



Universidad de Oviedo

Memoria del Trabajo Fin de Máster realizado por

IKER PLAZAOLA ORMAZABAL

para la obtención del título de

Máster en Ingeniería de Automatización e Informática Industrial

Análisis Arquitectura HW y Plataformas de Programación

FECHA DE PRESENTACIÓN: Enero del 2015

ÍNDICE

ÍNDICE ILUSTRACIONES.....	4
1 INTRODUCCIÓN.....	5
1.1 IDENTIFICACIÓN DEL PROYECTO.....	5
1.2 DESCRIPCIÓN DEL DOCUMENTO	5
2 GRUPO ULMA	6
2.1 ESTRUCTURA ORGANIZATIVA.....	7
2.1.1 ULMA Handling Systems.....	7
2.1.1.1 SISTEMAS DE PREPARACIÓN DE PEDIDOS	7
2.1.1.2 ALMACENAMIENTO AUTOMÁTICO.....	8
2.1.1.3 TRANSPORTE Y CLASIFICACIÓN AUTOMÁTICA.....	8
2.1.1.4 FINALES DE LÍNEA O PALETIZACIÓN AUTOMÁTICA.....	8
2.1.1.5 BAGGAGE HANDLING	8
2.2.-ORGANIGRAMA EMPRESARIAL UHS	8
3 CONCEPTO DE AUTOMATIZACIÓN DE ALMACENES.....	9
3.1. ESTRUCTURA DE UN ALMACÉN AUTOMATIZADO	9
4 DESCRIPCIÓN DEL PROYECTO	10
4.1.-OBJETIVOS DEL PROYECTO	10
4.2.-FASES LLEVADAS A CABO DURANTE EL PROYECTO.....	10
5 DESARROLLO DEL PROYECTO	11
5.1.-ESTUDIO DE LA NORMA IEC 61131	11
5.2.- INICIACIÓN A Programación Orientada a Objeto.....	12
5.2.1.-INTRODUCCIÓN A LA TÉCNICA	12
5.2.2.-CLASES EN POO	12
5.2.2.1.-PROPIEDADES DE LA CLASE	12
5.2.2.2.-MÉTODOS EN LAS CLASES.....	12
5.2.3.-OBJETOS EN POO	12
5.3.- RESULTADOS OBTENIDOS CON PROGRAMACION ORIENTADA A OBJETO EN CODESYS	13
5.4.- IMPLEMENTACIÓN Y SIMULACIÓN DE LA PROPUESTA	14
5.5.- ANÁLISIS DE DIFERENTES BUSES DE CAMPO Y ARQUITECTURAS HARDWARE	15
5.5.1.- CUADRO DE PRECIOS.....	18
5.5.2.- CÁLCULO DEL TIEMPO DE CICLO.....	19
5.5.2.1- SIEMENS.....	19
5.5.2.2- BECKHOFF.....	20
5.5.2.3- OMRON.....	21
6 DOCUMENTO DEL PROYECTO.....	22

7 CONCLUSIONES.....	23
8 LÍNEAS FUTURAS	24
8.1.- ¿QUÉ TRADUCIR A ST?	24
8.2.- ¿DISEÑO DE LA ARQUITECTURA?	24
9 BIBLIOGRAFÍA.....	25

ÍNDICE ILUSTRACIONES

<i>Ilustración 1: GRUPO ULMA por todo el mundo</i>	6
<i>Ilustración 2: Evolución de trabajadores</i>	6
<i>Ilustración 3: Porcentaje de ventas</i>	6
<i>Ilustración 4: Estructura Organizativa</i>	7
<i>Ilustración 5: Desglose de la estructura</i>	8
<i>Ilustración 6: Lenguajes IEC 61131</i>	11
<i>Ilustración 7: Aplicar herencia</i>	13
<i>Ilustración 8: Estructura jerárquica con o sin polimorfismo</i>	13
<i>Ilustración 9: Visualización modo online SIEMENS</i>	15
<i>Ilustración 10: Diseño arquitectura 1</i>	16
<i>Ilustración 11: Diseño arquitectura 2</i>	17
<i>Ilustración 12: Cuadro precios CPU SIEMENS</i>	18
<i>Ilustración 13: Cuadro precios OMRON</i>	18
<i>Ilustración 14: Cuadro precios Beckhoff</i>	19
<i>Ilustración 15: Tiempo de ciclo SIMENS</i>	19
<i>Ilustración 16: Tiempo de ciclo Beckhoff 1</i>	20
<i>Ilustración 17: Tiempo de ciclo Beckhoff 2</i>	20
<i>Ilustración 18: Tiempo de ciclo OMRON</i>	21

INTRODUCCIÓN

1 INTRODUCCIÓN

1.1 IDENTIFICACIÓN DEL PROYECTO

TÍTULO: Análisis de diferentes estructuras HW y de lenguajes de programación basados en el estándar IEC61131.

AUTOR: Iker Plazaola Ormazabal

FECHA: Julio de 2014

FINANCIACIÓN: Proyecto financiado por la empresa ULMA Handling Systems.

1.2 DESCRIPCIÓN DEL DOCUMENTO

El principal objetivo de este documento es el dar a conocer a los lectores los detalles del proyecto desarrollado por el alumno Iker Plazaola. Explicando cuales han sido los motivos para llevar a cabo este proyecto, detallando todos los pasos que se dieron para realizar dicho trabajo y sobre todo mostrando el alcance de este.

2 GRUPO ULMA

El grupo ULMA tiene sus orígenes en Oñati, una población pequeñita de 11000 habitantes en el corazón del País Vasco. Todo empezó cuando varias personas de dicha localidad dejaron sus empleos para embarcarse en un proyecto que, hoy en día, se ha convertido en uno de los Grupos Empresariales más importantes del norte de España, con filiales en los cinco Continentes.

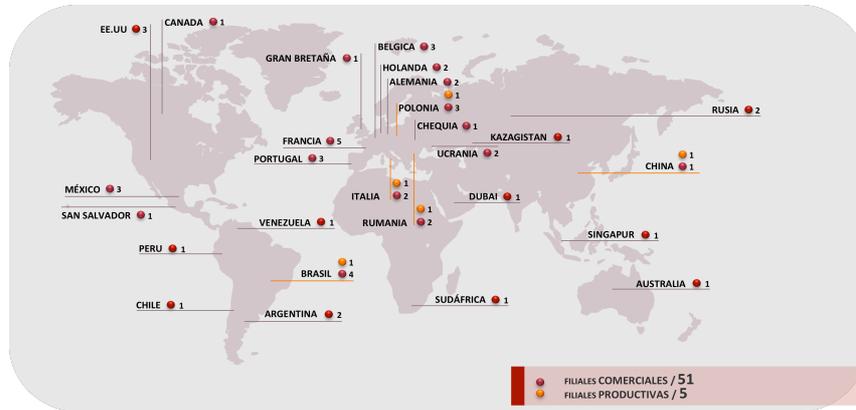


Ilustración 1: GRUPO ULMA por todo el mundo

A pesar de la situación de inestabilidad económica que vivimos a nivel mundial, El Grupo ULMA viene presentando unos resultados razonablemente positivos. Las ventas del Grupo alcanzaron en el ejercicio 2013 cerca de 679 millones de euros lo que es una clara muestra de la buena marcha del Grupo cuya estrategia de diversificación sigue siendo uno de sus principales activos. Del mismo modo, al cierre del ejercicio 2013 el empleo en el Grupo ULMA ascendía a 4.284 personas.

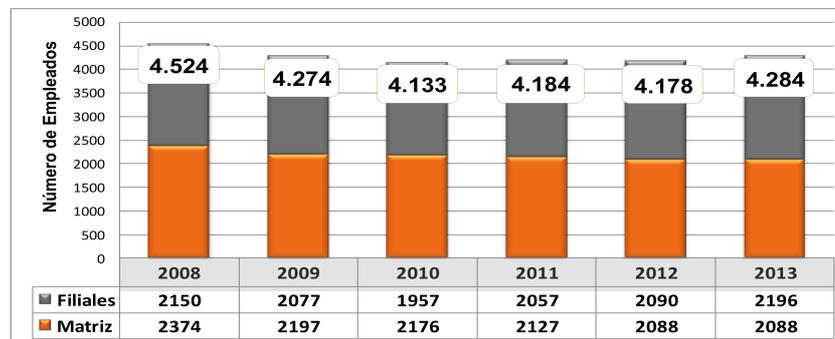


Ilustración 2: Evolución de trabajadores

Repartiendo sus ventas por todo el mundo:

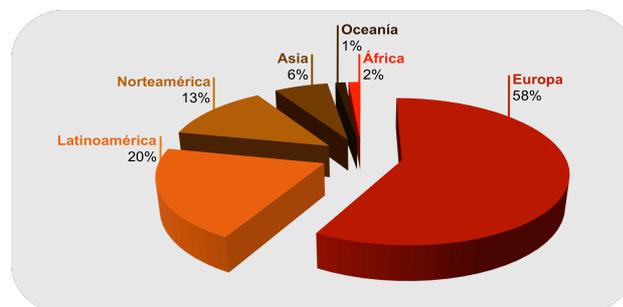


Ilustración 3: Porcentaje de ventas

2.1 ESTRUCTURA ORGANIZATIVA

La estructura organizativa del Grupo ULMA se divide en 8 líneas de negocios diferentes:



Ilustración 4: Estructura Organizativa

- **ULMA Agrícola:** Fabrica y comercializa estructuras y sistemas para cultivo protegido bajo cubierta.
- **ULMA Carretillas Elevadoras:** Ofrece una completa e innovadora gama de carretillas elevadoras diesel, gas o eléctricas diseñadas para aplicaciones de cargas tanto en entornos de interior como de exterior.
- **ULMA Construcción:** ofrece servicios y soluciones industrializadas de encofrados y andamios para identificación, obra civil y rehabilitación.
- **ULMA Conveyor Components:** Fabrica y comercializa componentes especializados (rodillos, tambores, guirnaldas y soportes) para una amplia gama de industrias.
- **ULMA Piping:** Fabrica y comercializa accesorios forjados para tuberías, bridas y accesorios para las industrias del petróleo y la energía.
- **ULMA Hormigón Polímero:** Diseña, fabrica y comercializa sistemas prefabricados en hormigón para el sector de la construcción.
- **ULMA Packaging:** Diseña y fabrica equipos, sistemas y servicios de envasado que proporciona a los productos de sus clientes una protección y presentación de alta calidad.
- **ULMA Handling Systems:** Ofrece soluciones integrales de logística mediante el diseño, desarrollo, montaje y mantenimiento de sistemas automáticos para logística.

2.1.1 ULMA Handling Systems

Como este proyecto se ha desarrollado para la empresa ULMA Handling Systems, a continuación se mencionaran las diferentes soluciones que ofrecen.

2.1.1.1 SISTEMAS DE PREPARACIÓN DE PEDIDOS

El Digital Picking System es una solución que permite al operario conocer de forma rápida e intuitiva la ubicación y cantidad exacta de la operación a llevar a cabo a través de LED luminosos y displays, mostrando la cantidad exacta demandada. Solución idónea para evitar búsquedas de referencias ilocalizables.

2.1.1.2 ALMACENAMIENTO AUTOMÁTICO.

La eficiencia de un almacén puede medirse juzgando cómo de bien se fusionan la función de preparación de pedidos con el envío de material a los muelles de expedición.

2.1.1.3 TRANSPORTE Y CLASIFICACIÓN AUTOMÁTICA

Los sistemas logísticos de Transporte y Clasificación Automática incrementan notablemente la productividad, eliminando mano de obra y manteniendo la cadencia y flujos necesarios en los procesos que van desde la producción hasta la preparación de pedidos.

2.1.1.4 FINALES DE LÍNEA O PALETIZACIÓN AUTOMÁTICA

ULMA Handling Systems ofrece Soluciones Fin de Línea integrando todos los subprocesos involucrados, entre los que podemos destacar: soluciones de paletización, encajonado, enfardado o flejado, etiquetado o transporte.

2.1.1.5 BAGGAGE HANDLING

Se diseñan y se desarrollan sistemas integrales de tratamiento de equipajes. UHS cuenta con una amplia experiencia en el sector y una extensa gama de productos y soluciones modulares que se ajustan a las necesidades funcionales de cualquier aeropuerto.

2.2. -ORGANIGRAMA EMPRESARIAL UHS

En el siguiente organigrama se muestra la estructura empresarial de ULMA Handling Systems para llevar a cabo todas estas soluciones:

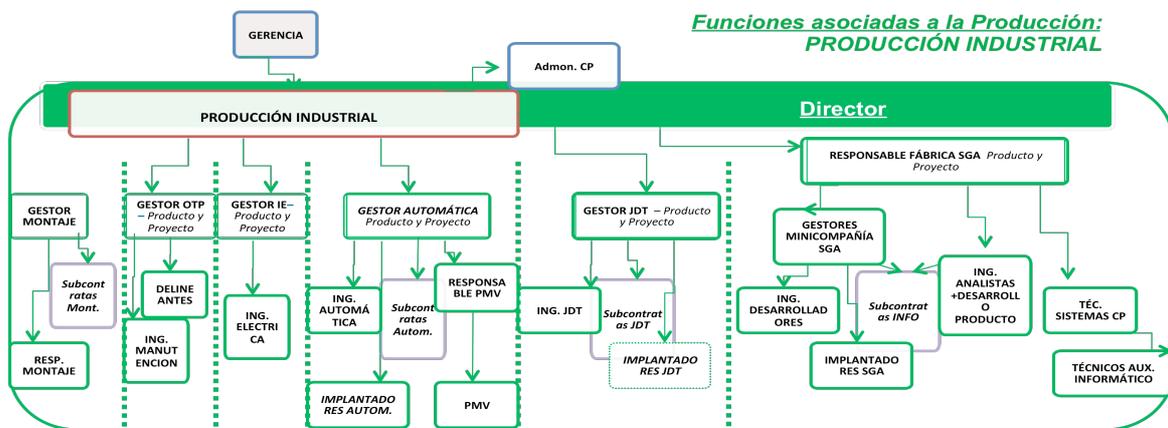


Ilustración 5: Desglose de la estructura

- Montaje→ Como su nombre indica son los encargados del montaje de la instalación.
- OTP→ Son los encargados del diseño de la solución teniendo en cuenta los requerimientos del cliente.
- IE→ Son los encargados de diseñar todo la parte eléctrica de la instalación.
- Automática→ Son los encargados de programar además los PLCs y sus respectivos elementos de campo, los robots y vehículos inteligentes.
- JDT→ El Jefe De Tráfico es el intermediario entre el PLC y el SGA se encarga en traducir al lenguaje adecuado para que cada uno entienda los mensajes del otro.
- SGA→ Sistema de Gestión de Almacenes como su nombre indica se dedican a gestionar mediante diferentes bases de datos los productos del almacén.

Por lo tanto, en UHS se lleva a cabo una organización en departamentos horizontal y totalmente dirigidas hacia el cliente. Creando grupos de trabajo enfocados a proyectos. Se dividen miembros de cada departamento en distintos proyectos y el responsable del proyecto organizara los recursos que tenga a su disposición.

3 CONCEPTO DE AUTOMATIZACIÓN DE ALMACENES

En términos generales se puede definir la automatización de almacenes como la técnica que hace posible que un producto cumpla con sus diversas fases de tratamiento antes de ser almacenado o expedido y todo esto de manera automatizada.

Estos tipos de almacenes están compuestos de diferentes elementos: sensores, accionamientos, finales de carrera, lectores de códigos, gálbos, etc. Es por tanto necesario una buena comunicación entre los diferentes elementos de campo, principalmente este es el mayor problema a la hora de montar una instalación de este tipo, ya que, es de vital importancia que el bus de campo escogido para la comunicación cumpla con los tiempos requeridos, porque de no ser así podríamos perder datos, afectando de manera directa al proceso.

Las arquitecturas pueden ser centralizadas o distribuidas dependiendo de la magnitud de la instalación. Hoy en día se apuesta más por la arquitectura descentralizada, facilitando la instalación de elementos a los integradores además de un ahorro sustancial en el cableado.

Los almacenes automáticos, por lo tanto, son una solución para aquellas empresas que manejen un gran número de pedidos y estén interesados en que el almacenamiento y expedición de estas sea más rápida, reduciendo así la mano de obra. La automatización de almacenes es muy ventajosa en:

- Ahorro de tiempo de producción
- Ahorro de personal
- Aumento de contratos de personal más especializado
- Trabajo menos físico
- Precio de productos más competitivos

3.1. ESTRUCTURA DE UN ALMACÉN AUTOMATIZADO

En general un almacén automatizado se divide en tres partes, PLC, JDT y SGA. El cometido del PLC es el de controlar que todos los accionamientos y controladores funcionen de manera adecuada, además de gestionar todas las seguridades de la instalación. También será el encargado de transportar las cargas según las necesidades marcadas por las capas superiores.

El JDT o Jefe De Tráfico es el encargado de que toda la información compartida entre el PLC y el SGA llegue a sus destinatarios. Para ello ULMA creó su propia agencia de comunicación. Además de comunicar esta agencia crea las rutas o camino que deben seguir los palets o cajas.

Y por último tenemos el SGA o Sistema de Gestión de Almacenes, gracias a este grupo podemos tener toda la información del almacén en una base de datos. Gracias a esta base de datos el cliente sabe en todo momento el stock del material que tiene en el almacén y puede hacer un seguimiento de cómo van sus ventas.

Pero hace dos años se decidió dar un cambio en la estructura. Ya que hasta ese momento el PLC era el encargado de la movimentación, y en cada posición de carga preguntaba a las capas superiores cual era el próximo destino. Pero debido a este cambio, ahora el PLC dentro del paquete de información que arrastra con cada caja o palet también llevará la ruta de la caja o palet. Así este sistema reduce la carga de comunicaciones entre capas además de facilitar la resolución de incidencias.

DESCRIPCIÓN DEL PROYECTO

4 DESCRIPCIÓN DEL PROYECTO

Este proyecto es el reflejo de la gran evolución que está sufriendo la automatización en los últimos años. Es más, en los últimos años las marcas que comercializan con PLCs han aumentado de manera exponencial, es tan amplio el mercado actual que las empresas deben hacer grandes trabajos de ingeniería, reinventándose cada poco tiempo.

Es por ello que ULMA ha optado por seguir creciendo y ha decidido mejorar su manera de programar haciéndolo más exportable, ya que, hasta ahora por exigencias del mercado la mayoría de las soluciones se ofertaban para la plataforma de SIEMENS. En gran medida, esta reforma se debe a la actualización de la plataforma de SIEMENS que utilizaban (Simatic STEP 7) y a las diversas peticiones que últimamente han ido haciendo los clientes para automatizar sus instalaciones con otras marcas de PLCs, por lo que se ha decidido hacer un estudio para ver cuánto supondría migrar todas las funciones a una plataforma cien por cien exportable.

4.1. -OBJETIVOS DEL PROYECTO

Este proyecto está orientado a desarrollar una solución que sea exportable a cualquier plataforma sin que nos importe que marca sea. Una vez se tenga la solución ULMA podrá ofertar una amplia gama de productos sin modificar el programa.

Y como las funciones de mesas, funciones para las pantallas y la mensajería no estarán ligadas a ninguna plataforma el mantenimiento de estas funciones será mucho más sencilla.

Con este cambio también se pretende simular cualquier tipo de programa sin que afecte en que plataforma esté programado, ya que, hasta ahora solamente se podía simular en un simulador creado por ULMA para SIEMENS.

4.2. -FASES LLEVADAS A CABO DURANTE EL PROYECTO

La ejecución del proyecto se ha dividido en las siguientes fases:

- La primera fase, la del análisis y diseño, consistirá en analizar los diferentes tipos de lenguaje existentes y ver cual se adapta mejor a la plataforma libre de Codesys. Por ello se estudiarán las diferentes alternativas de la norma IEC61131 además de la Programación Orientada a Objeto. Analizando los pros y contras de cada técnica mediante traducciones de funciones.
- La siguiente fase será la de la implementación y simulación, una vez diseñado como será el programa cien por cien exportable para la siguiente fase se ha decidido implementar un programa y migrarlo a diferentes plataformas. También se simulará en cada plataforma para ver que todo funciona correctamente.
- Y por último tendremos la fase del estudio de buses y arquitecturas, este apartado viene relacionado con los anteriores porque es la consecuencia de lo estudiado anteriormente. Una vez tengamos el programa en un lenguaje estándar y exportable, se aprovechará para analizar y diseñar que bus de campo y arquitectura se adapta mejor a las nuevas exigencias del mercado.

5 DESARROLLO DEL PROYECTO

En este apartado se resumirán los pasos que se dieron en cada fase sin entrar en detalles técnicos. Todo el desarrollo técnico lo podrán encontrar en el Manual del programador y en el instalador.

5.1. -ESTUDIO DE LA NORMA IEC 61131

Para la realización de la primera fase se estudió la norma IEC 61131 y los lenguajes que lo componen, ya que es un lenguaje unificado en el que se basan todas las plataformas de programación de PLCs.

Esta normativa está dividida en 5 apartados:

1. Vista general
2. Hardware
3. Lenguaje de programación
4. Guías de usuario
5. Comunicación

Por lo tanto la normativa que se estudiará será la tercera, IEC 61131-3.

El tercer apartado de esta normativa define cuatro lenguajes de programación, por tanto estos cuatro lenguajes no entienden de dialectos y como todos hablan el mismo idioma se podrán utilizar en más de una plataforma sin problemas.

Estos cuatro lenguajes se agrupan en dos tipos principalmente. Por un lado tenemos los lenguajes que son de tipo de texto como *Lista de Instrucciones (IL)* y *Texto Estructurado (ST)*. Y por otro, los de tipo gráfico que son *Diagrama de Contactos (LD)* y *Diagrama de Bloques Funcionales (FBD)*.

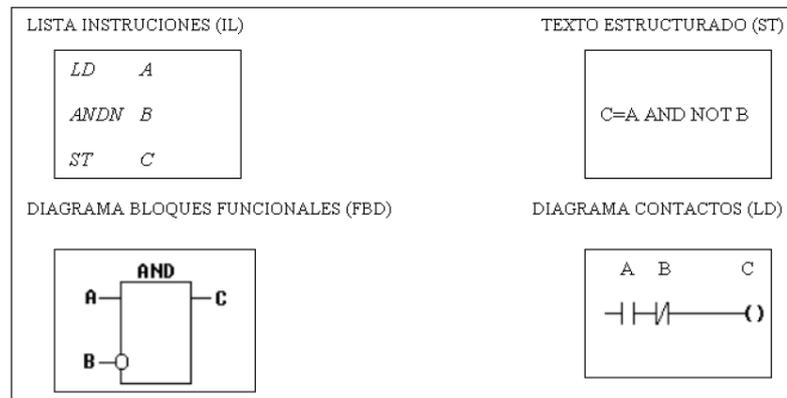


Ilustración 6: Lenguajes IEC 61131

Los cuatro lenguajes al estar interrelacionados, permiten programar combinando diferentes lenguajes en un mismo programa sin que esto suponga un motivo de conflicto.

DESARROLLO DEL PROYECTO

Una vez analizados los diferentes lenguajes que componen el estándar IEC61131, el siguiente paso será estudiar que plataforma es la que mejor se adapta a dicha normativa. Por ello la primera plataforma que se analizó fue la de Codesys, en su portal de internet vi que además de ser compatible cien por cien con la normativa IEC61131 también ofrecían la posibilidad de programar con la Programación Orientada a Objeto.

Aunque este estándar no reconozca la programación orientada a objeto como uno de los lenguajes unificados, debido a que en Codesys están apostando por esta técnica y como una de las asignaturas del máster estaba orientada al estudio de esta técnica, me pareció una buena idea aplicarla.

Por lo tanto, antes de empezar a programar se explicará en que consiste esta técnica, mencionando sus diferentes variantes.

5.2.- INICIACIÓN A Programación Orientada a Objeto

5.2.1.-INTRODUCCIÓN A LA TÉCNICA

La principal ventaja de esta técnica es que una vez se tenga definida la clase podemos crear instancias u objetos de cada clase todas las veces que se quiera, esta es la misma idea que se aplica con la técnica de Diagrama de Bloques Funcionales o FBD. Ya que, con el diagrama de bloques cada vez que llamemos a un bloque lo debemos instanciar como si fuera un objeto.

Pero esta técnica ofrece otra ventaja y es que a cada objeto se le asocian unas propiedades o atributos y métodos o acciones, estas propiedades serán específicas de cada clase, en cambio, los métodos se podrán compartir entre una clase y otra.

5.2.2.-CLASES EN POO

Las clases son declaraciones de objetos, también se podrían definir como abstracciones de objetos. Esto quiere decir que la definición de un objeto es la clase. Cuando se programe un objeto y definan sus características y funcionalidades en realidad lo que se está haciendo es programar una clase.

5.2.2.1.-PROPIEDADES DE LA CLASE

Las propiedades o atributos son las características de los objetos. A la hora de definir una propiedad normalmente se especifica su nombre y su tipo. Nos podemos hacer a la idea de que las propiedades son algo así como variables donde almacenamos datos relacionados con los objetos.

5.2.2.2.-MÉTODOS EN LAS CLASES

Los objetos son ejemplares de una clase cualquiera. Cuando uno programa las clases las llama métodos. Los métodos son como funciones que están asociadas a un objeto.

5.2.3.-OBJETOS EN POO

Los objetos son ejemplares de una clase cualquiera. Cuando se crea un ejemplar se tiene que especificar la clase a partir de la cual se creará. Esta acción de crear un objeto a partir de una clase se llama instanciar (que viene de una mala traducción de la palabra instance que en inglés significa ejemplar). Por ejemplo, un objeto de la clase coche es por ejemplo opel astra rojo.

También existen mecanismos como la herencia y el polimorfismo que son una de las posibilidades más potentes de la POO. El polimorfismo y la herencia son dos conceptos estrechamente ligados.

La herencia sirve para crear objetos que incorporen propiedades y métodos de otros objetos. Así podremos construir unos objetos a partir de otros sin tener que reescribirlo todo. La clase padre es la que transmite su código a las clases hijas, en muchos lenguajes de programación se declara la herencia con la palabra "extends".

```
Class COCHE extends VEHICULO {}
```

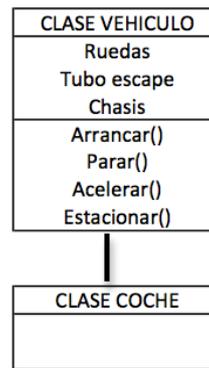


Ilustración 7: Aplicar herencia

El polimorfismo consiste en que un objeto de una clase se comporte como un objeto de cualquiera de sus subclases. O dicho de otra manera, el polimorfismo se da cuando se utiliza la referencia de una clase padre para referirse al objeto de la clase hijo.

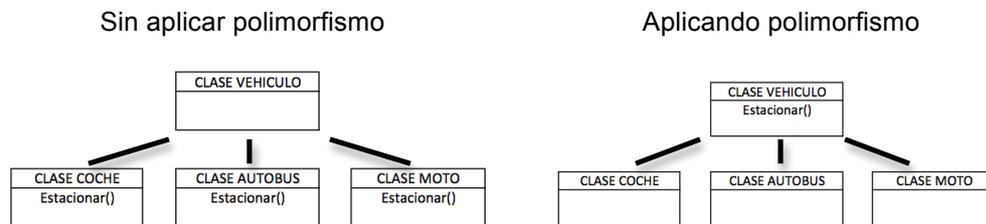


Ilustración 8: Estructura jerárquica con o sin polimorfismo

Y lo programaríamos de la siguiente manera:

Declarar el método: *Función estacionar (vehículo)}*

Llamar a la función: *estacionar (coche);*

estacionar (autobús);

estacionar (moto);

5.3. - RESULTADOS OBTENIDOS CON PROGRAMACION ORIENTADA A OBJETO EN CODESYS

Una vez estudiadas las diferentes formas de programar que existen, para dar una solución a la problemática planteada el siguiente paso fue el de realizar un programa donde se pudieran analizar las ventajas y desventajas que tiene programar con Programación Orientada a Objeto.

Pero como la técnica de Programación Orientada a Objeto es bastante diferente a las técnicas que hasta ahora se han aplicado en ULMA, se decidió traducir un par de funciones en vez de traducir un programa al completo porque con esto es suficiente para extraer conclusiones. En el manual del programador se explica con mayor detalle lo mencionado.

Como se mencionó al inicio del apartado de la Programación Orientada a Objeto la plataforma escogida para dicha traducción fue la de Codesys, ya que, las otras plataformas como TIA Portal aún no han incorporado la Programación Orientada a Objeto a su gama de lenguajes. Además, de esta manera a la vez que se adaptaban las funciones a la técnica POO se analizaba la plataforma Codesys. Porque como esta es una plataforma que cualquiera se puede descargar desde internet y no hace falta pagar licencias de mantenimiento ni de instalación, puede que sea una buena solución para el desarrollo del proyecto.

DESARROLLO DEL PROYECTO

Pero me encontré con los siguientes problemas:

- Después de muchos días conseguí compilarlo sin errores, pero no pude simular el programa al completo por falta de paquetes.
- Tampoco pude realizar una función donde obtenía la hora del sistema.
- Y por último me di cuenta que no tenía sentido seguir programando con Programación Orientada a Objeto, ya que, no estaba compartiendo los diferentes métodos.

Debido a todas estas circunstancias, no pude seguir adelante con la programación. Me vi falto de colaboradores, el único soporte que ofrecen los de Codesys es el de los manuales que tienen para descargar desde su página, además me di cuenta de que los resultados que iba obteniendo no eran suficientes como para seguir dedicando tanto tiempo a este tipo de programación.

5.4. - IMPLEMENTACIÓN Y SIMULACIÓN DE LA PROPUESTA

Después del intento fallido con la Programación Orientada a Objeto y visto que Codesys tenía muchas carencias, se continuó con la búsqueda de la plataforma que más se adapte a las necesidades futuras de ULMA. Se recopilaron en la siguiente lista las marcas que más piden los clientes.

- TIA Portal de SIEMENS, esta desde un principio se sabía que era la que menos se acercaba al estándar IEC61131
- Sysmac Studio de OMRON
- Twincat3 de BECKHOFF
- Unity de SCHNEIDER
- Estudio 5000 de ROCKWELL

Por lo tanto, comencé a analizar que plataforma era la que más se acercaba al estándar IEC61131 y después de unas cuantas reuniones y conferencias se decidió que Twincat3 de BECKHOFF era la que mejor se adaptaba a ese estándar de programación. Por lo que, la siguiente fase consistió primero en traducir un programa desde STEP7 a TIA Portal, y a su vez se aprovecho para adaptar el programa a un lenguaje de más alto nivel como es SCL o Texto Estructurado (ST). Ya que, programando en un lenguaje de alto nivel siempre es más fácil migrar de una plataforma a otra.

Una vez traducido y migrado el programa, se simuló para verificar que el simulador actual funcionaba sin problemas con programas escritos en TIA Portal y en lenguaje de alto nivel.

Una vez compilado y simulado el programa, se continuó con la migración a una plataforma cien por cien exportable. Por lo que el programa se migró a la plataforma Twincat3, así pudiendo cuantificar lo que supondría migrar cualquier programa desde SIEMENS a esta plataforma.

Y para finalizar se migró el programa base a otra plataforma diferente de Beckhoff y Siemens, en este caso se migró a Sysmac Studio. Gracias a esta maniobra se pudo realmente comprobar que el programa programado en la plataforma de Beckhoff era cien por cien exportable. Aunque, como queda reflejado en el manual del programador, se tuvieron que modificar algunas funciones, ya que en cada plataforma se programan de manera diferente. Por lo tanto la solución fue crear una librería con todas las funciones y dependiendo de la plataforma se llamará a una función o a otra.

También mencionar que en este caso se tradujo todo el programa previamente escrito en lenguaje de contactos a lenguaje de texto y aunque para migrar sea la mejor opción, a la hora de simular este tipo de lenguajes producen muchos inconvenientes debido a que no son lenguajes nada visuales.

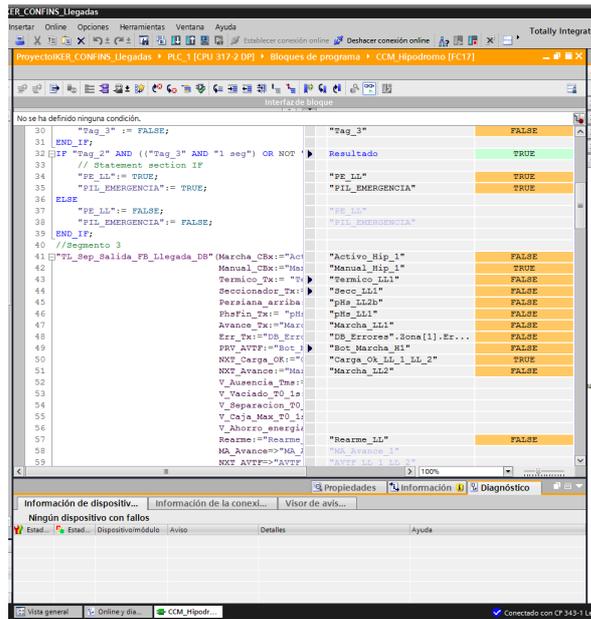


Ilustración 9: Visualización modo online SIEMENS

5.5.- ANÁLISIS DE DIFERENTES BUSES DE CAMPO Y ARQUITECTURAS HARDWARE

Esta fue la última fase realizada en este proyecto. A raíz del cambio del lenguaje de programación, se comenzó a analizar también las innumerables propuestas de buses de campo que ofrecen diferentes empresas del mundo de la automatización (Profinet, Ethernet, Ethercat. Etc.).

Después de varias consultas, finalmente se obtuvieron dos propuestas para el diseño de la arquitectura de la instalación. La primera propuesta está orientada a modular la programación, así tendríamos el programa dividido sin que nos importase la parte de tracking en que plataforma estuviera. Pero el principal inconveniente de esta propuesta es su elevado precio, debido a que se añade una red de comunicaciones que antes no existía entre los diferentes autómatas.

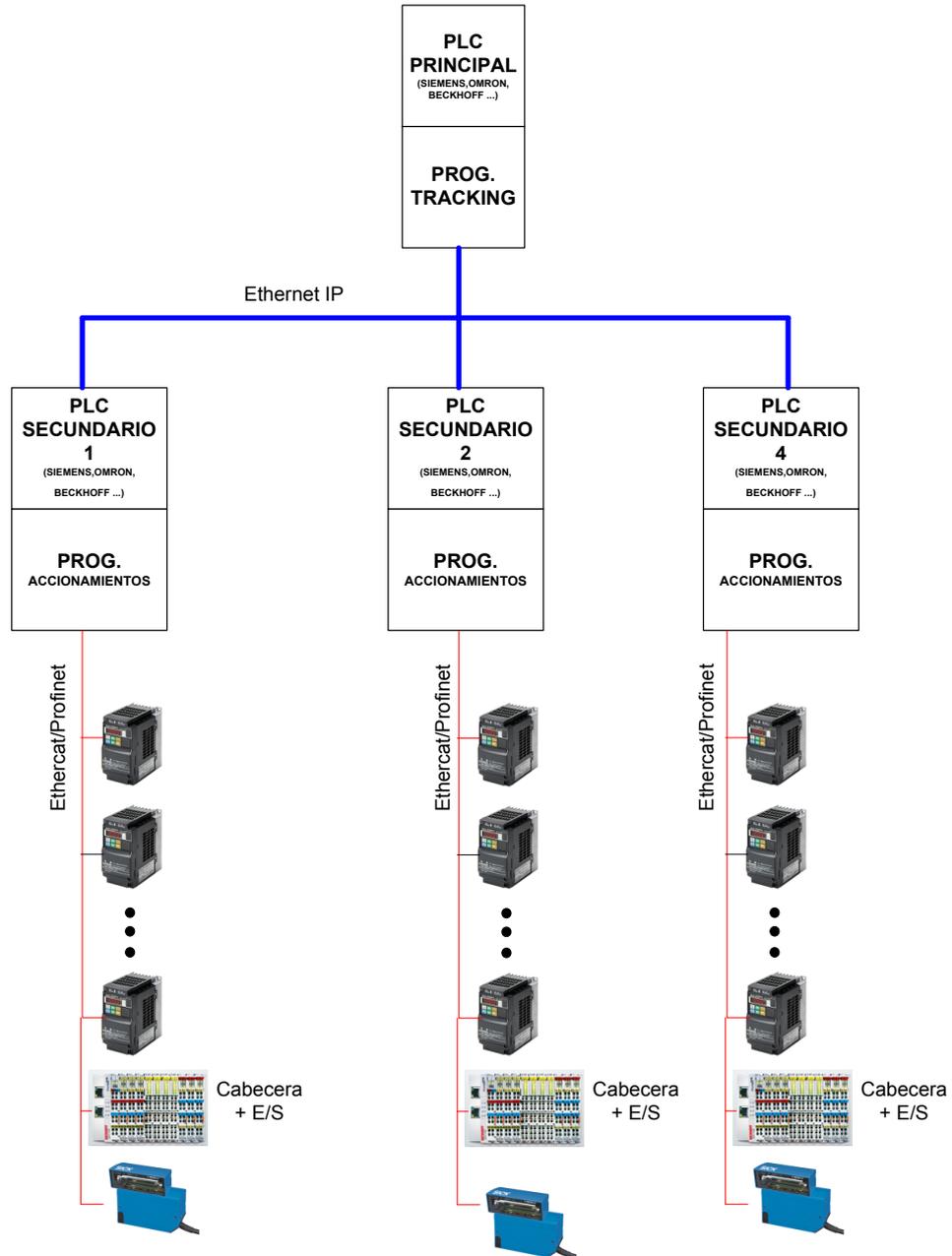


Ilustración 10: Diseño arquitectura 1

La segunda propuesta además de ser la más económica aprovecha la última tecnología de buses y gracias al switch evita que tras posible fallo se corte toda la línea.

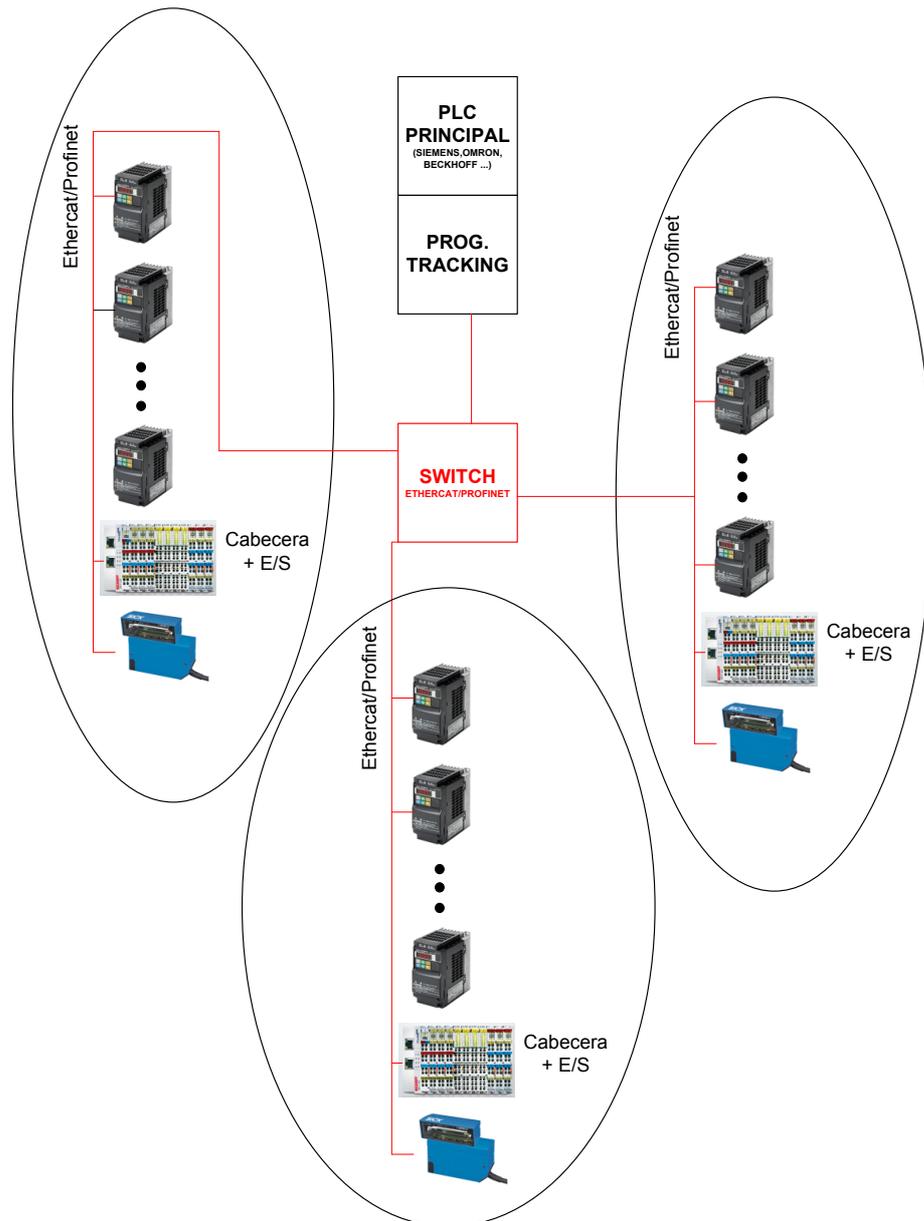


Ilustración 11: Diseño arquitectura 2

Mencionar también que la principal razón de modular la programación fue por la limitación de memoria que teníamos hasta ahora. Pero viendo las prestaciones de los PLCs de hoy en día esto deja de ser un problema. A continuación, se muestra una tabla con algunas de las instalaciones más pesadas diseñadas por ULMA.

	Memoria: Programa	Datos
Núcleo: Tracking(500mesas)+ Routing:	0,256Mb	0,5Mb
SQS con seguridad y Routing:	0,9 Mb	0,6 Mb
Ecenarro con seguridades y tracking:	1,09 Mb	0,4Mb
Silo3 con seguridades y tracking:	0,9 Mb	0,47Mb

5.5.1.- CUADRO DE PRECIOS

Una vez recopilados todos estos datos, se hizo un cuadro con aquellas CPUs que cumplieren con los requisitos y se pregunto por sus precios.

SIEMENS										
Controladores S7 1500	MEMORIA		Memory Card		OPERACIÓN	Controladores S7 300	Programa de trabajo		Precio ULMA	
	Datos	Carga	Carga	Bit			Carga	Bit		
CPU1518F-4PN/DP	4,5MB	10MB	32GB	1ns	6.001,00 €	CPU319F-3PN/DP	2,56MB	8MB	4ns	3.440,29 €
CPU1517F-3PN/DP	3MB	8MB	32GB	2ns	4.131,00 €	CPU317F-2PN/DP	1,5MB	8MB	25ns	3.165,00 €
CPU1516F-3PN/DP	1,5MB	5MB	32GB	10ns	3.196,00 €	CPU317F-2DP	1,5MB	8MB	25ns	2.521,00 €
CPU1515F-2PN	750KB	3MB	32GB	30ns	2.014,50 €	CPU315F-2PN/DP	512KB	8MB	50ns	1.984,00 €
CPU1518-4PN/DP	3MB	10MB	32GB	1ns	5.440,00 €	CPU315F-2DP	384KB	8MB	50ns	1.432,00 €
CPU1516-3PN/DP	1MB	5MB	32GB	10ns	2.890,00 €	CPU319-3PN/DP	2,0MB	8MB	4ns	3.374,00 €
CPU1515-2PN	500KB	3MB	32GB	30ns	1.751,00 €	CPU317-2PN/DP	1MB	8MB	25ns	2.806,00 €
CPU1513-1PN	300KB	1,5MB	32GB	40ns	1.175,00 €	CPU317-2DP	1MB	8MB	25ns	2.471,00 €
CPU1511-1PN	150KB	1MB	32GB	60ns	527,00 €	CPU315-2PN/DP	384KB	8MB	50ns	1.702,00 €
						CPU315-2DP	256KB	8MB	50ns	1.205,30 €
Controladores S7 1200	Programa de trabajo	Carga	Bit	Precio ULMA		CPU314C-2PH	192KB	8MB	60ns	
CPU1217C DC/DC/DC	125KB	4MB	85ns	581,00 €	CPU314C-2PN/DP	192KB	8MB	60ns		
CPU1215C DC/DC/Rly	100KB	4MB	85ns	402,00 €	CPU314C-2DP	192KB	8MB	60ns		
CPU1215C DC/DC/DC	100KB	4MB	85ns	402,00 €	CPU314	128KB	8MB	60ns		
CPU1215C AC/DC/Rly	100KB	4MB	85ns	402,00 €	CPU313C-2PH	128KB	8MB	70ns		
CPU1214C DC/DC/Rly	75KB	4MB	85ns		CPU313C-2DP	128KB	8MB	70ns		
CPU1214C DC/DC/DC	75KB	4MB	85ns		CPU313C	128KB	8MB	100ns		
CPU1214C AC/DC/Rly	75KB	4MB	85ns		CPU312C	64KB	8MB	100ns		
CPU1212C DC/DC/Rly	50KB	1MB	85ns		CPU312	32KB	8MB	100ns		
CPU1212C DC/DC/DC	50KB	1MB	85ns							
CPU1212C AC/DC/Rly	50KB	1MB	85ns							
CPU1211C DC/DC/Rly	25KB	1MB	100ns							
CPU1211C DC/DC/DC	30KB	1MB	85ns							
CPU1211C AC/DC/Rly	30KB	1MB	85ns							

Ilustración 12: Cuadro precios CPU SIEMENS

OMRON						
Controladores NJ	MEMORIA		Memory Card	OPERACIÓN		Precio
	Programa de trabajo	Carga		Bit	Motion control	
NJ0 ETC / EIP	3MB	2,5MB	5ns	0 ejes	855,00 €	
NJ1 ETC / EIP	3MB	2,5MB	5ns	2 ejes	??	
Controladores NJ 3	Programa de trabajo	Carga	Bit	Motion control	Precio	
NJ301-1100	5MB	2,5MB	3ns	4 ejes	??	
NJ301-1200	5MB	2,5MB	3ns	8 ejes	??	
Controladores NJ 5	Programa de trabajo	Carga	Bit	Motion control	Precio	
NJ501-1300	20MB	6MB	1,9ns	16 ejes	??	
NJ501-1400	20MB	6MB	1,9ns	32 ejes	??	
NJ501-1500	20MB	6MB	1,9ns	64 ejes	??	
Controladores CJ	Programa	Datos	Carga	Bit	Precio	
CJ1M	20KB	32KB	??	100ns	340,00 €	
CJ1G-P	60KB	128KB	??	40ns	950,00 €	
CJ2M	60KB	160KB	??	40ns	1.050,00 €	
CJ2H	400KB	832KB	??	16ns	1.875,00 €	

Ilustración 13: Cuadro precios OMRON

En Omron solo se puso el precio de una NJ, ya que con el modelo más pequeño ya es suficiente. También mencionaron que la gama de CJ no la iban a retirar del mercado, ya que es uno de sus productos estrella, pero que tampoco iban a desarrollar ni nuevos modelos ni nuevas versiones. La tendencia que se espera durante estos años es que se migre a los nuevos NJ, porque también se esperan próximamente nuevos modelos NJ sin motion logrando que el producto se ajuste más al mercado Handling.

PROGRAMA de Trabajo										
Controladores CX800	Micro SD	RAM	Procesador	Precio	EQUIVALENCIAS					
CX8010-EtherCAT	512MB, 1, 2 or 4GB	64MB	32 bit,400MHz	299,98 €	SIEMENS	BECKHOFF				
CX8030-PROFIBUS master	512MB, 1, 2 or 4GB	64MB	32bit, 400MHz	425,96 €	PLC	PLC	Seguridad	Comunicaciones	PRECIO	
CX8031-PROFIBUS slave	512MB, 1, 2 or 4GB	64MB	32bit, 400MHz	??	CPU 312	CX8010				299,98 €
CX8050-CANopen master	512MB, 1, 2 or 4GB	64MB	32bit, 400MHz	??	CPU 314	CX8010				299,98 €
CX8051-CANopen slave	512MB, 1, 2 or 4GB	64MB	33 bit,400MHz	??	CPU315-2DP	CX8030				425,96 €
CX8080-RS232/RS485	512MB, 1, 2 or 4GB	64MB	32bit, 400MHz	??	CPU315-2PN/DP	CX8030 Con Profibus		EL6631 Profinet		689,72 €
CX8090-Ethernet	512MB, 1, 2 or 4GB	64MB	32bit, 400MHz	??	CPU317-2DP	CX9020 + Licencia		EL6731 Profibus		753,19 €
CX8091-BACnet & OPC UA	512MB, 1, 2 or 4GB	64MB	32bit, 400MHz	??	CPU317-2PN/DP	CX9020 + Licencia		EL6631-EL6731		1.016,94 €
CX8093-PROFINET	512MB, 1, 2 or 4GB	64MB	34 bit,400MHz	??	CPU319-3PN/DP	CX9020 + Licencia		EL6631-EL6731		1.016,94 €
CX8095-EtherNet/IP	512MB, 1, 2 or 4GB	64MB	32bit, 400MHz	??						
CX8097-Sercos III slave	512MB, 1, 2 or 4GB	2,5MB	32bit, 400MHz	??	CPU315F-2DP	CX8030		EL6900		425,96 €
CX9020	2xMicro SD	1GB	1 GHz	456,89 €	CPU315F-2PN/DP	CX8030 Con Profibus		EL6900	EL6631 Profinet	689,72 €
CX5130	not included,		Intel® Atom™	853,78 €	CPU317F-2DP	CX9020 + Licencia		EL6900	EL6731 Profibus	753,19 €
Sistema operativo WESP7 requiere al menos 8GB	pero se oferta aparte Cfast Card 2,4,8,16GB	4GB	1,75 GHz, 2 cores	8GB->109,84€ 16GB->195,68€	CPU317F-2PN/DP	CX9020 + Licencia		EL6900	EL6631-EL6731	1.016,94 €
					GAMA 1500	CX5130+Licencia+Cfast card 4GB			2xEL6631	1.400,00 €

Ilustración 14: Cuadro precios Beckhoff

Beckhoff en cambio ofrece una amplia gama de PC embebidos pero en esta tabla se resumen aquellos modelos que son más parecidos a los modelos ofertados por Siemens para así poder hacer una comparación más directa.

5.5.2.- CÁLCULO DEL TIEMPO DE CICLO

Otro factor que tenemos pendiente hoy en día es el del ciclo de scan, ya que, con los actuales PLCs de Siemens se dispara dicho tiempo ralentizando la producción sensiblemente. Por lo que se ha decidido hacer una comparativa de tiempos de ciclos de los diferentes PLCs del mercado.

5.5.2.1- SIEMENS

Uno de los grandes problemas que ULMA tiene cada vez que va una instalación, es que se le dispara el tiempo de ciclo. Como Siemens no garantiza que pueda ser determinista en su respuesta, se hizo una prueba en TIA PORTAL para ver de cuánto tiempo estamos hablando en este caso. Para estas pruebas se utilizó una CPU 317-2DP y como se puede observar el tiempo de ciclo es de 10ms pero a veces se dispara hasta los 38ms lo cual supone un problema, y es que al ser 3 veces mayor al ciclo de scan puede ralentizar temporizadores e incluso no ver determinados pulsos, creando así diferentes errores.

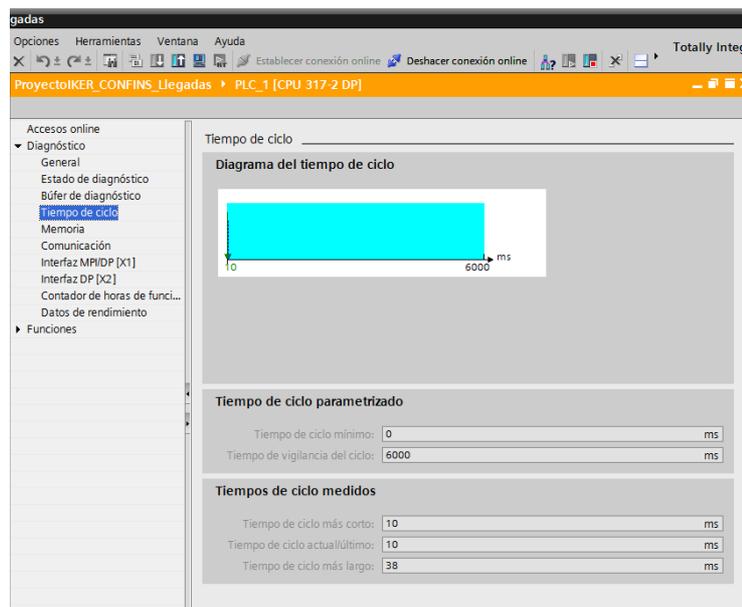


Ilustración 15: Tiempo de ciclo SIMENS

Visto que Siemens no permite configurar los tiempos de ciclo de las tareas, se quiso aprovechar para estudiar otras plataformas y otros buses como Ethercat.

5.5.2.2- BECKHOFF

Esta casa de autómatas dice que sus PLCs al ser deterministas son capaces de realizar la tarea en el periodo estipulado. A continuación una pequeña explicación de cómo lo hace posible Beckhoff esto. Antes de nada debemos configurar lo que ellos llaman el “Base Time”, el “Base Time” se puede configurar desde 50microsegundos hasta 1milisegundo y sirve para configurar de cuanto será el ciclo del programa, esto es, cada cuanto hará una lectura y escritura del programa. He aquí un ejemplo:

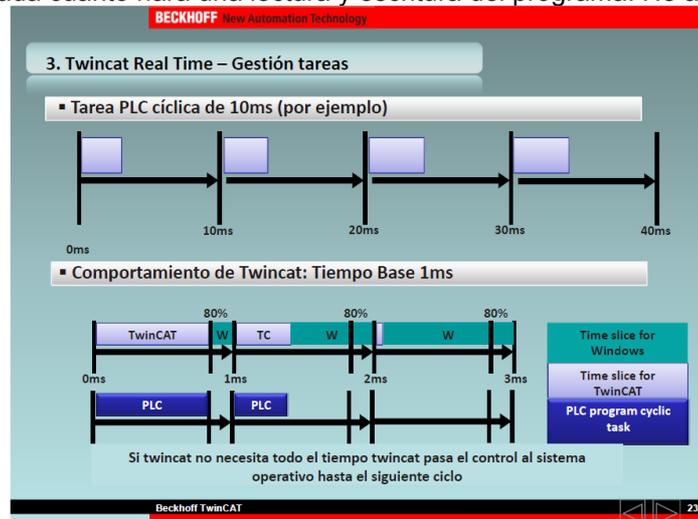


Ilustración 16: Tiempo de ciclo Beckhoff 1

Como se puede observar en este ejemplo si tenemos una tarea de 10ms y el Tiempo de Base es de 1ms, el programa puede ser que en un ciclo de la Base, esto es, en 1ms no se haya ejecutado al completo por lo que lo restante lo hace en el siguiente milisegundo y dejando todo a continuación libre para el sistema operativo. También observareis que del milisegundo del “Base Time” cierto porcentaje se deja libre para el sistema operativo, en este caso es el 20% pero se puede ampliar hasta lo que queramos pero siempre siendo múltiplo de diez.

¿Y qué pasa si tenemos varias tareas ejecutándose a la vez pero con diferente periodicidad? Sencillo, en la siguiente imagen explicamos de forma más visual.

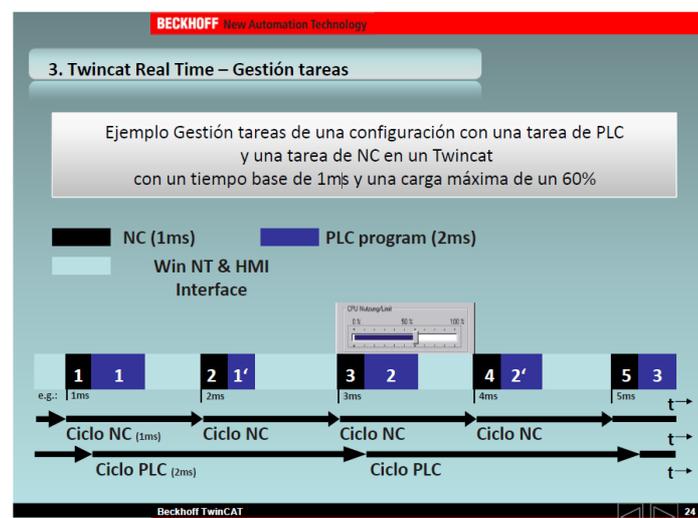


Ilustración 17: Tiempo de ciclo Beckhoff 2

DESARROLLO DEL PROYECTO

Como se puede observar en esta imagen la tarea con mayor prioridad se intuye que es la tarea con menos tiempo de ciclo. Por lo tanto, en este ejemplo se puede ver como la tarea principal se ejecuta cada vez que pasa el tiempo base y el secundario como no le daba tiempo a ejecutarse en un ciclo del tiempo de base se tuvo que poner de 2ms. Debido a este sistema de tareas prioritarias debemos definir muy bien en el programa las tareas de mayor a menor prioridad, si no queremos que el programa entre en error y no sea capaz de cumplir los tiempos.

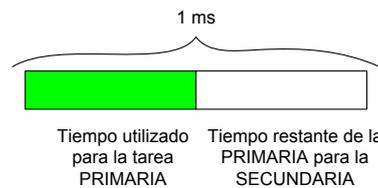
5.5.2.3- OMRON

Los PLCs de Omron, al igual que los de Beckhoff, tienen la posibilidad de comunicar mediante Ethercat lo cual hace que su tiempo de ciclo además de ser determinista es muy bajo en comparación con otros buses de campo. Para comprobar que esto era cierto se hizo un experimento pequeño, obteniendo las siguientes evidencias:

CALCULO DEL TIEMPO DE CICLO EN OMRON CON ETHERCAT

Tarea Periódica Principal → (0'5, 1, 2, 4) ms
 Tarea Periódica Secundaria → (1, 2, 3, 4, , 100) ms

Si ajustamos Primary Task a 1ms



Si ajustamos Periodic Task a 4ms

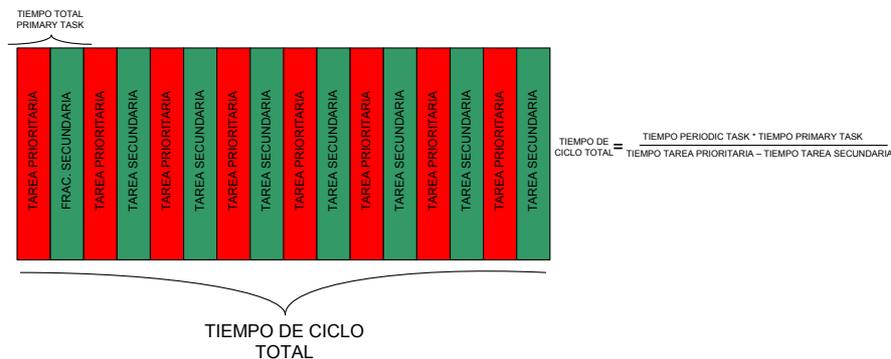
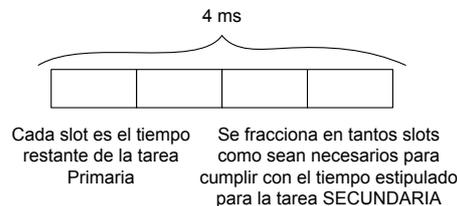


Ilustración 18: Tiempo de ciclo OMRON

Aunque el funcionamiento sea el mismo en Beckhoff y en Omron, debido a que en Omron se definen los tiempos para cada tarea y no para cada cuanto tiempo se debe ejecutar la tarea, se tuvo que sacar una fórmula matemática para saber de cuánto era el tiempo de ciclo en Omron o de lo contrario no había forma de saberlo.

6 DOCUMENTO DEL PROYECTO

El proyecto además de la memoria también estará compuesto de los siguientes manuales:

- Manual de planificación y presupuesto
- Manual del instalador
- Manual del programador

Todos estos manuales los encontraremos dentro del Anexo, por lo tanto, el Anexo estará dividido en dos carpetas:

1. **Carpeta Volumen I:**

Dentro de esta carpeta tendremos los siguientes manuales:

- Manual de planificación y presupuesto: En este manual se detallaran las diferentes fases llevadas a cabo para realizar el proyecto y el coste de este.
- Manual del instalador: Gracias a este manual sabremos con que versiones del programa se hizo el proyecto
- Manual del programador: El principal objetivo de este manual es el de explicar los pasos dados en el desarrollo del proyecto.

2. **Bibliografía del proyecto:**

En esta carpeta almacenaremos aquellos documentos, presentaciones o catálogos que han sido útiles para el desarrollo del proyecto.

CONCLUSIONES

7 CONCLUSIONES

Teniendo en cuenta que el estándar de programación de ULMA está dividido en dos partes (tracking y transporte), y toda la parte de tracking, encargada de gestionar las comunicaciones y los datos de las cargas está en AWL, y que a su vez, los trabajadores de Siemens nos comentaron que la tendencia de Siemens para los próximos años es la de hacer desaparecer el lenguaje AWL dejándolo obsoleto. Se decidió dar el salto a la programación de alto nivel, ya que este tipo de lenguajes nos ofrece una gran flexibilidad a la hora de migrar de una plataforma a otra. Además de ser más sencillo para la comprensión y manejabilidad de cualquier programador.

Se hizo una primera prueba con la Programación Orientada a Objeto pero los resultados no fueron tan buenos como se esperaban, por lo que, se decidió programar en Texto Estructurado.

De esta forma se pretende mediante un lenguaje unificado migrar de una plataforma a otra sin mayores inconvenientes. Se tradujo un programa completo, donde se hacen llamadas indirectas a DBs, se coge la hora del sistema, tenemos funciones que llaman a otras funciones, etc. Así, se pudo estimar cuanto podía costar traducir de un lenguaje a otro, además de valorar lo costoso que es migrar de una plataforma a otra.

Después de experimentar las dificultades que tiene programar en diferentes plataformas, la manera más sencilla para que todo este en un lenguaje unificado es programar todo en aquella plataforma que cumpla con el estándar y que a su vez sea migrable a otras plataformas. Como primera opción teníamos Codesys pero se descartó por falta de colaboradores.

La solución que considero óptima y que propongo a ULMA es programar en un lenguaje unificado como es el de texto estructurado y en la plataforma de Beckhoff, TwinCAT3. Beckhoff cumple al cien por cien con el estándar IEC61131-3 y mediante la opción de PLCOpenXML.xml lo hace exportable a otras plataformas como Codesys, además si se quiere migrar a Siemens al estar programado en texto estructurado no supone mayor problema, solamente copiando y pegando el código lo tenemos migrado de una plataforma a otra.

Después de mantener reuniones con diferentes expertos de Siemens, una idea nos quedó clara y es que el ciclo de vida de un PC embebido es bastante más corto que el de un PLC, ya que, el sistema operativo puede quedar desfasado en pocos años con la terrible consecuencia de dejar obsoleto el programa. Otra opción que se barajó era la de conectar un Runtime (es un PC que trabaja a tiempo real) en el mismo bastidor que el PLC, pero nos dijeron que era la misma idea que el PC embebido y que la descartásemos.

Por lo tanto, una vez analizadas las propuestas de los expertos y descartadas aquellas opciones que estuviesen ligadas a los PC embebidos, se optó por la gama de CPUs de última generación, hasta ahora estas estaban descartadas porque como aún no se había dado el salto al TIA Portal y la versión de STEP 7 que tienen instalada no es compatible con estas gamas de PLCs.

8 LÍNEAS FUTURAS

8.1. - ¿QUÉ TRADUCIR A ST?

Mi recomendación para un futuro es que primero se traduzca la parte del tracking, ya que toda su programación está en AWL y este lenguaje en muy pocos años quedara obsoleto.

Como uno de los grandes inconvenientes del texto estructurado es su compleja visualización a la hora de simular. Y como mencioné anteriormente desaconsejo traducir toda la parte de programación que tenga que ver con el transporte, porque en vez de facilitar la labor al técnico en puesta a punto se la complicaríamos notablemente.

8.2. - ¿DISEÑO DE LA ARQUITECTURA?

Para el apartado de la arquitectura y buses de campo, propongo que se haga alguna prueba con el nuevo bus de campo llamado Ethercat ya sea mediante algún prototipo. Porque, aunque Ethercat no sea un bus con el renombre de Profinet, pienso que sus prestaciones mejoran las de Profinet y con creces. Además los precios de los autómatas de Beckhoff pienso que son muy atractivos.

Por lo tanto, propongo que el PLC principal a poder ser sea una de Beckhoff y después mediante un switch distribuir toda la periferia así la parte de tracking, el apartado de la pantallas y la parte de seguridad podrán comunicar con la periferia mediante Ethercat o Profinet, ya que Beckhoff tiene una gran variedad de cabeceras para comunicar con diferentes buses.

9 BIBLIOGRAFÍA

Toda la información que me fueron facilitando para completar el proyecto lo encontraremos en los anexos.

- Manuales de OMRON y BECKHOFF
- Presentaciones de BECKHOFF
- PDF sobre CodeSys
- Varios PDF sobre IEC 61131-3

Además del material impartido en la asignatura de metodologías avanzadas también busqué en internet información sobre la Programación Orientada a Objetos. A continuación unos links interesantes:

- <http://www.desarrolloweb.com/manuales/teoria-programacion-orientada-objetos.html>
- http://librosweb.es/libro/python/capitulo_5/programacion_orientada_a_objetos.html