calculateur) et ne lui permet pas de répondre.

Cette procédure est uniquement utilisée lorsque l'on veut corriger l'anomalie et qu'une intervention manuelle est demandée.

Si on ne veut pas corriger l'anomalie

Si le calculateur sait répondre ou si sa réponse est facultative pour l'automate (point de passage par exemple), alors dans ce cas le calculateur constate l'anomalie, édite un message et répond VAL CAL = 1.

Si la caisse ne peut continuer sans être correctement identifiée (pour des besoins process ou calculateur) et qu'on ne veut pas immobiliser la caisse (jusqu'à l'intervention manuelle), alors dans ce cas, le calculateur constate l'anomalie, édite un message et écrit une information dans le mot "orientation flux" du bloc d'échange et répond VAL CAL = 1.

3.4.2 Anomalie signalée par l'automate

Information fournie par l'automate qui constate que des informations d'identification du véhicule sont incorrectes (car son poste de lecture de l'étiquette d'identification est en panne, par exemple). Cette information permet au calculateur de demander (par exemple) à un opérateur d'effectuer cette lecture manuellement et de renseigner ces informations sur une console liée au calculateur.

IMPORTANT :

Une analyse au cas par cas définit si l'on doit corriger l'anomalie ou non.

Par exemple, sur un point de passage simple ne délimitant pas un encours strict, il n'est pas indispensable de vouloir corriger l'anomalie. (Voir paragraphes 5.4, et 6.4).

3.5 Gestion "double dialogue"

Information complémentaire à la gestion "marche normale" ou "anomalie".

Elle est fournie par l'automate et/ou le calculateur et renseigne de la fin et du bon déroulement de son deuxième cycle de dialogue.

On trouve cette information :

- sur une gestion d'anomalie (dans certains cas)
- sur un point de passage nécessitant un ordre du calculateur suivi d'un acquit de l'automate.

Par exemple, à l'entrée d'un tristock en ligne

- l'automate informe de l'arrivée et l'identification d'une caisse
- le calculateur donne un numéro de ligne de destination
- l'automate confirme (ou infirme) la case demandée (double dialogue API)
- le calculateur enregistre (double dialogue calculateur).