



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



Escola Tècnica
Superior d'Enginyeria
Informàtica

Escola Tècnica Superior d'Enginyeria Informàtica
Universitat Politècnica de València

Transformación digital en la industria

Trabajo Fin de Grado

Grado en Ingeniería Informática

Autor: Jaime Martínez Sánchez

Tutor UPV: José Salvador Oliver Gil

Tutor Empresa: Álvaro Roig Covas

Curso 2020-2021

Resumen

En este proyecto trataremos la importancia que tiene la transformación digital en la industria, lo necesaria que se ha vuelto hoy en día, y los diferentes servicios y soluciones que se usan para conseguirla. Entre ellos destacan las soluciones MES/MOM, que son las encargadas de unir la capa empresarial o de negocio con la capa de control, donde se encuentran las máquinas encargadas de llevar a cabo la producción en planta. Por último, veremos un caso de transformación digital en una empresa real gracias a los servicios LMS de Siemens, que sirven para monitorizar y tener visibilidad del proceso de producción. Esto se consigue gracias al análisis de los datos de planta y a la elaboración de una interfaz web desde la que visualizar toda la información.

Palabras clave: transformación digital, industria, soluciones MES/MOM, servicios LMS, Siemens, proceso de producción, interfaz web.

Abstract

In this project we are going to treat the importance that digital transformation has in the industry, how necessary it has become nowadays, and the different services and solutions used to achieve it, among which MES/MOM systems stand out, which are the attendants to join the business layer with the control layer, where the machines in charge of production in plant are ubicated. Finally, we are going to see an example of digital transformation in a real enterprise thanks to Siemens LMS services, used to oversee and visualize the production process. This is achieved thanks to data analysis of plant and creating a web interface where visualize all the information.

Keywords: digital transformation, industry, MES/MOM solutions, LMS services, Siemens, production process, web interface.

Resum

En aquest projecte tractarem la importància que té la transformació digital en la indústria, el necessària que s'ha tornat hui dia, i els diferents serveis i solucions que s'usen per a aconseguir-la. Entre ells destaquen les solucions MES/MOM, que són les encarregades d'unir la capa empresarial o de negoci amb la capa de control, on es troben les màquines encarregades de dur a terme la producció en planta. Finalment, veurem un cas de transformació digital en una empresa real gràcies als serveis LMS de Siemens, que serveixen per a monitorar i tindre visibilitat del procés de producció. Això s'aconsegueix gràcies a l'anàlisi de les dades de planta i a l'elaboració d'una interfície web des de la qual visualitzar tota la informació.

Paraules clau: transformació digital, indústria, solucions MES/MOM, serveis LMS, Siemens, procés de producció, interfície web.

Tabla de contenidos

1.	Introducción	1
1.1	Motivación. Automatización industrial y Maval como Siemens Partner.....	1
1.2	Objetivos del trabajo	3
1.3	Estructura del documento.....	4
2.	Evolución de la industrialización	7
2.1.	Historia de las revoluciones en la industria.....	7
2.1.1.	Primera revolución industrial (1785)	7
2.1.2.	Segunda revolución industrial (1870)	9
2.1.3.	Tercera revolución industrial (1969).....	10
2.1.4.	Cuarta revolución industrial (Presente).....	12
2.2.	Actualidad y transformación digital.....	13
2.2.1.	Big Data y BI.....	13
2.2.2.	Máquinas, robots y sistemas autónomos	14
2.2.3.	Internet de las cosas.....	14
2.2.4.	Computación en la nube.....	14
2.2.5.	Inteligencia artificial y Machine Learning	15
2.2.6.	Ciberseguridad	15
2.3.	Retos de la industria de procesos	16
2.3.1.	Retos tecnológicos.....	16
2.3.2.	Retos sociales	17
2.3.3.	Retos regulatorios.....	18
3.	Servicios de automatización en la industria	21
3.1.	Tipos de industria y áreas de negocio	21
3.1.1.	Industria de alimentación y bebidas	21
3.1.2.	Industria aeronáutica	22
3.1.3.	Industria de productos químicos.....	23
3.2.	Necesidades de digitalización y automatización de las empresas.....	24
3.2.1.	Digitalización de la cadena de valor.....	24
3.2.2.	Diseño del Gemelo Digital	27
3.3.	Soluciones para la transformación digital	28
3.3.1.	Estándar ISA 95	28
3.3.2.	Level 1,2: Control y Automatización.....	28
3.3.3.	Level 3: Operacional y Control de producción	30



3.3.4.	Level 4 y 5: Capa de negocio y análisis y explotación de datos	31
4.	Sistemas MES/MOM	35
4.1.	Situación de un sistema MES/MOM dentro de la cadena de valor	35
4.2.	Integración con sistemas de la capa operacional y estratégica.....	36
4.3.	Operaciones integradas de manufactura.....	37
4.3.1.	Gestión de los Recursos de Producción	37
4.3.2.	Gestión de la Definición del Producto	38
4.3.3.	Programa detallado de Producción.....	39
4.3.4.	Despacho a Producción	41
4.3.5.	Gestión de la Ejecución de la Producción	42
4.3.6.	Trazabilidad de la Producción.....	43
4.3.7.	Adquisición y Recolección de Datos de Producción.....	44
4.3.8.	Análisis del Desempeño de Producción	44
5.	Caso práctico (Análisis del Desempeño de Producción).....	51
5.1.	Empresa Acram	51
5.2.	Necesidades de transformación digital del cliente	51
5.3.	Tecnologías utilizadas para la Interfaz Web	52
5.3.1.	Vue.js	52
5.3.2.	Vuex	58
5.3.3.	Quasar	61
5.3.4.	Base de datos.....	63
5.4.	Implantación de servicios LMS.....	66
5.4.1.	Tecnología Siemens LMS	66
5.5.	Diseño de la interfaz de usuario del LMS	69
5.5.1.	Inicio de sesión.....	69
5.5.2.	Página principal.....	71
5.5.3.	Indicadores operacionales	76
5.5.4.	Paneles complementarios	82
5.5.5.	Control de calidad	83
6.	Pruebas	87
7.	Conclusiones	91
7.1.	Cumplimiento de los objetivos.....	91
7.2.	Problemas y contratiempos surgidos.....	92
7.3.	Líneas futuras	92
8.	Bibliografía	95
	Apéndice A. Configuración inicial de la página	97

A.1. Fichero de enrutamiento.....	97
A.2. Configuración inicial de Quasar.....	99
Apéndice B. Código página web.....	103
B.1. Código de la página de inicio de sesión.....	103
B.2. Módulo <i>login</i> de Vuex.....	106
B.3. Llamadas a API <i>Rest</i>	107
B.4. Manejo de Pop-ups.....	108
B.5. Menús expandibles.....	109
B.6. Manejo del svg del mapa.....	110
B.7. Uso de <i>slots</i>	112
sideMenu.vue.....	112
OperacionalLayout.vue.....	113
B.8. <i>Select</i> intervalo.....	114
B.9. Conexión WebSocket.....	115
B.10. Gráfica OEE.....	117
B.11. Búsqueda en una tabla.....	122
B.12. Creación de una tabla.....	125

Agradecimientos

Antes que nada, me gustaría agradecer a la empresa para la que he realizado este proyecto, Maval, por haber confiado en mí para este trabajo y por haberme dado la idea. En primer lugar, me gustaría agradecer a mis dos tutores: Álvaro por toda la ayuda que me ha dado para la realización de este y todas las reuniones que hemos tenido, a pesar de que en momentos haya podido llegar a ser muy pesado. También me gustaría agradecerle a Jose por haber aceptado ser mi tutor y haber revisado que se haya llevado a cabo correctamente.

Gracias a todos mis compañeros de empresa que me han ayudado y enseñado una gran cantidad de cosas de algo que era relativamente nuevo para mí. Gracias Jose Luis por haberme ayudado y acompañado en esta nueva etapa de mi vida, lo he pasado muy bien contigo, y para mí siempre serás mi verdadero jefe. Gracias a Navarro por ayudarme tanto con el diseño de esta página, aunque a veces me dieras mucho la chapa, has sido clave para que quedara tan bonita la página, espero que te vaya bien en tu nueva aventura. También en este sentido agradecer a Bea por la ayuda a la hora del diseño, no sé cómo habría sido posible algunas de las cosas de la página, así que gracias. También me gustaría agradecer a Saúl por haberme ayudado con todo el tema del back, ya que quedaba poco tiempo y yo no tenía ni idea y de no haber sido por ti jamás habría llegado a tiempo. También agradecer al resto de mis compañeros de trabajo, Raúl, Silvia, Borja, etc. por el buen rato que he tenido en el poco tiempo que he podido estar en la oficina.

No me quiero olvidar tampoco de los amigos que me han acompañado durante todo este camino, tanto los de Caravaca como los de Valencia y los del Erasmus. Puede que vosotros no hayáis ayudado directamente a lo largo de la carrera, pero siempre habéis estado ahí cuando he necesitado despejarme y olvidar un rato mis problemas. No os nombro a todos porque sois demasiados y no quiero dejarme a nadie, pero vosotros sabéis quiénes sois. Gracias por todos estos años.

Y, por último, dar las gracias a mi familia que siempre han estado ahí para lo que sea desde el día que nací, tanto mis abuelos/as, primos/as, tíos/as y, por supuesto, mi hermana. Y para acabar, la mención más importante en estos agradecimientos. Gracias a mis padres por haber estado siempre ahí, haber sacrificado y haber trabajado tanto para poder conseguir esto. Sin vuestro esfuerzo esto nunca hubiera sido posible.

MUCHAS GRACIAS



1. Introducción

Hoy en día vivimos en un mundo en el que la tecnología la podemos encontrar en cualquier sitio. El mundo está tan conectado y digitalizado, que no nos imaginaríamos vivir sin un teléfono móvil o sin un ordenador. La aparición de estos dispositivos nos ha proporcionado una gran comodidad, y nos permite hacer muchas cosas que en épocas anteriores sería impensable, como por ejemplo poder comprar casi cualquier cosa sin tener que movernos de nuestra casa.

Antiguamente, los trabajos que existían eran muy manuales, pero con el tiempo, las empresas han ido evolucionando, y cada vez son más las que van reduciendo estos trabajos manuales, sustituyéndolos por trabajos digitalizados, lo que ha permitido una gran eficiencia y rentabilidad para muchas empresas.

En este proyecto, llevaremos a cabo una transformación digital de una empresa real. Como por motivos de privacidad y de confidencialidad no se puede nombrar la empresa, la llamaremos Acram. Esta transformación servirá para ayudar a mejorar sus procesos productivos, y veremos por qué es tan importante en la actualidad que las empresas continúen digitalizándose.

1.1 Motivación. Automatización industrial y Maval como Siemens Partner

La necesidad de digitalización en las empresas es lo que me ha llevado a la realización de este proyecto. Las tecnologías cada vez van ganando más y más terreno y no podemos quedarnos atrás, ya que nos permiten producir de una forma más veloz. Además, la demanda es cada vez mayor gracias a la globalización, y esto hace que, junto con la gran competencia que existe, deba de haber una buena puesta tecnológica para aquellas empresas que quieran tener presencia en el mercado.

De entre todos los sectores que pueden abarcar las tecnologías de la información, la automatización industrial es una de las más importantes. La industria necesita que la producción sea rápida y fiable, ya que la demanda en estos días es muy alta. Es por eso por lo que se necesita la incorporación de máquinas y distintos dispositivos en las plantas de trabajo, que realicen procesos de manera automática, y que ayuden a los trabajadores

Maval¹ es una empresa que lleva en activo desde 1998, y que lleva ofreciendo desde entonces soluciones de automatización, entre otros sectores, en la industria de procesos. Además, lleva siendo Siemens Partner desde el año 2000, lo que demuestra el gran nivel de calidad que ofrece en sus soluciones.

¹ MAVAL Industria Viva. <https://www.maval.es/> .





Figura 1.1: Logo de Maval.

En sus 22 años en activo, Maval ha estado colaborando con muchas empresas de los distintos sectores que hay en la industria. En el año 2000 hizo su primer proyecto PCS7² para Mercadona, convirtiéndose en PCS7 *Solution Partner* en el 2007, y más tarde, en el 2013, en SIMATIC PCS7 *Specialist Partner*³, año en el que también se convirtió en Siemens *Food&Beverage Industry Partner*, siendo la primera empresa de España en lograrlo. También en 2013 empezó su colaboración con Airbus Stade-Illescas, y más tarde, en 2016, con Airbus Toulouse. En 2018 abrió su oficina de ventas en Madrid, y en el 2019 una oficina en Barcelona. Su último gran logro fue también en el 2019, ampliando su gama de implantación de soluciones con SIMATIC IT PREACTOR.

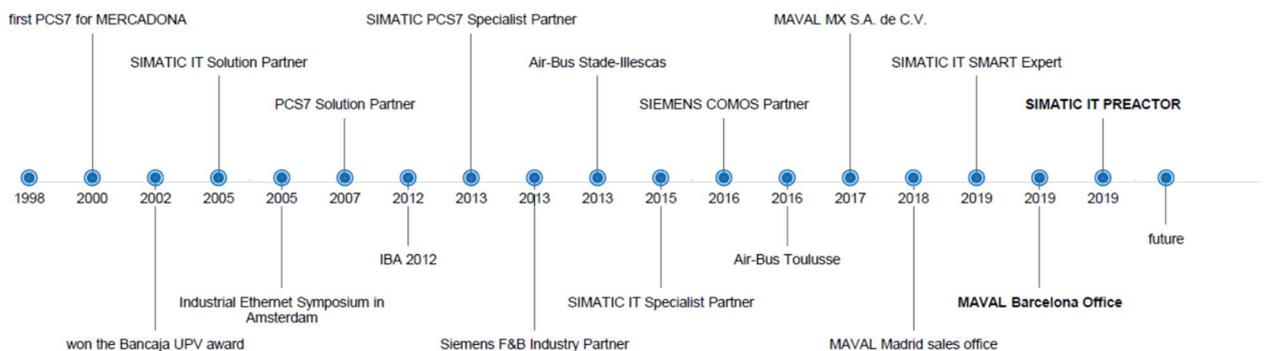


Figura 1.2: Línea temporal de Maval.

Es una empresa que se enfoca en muchos campos de la industria: desde el sector de la alimentación, hasta el de los productos químicos, pasando por el de la aeronáutica y el papelero.

Entre las distintas soluciones que propone, se encuentra la de los sistemas de gestión de producción MES/MOM, que es un sistema que sirve para comunicar la capa de

² Siemens Simatic PCS7.

https://www.automation.siemens.com/w2/efiles/pcs7/pdf/78/prdbrief/kb_pcs7_v70_es.pdf .

³ Niveles de Siemens *Solution Partner* https://www.maval.es/?page_id=505 .

negocio de una empresa con la capa de control (las máquinas que se encuentran en las plantas). Estos sistemas los veremos con detalle a lo largo de este proyecto.

Otra de las motivaciones es el entendimiento de las tecnologías y de los lenguajes que se usarán para la realización de este proyecto, entre los que destaca el *framework* Vue.js, que facilita la creación del diseño, los estilos y la lógica de una página web (*front-end*), y que está basado en el lenguaje JavaScript, uno de los lenguajes más utilizados actualmente. También busco aprender de la tecnología OPC UA, que sirve para llevar datos de máquinas a cualquier plataforma de nuestra empresa. Para obtener estos datos deberemos de comunicarnos de forma eficiente con el servidor, reduciendo lo máximo posible los tiempos de espera para obtener los datos.

Como se puede ver, se trata de una empresa que busca diversas soluciones para la automatización y la transformación digital en la industria, y con la tecnología de Siemens, ofrece una gran calidad en su trabajo y en sus proyectos. Estas son las motivaciones que me han llevado a decantarme por este trabajo y por este proyecto en particular.

1.2 Objetivos del trabajo

Desde la comercialización de los primeros ordenadores, las empresas han ido digitalizándose con la incorporación de distintos dispositivos que les permiten hacer su trabajo de una manera más fácil. Con este trabajo buscamos que se entienda lo importante que es la transformación digital en la industria, y la implementación de esta digitalización en un caso práctico.

El objetivo principal del proyecto es la implementación de un entorno web para un sistema LMS, que permita obtener visibilidad y monitorizar un proceso de producción. Se deberá de conseguir implementar una estructura estándar que cumpla con las siguientes pautas:

- Desarrollar una interfaz web estándar que sea *responsive*, es decir, que se pueda ver correctamente en cualquier navegador o dispositivo que se desee, usando las tecnologías necesarias para ello.
- Representar los datos obtenidos durante el proceso de producción de forma clara y concisa, a nivel de planta, área, línea y máquina. Para ello se deberán usar distintos indicadores de eficiencia de un proceso de producción.
- Obtener estos datos en tiempo real para tener conocimiento en todo momento de cómo está transcurriendo la producción.

Este es el objetivo básico que se tiene que cumplir para que el cliente quede satisfecho. La empresa en la que se llevará a cabo esta digitalización (recordemos que por

motivos de privacidad vamos a usar un nombre e imagen ficticios), es Acram, empresa dedicada a la elaboración de productos químicos. Además, al ser una empresa española, también busco poner un granito de arena para la mejora de la digitalización de la industria en este país. El proyecto parte de la necesidad de Acram por obtener una visibilidad de su proceso de producción y poder monitorizarlo, ya que hasta ahora el cliente no tiene nada que implemente estos servicios. Con esta implementación se busca mejorar la eficiencia y la fiabilidad del proceso, además de poder detectar rápidamente los errores, o aquella parte del proceso que está ralentizando la producción, entre otras cosas.



Figura 1.3: Logo de Acram.

Como objetivo secundario se busca ser capaces de comprobar los cuellos de botella que aparecen en un proceso de producción, los problemas más comunes, o comprobar el tiempo que pasa parada una máquina. De esta manera, si se ve que una máquina pasa mucho tiempo parada, se podrá tener un conocimiento de por qué ocurre, en qué máquina ocurre y buscar una solución.

Otro de los objetivos que se deben llevar a cabo es realizar una interfaz que sea intuitiva y de fácil manejo, ya que debemos pensar que nuestra página web debe de ser utilizada por muchos operarios, y que no todos deben tener un gran conocimiento de esta. Necesitamos que los usuarios sepan cómo moverse por la página y por las diferentes vistas sin tener un conocimiento muy extenso de nuestra web.

Por último, a nivel personal, el objetivo que busco con este trabajo, aparte de mejorar mi entendimiento en las tecnologías que se usarán, como se ha comentado anteriormente, es que la empresa confíe en mí para más proyectos de este estilo, y la satisfacción que produce la realización de un proyecto de tanta envergadura.

1.3 Estructura del documento

El proyecto se compone de seis capítulos, en los que se define, de forma clara y concisa, el tema que estamos tratando; en este caso, la transformación en la industria.

- ❖ **Capítulo 1: Introducción.** En este capítulo nos enfocaremos en introducir al lector en el tema del que vamos a hablar, exponiendo lo que nos ha motivado a la realización del trabajo y los objetivos que buscamos lograr.
- ❖ **Capítulo 2: Evolución de la industrialización.** Aquí vamos a hablar del estado del arte en la industria, de los cambios que ha sufrido a lo largo de su historia, los retos que surgen a raíz de esta transformación digital, y lo que esperamos para el futuro.
- ❖ **Capítulo 3: Servicios de automatización en la industria.** En este capítulo ahondaremos en lo que es la automatización en sí, la necesidad de tenerla en nuestra empresa, y las distintas soluciones que se ofrecen para ello.
- ❖ **Capítulo 4: Sistemas MES/MOM.** Aquí hablaremos de las soluciones MES, y de cómo se acoplan a las distintas capas que se pueden encontrar dentro del proceso de producción.
- ❖ **Capítulo 5: Caso práctico (Análisis del Desempeño de Producción).** En este capítulo hablaremos del caso práctico al que vamos a aplicar esta digitalización; en nuestro caso, la empresa Acram. Hablaremos de por qué necesita esta transformación digital, de los servicios que se han usado para lograr este objetivo, y de cómo se ha implementado esta solución.
- ❖ **Capítulo 6: Pruebas:** En este capítulo realizaremos distintas pruebas que demuestren el correcto funcionamiento de la solución que hemos implementado.
- ❖ **Capítulo 6: Conclusiones.** Aquí hablaremos de los problemas que nos hemos encontrado durante el transcurso del proyecto, y de si se han llegado a cumplir los objetivos marcados al principio. También veremos cómo se podría mejorar este proyecto en el futuro.
- ❖ **Capítulo 7: Bibliografía.** En este capítulo se encontrarán todos aquellos libros y sitios web que se han usado para recabar la información usada en la realización de este proyecto.
- ❖ **Capítulo 8: Apéndice.** Por último, aquí se mostrarán distintos fragmentos de código que se han utilizado para la implementación de la transformación digital en Acram.

2. Evolución de la industrialización

La industria es una de las razones principales por las que la calidad de vida de las personas cada vez es mayor. Sin ella, no podríamos soportar la alta demanda que existe en la actualidad. Sin embargo, sus inicios fueron muy distintos a lo que hoy en día conocemos, ya que, a lo largo de la historia, es un sector que ha ido evolucionando y cambiando de forma drástica, para cubrir las necesidades de la humanidad.

2.1. Historia de las revoluciones en la industria

A lo largo de la historia, la industria ha experimentado varias revoluciones que la ha llevado a ser lo que es hoy, provocado por cambios drásticos en los medios de producción, que a su vez ha provocado cambios económicos y sociales. En este capítulo haremos un recorrido por todas las revoluciones que ha habido a lo largo de la historia, y que han cambiado drásticamente la concepción que teníamos de la industria [1].

2.1.1. Primera revolución industrial (1785)

Desde los primeros tiempos, la humanidad ha sabido explotar su entorno para satisfacer sus necesidades. Ha explotado los elementos que la naturaleza le ha ofrecido para crear herramientas, y obtener productos que no se podrían haber obtenido de forma natural. Durante siglos, ha habido invenciones muy importantes para la vida de los seres humanos, como puede ser la invención de la agricultura en la prehistoria, gracias al uso de sílex y de metales.

Ya en la edad media, las artesanías manuales sobre fibras naturales, pieles, cerámicas, vidrio, metales y maderas fueron el motor principal de la economía de esa época. Las fuentes de energía que servían para la producción era el fuego, la fuerza animal y el viento. No fue hasta el siglo XVIII que empezó a hablarse de industria como hablamos hoy en día, cuando se produjo una serie de invenciones que supuso una auténtica revolución socioeconómica en la sociedad.

En el año 1712, Thomas Newcomen [17] inventó la máquina de vapor que, más tarde, en 1769, fue perfeccionada por James Watt [18]. Este invento propició la automatización de los telares, apareciendo así la producción masiva de productos idénticos. El primer telar mecánico fue patentado en 1785 por Edmund Cartwright [19] que marca el inicio de la primera revolución industrial (aunque el año de inicio no está claro, lo que sí se sabe con certeza es que se produjo en el siglo XVIII).

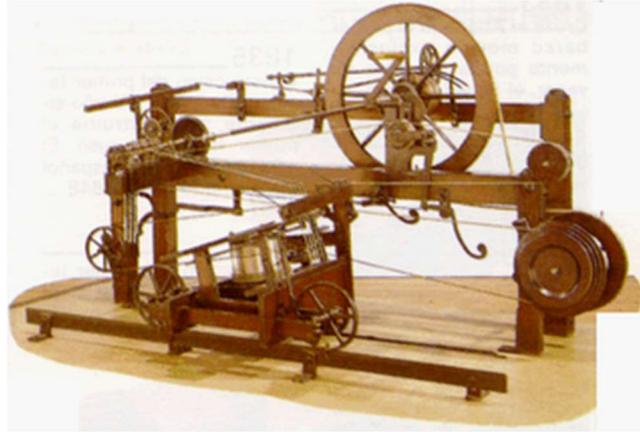


Figura 2.1: Telar mecánico de Cartwright (1785)

La invención de estos telares mecánicos propició que las fuentes de energía con las que se producía antiguamente quedaran desfasadas, y se empezaron a crear grandes fábricas que usaban estas máquinas de vapor. Estas fábricas se construían preferiblemente en las orillas de los ríos, ya que necesitaban abundantes suministros de agua y combustible.

De esta manera surgió la clase obrera y una nueva burguesía, que se enriqueció con estas fábricas. También propició la migración de los trabajadores del campo hacia las grandes ciudades, que se vieron desbordadas, con la consiguiente aparición de suburbios con servicios muy pobres.

Entre todos los cambios que se produjeron durante esta primera revolución industrial, destacan los siguientes:

- La aplicación de la ciencia para crear máquinas y para los procesos productivos, gracias al aporte de capital de burgueses y aristócratas para llevar a cabo las investigaciones necesarias. Esta mejora técnica en la agricultura provocó un gran paro laboral en el campo.
- Crecimiento de la industria del carbón y del hierro.
- Gran crecimiento demográfico provocado por la mejora de higiene de la población.
- Eliminación de muchas costumbres feudales en el sistema social, sentándose las bases de la igualdad ante la ley, la libertad civil, y la libertad religiosa de la población.

El primer país en industrializarse fue Gran Bretaña, debido a diversos factores como su gran fuerza naval, su gran imperio colonial, o una buena situación geográfica. A ellos le siguieron países como Bélgica, España, Alemania y Estados Unidos.

Todos estos cambios tuvieron un gran impacto en la sociedad, tanto a nivel económico e industrial, como ideológico, sentando las bases de lo que en el futuro sería el liberalismo y el socialismo.

2.1.2. Segunda revolución industrial (1870)

Desde la primera revolución industrial, la industria fue evolucionando sin parar hasta que, a finales del siglo XIX y principios del siglo XX, se produjeron importantes innovaciones que llevaron a la segunda revolución industrial. Entre estas innovaciones destacan la introducción del gas como combustible o la aparición de la electricidad como nueva fuente de energía, lo que provocó que dejara de ser necesario construir las fábricas junto a los ríos, incrementándose la diversidad en la industria.

Aparecieron tres nuevos sectores industriales: el químico, el metalúrgico y el eléctrico. Gracias a la industria siderometalúrgica se consiguieron metales más puros, además de comenzar a producirse acero más barato y a utilizar aluminio.

En cuanto al transporte, se crearon grandes vías ferroviarias para el transporte de productos y materias primas, lo que hizo que se ampliara enormemente el mercado. También aparecieron barcos de vapor lo que, junto a la apertura de los canales de Suez y de Panamá hizo que el mercado marítimo fuera mucho más eficiente.

Esta expansión comercial por todo el mundo hizo que se consolidara el capitalismo como principal modelo económico y, al haber un gran incremento en los beneficios de las empresas, aparecieron grandes monopolios. Al producirse a tanta velocidad y tan masivamente, se inició una gran búsqueda de lugares donde poder invertir y de donde extraer las materias primas necesarias para abastecer este alto nivel de producción. Esto provocó que surgieran muchos movimientos colonizadores de los países industrializados, centrados sobre todo en Asia y en África.

Estados Unidos fue la potencia que lideró este cambio, con inventos como el de la industria cárnica de Cincinnati en 1870 [2]. Las empresas de esta industria crearon la primera producción en línea o en cadena, invento con el que situamos el inicio de esta segunda revolución industrial, aunque, como pasa con las demás, el año y el invento clave pueden variar. Estas empresas usaban una cinta transportadora para mover sus productos, con muchos trabajadores a su alrededor, cada uno con una tarea concreta.

La industria no se fijó en las ganancias en productividad y eficiencia de la industria cárnica de Cincinnati hasta que, en el 1908, Henry Ford [20], propietario de una de las industrias automovilísticas más grandes de la historia, Ford Motor Company, uso su modelo de producción en línea para la creación de su Ford Modelo T. Gracias al uso de este modelo se rebajó el tiempo de ensamblaje de un coche de 12,5 horas a 93 minutos, lo que significó una gran ganancia en términos de eficiencia, una gran rebaja en el coste de producción y, por consiguiente, una drástica caída en el precio de sus coches. Esto hizo que los demás sectores de la industria asumieran este modelo de producción, lo que hizo que se popularizara enormemente.



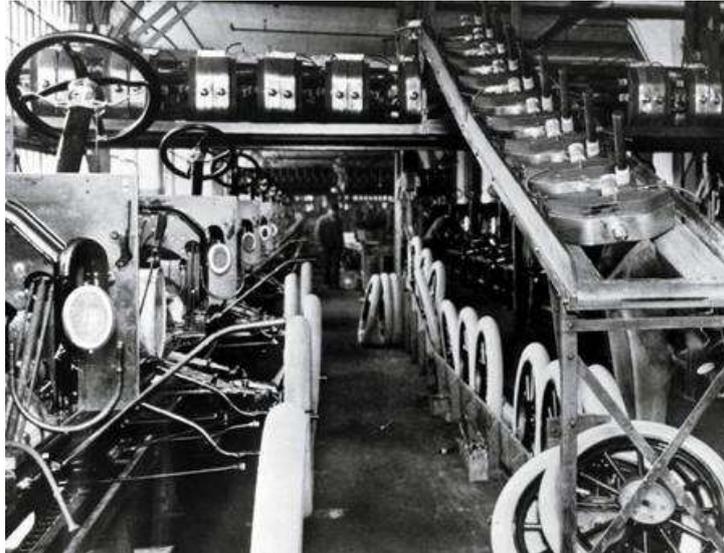


Figura 2.2: Cadena de montaje del Ford Modelo T.

Además del Modelo T de Ford, se produjeron otros grandes inventos durante la segunda revolución industrial, como puede ser la televisión en el 1924, la penicilina en el 1928, el radar en el 1935, o la fibra óptica en el 1955. Por otro lado, otros inventos como la cámara de fotos, la máquina de escribir, el teléfono o el automóvil empezaron a ser accesibles para una parte de la población, lo que provocó que apareciera una clase media.

2.1.3. Tercera revolución industrial (1969)

A finales del siglo XX, las tecnologías de la información y la comunicación (TIC) se popularizaron en la industria, dando lugar a la sociedad de la información y desencadenando esta tercera revolución industrial, en la que se encuentran la mayoría de los países industrializados.

Uno de los mayores inventos que se consiguieron gracias a las TIC fue el primer controlador lógico programable (comúnmente llamado PLC por sus siglas en inglés), llamado Modicon 084, en el 1969, año en el que ponemos el inicio de la tercera revolución industrial. Estos controladores guardan instrucciones que se usan para la implementación de ciertas funciones y cálculos, que servían para controlar las distintas máquinas y procesos en las fábricas. Gracias a ello, se consiguió la automatización en la cadena de montaje de las distintas empresas, lo que permitía una mayor velocidad y fiabilidad de la producción.



Figura 2.3: Modicon 084.

Durante esta época, se produjeron grandes cambios en la industria. Aparecieron grandes innovaciones en microelectrónica y medios de comunicación, como puede ser los inicios de la televisión multicanal que hoy en día conocemos. Los dispositivos analógicos y mecánicos que se usaban antiguamente fueron sustituidos por dispositivos digitales y equipos electrónicos, desarrollándose los primeros robots industriales. Esto trajo como consecuencia una demanda más fuerte de trabajadores cada vez más especializados.

Las fuentes de energía convencionales comenzaron a ser sustituidas por fuentes de energía renovables, para adaptarse a una sociedad cada vez más concienciada con la sostenibilidad del planeta. Recursos como los combustibles fósiles comenzaron a ser sustituidos, ya que son muy perjudiciales para el medio ambiente, además de cada vez más escasos.

Otro de los grandes inventos que se produjeron en esta tercera revolución industrial fue el World Wide Web (más conocido por sus siglas WWW), en 1989, por Tim Bernes-Lee [21]. Este invento permitió que cualquiera pudiera acceder a la información de manera muy rápida, cuando y donde sea, lo que supuso un gran avance para la sociedad. A consecuencia de esto aparecieron las redes sociales, elemento casi indispensable a la hora de comunicarnos en la actualidad.

Las TIC ganaron gran importancia a la hora de la gestión de las empresas, con la consiguiente aparición del *marketing* y de los comercios electrónicos, aprovechando la globalización que había surgido gracias a Internet. Esto también provocó que la producción fuera más descentralizada, surgiendo de esta manera el teletrabajo, y una comunicación entre las empresas cada vez más digitalizada.

En el plano económico, durante la tercera revolución industrial creció enormemente la economía de los servicios, además de crearse un mercado común entre aquellos países que tenían buenas relaciones.

Dentro de esta revolución industrial es donde aparece Maval, fundada en 1998 y que, como hemos visto, usando estas tecnologías, ofrece soluciones útiles a la industria para ir adaptándose a todos estos cambios que surgieron a partir de la tercera revolución industrial.

2.1.4. Cuarta revolución industrial (Presente)

Esta última revolución industrial es la que se está produciendo hoy en día. Como en las demás, tampoco se sabe con exactitud cuando arranca. Una de las primeras definiciones que recibió esta revolución fue en 2011 en la Feria de Hannover, que es un encuentro industrial mundial de todas las empresas relacionadas con este sector. Sin embargo, no recibió el nombre comúnmente conocido de Industria 4.0 hasta el 2013.

En esta Industria 4.0, la creación de los distintos productos se hace de forma sostenible, usando materiales reciclables y energías renovables o limpias. En cuanto al diseño de los productos, nos encontramos con un diseño ecológico y colaborativo, cuyos clientes ya no son consumidores únicamente, si no que sirven a la hora del diseño de los nuevos productos, teniendo, en ocasiones, potestad para modificarlo.

También destaca la aparición del Internet de las Cosas (*Internet of Things*, IoT, en inglés), que lleva las tecnologías de la información a muchos objetos de la vida cotidiana que creíamos imposible, como frigoríficos o aspiradoras. De esta manera, podemos manipular estos dispositivos para que funcionen de forma automática, además de poder monitorizarlos.

En cuanto a los medios de comunicación, cobran gran importancia las redes sociales, que permite conectar con cualquier persona del mundo, surgiendo de esta manera el marketing digital. Gracias a estas nuevas formas de comunicarse por internet se generan una gran cantidad de datos y de información a la que se llama *Big Data*. A través de esta gran cantidad de datos, se puede ofrecer una mejor experiencia al usuario, como la aparición de publicidad relacionada con sus gustos, o sugerirles usuarios con los que tengan cosas en común. En la industria, el Big Data puede ayudar analizando la evolución de los mercados y, de esta manera, que las empresas tomen las decisiones adecuadas basándose en estos datos.

Mientras que en la tercera revolución industrial se perfeccionaron las TIC en la sociedad digital, en esta cuarta revolución las TIC se integran durante todo el ciclo de vida de un producto. En términos de logística, se ha mejorado y abaratado el almacenamiento y los medios de transporte, con la aparición de nuevos inventos como los drones, que poco a poco van ganando más peso.

En el plano de los sistemas de producción, aparece la Inteligencia Artificial (IA), de la que derivan la robótica, los sensores o las comunicaciones inalámbricas. De esta forma, las distintas fases de un proceso de producción se realizan con estos dispositivos inteligentes, que se comunican entre ellos de manera automática, ayudando a los trabajadores que antes se encargaban de estas funciones. Gracias a esto se abarata mucho los costes de tiempo, de procesamiento, de trabajo, de comunicación, etc.

Maval supo ver el cambio de paradigma que se estaba produciendo en la industria, introduciéndose, de esta forma, en la Industria 4.0, y ofreciendo soluciones a sus distintos clientes para automatizar todo su proceso productivo.

Como se puede ver, esta Industria 4.0 está cada vez más presente en los distintos sectores, sin embargo, aún queda mucho trabajo por hacer hasta conseguir la digitalización total de la industria, aunque parece algo inevitable.



Figura 2.4: Industria 4.0: La cuarta revolución industrial

2.2. Actualidad y transformación digital

En la actualidad cada vez son más las empresas que buscan esta digitalización, pues vivimos en un mundo que está muy conectado, y esto hace que cada vez sea más necesario esta transformación.

Para conseguir esta transformación digital es necesaria la conexión de todos los elementos que forman parte de una empresa a través de los distintos dispositivos y plataformas digitales.

Pese a que muchos de los avances tecnológicos que se usan en la industria 4.0 ya se usaban tras la tercera revolución industrial, la gran diferencia es que ahora estos avances están conectados entre sí, lo que crea un proceso de producción totalmente automatizado, con lo que se consigue una gran eficiencia y productividad en las empresas. Para alcanzar esta transformación digital, es necesario el uso de las siguientes tecnologías, entre otras [3].

2.2.1. Big Data y BI

Con el Big Data, las empresas recogen una gran cantidad de datos de Internet y, gracias a ello, intentan escoger las mejores estrategias empresariales. Estos datos pueden ser tanto tendencias de mercado, como datos de usuarios. En la industria, el Big Data se usa para recopilar la mayor de datos posible sobre la evolución de mercado y más tarde, con BI (*Business Intelligence*) y usando estos datos, se llevarán a cabo las mejores decisiones para mejorar el sistema de producción en nuestra empresa. En la actualidad, hay empresas que llegan a procesar datos de incluso varios *petabytes* (lo que correspondería a más de 1000 terabytes de datos).

2.2.2. Máquinas, robots y sistemas autónomos

Estas máquinas automatizadas también son muy importantes para la transformación digital, ya que son mucho más rápidas y eficientes que los humanos a la hora de automatizar algunas de las tareas que se llevan a cabo dentro de una empresa. En la industria actual, los robots sirven para prácticamente cualquier paso del proceso productivo, gracias a su navegación autónoma, a la integración de sensores y actuadores, y a la comunicación con el resto de las máquinas. La principal idea para el futuro es seguir implementando estos robots para que prácticamente cualquier área de una empresa acabe completamente automatizada.

2.2.3. Internet de las cosas

Gracias al Internet de las cosas o *Internet of Things* se consigue una gran comunicación entre todas las máquinas que hay en una empresa, recogiendo información que posteriormente sirve para identificar algún fallo en alguna máquina o cómo podría ser más eficiente [10]. Aquellos dispositivos que tienen sistemas IoT pueden adquirir datos con los sensores que tengan instalados y mandar órdenes a los dispositivos que interactúan con el mundo real. Estos dispositivos entonces ejecutarán aquellas instrucciones que reciban de manera automática, lo que en una industria puede servir para reducir la mano de obra humana en una empresa y, por consiguiente, ganar una mayor eficiencia.

2.2.4. Computación en la nube

La computación en la nube nos sirve para poder almacenar aquellos recursos informáticos de una empresa sin la necesidad de tener que disponer de una infraestructura física en la propia empresa, lo que ofrece una gran agilidad, interoperabilidad y escalabilidad, además de una gran facilidad para acceder a estos recursos y de una gran disponibilidad de estos. Este servicio es muy importante en aquellas empresas que usan grandes cantidades de datos o de recursos ya que, al estar

en la nube, no necesitan servidores para el almacenamiento y el acceso de estos recursos, con el consiguiente ahorro energético y de mantenimiento que ello conlleva. Gracias a la computación en la nube se consigue una alta disponibilidad al querer acceder a la información y una buena escalabilidad, aunque dependen de tener un buen acceso a Internet.

2.2.5. Inteligencia artificial y *Machine Learning*

Con la llegada de las tecnologías de la información y de los primeros dispositivos tecnológicos surgió la inteligencia artificial. Lo más característico de esta es que sus algoritmos van aprendiendo automáticamente de la experiencia, gracias a los datos que van recogiendo. Con este aprendizaje, que es llamado *machine learning*, la inteligencia artificial desarrolla redes neuronales que hace que se modifiquen los procesos productivos en una industria en tiempo real con la información que se va recogiendo. La inteligencia artificial también puede tener otras funciones, como es el procesamiento de imágenes de manera automática, o la predicción de la evolución de un mercado.

2.2.6. Ciberseguridad

Esta digitalización de la industria no está exenta de problemas, y es por ello por lo que la ciberseguridad es tan importante. Al estar trabajando con dispositivos que se comunican a través de la red, se necesitan más herramientas para ayudar a prevenir los posibles ciberataques que se puedan hacer contra estos sistemas, o para recuperar estos dispositivos una vez han sido atacados. En la actualidad, la seguridad es lo que más preocupa a las empresas a la hora de cambiar su infraestructura de sistemas físicos por una infraestructura digital, ya que esta industria 4.0 es relativamente nueva y aun no se tiene una completa certeza de las vulnerabilidades que pueden ocurrir en estos sistemas, y de los ataques que pueden sufrir y cómo defenderse de ellos.





Figura 2.5: Pilares tecnológicos de la Industria 4.0

2.3. Retos de la industria de procesos

Esta cuarta revolución industrial o industria 4.0 no ha hecho más que arrancar, por lo que se nos presentan varios retos de diversa índole hasta alcanzar la digitalización total y la creación de un Gemelo Digital completo. Aquí vamos a ver algunos de los retos más importantes que hay en la industria de procesos.

2.3.1. Retos tecnológicos

Uno de los retos que existe en esta automatización industrial, es alcanzar un acuerdo para la estandarización de las interfaces de comunicación entre dispositivos. Esto es importante, ya que, en una industria 4.0, favorecería mucho a la comunicación entre equipos y maquinaria que la interfaz de estas fuera compatible. En la actualidad, bajo el liderazgo de Alemania, hay varios países negociando para llegar a este fin.

En la búsqueda de la estandarización de las interfaces aparecen protocolos orientados a la comunicación entre equipo tales como OPC⁴ UA, la última versión de la tecnología OPC, o el protocolo S7 de Siemens⁵. Estos protocolos hacen uso de conectores estándar que pueden ser de tipo Ethernet RJ45, los cables de red más comunes, o de otros tipos, como Profibus o ProfiNet⁶, esta última también basada en Ethernet. Estos

⁴ ¿Qué es un Servidor OPC? Disponible en <https://www.matrikonopc.es/opc-servidor/index.aspx>.

⁵ Siemens communication overview. Disponible en http://snap7.sourceforge.net/siemens_comm.html.

⁶ 1 de marzo de 2018. PROFIBUS vs PROFINET: Estrategias de Comparación y Migración. Disponible en: <https://profibus.com.ar/profibus-vs-profinet-estrategias-de-comparacion-y-migracion/>.

tipos de conectores sustituyen a los antiguos puertos serie de 9 pines, que poco a poco van quedando obsoletos.



Figura 2.6: Puerto serie de 9 pines.

Al ser todo tan digital y automático, es muy importante que los algoritmos cometan cada vez menos errores, ya que cuanto más se use, más costoso será cualquier fallo que se encuentre en estas instrucciones.

Otra de las tareas pendientes de la industria 4.0 es la mejora en la ciberseguridad, que debe de ser directamente proporcional a la digitalización de la industria, ya que cada vez habrá más ciberataques. Solo en 2017, en España se registraron 1200 ciberataques⁷, y este número no hará más que subir con el tiempo, pues cada vez serán más las empresas que irán digitalizándose. Es por esto por lo que es tan importante aumentar el esfuerzo en ciberseguridad a medida que se avanza en la transformación digital.

2.3.2. Retos sociales

En el aspecto social, se necesita contribuir a que los trabajadores se vayan familiarizando con las nuevas tecnologías que se usen en sus respectivas empresas. Esto es una tarea difícil, ya que hay personas que no usan normalmente los dispositivos tecnológicos que tenemos a nuestra disposición hoy en día. Es por eso por lo que es clave tutorizar a los trabajadores de las empresas que van digitalizándose en estas nuevas tecnologías. En el aspecto de la educación, se debe promover el uso de estos dispositivos en colegios e institutos para que los alumnos salgan preparados para afrontar los nuevos retos que surgen con la industria 4.0.

Otro de los temas a tener en cuenta es el riesgo de que un trabajador pierda su empleo por culpa de la digitalización de su empresa. Es por ello por lo que aquellos trabajadores

⁷ Albert López (7 de marzo de 2018) España duplica hasta los 1.200 el número de ciberataques de alto riesgo en 2017. Disponible en: [https://www.lavanguardia.com/seguros/empresa/20180307/462104963921/espana-duplica-hasta-los-1-200-el-numero-de-ciberataques-de-alto-riesgo-en-2017.html#:~:text=Los%20ciberincidentes%20registrados%20por%20el,han%20duplicado%20en%20un%20a%C3%B1o&text=Espa%C3%B1a%20acabar%C3%A1%20el%20a%C3%B1o%202017,Nacional%20\(CCN%2DCERT\) .](https://www.lavanguardia.com/seguros/empresa/20180307/462104963921/espana-duplica-hasta-los-1-200-el-numero-de-ciberataques-de-alto-riesgo-en-2017.html#:~:text=Los%20ciberincidentes%20registrados%20por%20el,han%20duplicado%20en%20un%20a%C3%B1o&text=Espa%C3%B1a%20acabar%C3%A1%20el%20a%C3%B1o%202017,Nacional%20(CCN%2DCERT) .)

que sean más vulnerables a este cambio tienen que recibir una buena capacitación para que tengan las habilidades necesarias para afrontar esta evolución en su trabajo.

La paridad de género también hay que tenerla en cuenta en esta transformación digital, ya que, como se sabe, en los trabajos relacionados con la informática y con la tecnología, el número de hombres es bastante superior al de mujeres. Por ello se debe incentivar la incorporación de la mujer al sector de la tecnología y, además, su ingreso en aquellas carreras que son más técnicas, como las ingenierías o las carreras de computación.

2.3.3. Retos regulatorios

Como en todos los ámbitos de la vida, en el aspecto de la transformación digital también se necesitan nuevas normas, leyes e instituciones que regulen esta digitalización. En concreto se necesitan leyes que garanticen el acceso de todo el mundo a estas tecnologías, y leyes que protejan los datos e información de las personas y de las empresas, pues en la actualidad existe muchísima información personal en la red que es necesario proteger.

Otros aspectos a tener en cuenta en el tema jurídico es la ley de propiedad intelectual⁸, donde habría que revisar la efectividad de las patentes o del registro de las marcas, ya que en el mundo digital no queda claro cómo se protege esta propiedad intelectual.

Una de las cosas más útiles que han surgido gracias a las tecnologías de la información y la comunicación son las redes sociales, ya que nos permite estar en contacto con cualquier persona del mundo sin movernos de casa. Sin embargo, como todo, también tiene su parte mala, y es que son muy útiles para los criminales, ya que con el anonimato y el alcance que puede llegar a tener su mensaje, estos grupos criminales pueden alentar más usuarios para sus causas, y de forma mucho más rápida y sencilla.

Esta transformación digital también necesita de una correcta regulación en el libre mercado, ya que se debe evitar que aparezcan monopolios que acaparen todo el mercado de un sector. Para ello se debe de promover una política tributaria adecuada basado en los ingresos que una empresa obtenga con este modelo de negocio, y se debe de facilitar la participación en este mercado, teniendo en cuenta que hay empresas que solo entran de manera esporádica en este mercado. Por lo tanto, se debe de revisar el sistema tributario en este mercado para evitar que a una empresa le desanime entrar en este modelo de negocio si debe de cambiar su régimen tributario o darse de alta en uno nuevo.

Una de las soluciones que han propuesto algunos países y personalidades es una tributación en base a la robotización de cada empresa. De esta forma, la pérdida de poder adquisitivo que supondría estas pérdidas de trabajo se podría suplir con un impuesto que permita mejorar los servicios sociales de un país, y que permita que las empresas sigan

⁸ Texto Refundido de la Ley de Propiedad Intelectual (22 de abril de 1996).
<https://www.boe.es/buscar/act.php?id=BOE-A-1996-8930>

aportando la gran cantidad de ingresos que se pierden al sustituir a los trabajadores por estas máquinas y robots [4].

Por otra parte, gracias a tecnologías como *blockchain*⁹, también es posible la aparición de monedas digitales, como es el *Bitcoin*¹⁰. Este tipo de monedas, al ser relativamente nuevas, no están reguladas, y pueden ser usadas de forma ilegal, ya que las transacciones son anónimas y no se hacen a través de bancos como sí se hace con el dinero físico. Es por esto por lo que los países ya están manos a la obra para intentar tener controlado el flujo de estas criptomonedas.

⁹ Javier Pastor (23 de septiembre de 2018). Qué es *blockchain*: la explicación definitiva para la tecnología más de moda. Disponible en: <https://www.xataka.com/especiales/que-es-blockchain-la-explicacion-definitiva-para-la-tecnologia-mas-de-moda>

¹⁰ Más información sobre *Bitcoin*: <https://bitcoin.org/es/faq#que-es-bitcoin>



3. Servicios de automatización en la industria

La automatización en la industria se compone de diversos y complejos procesos. La industria es un área de la economía que tiene muchos sectores y áreas de negocio distintas, por lo que los servicios que se usan en un sector no tienen por qué ser los más apropiados en otro.

3.1. Tipos de industria y áreas de negocio

La industria es un área muy grande, que cubre muchos y muy variados sectores de la sociedad. Es por eso por lo que es necesario dividirla en sectores basándose en los productos que elaboran cada una. En Maval se encuentran tipos de industria distintos en las que, a su vez, se encontrarán áreas de negocio distintas, que usarán distintas soluciones para la automatización de sus procesos de producción.

3.1.1. Industria de alimentación y bebidas

La industria de alimentación y bebidas es una de las más importantes, ya que nos permite abastecer toda la demanda de alimentos que tenemos en la actualidad. Para esta industria se necesitarán, además de tecnologías para la automatización, tecnologías para la gestión de producción, que deberán estar integradas con el resto del proceso de producción.

En esta industria se necesita tener buen control de la producción para comprobar qué es lo que se puede mejorar en este proceso, como productos cuyo margen de beneficio sea mayor, o para detectar aquellos procesos que estén ralentizando la producción. Para ello, se tienen que implementar soluciones que mejoren el tratamiento de la materia prima, que dosifiquen de forma automática los ingredientes que se usarán para cada producto, o que mejoren el sistema de almacenamiento y envasado del producto.

Gracias a esta industria de la alimentación se pueden obtener los distintos alimentos que ingerimos todos los días, como son las bebidas y alimento líquido, la fruta y verdura, la panadería y la bollería, los productos de pescado, los platos preparados, etc. Además, en esta industria también se tiene un control de los mataderos, con su consiguiente procesamiento de la carne obtenida de los animales.



Figura 3.1: Proceso de producción en la industria de la alimentación.

3.1.2. Industria aeronáutica

Otro tipo de industria muy importante es la industria aeronáutica, pues cada vez tiene más peso en lo que a desplazamientos y transporte de mercancías se refiere, y se está convirtiendo en el medio de transporte de referencia, tanto para el transporte de personas, como de logística.

La industria aeronáutica necesita soluciones para monitorizar el proceso de fabricación de sus productos, por lo que usa sistemas LMS (las siglas de *Line Monitoring System* en inglés que se verá más adelante), PDR (*Preliminary Design Review*), CDR (*Critical Design Review*), y FDR (*Final Design Review*). Todos ellos sirven para supervisar el proceso de fabricación de los distintos productos de esta industria. Además, como todo proyecto que se lleva a cabo en la industria, también será necesario un análisis funcional del proceso de fabricación. Gracias a este análisis se podrá saber qué hace nuestro cliente y de qué forma, y lo que se espera que hagamos.

Con estas soluciones se pueden controlar los sensores, actuadores, robots, PLCs, etc. que se usen en el proceso de producción, o crear interfaces Web que supervisen cada parte del proceso en todo momento, notificando cada vez que haya alguna incidencia en este proceso. Para ello integran, por ejemplo, los sistemas ANDON [5], que son los encargados de dar la señal de alarma cuando algo ha fallado. También se tienen que integrar aquellas máquinas que se usan en planta, como pueden ser cortadoras, plegadoras, prensas, láseres, etc.

En cuanto a la gestión de la producción MOM (que es la encargada de conseguir una mayor eficiencia y flexibilidad en la fabricación de los productos), se usarán órdenes de producción automatizadas, sistemas *PULL*¹¹ para solicitar los materiales necesarios para los productos dentro de la cadena de valor, o la recogida de datos que sirvan de forma

¹¹ 20 de febrero de 2018. ¿En qué consiste el sistema *pull*? Disponible en: <https://blog.infaimon.com/sistema-pull-ventajas-desventajas/>

estadística para indicar la calidad de estos productos (KPI y SPC). También se deben de gestionar los tiempos de ejecución de las operaciones de trabajo OEE/DTM¹²

Esta industria es de las más complejas, ya que cualquier fallo en el ensamblaje o la manufactura de los componentes que se usan para el transporte por el aire puede resultar muy caro. Es por eso por lo que las soluciones utilizadas en esta industria deben de ser de la mayor calidad posible.



Figura 3.2: Producción en la ingeniería aeronáutica.

3.1.3. Industria de productos químicos

También se pueden encontrar otros tipos de industria, como la industria de productos químicos, en la que se necesitarán distintas soluciones para, por ejemplo, llevar un control de la producción de biocarburantes y de nuevas energías. También hay que controlar el proceso de transformación de plásticos a través de sistemas de inyección, de soplado o de calandrado, y gestionar las materias primas o el almacén y el stock de estas. La gestión de fertilizantes también es necesaria dentro de este tipo de industria, con el formulado de ingredientes, los sistemas de filtración y el tratamiento del agua.

Esta industria es clave en nuestra economía, pues los productos que aquí se obtienen además de para el uso cotidiano de los consumidores, también pueden servir para otras industrias como materias primas. Sin embargo, por la naturaleza de esta industria, a veces es criticada por el impacto que tiene en el medio ambiente o por lo peligroso que

¹² 13 de septiembre de 2005. Siemens presenta un nuevo software para analizar la eficiencia de plantas. Disponible en: <http://www.interempresas.net/MetalMecanica/Articulos/11132-Siemens-presenta-un-nuevo-software-para-analizar-la-eficiencia-de-plantas.html>

puede ser para la seguridad de las personas si estos productos no son tratados de forma correcta. Pese a ello, esta industria debe de seguir funcionando, pues es muy importante para la fabricación de productos que se usan en nuestra vida cotidiana como pueden ser detergentes, cosméticos, etc.



Figura 3.3: Estación de producción en la industria de productos químicos.

3.2. Necesidades de digitalización y automatización de las empresas

Las empresas cada vez tienen que soportar una demanda mucho mayor por parte de los clientes, por lo que necesitan que sus procesos de producción sean más eficientes y fiables. Con los avances tecnológicos que se han producido en los últimos años, cada vez son más las empresas que están pensando en digitalizarse y automatizarse para buscar este objetivo. Sin embargo, este proceso no es fácil, ya que, para lograr los resultados óptimos, requiere de digitalizar por completo su cadena de valor.

3.2.1. Digitalización de la cadena de valor

Para lograr esta digitalización y que una empresa funcione de forma automática, es necesaria la digitalización de su cadena de valor.

La cadena de valor sirve para planificar la producción de un producto y, de esta manera, conseguir que sea rentable para una empresa [6]. Los objetivos de la cadena de valor es mejorar el proceso de producción, reduciendo los costes y valorizando el producto. Para ello, se siguen 5 pasos que tendrán que ser digitalizados:

- **Diseño del Producto:** Para llevar a cabo un producto, primero hay que tener un diseño de lo que se quiere hacer. Para digitalizar este diseño se pueden usar servicios que diseñen aquellas partes del producto final que el cliente quiera, y que produzca un prototipado digital que le permita ver su diseño en 3D. Uno de los servicios más usados en Siemens para este fin es el *software* NX¹³. Gracias a este *software* se puede gestionar el ciclo de vida de un producto 3D, con herramientas que preserva la integridad de los datos, y que da soporte para optimizar este diseño.

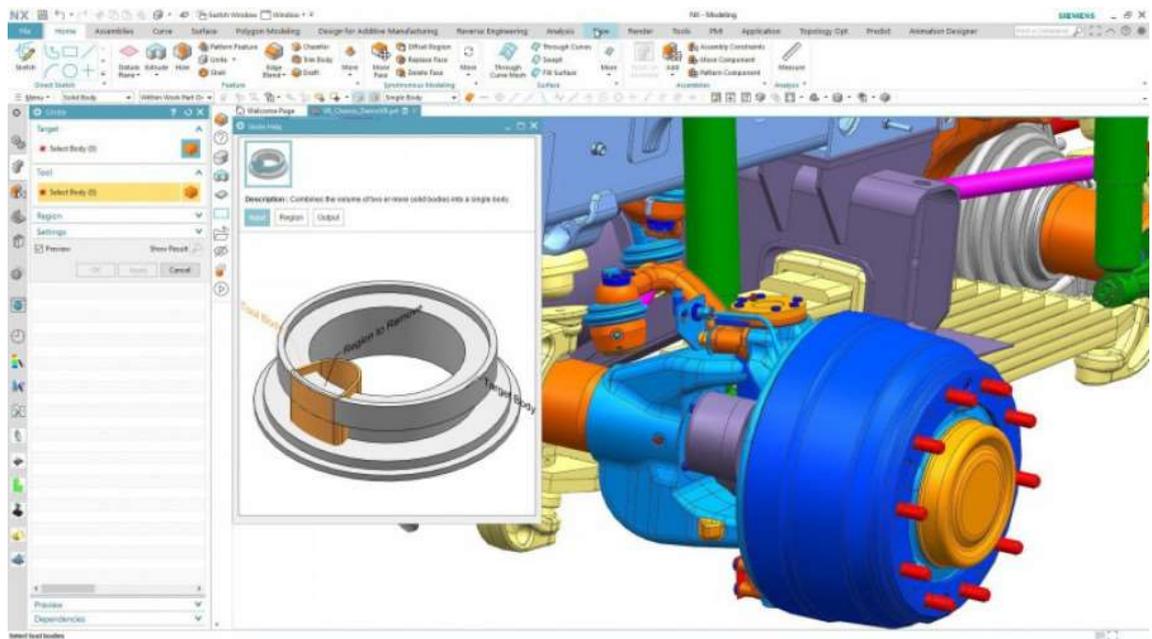


Figura 3.4: Ejemplo de prototipado digital en el *software* NX

- **Planeación de la Producción:** Otro de los aspectos a tener en cuenta al crear un producto es cómo se va a producir este. En este paso hay que decidir qué y cómo se fabrica, en qué orden hay que hacerlo, y con qué recursos. Se deberá de estar al tanto de los recursos de los que se dispone en todo momento.
- **Ingeniería de Producción:** En esta etapa entra en juego el I+D (investigación y desarrollo) de un producto. Se deberá definir la especificación del producto, y los parámetros de configuración que deberán de tener las máquinas que se usen en el proceso productivo. Será necesario tener un buen diseño y simulación eléctrica para evitar que haya un fallo en la planta y se deberá llevar a cabo una puesta en marcha virtual de la producción del producto.

¹³ Más información sobre NX:

<https://www.plm.automation.siemens.com/global/es/products/nx/>

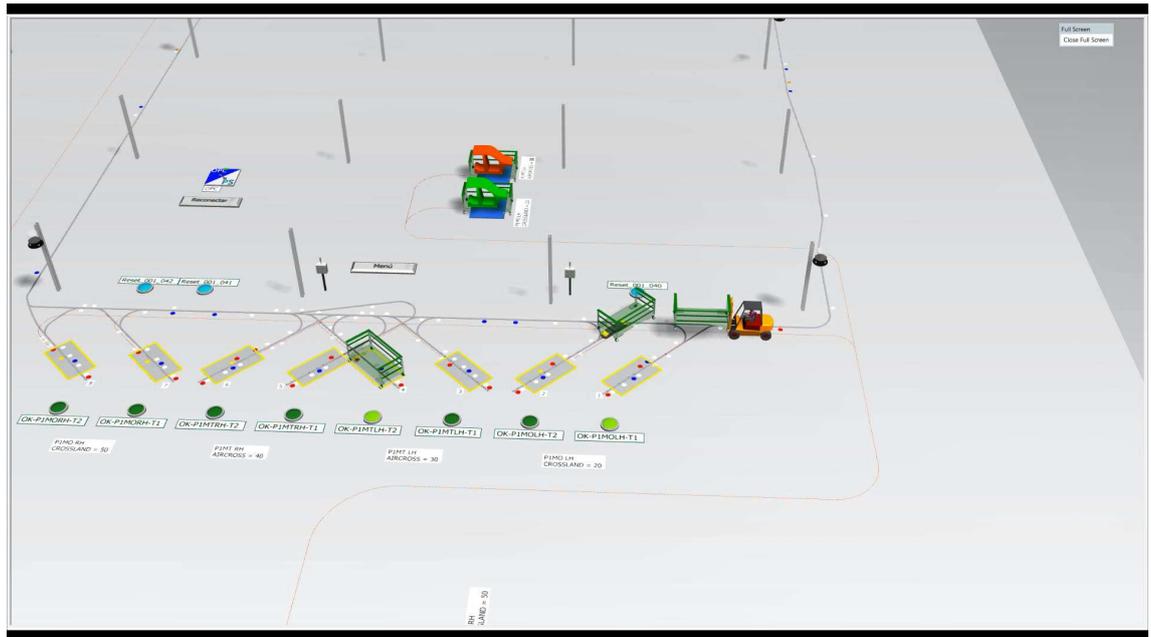


Figura 3.5: Ejemplo de gestión del ciclo de vida de un producto (PLM)¹⁴

- Ejecución de la Producción:** Aquí se ejecutará el proceso de producción que hemos ido definiendo en los pasos anteriores. Al digitalizarlo se tendrá que poder controlar aquellas órdenes que realicen algún proceso automático en la planta, y la elaboración constante de informes que sirvan para comprobar que todo funciona correctamente y, en caso de que no, arreglarlo. También se podrá vigilar la eficiencia energética de nuestro proceso para buscar soluciones en caso de que no sea la correcta. En este paso se deberá de tener una comunicación directa con los elementos de la capa de control y con la capa de negocio de las empresas (ERP).
- Servicios:** Por último, un producto necesita de un monitoreo y supervisión constante para buscar la máxima eficiencia y, una vez que se haya enviado el producto al cliente, llevar a cabo una mejora continua del mismo, buscando siempre las mejores soluciones. Para digitalizarlo se necesitarán plataformas como *MindSphere*¹⁵, que es un ecosistema basado en la nube de Siemens que conecta todos los dispositivos que se encuentran en un proceso de producción para obtener un buen monitoreo de todas aquellas máquinas y variables que participan en este proceso.

¹⁴ Software de gestión del ciclo de vida del producto (PLM). Disponible en: <https://www.plm.automation.siemens.com/global/es/our-story/glossary/product-lifecycle-management-plm-software/12506>

¹⁵ Más información sobre *MindSphere*: <https://new.siemens.com/es/es/productos/software/mindsphere.html>

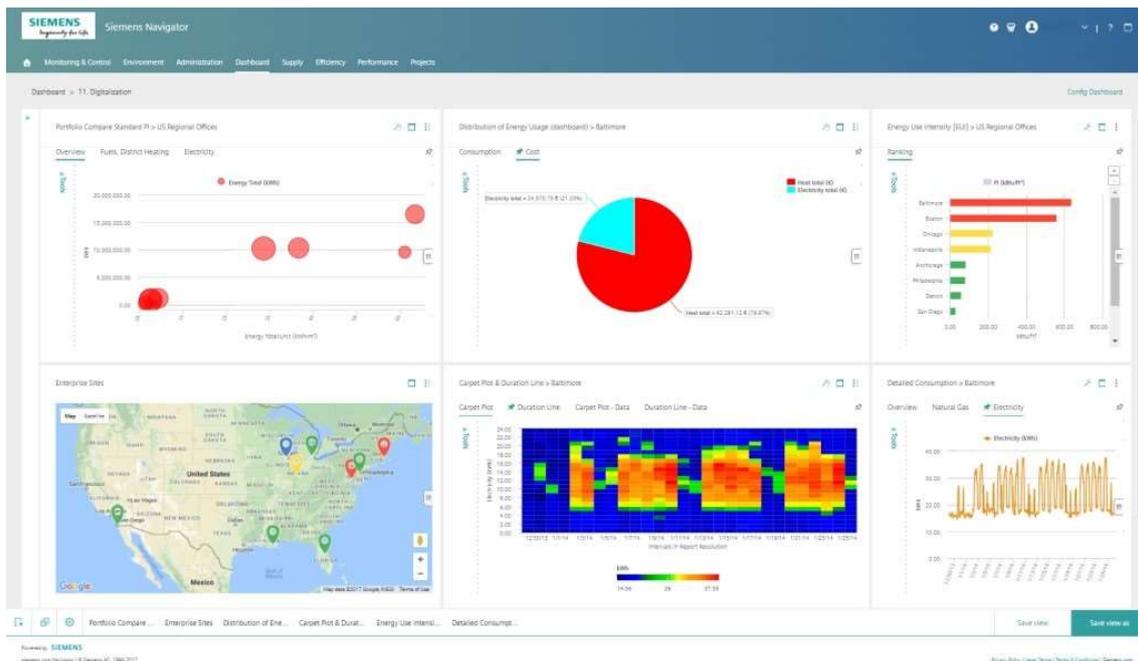


Figura 3.6: Ejemplo de *Energy Analytics* de Siemens basado en *Mindsphere*.

Gracias a la digitalización de esta cadena de valor, se puede obtener una simulación virtual de lo que pasará en el proceso de producción real. Con los datos que se recogen de estas simulaciones, se irá mejorando este proceso en cada simulación que se haga hasta conseguir un proceso de producción que consideremos óptimo.

3.2.2. Diseño del Gemelo Digital

Con esta digitalización de la cadena de valor, se acabará obteniendo un Gemelo Digital del producto que se quiera producir. Un Gemelo Digital es un ente virtual del objeto físico que se busca producir, y que irá evolucionando con los datos que reciba de este. Esto es muy importante para la mejora continua del producto [7].

El Gemelo Digital no es único en el proceso de producción, sino que se producen varios en la cadena de valor. Por una parte, está el Gemelo Digital del producto, que se trata del producto virtual, que debe de ser verificado por el siguiente Gemelo Digital, que es el Gemelo Digital de la producción. En Gemelo Digital, y con las especificaciones obtenidas del producto virtual se debe llevar a cabo una producción virtual, que servirá de base para la producción real. Al validar la producción real este proceso podrá ser automatizado. Por último, una vez hayan sido validados ambos gemelos digitales, se obtiene el Gemelo Digital del rendimiento, que se consigue al obtener el producto real, y que permite predecir el rendimiento futuro del producto.

El objetivo del Gemelo Digital es conseguir un bucle cerrado entre el mundo virtual y el mundo físico, que permita tomar decisiones a lo largo del ciclo de vida de un producto y de las operaciones de producción.

3.3. Soluciones para la transformación digital

Para alcanzar esta transformación digital se deben de disponer de los servicios y de las soluciones adecuadas. Ahora se verán algunas de las más importantes.

3.3.1. Estándar ISA 95

El estándar ISA 95 es un estándar internacional creado por la ISA¹⁶ en el 1990 que busca la unión entre las funciones empresariales y los sistemas de control para reducir los errores y los costes que tienen la implementación de interfaces entre ambos sistemas [8].

Para lograr esta unión, el modelo ISA 95 propone una jerarquía, separada en cinco niveles, que a su vez se pueden agrupar en otras tres.

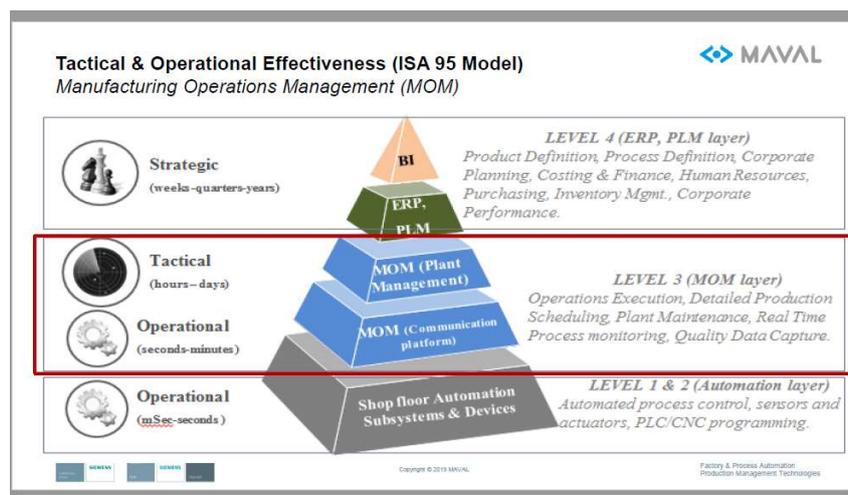


Figura 3.7: Pirámide del modelo ISA 95.

3.3.2. Level 1,2: Control y Automatización

En esta primera parte del estándar ISA 95 se encuentran aquellos procesos que han sido automatizados gracias a controladores, sensores y actuadores. En esta capa se

¹⁶ ¿Qué es ISA? Disponible en: <https://isa.org/sv/que-es-isa/#:~:text=Fundada%20en%201945%2C%20La%20Sociedad,mundo%2C%20m%C3%A1s%20de%2030.000%20miembros.&text=ISA%20es%20el%20patrocinador%20de%20la%20Federaci%C3%B3n%20de%20Automatizaci%C3%B3n>.

utilizan PLCs y CNCs¹⁷, que deberán de ser programadas para realizar de manera automática aquellas tareas que le corresponderían a un operario [9].

En estos niveles se logra la automatización de la planta de producción, gracias a los diferentes sistemas, máquinas y dispositivos de los que se disponen, obteniendo grandes beneficios como la reducción del coste de producción, el incremento de la productividad, la mejora de calidad del producto, un uso óptimo de las instalaciones disponibles, control del inventario o control de la contaminación medioambiental. De esta manera, se aumenta ostensiblemente la eficiencia del proceso de producción.

Para lograr esta automatización industrial se usan DCS (*Distributed Control Systems*). Existen una gran variedad de estos sistemas de control distribuidos, lo que dificulta identificar qué elementos son comunes en un DCS. Una posible clasificación de los niveles de estos sistemas podría ser la que se ve en la Figura 3.8, donde existe un ordenador central que maneja la planificación de la producción, y que va bajando hasta llegar al nivel de campo, pasando por el control de producción o por el nivel de supervisión de la planta.

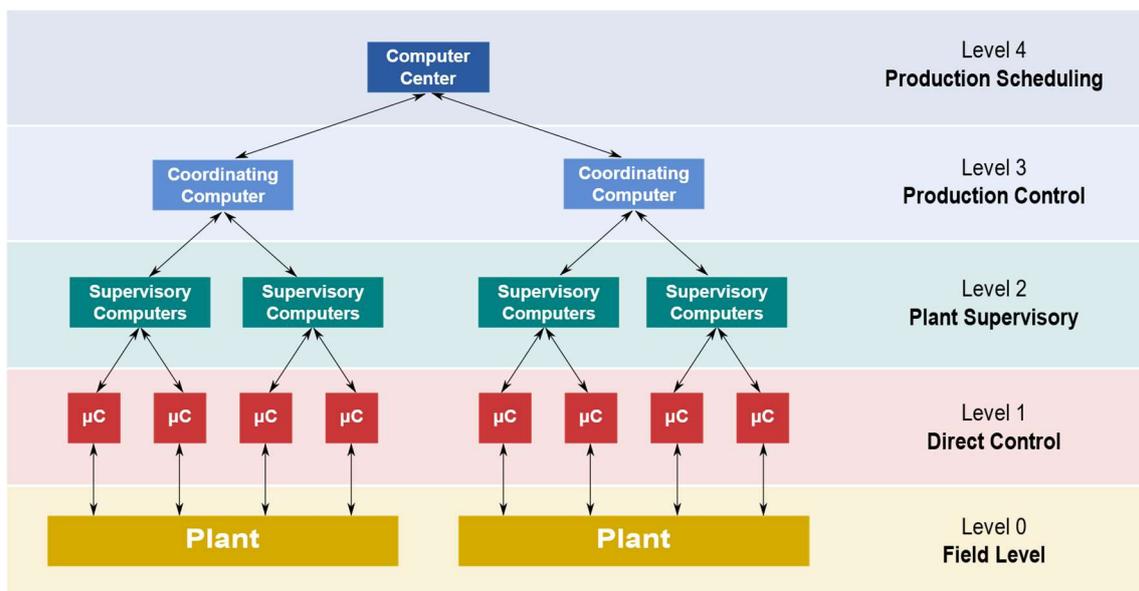


Figura 3.8: Niveles funcionales de un DCS.

Gracias a la *Manufacturing Intelligence* (MI) los datos que se obtienen en la planta se transforman en información que más tarde es enviada a los gerentes para mejorar la fabricación del producto. Con este *software* se podrá tener una visión de todas las máquinas que trabajan en cada una de las plantas de las que dispone una empresa, permitiendo resolver los problemas de calidad que se encuentren y obteniendo datos del proceso e información empresarial, junto a indicadores claves de rendimiento, con los que se podrán tomar las mejores decisiones empresariales. En Siemens se dispone de

¹⁷ 28 de diciembre de 2015. Introducción a la tecnología CNC. Disponible en: <https://www.demaquinasyherramientas.com/mecanizado/introduccion-a-la-tecnologia-cnc>



una herramienta de MI llamada *Opcenter Intelligence* que se verá más adelante con detalle.

3.3.3. Level 3: Operacional y Control de producción

Este es el nivel en el que se llevan a cabo las operaciones de manufactura (MOM) en el proceso de producción.

Esta fase se compone de varias operaciones. Por una parte, hay que definir un planteamiento detallado del proceso de producción para saber cómo se va a producir nuestro producto. Para ello, primero se tendrá que hacer una definición del producto, y un análisis de los recursos de producción de los que se disponen, para saber qué se puede, o no, producir.

Una vez que se tenga perfectamente definido el producto y el proceso de producción que se va a seguir, se podrán ejecutar las operaciones de producción que se han definido. Se deberá realizar un buen mantenimiento en tiempo real de este proceso, para recoger los datos necesarios que sirva para mejorar el proceso de producción. También se tendrá que realizar un análisis del rendimiento de este proceso.

Gracias a esta fase se podrán enviar los datos que hayan sido recabados de todo este proceso a la capa de negocio, donde se tomarán las decisiones adecuadas para mejorar el producto. Una vez que se disponga de las mejoras a realizar, se revisará la programación de los PLCs y las distintas máquinas automáticas que se tengan en la planta, con el fin de obtener un producto final de gran calidad.

De esta manera, se puede ver cómo el modelo ISA 95 conecta las capas de negocio y la de control a través de esta capa intermedia, la de MES/MOM. Gracias a este modelo, las operaciones que se llevan a cabo en un proceso de producción quedan perfectamente conectadas entre sí para obtener el mejor resultado final.

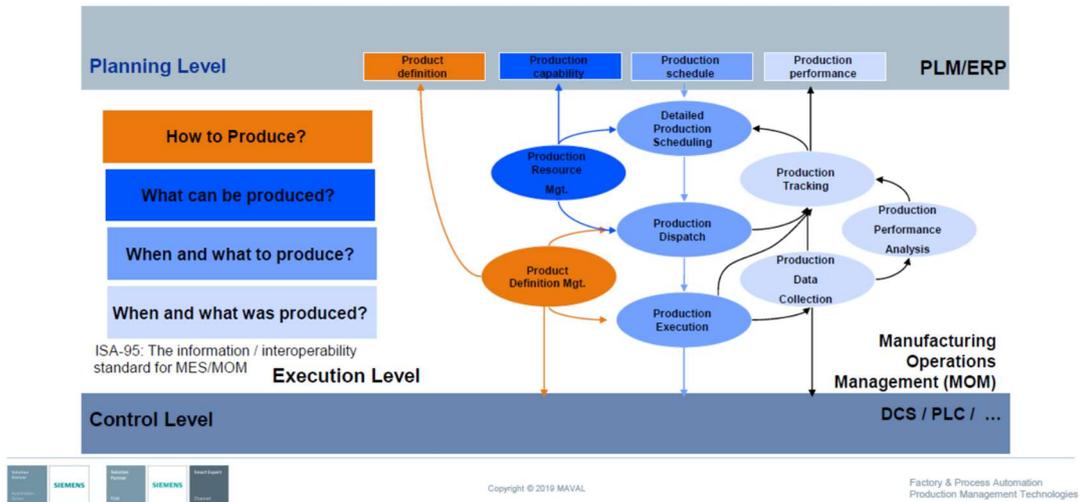


Figura 3.9: Proceso de producción en el modelo ISA 95.

3.3.4. Level 4 y 5: Capa de negocio y análisis y explotación de datos

Por último, se encuentran los dos últimos niveles, que se corresponden con la capa de negocio.

En el nivel 4 está la capa en donde se lleva a cabo un sistema de planificación de recursos empresariales (ERP)¹⁸, que incluye también los recursos humanos, y que se encarga de distintas operaciones dentro de una empresa, como puede ser la gestión del inventario o la compra de los distintos recursos necesarios para la producción de nuestro producto, teniendo en cuenta los costes de estos y gestionando las finanzas de la empresa.

También es en este nivel donde se encuentra el *Product Lifecycle Management* (PLM), con el que se gestiona todo el ciclo de vida de un producto, desde su definición hasta la obtención del producto final [11]. Este proceso es el encargado de, con los datos que se obtienen de la capa MOM, tomar las mejores decisiones en torno a lo que se quiere producir.

¹⁸ ¿Qué es un sistema ERP y para qué sirve? Disponible en: <https://www.ticportal.es/temas/enterprise-resource-planning/que-es-sistema-erp>



Figura 3.10: Componentes del diseño de un producto.

Con el PLM se podrá diseñar un producto, que consta de varios componentes. Primero está la gestión de las especificaciones, que se encarga de los materiales del producto. Después está la gestión de datos de proyecto de I+D, que se encarga de la investigación y de la modificación y mejora del proyecto que se quiera llevar a cabo. También se encuentra la Formula *Workbench*, que sirve para desarrollar la formulación del producto en tiempo real antes de llevar la fórmula a producción. Luego se encuentra el Cumplimiento Normativo, que es donde se confecciona el contenido de las etiquetas que tendrán los productos a producir, que tendrán que cumplir con la normativa existente en ese momento. También está el componente de Colaboración con Proveedores, donde se recibe información por parte de los proveedores para conseguir una mejor calidad en un producto, y por último la Gestión de Laboratorio, donde se hace un seguimiento de calidad de la fórmula para validar el producto.

Todos estos componentes tienen que estar conectados entre ellos para que el proceso de desarrollo quede simplificado. Gracias al I+D colaborativo no se adoptará una actitud conformista con el producto, y se seguirá desarrollando siempre dentro del cumplimiento de la regulación existente.

Los datos que se recogen en la capa del MES son analizados y explotados en el nivel 5, que es el que se corresponde con la *Business Intelligence* (BI), para tomar las mejores decisiones en cuanto al producto que se quiera producir. Es en este nivel en donde se define el producto que posteriormente se va a pasar al nivel del PLM/ERP, junto a la definición del proceso que se va a llevar a cabo para producir dicho producto.

Como se puede ver, gracias a este estándar ISA 95, todo el proceso de producción queda perfectamente conectado, desde los sistemas de control automatizados hasta la capa de negocio, gracias a los sistemas MES/MOM. Todos los niveles necesitan al anterior y, a su vez, también necesitan al posterior.

Cuando queramos producir un producto en una empresa, en el nivel 5 se analizarán y explotarán los datos de los que se disponen, pasándolos al nivel 4, donde en el proceso PLM se definirá nuestro producto y el proceso de producción que se llevará a cabo, y con ERP se tendrá en cuenta los recursos de los que se disponen para producir dicho producto.

Una vez que se tenga definido nuestro producto, se deberá de pasar a la capa operacional, donde se encuentran los sistemas MES/MOM, que son los encargados de

comunicarse con el nivel 1 y 2, donde se encuentran los sistemas de control automatizados, que son monitorizados constantemente por el nivel 3 para recoger los datos que se utilizarán después en la capa de negocio para tener una mejora continua de nuestro producto.

4. Sistemas MES/MOM

Con el paso del tiempo, cada vez se necesitan procesos de producción que sean más eficientes debido a la alta demanda y a la alta competencia que hay en la industria. Es por eso por lo que se necesita una forma de que los sistemas de control y la capa de negocio puedan comunicarse. Gracias a esto aparecieron los sistemas MES/MOM, que cumplen con esta función [12].

Al tener las operaciones de manufactura integradas, el tiempo de comercialización es más rápido, ya que los productos nuevos se introducen rápidamente al estar conectadas la capa de negocio con la operacional. También se consigue ahorrar en colaboración al optimizar los procesos, y se obtiene una gran consistencia en los datos que posteriormente deberán de ser analizados para tomar decisiones en la capa estratégica.

Con esta integración también se consigue que se incremente la flexibilidad y la transparencia del proceso, además de reducir la complejidad que tenían estos procesos anteriormente. Gracias a las herramientas PLM existe la posibilidad de llevar un historial con todas las versiones de un producto y, de esta forma, se podrán recuperar o revisar las distintas alternativas que se han introducido en el sistema. Esto es muy útil en el caso de que se encuentren problemas en el producto.

4.1. Situación de un sistema MES/MOM dentro de la cadena de valor

Siempre que se quiera crear un Gemelo Digital del producto a producir, se deberá tener en cuenta todos los elementos que integran la cadena de valor de una empresa. Estos sistemas MES/MOM se encuentran dentro de esta cadena de valor, facilitando las operaciones que se ubican dentro de la etapa de ejecución de la producción.

Como se ha visto anteriormente, la cadena de valor define varios pasos que sirven para obtener un Gemelo Digital de calidad. Entre todos ellos se encuentra la ejecución de la producción, que es el penúltimo paso de esta cadena de valor. Una vez que ya se tenga la especificación de nuestro producto definido, y que sabemos cómo se va a llevar a cabo nuestro proceso de producción, habrá que comunicarlo a la capa de control.

Los sistemas MES/MOM son los encargados de volcar la información que viene de la etapa de ingeniería de producción, y coordinar la maquinaria necesaria para que cumpla sus funciones de forma automática. De la misma forma, estos sistemas deberán de recoger los datos que les lleguen de estos sensores, actuadores, etc., para pasárselos a la capa superior y que sean analizados.





Figura 4.1: Esquema de un sistema MES/MOM

Gracias al sistema MES/MOM, todo el proceso de producción de una empresa queda completamente conectado, desde la capa de negocio, donde se toman las decisiones para un proceso de producción, hasta las máquinas y equipos automatizados que son los encargados de llevar a cabo dicho proceso.

4.2. Integración con sistemas de la capa operacional y estratégica

Para que este modelo de producción funcione, se necesita que los sistemas MES/MOM se integren correctamente con la capa operacional, que es en la que se encuentran las máquinas automatizadas, así como con la capa estratégica, donde se toman las decisiones que afectan al proceso de producción de la empresa.

En la capa de negocio, donde se encuentra el ERP, y con los datos que se tienen, se toman las decisiones sobre qué se va a producir y de qué recursos se dispone. Esta información es enviada a los sistemas MES/MOM y, desde el módulo de I+D que estos integran, se indica cómo se debe de producir el producto. Una vez que se tenga definido este proceso, el MES vuelca la información en los PLCs y demás máquinas automáticas que se tengan en las plantas de producción para coordinar las actividades a realizar.

Estos sistemas entonces, entre otras funciones, monitorizan en tiempo real el funcionamiento de estas máquinas, obteniendo una gran cantidad de datos y dejando un registro de las actividades que se han realizado en el MES. Posteriormente, estos datos serán volcados en la capa estratégica para que sean analizados y, de esta forma, intentar mejorar aquellos pasos del proceso de producción que tengan algún margen de mejora.

Los datos que se recojan en el MES se almacenan y se pasan a la capa de negocio en caso de que le sirvan al ERP de la empresa. El resto de datos relacionados con el proceso de producción se quedan almacenados para que desde el MES se intente mejorar este proceso.

Los sistemas MES/MOM son necesarios en esta transformación digital de la industria, ya que es muy importante una buena integración de los sistemas de la capa operacional y estratégica para ganar en eficiencia.

4.3. Operaciones integradas de manufactura

Gracias a estos sistemas, se ha conseguido que muchas de las tareas de manufactura que antes eran independientes queden correctamente conectadas. A continuación, se verá en detalle cada una de estas operaciones [13].

4.3.1. Gestión de los Recursos de Producción

Una de las operaciones que realizan es la de gestionar aquellos recursos que se destinan a la producción de cierto producto. De esta manera, los sistemas MES/MOM optimizan los recursos que se tengan a disposición, haciendo un uso más eficiente de los mismos. Una de las ventajas de estas operaciones es que permite optimizar el tiempo productivo de lo que se vaya a producir.

Entre el manejo de estos recursos se encuentran la gestión de máquinas, materiales e incluso las habilidades que posee un trabajador. También gestiona aquellos documentos que son necesarios para iniciar alguna de las actividades que se precisen en una planta de producción.

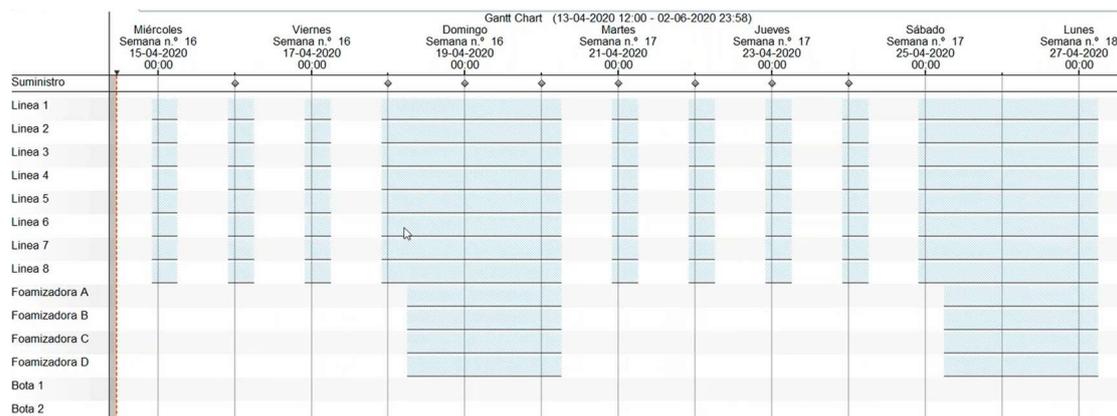


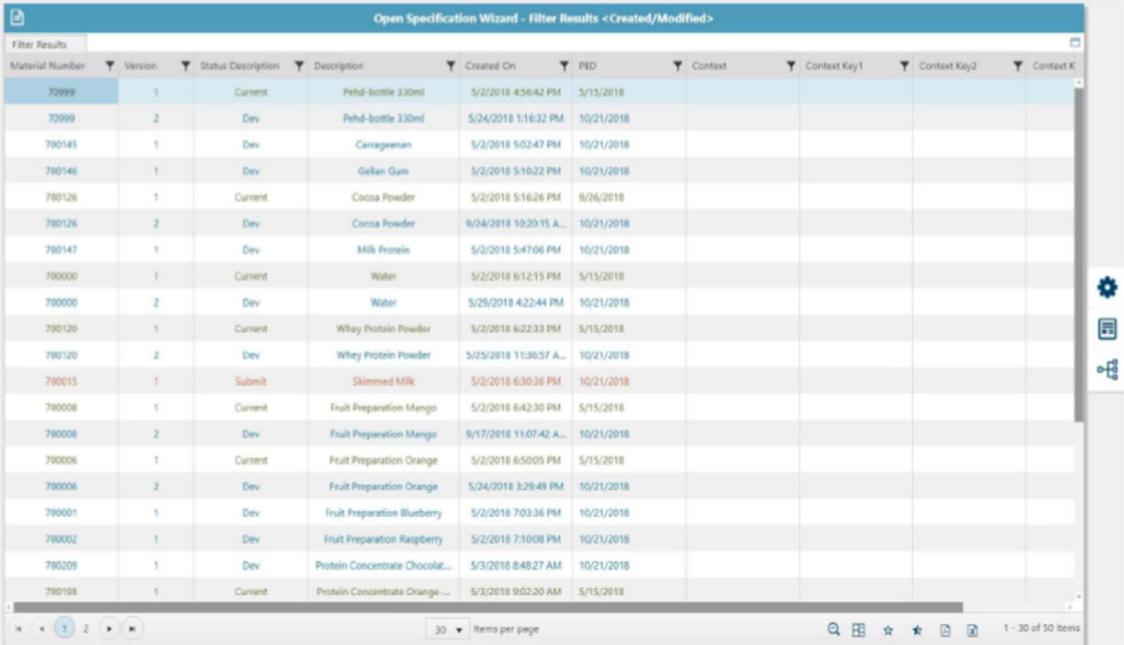
Figura 4.2: Ejemplo de gestión de recursos en una planta.

Como se puede ver en la Figura 4.2, los recursos se deben gestionar indicando los recursos de los que se dispone, y el tiempo que van a estar produciendo dichos recursos. El calendario y el horario de estos recursos se deberá de ajustar a los plazos que tenemos para producir el producto que queramos, obteniendo el sistema de producción más eficiente posible.

4.3.2. Gestión de la Definición del Producto

Otra de las funciones de los sistemas MES/MOM es gestionar la definición del producto que se vaya a producir. Cuando en una empresa se quiere producir algo, antes hay que definir las características del producto que se quiera producir y analizar las especificaciones del producto formulado que se han definido, produciéndose de esta manera una retroalimentación que sirve para mantener una mejora continua del producto.

Un producto tiene una gran cantidad de especificaciones que deben de ser gestionadas, pues existen muchas normativas referentes a un producto, y que pueden ser legales, de calidad, logísticas, nutricionales, medioambientales, etc. En la gestión de estas especificaciones hay que diferenciar entre aquellas que son globales y las que son específicas para una planta. Esta gestión debe de ser flexible y configurable, pero tiene que estar controlada para que no aparezcan problemas cuando cambien las especificaciones.



The screenshot displays the 'Open Specification Wizard - Filter Results <Created/Modified>' window. It features a table with the following columns: Material Number, Version, Status, Description, Created On, PID, Context, Context Key1, Context Key2, and Context Key3. The table lists various materials such as 'Pehd-bottle 330ml', 'Carnegeman', 'Gellan Gum', 'Cocoa Powder', 'Milk Protein', 'Water', 'Whey Protein Powder', 'Skimmed Milk', 'Fruit Preparation Mango', 'Fruit Preparation Orange', 'Fruit Preparation Blueberry', 'Fruit Preparation Raspberry', 'Protein Concentrate Chocolate', and 'Protein Concentrate Orange'. The status of each material is indicated, with some marked as 'Current' and others as 'Dev' or 'Submit'.

Material Number	Version	Status	Description	Created On	PID	Context	Context Key1	Context Key2	Context Key3
70999	1	Current	Pehd-bottle 330ml	5/2/2018 4:56:42 PM	5/15/2018				
70999	2	Dev	Pehd-bottle 330ml	5/24/2018 1:16:32 PM	10/21/2018				
700145	1	Dev	Carnegeman	5/2/2018 5:02:47 PM	10/21/2018				
700146	1	Dev	Gellan Gum	5/2/2018 5:10:22 PM	10/21/2018				
700126	1	Current	Cocoa Powder	5/2/2018 5:16:26 PM	9/26/2018				
700126	2	Dev	Cocoa Powder	8/24/2018 10:20:15 A.	10/21/2018				
700147	1	Dev	Milk Protein	5/2/2018 5:47:06 PM	10/21/2018				
700000	1	Current	Water	5/2/2018 6:12:15 PM	5/15/2018				
700000	2	Dev	Water	5/25/2018 4:22:44 PM	10/21/2018				
700120	1	Current	Whey Protein Powder	5/2/2018 6:22:33 PM	5/15/2018				
700120	2	Dev	Whey Protein Powder	5/25/2018 11:36:57 A.	10/21/2018				
700015	1	Submit	Skimmed Milk	5/2/2018 6:30:36 PM	10/21/2018				
700008	1	Current	Fruit Preparation Mango	5/2/2018 6:42:30 PM	5/15/2018				
700008	2	Dev	Fruit Preparation Mango	9/17/2018 11:07:42 A.	10/21/2018				
700006	1	Current	Fruit Preparation Orange	5/2/2018 6:50:05 PM	5/15/2018				
700006	2	Dev	Fruit Preparation Orange	5/24/2018 3:29:49 PM	10/21/2018				
700001	1	Dev	Fruit Preparation Blueberry	5/2/2018 7:03:36 PM	10/21/2018				
700002	1	Dev	Fruit Preparation Raspberry	5/2/2018 7:10:08 PM	10/21/2018				
700209	1	Dev	Protein Concentrate Chocolat...	5/3/2018 8:48:27 AM	10/21/2018				
700198	1	Current	Protein Concentrate Orange ...	5/3/2018 9:02:20 AM	5/15/2018				

Figura 4.3: Software de gestión de especificaciones de Siemens.

Gracias a esta gestión se podrán definir de forma más precisa los objetivos que se marquen a este producto, que sirve para tomar mejores decisiones a nivel estratégico. También se deben definir las fortalezas, debilidades, oportunidades y riesgos que tendrá el producto antes de comenzar el proceso de producción, para saber las decisiones que se tendrán que tomar con respecto al producto a producir.

Para gestionar la definición del producto se tendrán que definir las especificaciones y realizar distintas pruebas y ensayos de laboratorio, que podrán ser gestionadas con LIMS¹⁹. Para la realización de estas pruebas a veces se necesita que se envíe una parte de las especificaciones al MES para obtener información de planta. Gracias a estas pruebas se podrán optimizar los datos de calidad de un producto o la generación de informes, probando la ejecución de la producción del producto en distintos entornos.

4.3.3. Programa detallado de Producción

Para un buen proceso de producción, se tendrá que definir un programa detallado de dicho proceso. Este programa deberá indicar claramente la mejor forma de realizar cada tarea, y la manera de actuar cuando surja algún problema. Esta programación debe de proporcionar una buena secuenciación y sincronización entre sus tareas, y también debe de tener claras sus prioridades, las restricciones que va a sufrir o que puede sufrir, al igual que los conflictos que puedan aparecer. Se debe de hacer un monitoreo constante de la ejecución de dicho programa para minimizar la aparición de problemas no previstos.

Para la realización de este programa se necesita tener una buena planificación sobre nuestro proceso de producción. Es por esto por lo que se deberá tener claro qué es lo que se va a hacer, cuándo se quiere hacer, dónde, y los materiales y recursos que se necesitarán para llevar a cabo este proceso. Para cumplir con estos requisitos, se necesitará un modelo detallado de la planta en la que se van a producir los productos.

Hay que tener en cuenta que estos programas pueden cambiar, pues en ocasiones pueden aparecer conflictos que deban ser subsanados. Para estos casos se deberán usar, siempre en la medida de lo posible, reparaciones automatizadas en el programa, para que sea más eficiente.

¹⁹ ¿Qué es un Sistema de gestión de información de laboratorio (LIMS) Disponible en: <https://www.engusa.com/es/product/siemens-simatic-it-unilab>



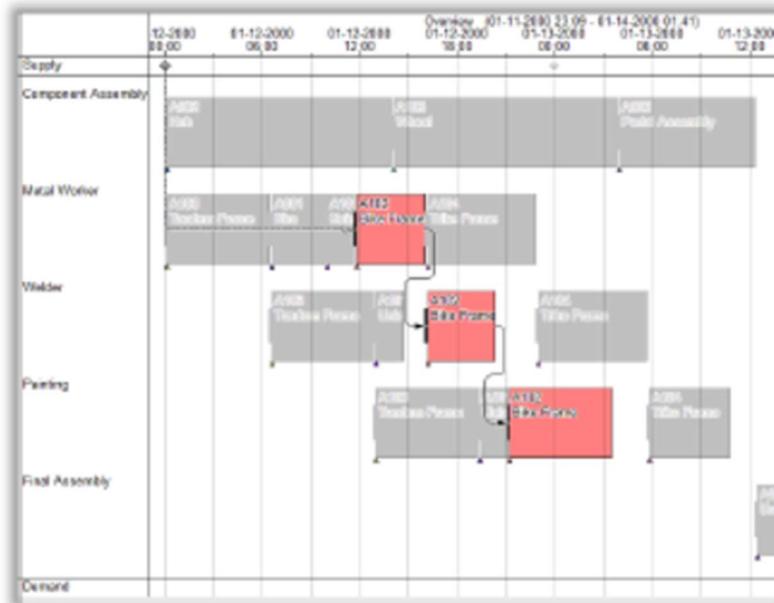


Figura 4.4: Ejemplo de un diagrama de Gantt²⁰

Una de las soluciones de Siemens que Maval implementa para la programación de la producción es la herramienta Preactor²¹. Con Preactor se pueden realizar planes de producción que tengan en cuenta aquellos imprevistos que puedan aparecer durante un proceso de producción, como pueden ser paradas o averías en las máquinas, ya que disponen de un sistema de simulación en tiempo real para analizar cada una de las situaciones que puedan surgir durante este proceso.

Desde ERP se le puede mandar la orden a Preactor de que ha habido un cambio en los pedidos o en la disponibilidad del material necesario para la producción, y Preactor responderá ajustando la programación para abordar estos inconvenientes. De la misma manera, también cambiará la planificación en caso de que desde MES se le indique que ha habido un problema durante el proceso de producción del pedido.

Gracias a esta herramienta se pueden crear planes y programas en minutos o incluso en segundos, y se pueden identificar problemas que pueden surgir durante la producción y anticiparse a ellos. Además, esta solución se puede adaptar prácticamente a cualquier tipo de programación que queramos, y tiene otras funcionalidades como la de recordar las secuencias preferidas, ser capaz de integrar casi cualquier origen de datos y la posibilidad funcionar de manera autónoma.

²⁰ Carmen Villanueva, 7 de diciembre de 2018. ¿Qué es y para qué sirve un diagrama de Gantt? Disponible en: <https://blog.teamleader.es/diagrama-de-gantt>

²¹ Más información sobre Preactor: <https://www.maval.es/preactor/>

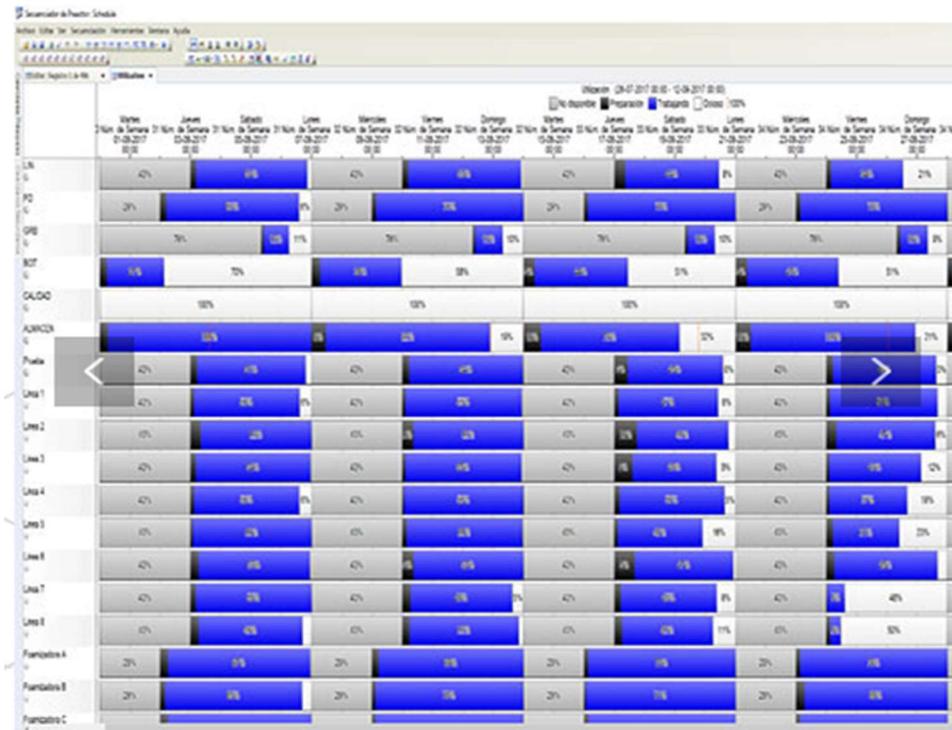


Figura 4.5: Simatic IT Preactor AS de Siemens

4.3.4. Despacho a Producción

Una vez que se tenga el programa que se va a seguir para llevar a cabo la producción del producto, es el momento de dar la orden para empezar a producir. En esta operación, los sistemas MES/MOM serán los encargados de indicar a las máquinas que comiencen a producir aquello que se requiera.

Para empezar, se tendrán que descargar las órdenes que tienen que ser enviadas a las máquinas, que tendrán que ser secuenciales en cada máquina para que no se empiece a hacer una orden antes de que termine la anterior. El programa de producción se tendrá que modificar de forma manual cada vez que se quiera añadir un nuevo cambio al mismo.

Una vez que se tenga todo esto configurado, las órdenes ya estarán listas para ser enviadas a las máquinas. En el sistema MES/MOM se podrán encontrar las órdenes que van a ser ejecutadas, con distintas propiedades como un identificador (que será único para cada orden), un nombre, el material del que se dispondrá para su realización, el estado en el que se encuentra, el momento en el que se quiera que empiece, o cuándo debería de acabar la orden, entre otras cosas.



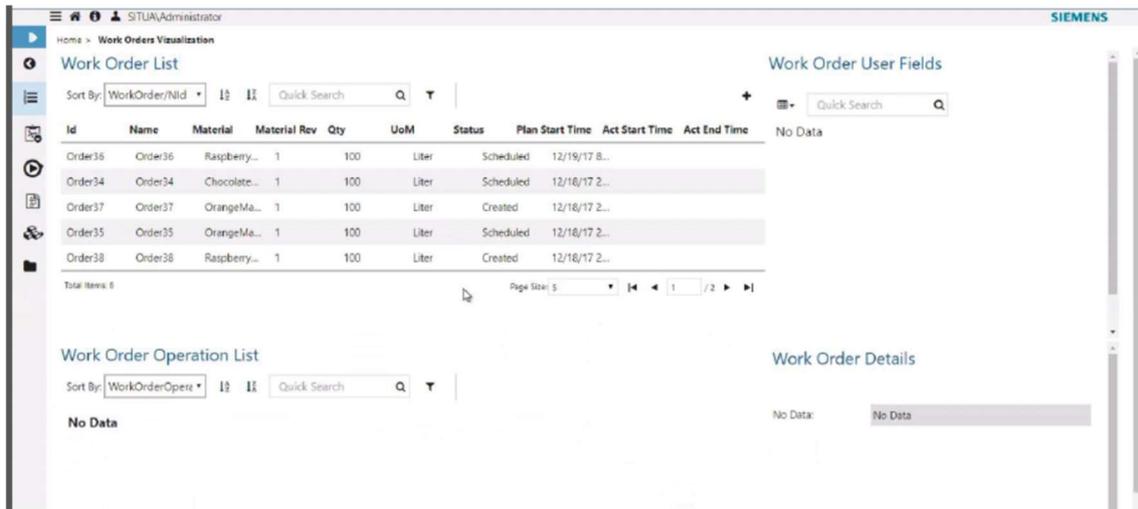


Figura 4.6: Ejemplo de la operación de despacho a producción.

4.3.5. Gestión de la Ejecución de la Producción

Una vez que se tenga el proceso de producción completamente definido, toca ponerlo en marcha, ejecutar las órdenes que se han definido anteriormente y gestionar todo este proceso.

Para un buen proceso de producción las tareas se deben ejecutar de manera orquestada y en el orden que se ha definido. Estas tareas deben de integrarse correctamente con la automatización para conseguir sacar la máxima eficiencia de estas operaciones, pero siempre asegurando la calidad del producto. En este paso se deben de gestionar todas las tareas presentes en la producción, tanto las tareas automáticas como las manuales.

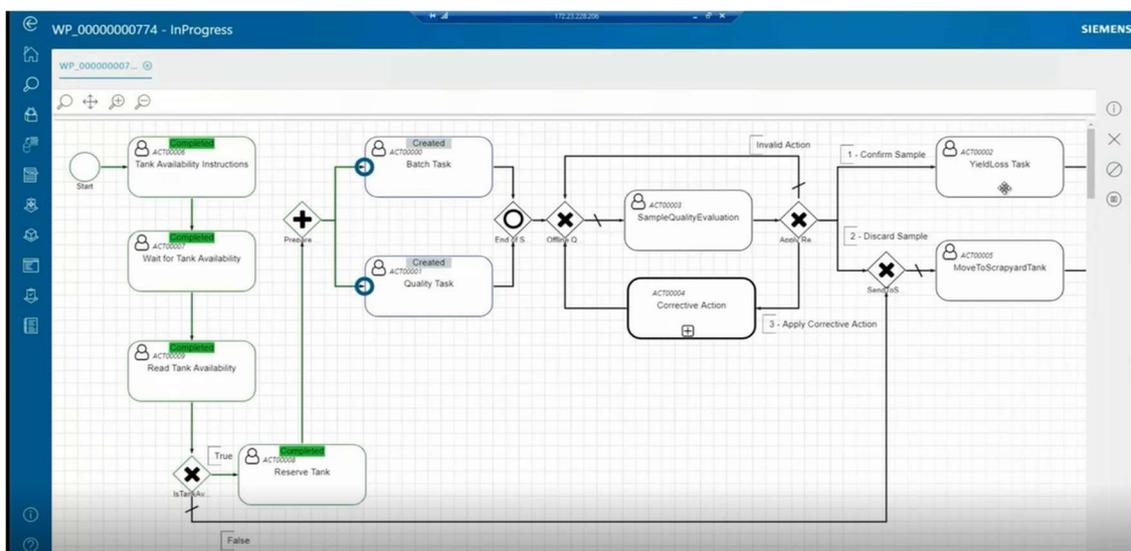


Figura 4.7: Ejemplo del flujo de tareas en un proceso de producción.

Para asegurar la calidad de nuestro producto se deben realizar operaciones de laboratorio, y hacer pruebas teniendo en cuenta las desviaciones que se pueden producir durante la producción. Para monitorizar la calidad del producto se deberán de producir reportes e informes, que serán utilizados más tarde para mejorar la calidad del producto. Esta calidad integrada se lleva a cabo durante el proceso de producción, y se consigue gracias a herramientas como Siemens Opcenter RD&L²²

Por último, para gestionar correctamente el proceso de producción, se tendrá que monitorizar en tiempo real el progreso de las órdenes que se están ejecutando. De esta manera se podrán recoger datos que posteriormente se transmitirán al equipo de I+D para continuar con la mejora del producto y la mejora del proceso.

4.3.6. Trazabilidad de la Producción

Una vez que se haya mandado a producción el producto definido en los pasos anteriores, se debe de seguir el proceso de producción en todo momento, teniendo conocimiento de dónde se encuentra el proceso en cada instante, y las tareas que se están ejecutando. Entre esta información se tiene que incluir apartados como las personas que están trabajando en estas tareas o los recursos de los que se está disponiendo para dichas tareas.

Gracias al seguimiento que se hace de la producción se generan informes para posteriormente añadir esta información a una base de datos. Este paso es muy importante para el equipo de I+D, ya que estos datos recogidos en producción les permite seguir desarrollando y mejorando este proceso.

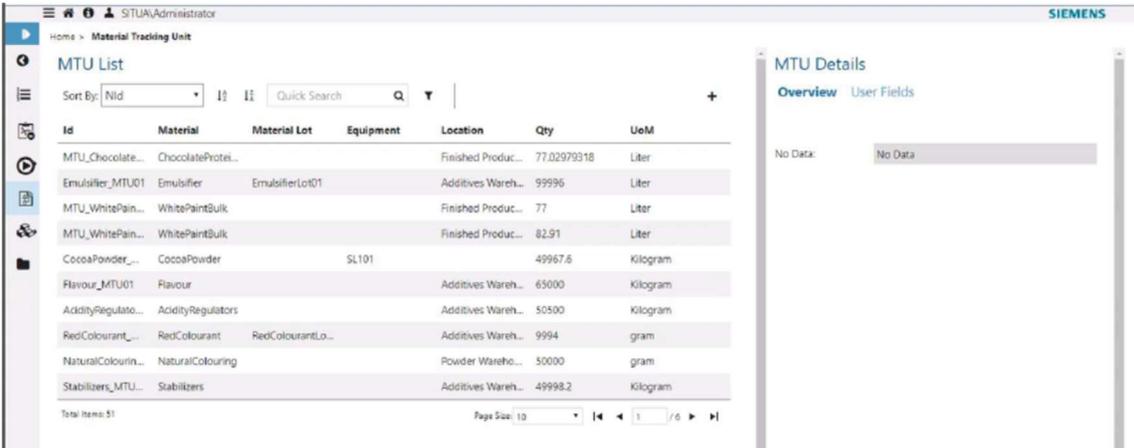
También se tiene que hacer un seguimiento de los proveedores que intervienen en este proceso, controlando la composición de los materiales, los lotes o los números de serie de estos materiales que son enviados por los proveedores.

Existen dos tipos de trazabilidad:

- **Trazabilidad hacia atrás:** esta es la trazabilidad que va desde el producto final hasta las materias primas que componen el producto. Con la trazabilidad hacia atrás, en caso de que exista una reclamación sobre uno de los productos, se puede saber al momento cuáles son los ingredientes que componen dicho producto.
- **Trazabilidad hacia delante:** lo mismo que la trazabilidad hacia atrás, pero al revés. En este caso, va desde una materia prima hasta los productos finales que se han obtenido con dicha materia. Gracias a esto, si tras producir un producto se detecta que es defectuoso, se puede saber cuáles son los lotes en los que se encuentra esa materia o componente y bloquear su comercialización.

²² *Opcenter Research, Development & Laboratory (RD&L)*. Disponible en: <https://www.plm.automation.siemens.com/global/es/products/manufacturing-operations-center/simatic-it-r-d-suite.html>

Como se puede ver, ambos tipos son importantes y cumplen su rol dentro del proceso de producción y, en específico, dentro de la etapa de trazabilidad de la producción.



The screenshot shows a Siemens Material Tracking Unit (MTU) interface. The main window displays a table of materials with columns for Id, Material, Material Lot, Equipment, Location, Qty, and UoM. The table lists various materials such as MTU_Chocolate, Emulsifier, WhitePaintBulk, CocoaPowder, Flavour, AcidityRegulators, RedColourant, NaturalColouring, and Stabilizers. The right-hand panel shows 'MTU Details' with 'Overview' and 'User Fields' tabs, and a 'No Data' message.

Id	Material	Material Lot	Equipment	Location	Qty	UoM
MTU_Chocolate...	ChocolateProtei...			Finished Produc...	77.02979318	Liter
Emulsifier_MTU01	Emulsifier	EmulsifierLot01		Additives Wareh...	99996	Liter
MTU_WhitePain...	WhitePaintBulk			Finished Produc...	77	Liter
MTU_WhitePain...	WhitePaintBulk			Finished Produc...	82.91	Liter
CocoaPowder_...	CocoaPowder		SL101		49967.6	Kilogram
Flavour_MTU01	Flavour			Additives Wareh...	65000	Kilogram
AcidityRegulato...	AcidityRegulators			Additives Wareh...	50500	Kilogram
RedColourant_...	RedColourant	RedColourantLo...		Additives Wareh...	9994	gram
NaturalColourin...	NaturalColouring			Powder Wareho...	30000	gram
Stabilizers_MTU...	Stabilizers			Additives Wareh...	49998.2	Kilogram

Figura 4.8: Trazabilidad de los materiales de un proceso de producción

4.3.7. Adquisición y Recolección de Datos de Producción

Una de las partes más importantes del proceso de producción es la de recolección de datos, ya que al realizar un análisis sobre estos se facilita la mejora de este proceso. En este paso, el sistema MES recoge la información que viene de la capa de control y automatización, es decir, de las máquinas y robots automáticos que se encuentran en la planta de producción. Para realizar estos análisis se tendrá que hacer uso de los KPI²³ (siglas en inglés de *Key Performance Indicator*), que te indica lo que se debe hacer para incrementar el rendimiento de alguna parte del proceso de producción [15].

Estos KPIs deben de ser monitorizados constantemente, pues son un elemento clave para el futuro de nuestra empresa. Para que una medida de rendimiento se convierta en KPI debe de ser sometida a pruebas, ya se debe asegurar que esta medida crea el comportamiento deseado. De no ser así, pueden producirse casos de que una medida de rendimiento de lugar a un empeoramiento en la producción de un producto.

Las máquinas dejarán datos relacionados con su fiabilidad, la calidad de los productos que está produciendo o el rendimiento de la máquina correspondiente. Estos datos serán entonces analizados en el MES para crear KPIs que mejoren el comportamiento de las máquinas y de los sistemas automáticos y, de esta manera, mejorar la eficiencia del proceso de producción.

4.3.8. Análisis del Desempeño de Producción

²³ Manuel Porras Blanco, 29 de septiembre de 2017. KPI's ¿Qué son, para qué sirven y por qué y cómo utilizarlos? Disponible en <https://blog.es.logicalis.com/analytics/kpis-qu%C3%A9-son-para-qu%C3%A9-sirven-y-por-qu%C3%A9-y-c%C3%B3mo-utilizarlos>

Una vez que se han recolectado suficientes datos del proceso de producción, se podrá llevar a cabo un análisis de estos datos para mejorar el desempeño de la producción. Este análisis del desempeño es el que vamos a implementar en este proyecto para la empresa Acram.

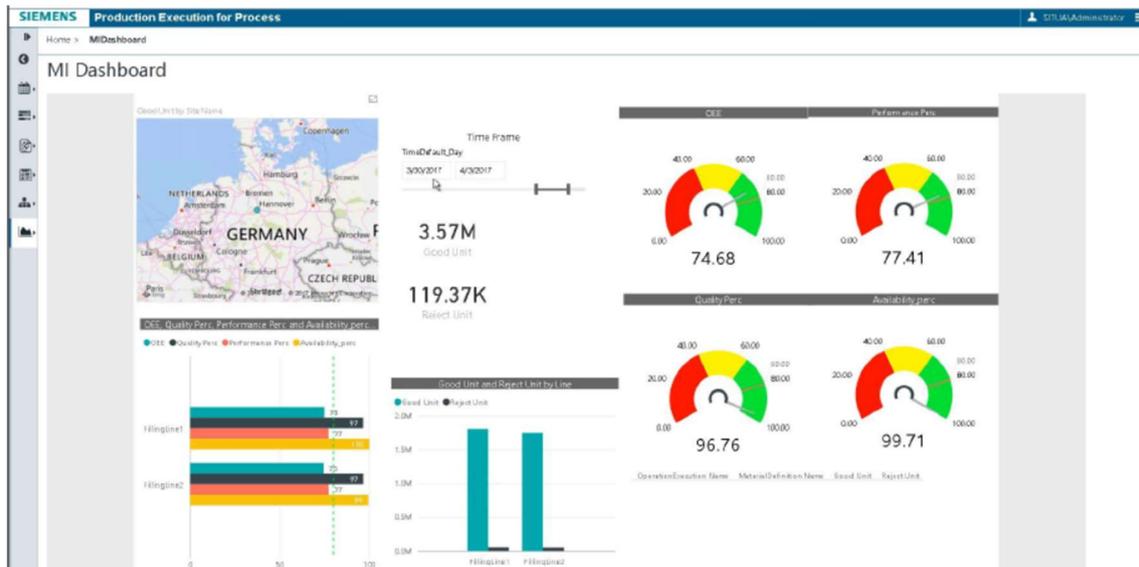


Figura 4.9: Ejemplo de análisis del desempeño de la producción

Como se puede ver en la Figura 4.9, con los datos que se han obtenido de las operaciones que se llevan a cabo en la planta, se pueden crear distintas gráficas que permitan analizar cómo se está desarrollando la producción. En caso de que haya algún fallo en el proceso se indicará en el MES, al igual que habrá distintos indicadores que nos muestren si una máquina está operando con un rendimiento por debajo de cierto límite, o si la calidad de los productos es la correcta.

Para la mejora de estas operaciones también se tendrá que hacer uso de los KPI, que se crean en base a los datos que se obtienen en el MOM durante el proceso de producción. Los KPI tendrán que ser analizados para los *stakeholders*²⁴, ya que al participar en una empresa de una u otra manera, se tendrá que decidir junto a ellos las estrategias y decisiones que se tomen en una empresa [14]. De esta manera, se conseguirá una mayor visibilidad del proceso de producción, lo que favorecerá la mejora continua del producto.

El análisis del desempeño de la producción se puede dividir en cuatro categorías según el tipo de operaciones que se vayan a gestionar.

²⁴ Camilo Parra, 8 de junio de 2020. Qué son los *stakeholders*, qué tipos existen y de qué manera impactan a una empresa. Disponible en <https://rockcontent.com/es/blog/que-es-un-stakeholder/>

4.3.8.1. Gestión de las Operaciones de Producción

Una vez que se estén ejecutando las órdenes del proceso anteriormente definido, se deben monitorizar estas operaciones para mejorar el proceso de producción. Para la gestión de estas operaciones se deberá hacer un análisis de rendimientos con los datos obtenidos de las operaciones. Tras estos análisis se tomarán las decisiones oportunas para tratar de mejorar estas operaciones de producción.

Una solución utilizada para esta gestión de las operaciones de producción o de fabricación es Opcenter *Intelligence* [16]. Esta solución tiene dos funcionalidades para este tipo de gestión. Una de ellas es la de transformar los datos que llegan de la producción en información que pueda ser utilizada para la mejora del proceso, como pueden ser los KPIs vistos anteriormente.

Otra de las funcionalidades que tiene Opcenter *Intelligence* es la de generar reportes de las distintas operaciones de producción, como se puede ver en la Figura 4.10, que nos muestra la eficiencia con la que está funcionando el proceso.

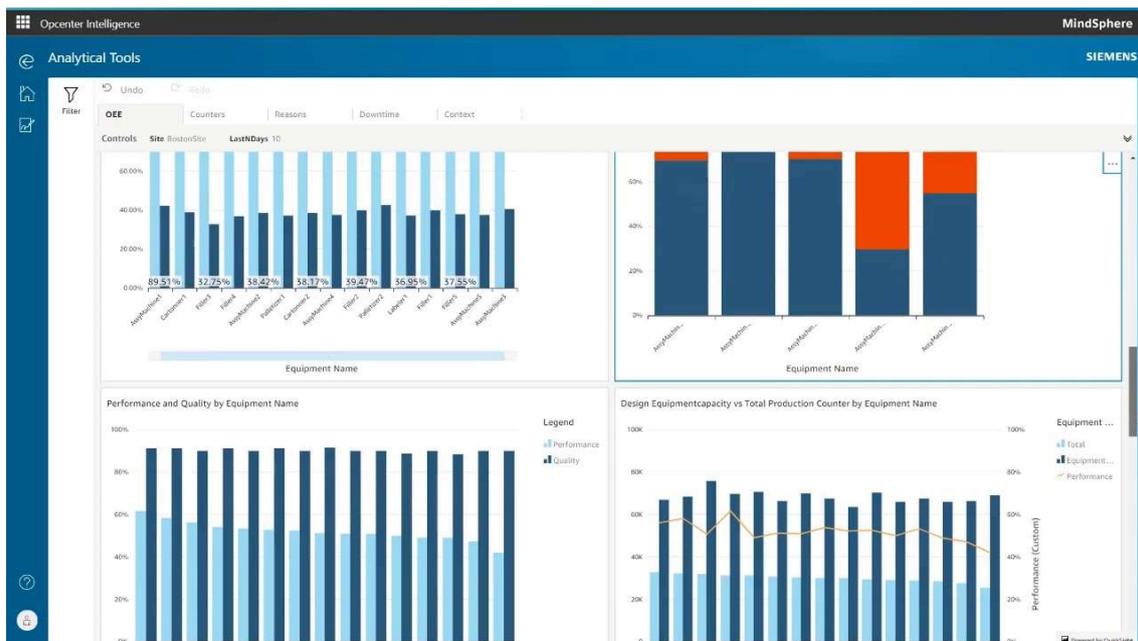


Figura 4.10: Ejemplo de reportes de OEE en un proceso de producción

4.3.8.2. Gestión de las Operaciones de Calidad en Producción

Durante la ejecución de la producción de un producto hay que llevar un control de calidad de este producto, además de la calidad de los proveedores. Para ello existen herramientas como Opcenter RD&L, que permite a las empresas crear nuevos productos de forma más rápida, y que tiene distintos componentes muy útiles.

Para la gestión de estas operaciones de calidad se usa la solución LIMS (*Laboratory Information Management System*), con las que se consigue una gestión eficiente de laboratorio. Una de las funciones de esta solución es la de gestionar la calidad de los datos que se obtienen de los laboratorios, optimizando su análisis y los informes generados.

LIMS también ayuda en los procesos de I+D que se realizan en los laboratorios con soporte en las pruebas, definiendo los métodos y ejecutando estas pruebas en distintos contextos. Gracias a esta solución, los costos en estas pruebas de calidad se reducen, ya que se produce un gran ahorro estructural y se simplifican estas pruebas.

The screenshot displays the LIMS software interface. At the top, there is a navigation bar with options like 'Home', 'Workspaces', 'New Search', 'Create Shortcut', 'Refresh', 'Status to Cancelled', 'Open SOP Document: Method', 'Assign Full Test Plan', 'Reanalysis', and 'Switch Screens'. Below this, the main area is titled 'Open Sample Wizard - Filter Results < Basic Task >'. It features a table of sample results with columns for 'ShortDescription', 'Description', 'CreationDate', 'StatusDescription', and 'Start Date'. The table lists several samples, including 'Protein Drink Chocolate Flavour' and multiple 'Cocoa Butter' samples with various statuses like 'In Execution', 'On Hold', and 'Available'. Below the table, there is a 'Results of Sample RQ20181204-0-001' section, which includes a search bar and a detailed table for 'Cocoa Butter (Chemical-Physical)'. This table has columns for 'Parameter', 'Low Limit', 'Low Spec', 'Value', 'Target', 'Unit', '#R', 'DueDate', and 'Status'. The data shows values for Oleic acid, Palmitic acid, and Stearic acid, with a total acids result of 102.80. To the right, a window titled 'SOP (Method): Acids' is open, displaying a PDF document titled 'Standard Operating Procedure for Gas Chromatography (GC)'. The document includes a file name, file type, and a download link, along with a preview of the SOP content.

Figura 4.11: Ejemplo de pruebas de laboratorio con la solución LIMS

Dentro del conjunto de soluciones que Siemens ofrece, se encuentra la solución Siemens Opcenter *Laboratory* para LIMS. Esta solución sirve para mejorar la eficiencia en los laboratorios de control de calidad y en los de I+D. En lo que a control de calidad respecta, con *Laboratory* se puede comprobar la calidad que hay en línea mientras está en marcha el proceso de producción, para luego analizar la muestra fuera de línea siguiendo los procedimientos marcados.

Esta solución tiene grandes beneficios, ya que es muy escalable y se puede ejecutar en muchos contextos, como pueden ser distintos lenguajes o distintos laboratorios. También permite comparar un producto con el de un competidor, siempre que se conozcan las especificaciones de este, para así comprobar cuales son las debilidades y fortalezas del producto.

4.3.8.3. Gestión de las Operaciones de Inventario en Producción



Durante un proceso de producción, es importante tener un control del inventario del que se dispone para este proceso. Esta gestión del inventario depende de las operaciones de producción que se realicen pues, dependiendo de las órdenes que se vayan a llevar a cabo, se podrá disponer de unos recursos u otros.

Los sistemas MES tienen funcionalidades para esta gestión de inventario, como lo es la gestión de inventario en proceso (WIP²⁵). El WIP permite la gestión del proceso de producción, evitando los cuellos de botella que se producen cuando se tiene un gran volumen de trabajo y los recursos son insuficientes. A los procesos se les puede añadir un límite WIP para evitar estos cuellos de botella y de esta manera conseguir un proceso más eficiente al detectar cuáles son las tareas que necesitan recursos adicionales.

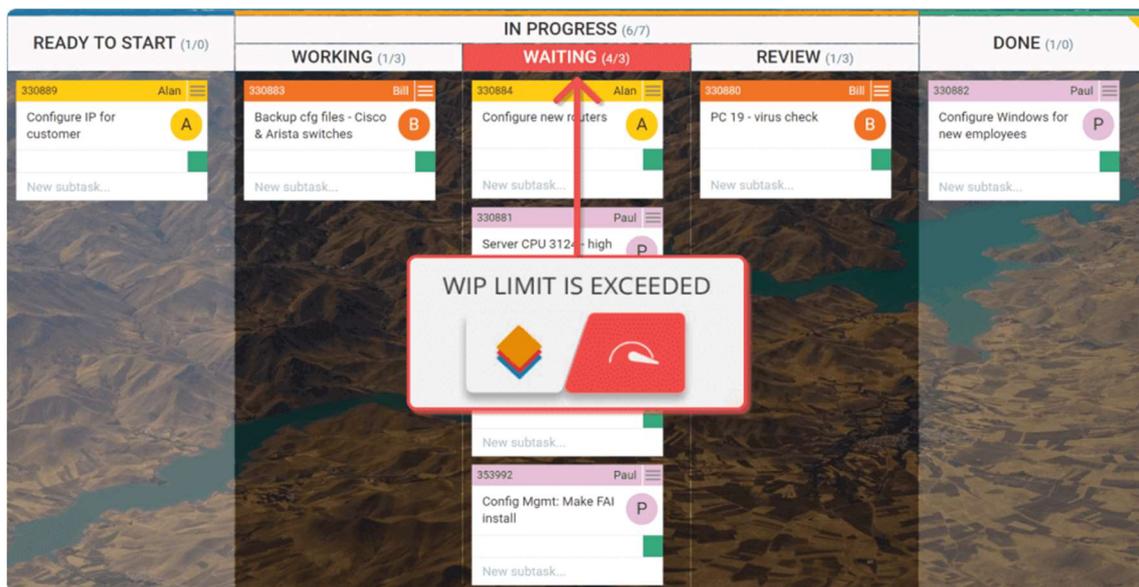


Figura 4.12: Ejemplo de sobrepaso del límite WIP en un proceso

Al tener un control de todo el proceso de producción, el trabajador que use el MES puede gestionar el inventario asignando recursos a aquellas tareas que estén ralentizando al proceso y, de la misma manera, puede retirar recursos a aquellas fases que no necesiten tantos de estos.

4.3.8.4. *Gestión de las Operaciones de Mantenimiento en Producción. Gestión de materiales*

Una vez que se ha iniciado la producción de un producto, es necesario llevar a cabo un correcto mantenimiento del proceso para evitar posibles problemas que puedan aparecer. Esta gestión del mantenimiento es muy importante, ya que para que un

²⁵ Miguel Ángel Olimpo. ¿Qué es WIP? ¿y cuáles son sus beneficios? Disponible en: <https://www.librered.net/que-es-wip-y-cuales-son-sus-beneficios/>

producto tenga una buena calidad y se le pueda sacar la máxima rentabilidad posible, hay que cuidarlo e ir mejorándolo para que no quede desfasado y para que cumpla con aquellas regulaciones que se impongan en un momento dado.

Una de las funcionalidades que tiene el MES para facilitar estas operaciones de mantenimiento es la genealogía del producto. Gracias a esta funcionalidad se puede tener un registro de todo el proceso de producción, de manera que se puede comprobar el estado del producto en cualquiera de sus fases, y realizar los cambios necesarios en aquellas etapas que no sean correctas.

Para esta gestión de las operaciones de mantenimiento es muy útil disponer de un buen *software* de GMAO²⁶ que facilite las operaciones de mantenimiento. Este *software* no está incluido en los sistemas MES, sino que se integra con estos para ayudar con la monitorización y análisis de los equipos.

Entre sus funcionalidades y módulos, GMAO tiene un módulo de mantenimiento que es el que se utiliza para esta gestión. Las principales funciones de este módulo son gestionar los activos, evitando así averías producidas durante la producción, y el mantenimiento preventivo o correctivo, que se usa para intentar adelantarse al fallo que se pueda producir en alguna máquina o sistema de la planta. Estas funciones se pueden mejorar siempre que los activos de la planta usen sensores que se puedan conectar a los sistemas GMAO, ya que permitiría la automatización de las órdenes de mantenimiento.

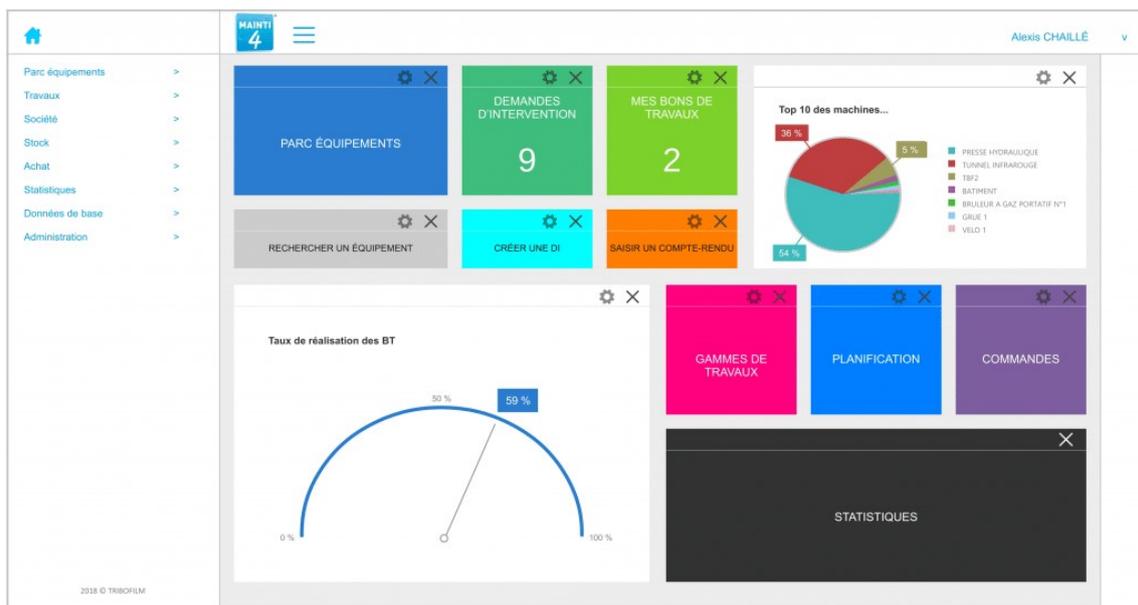


Figura 4.13: Ejemplo de GMAO de un proceso de producción

²⁶ Marc Mata. *¿Qué es y para qué sirve un GMAO?* Disponible en <https://www.wonderware.es/apm-asset-performance-management/que-es-y-para-que-sirve-un-gmao/>

5. Caso práctico (Análisis del Desempeño de Producción)

Como se ha visto hasta ahora la transformación digital consta de varias partes y usa diversas herramientas para alcanzar los objetivos que se buscan. En este caso práctico se va a ver un ejemplo de transformación digital para la operación de análisis del desempeño de producción en la empresa Acram.

5.1. Empresa Acram

Acram, que recordemos, es una empresa ficticia usada por temas de privacidad, es una empresa dedicada a la fabricación de productos químicos que lleva en activo más de 90 años, y cuya sede se sitúa en Barcelona. Además, Acram tiene varios centros en distintos países del mundo aparte de España, como pueden ser Portugal, Austria o incluso en países de otros continentes como Colombia.

Esta empresa ha tenido varias marcas con las que ha trabajado para sacar sus productos, teniendo cierta relevancia a nivel nacional con las grandes innovaciones que ha presentado en dichos productos, y lleva 20 años siendo una empresa que opera a nivel internacional.

Dentro de la industria de productos químicos ha adquirido varias marcas que le han permitido abrirse a distintas áreas de negocio, ofreciendo soluciones innovadoras y de alta calidad en cada una de ellas.

Hoy en día, Acram tiene mucho crédito internacional, lo que le ha permitido tener una gran presencia por todo el mundo, estando presente en hasta 50 países entre filiales y distribuidores. La mayoría de estos países son europeos, aunque también comienza a tener cierta presencia en América.

El grupo Acram es uno de los líderes del mercado en la industria de productos químicos, y sus productos se pueden encontrar en muchos países del mundo. Es por esto por lo que la empresa tiene que seguir mejorando para asegurar la calidad de sus productos, y esto implica adaptarse a las nuevas tecnologías que van apareciendo con el paso del tiempo.

5.2. Necesidades de transformación digital del cliente

Como ya se ha visto en capítulos anteriores, las empresas necesitan ir modernizándose y adaptándose a los tiempos actuales. Los trabajos manuales poco a poco van quedando obsoletos, y se necesitan máquinas y sistemas autónomos con



sensores, actuadores, etc., que ayuden a los trabajadores de la industria en la realización de sus tareas para, de este modo, ganar en eficiencia y evitar aquellos errores humanos que puedan surgir. La necesidad de transformación digital en las empresas aparece cuando se intenta alcanzar este objetivo.

En este caso, el cliente, Acram, busca esta transformación digital en la parte del análisis del desempeño de la producción, o lo que es lo mismo, busca aportar visibilidad a sus datos de producción. Actualmente, la empresa no tiene una visión global de todo el proceso de producción, por lo que no tiene un alto conocimiento sobre el estado en el que se encuentra la planta.

Para buscar este fin, se van a montar los KPI correspondientes, implementando información como pueden ser los problemas más comunes del proceso, u otras características de la planta, como puede ser el tiempo de uso de cada máquina o su rendimiento. De esta forma, se podrá detectar más fácilmente los problemas de la producción, y será más sencillo encontrar una solución para ellos.

Con esta implementación se podrá comprobar cuales son los puntos de mejora y como se puede ser más eficiente, además de ver donde se puede ahorrar recursos y, de esta manera, hacer un uso más eficiente de estos recursos de los que se dispone.

5.3. Tecnologías utilizadas para la Interfaz Web

Para la realización de esta transformación digital se necesitará una interfaz web con la que pueda interactuar el usuario, ya que esto permitirá una visualización ágil, sencilla, escalable y, además, multiplataforma, pues se podrá entrar en ella tanto en un PC, como en una *tablet*, una pantalla ANDON, y demás sistemas tecnológicos que sean compatibles con la interfaz . Esta interfaz deberá de tener distintos elementos para tratar de ofrecer la mejor experiencia al usuario.

Para crear una interfaz web existen diversas herramientas. Dependiendo de lo que se busque o de las preferencias del cliente, se usarán unas u otras. En este caso se ha usado el editor de código *Visual Studio Code*²⁷, ya que tiene múltiples funcionalidades para las herramientas utilizadas en la creación de esta interfaz. A continuación, se van a ver aquellas usadas para la creación de la interfaz de este proyecto.

5.3.1. Vue.js

Para la creación de esta interfaz se ha utilizado el *framework* Vue.js²⁸ (a partir de ahora Vue), una herramienta muy útil para la realización de páginas web y de interfaces de usuario, basado en JavaScript. Este *framework* tiene herramientas muy útiles como pueden ser los componentes, que sirven para diferenciar los distintos elementos que se

²⁷ Más información sobre *Visual Studio Code*: <https://code.visualstudio.com/docs>

²⁸ Más información sobre Vue: <https://vuejs.org/>

encuentran dentro de una página, y de esta forma dividir estos elementos para que el código sea más legible y liviano.

Para la gestión de los paquetes necesarios para este proyecto se ha usado el manejador de paquetes de *Node.js npm*²⁹ (aunque también se podría haber utilizado *yarn*). Gracias a *npm* se ha podido utilizar *CLI*³⁰, que es un paquete que permite tener una base para un proyecto de Vue, o *Lodash*³¹, que es una biblioteca de JavaScript que proporciona varias facilidades en un proyecto, como se verá en los siguientes capítulos.

Dentro de cada uno de estos componentes existe la posibilidad de tener la estructura del componente, junto con la lógica y los estilos, lo que los hace muy accesibles al encontrarte todo en un mismo fichero, además de ser fácilmente reutilizables en cualquier proyecto y fácilmente escalables. Vue también tiene herramientas útiles que favorecen una comunicación entre componentes muy sencilla, que se verá a continuación.

Para la creación de la interfaz web se tendrán que usar distintos lenguajes de programación. Dentro de la creación de una página web se pueden distinguir tres capas: la plantilla de nuestra página web, donde se definirá la posición de cada elemento de la página, la lógica de nuestra página, donde se definirán las acciones que puede realizar el usuario y las consecuencias que tendrán, y los estilos de la página web, donde se definirá el diseño que tendrá.

Para la elaboración de esta página usaremos el lenguaje HTML (*HyperText Markup Language*). Este lenguaje permite ordenar el contenido que se tenga en una página web mediante marcas de hipertexto, también llamadas etiquetas. Gracias a estas etiquetas se pueden crear barras, tablas, insertar imágenes o crear los botones de esta página, entre otras muchas cosas.

Este lenguaje también permite indicar cómo va a ser la estructura del documento usando los elementos que pone a disposición, la lógica que existirá en la página gracias a los eventos de JavaScript, y también permite asignar estilos a los distintos elementos mediante clases o identificadores.

Para la lógica de esta página se utilizará Vue, que, como se ha dicho anteriormente, se basa en el lenguaje JavaScript. JavaScript es un lenguaje orientado a objetos muy útil a la hora de elaborar páginas web, ya que permite comunicarse con los elementos de la página a través del DOM³² (*Document Object Model*), gracias a identificadores en los distintos elementos y gracias a los manejadores de eventos que proporciona. De esta manera, con JavaScript y DOM, se puede acceder e interactuar con los elementos creados anteriormente en HTML.

Vue facilita que esta página web sea dinámica, lo que permite modificar el DOM en el momento. Una de las funciones más importantes de este lenguaje es la de interactuar con el servidor, lo que permite unir el *frontend* (la parte que ve el usuario) con el *backend* (la parte del servidor).

²⁹ Más información sobre npm: <https://docs.npmjs.com/about-npm>

³⁰ Más información sobre CLI: <https://cli.vuejs.org/guide/> .

³¹ Más información sobre Lodash: <https://lodash.com/>

³² Más información sobre DOM: <https://developer.mozilla.org/es/docs/Glossary/DOM>



Para insertar Vue en el HTML se tendrá que usar la etiqueta `<script>` `</script>`, dentro de donde se introducirá el código de este lenguaje. En Vue, esta es la parte en la que se permite la comunicación entre los componentes. Para declarar variables globales dentro de cada componente, se deberán inicializarlas en un elemento llamado *data*, para poder usar las variables dentro del HTML.

```
data: {
  newTodoText: '',
  visitCount: 0,
  hideCompletedTodos: false,
  todos: [],
  error: null
}
```

Figura 5.1: Declaración de variables globales en Vue.

A estas variables se les asigna un valor por defecto que posteriormente podrá ser modificado con las funciones que se definan. Para utilizar estas variables dentro del DOM se deberá insertar la variable dentro de llaves dobles.

En una página web creada con componentes, a veces es necesario pasar datos entre los componentes. Para ello, en la declaración de nuestro componente se deberá permitir que se acepten propiedades, con el atributo “props”. Dentro de este atributo se recibirán las variables del componente padre, indicando el tipo que va a tener esa variable y, de forma opcional, el valor que tendrá por defecto esta variable, ya que dependiendo de la página en la que se encuentre el usuario, un componente podrá tener unas propiedades u otras.

```
<active-warnings :warnings-data="warnings"></active-warnings>
```

```
const props = {
  warningsData: {
    type: Array,
    default() {
      return [];
    }
  }
};
```

Figura 5.2: Comunicación padre-hijo.

En este ejemplo se puede ver que dentro de un componente se quiere insertar otro llamado “active-warnings”, y para pasar una variable que será utilizada en este

componente hijo usan la propiedad “warnings-data”, en *kebab-case*³³, y en el otro componente se recoge gracias al atributo “props”. De esta manera, se permite que un componente padre se comunique con un componente hijo de forma sencilla.

De la misma manera, un componente hijo puede comunicarse con un componente padre gracias al evento “\$emit()”. Dentro de los paréntesis irá el nombre del evento que se quiera lanzar en el componente padre, seguido de los parámetros que se quieran pasar, o solo el nombre del evento en caso de que no se quiera pasar nada. En el componente padre se indicará el evento junto con las acciones que se quieren llevar a cabo.

```
    this.$emit('myEvent')  
  
<my-component v-on:my-event="doSomething"></my-component>
```

Figura 5.3: Comunicación hijo-padre

También cabe destacar que en el momento de pasar las propiedades se usa “v-bind”, cuyo *shorthand* es :, que vincula una propiedad con la variable que se quiere pasar. Otra de estas directivas es “v-on”, cuyo *shorthand* es “@”, que sirve para escuchar eventos.

Además de estos atributos, Vue también dispone de funciones que permiten manejar el ciclo de vida de un componente, como puede ser la función *mounted*, que permite ejecutar código en el momento en que se está construyendo el componente, entre otras funciones.

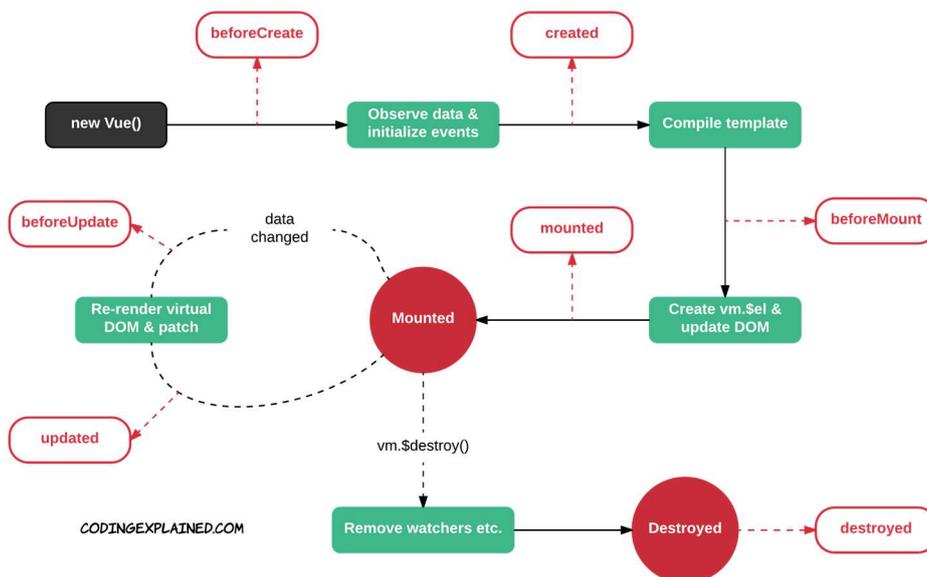


Figura 5.4: Esquema ciclo de vida de Vue

³³ Convención de nombres: desde el CamelCase hasta el kebab-case. Disponible en: <https://adrianalonso.es/cajon-desatre/convencion-de-nombres-desde-el-camelcase-hasta-el-kebab-case/>



Dentro del ciclo de vida de un componente de Vue, se puede distinguir entre el momento antes de empezar a crear los eventos, *beforeCreate*, y posteriormente *created*. Tras esto, y justo antes de que aparezca la plantilla, se encuentra *mounted*. Cada vez que se vuelva a renderizar el DOM se puede lanzar el evento *updated* y, por último, se puede lanzar el evento *destroyed* al salir de la página.

Una de las principales ventajas que tiene el uso de componentes es la capacidad de poder reutilizar el código de un componente para, de esta manera, ganar en eficiencia. Por lo tanto, hay componentes que podrán ser usados en más de una ocasión, con lo que ello conlleva. Aquí es donde aparecen los *slots*, que son muy útiles a la hora de esta reutilización de componentes. Hay ocasiones en las que queremos usar el mismo componente, pero existen pequeñas diferencias que lo impiden. Esto se soluciona con los slots, ya que permiten modificar un componente hijo desde un componente padre.

```
<navigation-link url="/profile">
  Su Perfil
</navigation-link>
```

```
<a
  v-bind:href="url"
  class="nav-link"
>
  <slot></slot>
</a>
```

Figura 5.5: Uso de slots para modificar un componente

En este ejemplo, se tiene un componente con las etiquetas `<slot></slot>`. Estas etiquetas serán sustituidas por el texto que se indique cuando queramos insertar este componente. De esta manera, cada vez que se llame a este componente, podrá aparecer un texto distinto. Estos slots podrán ser sustituidos por un texto, una plantilla HTML o incluso otro componente. En caso de que se usen varios slots en un componente, se deberá de usar el atributo *name* para diferenciarlos. En el componente padre se usará la directiva “v-slot:” seguido del nombre del *slot* que se quiera sustituir.

```
<div class="container">
  <header>
    <slot name="header"></slot>
  </header>
  <main>
    <slot></slot>
  </main>
  <footer>
    <slot name="footer"></slot>
  </footer>
</div>
```

```

<base-layout>
  <template v-slot:header>
    <h1>Aquí podría ir un título de página</h1>
  </template>

  <p>Un párrafo para el contenido principal.</p>
  <p>Y otro más.</p>

  <template v-slot:footer>
    <p>Aquí va alguna información de contacto</p>
  </template>
</base-layout>

```

Figura 5.6: Slots con atributo *name*

En este ejemplo, dentro de la etiqueta del componente que se quiere utilizar, hay un HTML que sustituirá el *slot* por el HTML que tenga el mismo nombre.

Dentro de un proyecto de Vue, en caso de que se quiera que en una web se pueda navegar por distintas rutas, se tendrá la configuración de enrutamiento de la página en una carpeta llamada *router*. Dentro de esta carpeta se deberán definir las rutas a las que se pueda navegar en la página, y se indicará las plantillas o páginas que se utilizarán en esta ruta. Gracias a esta funcionalidad se pueden usar las mismas plantillas o páginas para distintas rutas.

```

routes: [
  { path: '/user/:id', component: User,
    children: [
      {
        // UserProfile will be rendered inside User's <router-view>
        // when /user/:id/profile is matched
        path: 'profile',
        component: UserProfile
      },
      {
        // UserPosts will be rendered inside User's <router-view>
        // when /user/:id/posts is matched
        path: 'posts',
        component: UserPosts
      }
    ]
  }
]

```

Figura 5.7: Ejemplo de la variable *routes*

Dentro de una ruta se puede pasar un identificador para que el componente obtenga los datos que se correspondan con ese identificador. Además, una ruta podrá tener otras rutas dentro de ella gracias al atributo *children* y, de esta manera, poder reutilizar la plantilla que utilice la ruta padre. También se cuenta con la propiedad *redirect*, que

permite ir a una ruta por defecto de las que tenemos en *children*. Para usar una página o una plantilla deberá de estar referenciada en el archivo “/src/router/routes.js”, dentro de la variable *routes*.

Para que estas páginas sean imbuidas dentro de la página actual se necesitará el componente “<router-view>”. Si se pone esta etiqueta en la página, se colocará el componente correspondiente con la ruta en la que se encuentra, sustituyendo la etiqueta por el componente que se haya definido en la variable *routes*. En el [Apéndice A.1](#) se podrá encontrar la variable *routes* del sitio web creado en este proyecto.

```

<!-- /src/layouts/User.vue -->
<template>
  <q-layout>
    ...

    <!-- this is where the Pages are injected -->
    <q-page-container>
      <router-view></router-view>
    </q-page-container>

    ...
  </q-layout>
</template>

```

Figura 5.8: Ejemplo de uso del componente “<router-view>” .

En esta página web se usará *lazy loading* de plantillas y páginas, donde los componentes se cargan solo cuando se quiera acceder a la ruta correspondiente que los use y, de esta manera se evita la sobrecarga de componentes en proyectos relativamente grandes como este.

Para los estilos de la página se usará el lenguaje de diseño CSS, cuyos estilos se pueden insertar asignando a los elementos del DOM clases o identificadores. Estos estilos irán dentro de la etiqueta <style></style> .

Dentro de estos estilos se podrá definir si queremos usar SCSS³⁴, que tiene todas las ventajas de CSS, pero además posee otras características como puede ser el uso de variables definidas de forma global, que se verán en adelante con más detalle. También se puede añadir el atributo *scoped* en la etiqueta si se quiere que estos estilos solo se apliquen en el componente en el que se encuentran definidos.

5.3.2. Vuex

³⁴ Más información sobre SCSS: <https://www.hiberus.com/crecemos-contigo/diarios-de-un-junior-css-vs-bsass/#:~:text=CSS%20%20ABCascading%20Style%20Sheets%20BB%20oes,metalenguaje%20ode%20hojas%20ode%20estilo> .

En este proyecto también se usará Vuex³⁵, que es una librería de Vue muy útil a la hora de manejar estados en una página. Esta librería es importante cuando se tienen muchos componentes, y pasar propiedades de un componente a otro se convierte en algo muy tedioso. Gracias a Vuex, el estado se convierte en algo global para toda la página, y se puede acceder fácilmente desde cualquier componente.

Vuex cuenta con varias partes. Dentro de un componente, para conseguir los datos de una API se deberá ejecutar un *dispatch* con la acción que se quiera ejecutar y, desde un fichero *actions*, se tendrá que acceder a dicha API. Una vez que se obtenga la información, en caso de que se quiera modificar el estado, se realizará un *commit* con la función que se quiera ejecutar para que desde el fichero *mutations* modifique el estado que se quiera, y renderice los componentes. Por último, se encuentra el fichero *state*, que recoge las variables globales que se pueden recuperar o modificar en cualquier componente de una página.

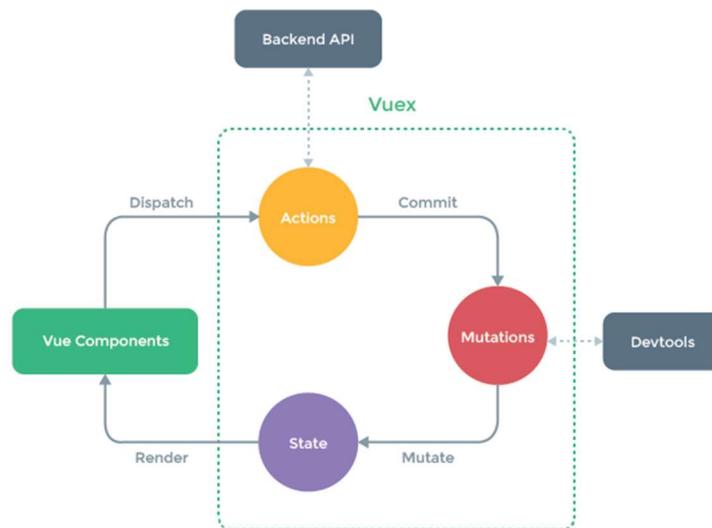


Figura 5.9: Estructura de Vuex

Cuando se quiera obtener el estado en alguno de los componentes, se usará la propiedad *computed*. Una de las ventajas que tiene el uso de esta propiedad es que, si el estado que se quiere obtener cambia, la propiedad computada se volverá a evaluar y cambiará la variable o variables asociadas a dicho estado.

Para llevar a cabo un cambio en un estado se deberá de usar *mutations*, donde se tendrá la función que cambiará el estado que se necesite. Para acceder a *mutations* se tendrá que ejecutar un *commit*, con el nombre de la función que se quiera ejecutar, además de añadir los argumentos que se quieran gracias al *payload*.

³⁵ Más información sobre Vuex: <https://vuex.vuejs.org/#what-is-a-state-management-pattern>



```
// ...
mutations: {
  increment (state, n) {
    state.count += n
  }
}

store.commit('increment', 10)
```

Figura 5.10: Ejemplo de *commit* con *payload*.

A veces en nuestra página se necesitarán hacer llamadas asíncronas a una API. Ya que *mutations* es síncrono, no se puede usar para ello, por lo que se deberá buscar algo distinto. Es aquí donde aparece *actions*, que es el encargado de obtener esta información. Para ello, se llamará a la *action* con *await*, dentro de una función que use *async*.

```
actions: {
  async actionA ({ commit }) {
    commit('gotData', await getData())
  },
  async actionB ({ dispatch, commit }) {
    await dispatch('actionA') // wait for `actionA` to finish
    commit('gotOtherData', await getOtherData())
  }
}
```

Figura 5.11: Ejemplo de ejecución de una acción asíncrona.

Para que una página sea escalable, se deberá usar los módulos de Vuex. De esta forma, cada módulo que se cree tendrá su propio *state*, *mutations* y *actions*. Dentro de la carpeta *store* deberá de haber un archivo llamado “index.js”, donde se tengan bien importados el lugar en el que se encuentran estos módulos para poder ser utilizados en un proyecto.

Para usar estos estados, acciones o mutaciones, se pueden usar *helpers* para facilitar su inserción cuando se tienen muchos componentes en un proyecto. Para ello se usará *mapState*, *mapActions* y *mapMutations*.

```
methods: {
  ...mapActions('option', ['getOptionInfo']),
  ...mapActions('plant', ['getMenuInfo']),
  ...mapMutations('plant', ['setPlant']),|
```

Figura 5.12: Uso de *helpers* en Vuex.

Dentro de estos *helpers* se tendrán que introducir dos argumentos: el primero será el nombre del módulo donde estará la acción o la mutación. El segundo será un array con todas las acciones o mutaciones del módulo anteriormente especificado que se quieran usar en este componente.

Gracias a Vuex se podrá unir el *frontend* con el *backend*, además de guardar el estado que se desee. Más adelante se verá cómo *actions* obtiene los datos que se usarán en la página, y la base de datos donde se encuentran guardados estos datos.

5.3.3. Quasar

Para este proyecto también se ha usado Quasar³⁶, que es un *framework* basado en Vue y enfocado en el desarrollo web. En este proyecto se ha utilizado para la creación de múltiples componentes, ya que proporciona estilos muy prácticos para los elementos de una página web y, además, son fácilmente modificables. También facilita la obtención de datos recibidos de un servidor, siendo sencillo insertarlos en sus componentes.

Dentro de este proyecto se encontrarán muchos elementos de Quasar, desde componentes hasta directivas, e incluso *plugins*, que se verán durante todo el proyecto.

Quasar tiene componentes y estilos ya definidos, que facilita la creación de una página web, y que se pueden asignar a los elementos de la página mediante clases. Una de sus mejores herramientas, y de las que más se usan durante este proyecto, es un sistema de *grid* para las páginas. Gracias a este sistema, se puede dividir cada elemento de nuestra página en filas y columnas, lo que permite una mejor distribución de nuestros elementos dentro de un contenedor, ya que al ser dinámico se adapta fácilmente a los distintos tamaños que debe adoptar una página.

En caso de que se quiera usar estas clases, se deberán asignar al contenedor que se quiera con la clase *row*. Dentro de este contenedor se podrán tener otros contenedores hijos a los que se le asignará la clase *col*.

Dentro de cada fila se pueden tener hasta 12 columnas. El tamaño de cada una se obtendrá con el número que se le dé a la clase, de manera que la clase “col-6” ocupará la mitad del tamaño total de la página. Por lo tanto, en caso de que se tenga una clase “col-X”, el tamaño de la columna será X/12, siendo X siempre menor o igual a 12. Además de estas clases, también se podrá usar la clase “col-auto”, que ocupará únicamente el espacio que necesite, y la clase “col”, que ocupará todo el espacio que haya disponible en la fila.

³⁶ Más información sobre Quasar: <https://quasar.dev/introduction-to-quasar#What-is-Quasar%3F>.



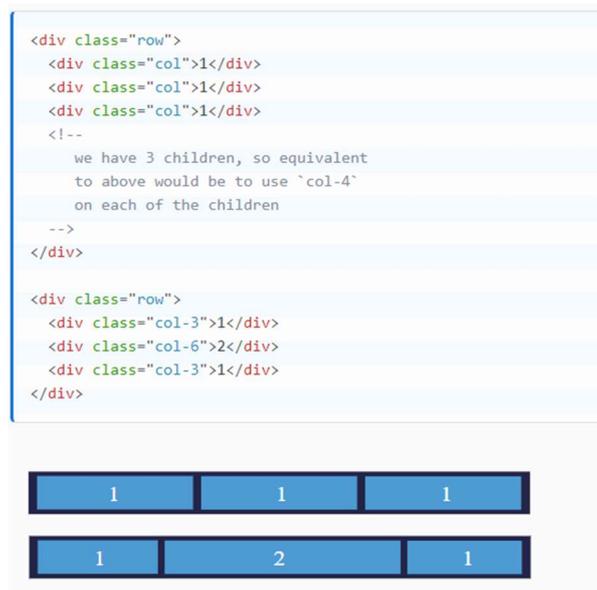


Figura 5.13: Ejemplo de uso del *grid* de Quasar.

Al usar este tipo de *grid* se puede contar con la ayuda de *Flexbox*³⁷, que es muy útil a la hora de alinear, ordenar o usar el espacio a disposición. Con *flex* se pueden ordenar los elementos de las filas o alinear las columnas. A lo largo de este proyecto se irán presentando ocasiones en las que será necesario usar estas funcionalidades.

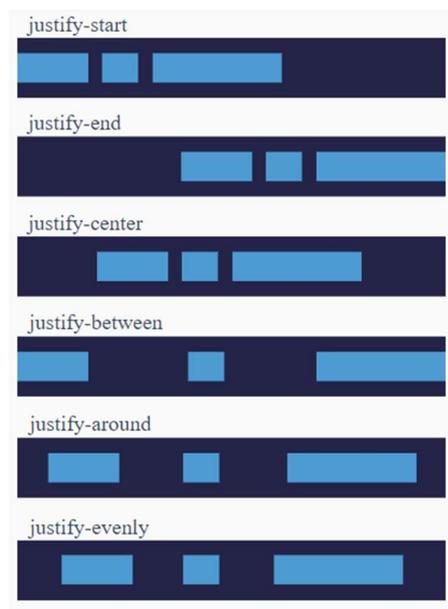


Figura 5.14: Distribución de los elementos de una fila

³⁷ Más información sobre *Flexbox*: <https://css-tricks.com/snippets/css/a-guide-to-flexbox/>.

5.3.4. Base de datos

Para esta página web se ha necesitado una gran cantidad de datos. Estos datos pueden ser tanto información de la planta, como la disponibilidad, rendimiento, etc. , o los registros que han hecho los distintos encargados que hay en dicha planta. Para la obtención de datos se ha usado tanto un WebSocket como una API Rest³⁸.

5.3.4.1. WebSocket

En esta página hay momentos en los que se necesita obtener datos en tiempo real. Por ejemplo, cuando hay que obtener el OEE de una máquina o de una línea, se necesita que estos datos vayan actualizándose de manera automática cada cierto tiempo.

Con un WebSocket se puede tener una comunicación bidireccional con el servidor, en el que se encuentra esta base de datos, con los datos que se necesiten en tiempo real.

Para introducir esta comunicación con nuestro servidor a través de un WebSocket, Vue tiene una API que permite abrir una conexión WebSocket con un servidor, y manejar el envío y recepción de mensajes con el mismo.

```
created: function() {
  console.log("Starting connection to WebSocket Server")
  this.connection = new WebSocket("wss://echo.websocket.org")

  this.connection.onmessage = function(event) {
    console.log(event);
  }

  this.connection.onopen = function(event) {
    console.log(event)
    console.log("Successfully connected to the echo websocket server...")
  }
}
```

Figura 5.15: Manejo de WebSockets en Vue

Para abrir la conexión, se tendrá que crear un WebSocket, indicando la url a la que se desea conectar, y asignarlo a una variable para poder manejarlo. Una vez hecho esto, se dispondrá de varios eventos, como puede ser el “onopen”, que se lanza cuando se ha abierto la conexión. Para manejar un mensaje recibido por el servidor se usa el evento “onmessage”, que recibirá en la variable *event* la información enviada por el servidor.

³⁸ REST vs WebSocket. ¿Qué diferencias hay? Disponible en: <https://www.itdo.com/blog/rest-vs-websocket-que-diferencia-hay/> .

Para recibir los datos del servidor, a veces se tendrá que enviar un mensaje que indique el tipo de respuesta que se espera recibir. Por ejemplo, a veces se querrá obtener las propiedades de una planta y otras veces se solicitará las de un área, por lo que hay que especificarlo con el mensaje que se envíe.

```

methods: {
  sendMessage: function(message) {
    console.log(this.connection);
    this.connection.send(message);
  }
},

```

Figura 5.16: Envío de un mensaje a un servidor

Con el método *send*, se enviará el mensaje deseado a nuestro servidor, y se esperará a recibir su respuesta con el evento “onmessage”, como se ha visto anteriormente.

En este caso, lo que se quiere es obtener información cada cierto tiempo, por lo que, al enviar cierto mensaje, se deberá obtener respuesta cada vez que pase el tiempo que haya sido especificado en el servidor. Para ello, en el servidor se ha definido un fichero XML donde se encontrarán todos los WebSocket que podrán ser utilizados para obtener datos en nuestra aplicación.

```

<WebSocket>
  <ID>BynsaWB-StatusMachines</ID>
  <Port>8081</Port>
  <Class>MaterialesWeb</Class>
  <Method>CheckMachineState</Method>
  <ElapsedTime>20000</ElapsedTime>
  <LogEnabled>true</LogEnabled>
</WebSocket>

```

Figura 5.17: Definición de WebSocket en un XML

Para poder utilizar cada WebSocket se deberán de tener los siguientes elementos:

- **ID:** El identificador que tendrá cada WebSocket que se quiera usar, y que deberá de ser único.
- **Port:** El puerto en el que se encontrará el WebSocket, necesario a la hora de indicar la dirección a la que conectarse.
- **Class:** La clase que usará este WebSocket dentro del servidor.
- **Method:** Dentro de cada clase, el método al que hace referencia, y el que se encarga de obtener la información a devolver.

- **ElapsedTime:** El intervalo, en milisegundos, en el que se ejecutará el método indicado para obtener los datos que se quieran en tiempo real.
- **LogEnabled:** Esta variable indica si se quiere que la ejecución de este WebSocket deje un registro para que se pueda monitorizar.

Como se ha visto, WebSocket es necesario en la página a la hora de obtener datos en tiempo real, sin embargo, no es la única forma de obtener datos que será utilizada en este proyecto.

5.3.4.2. API Rest

Dentro de una página web, hay veces en las que se quiere obtener datos estáticos, es decir, que no cambien los datos cada cierto tiempo, sino que se lance un evento en un momento dado, y que no se actualice hasta que no se vuelva a lanzar otro evento. Aquí es donde aparece el API Rest, que a diferencia del WebSocket, obtiene los datos de forma estática.

Para obtener los datos de una API se deberá de llamar a una acción, que a su vez llamará al servicio que haga la llamada a la API que se quiera. Al realizarse estas llamadas a un servidor externo, las llamadas tendrán que ser asíncronas, para que espere la respuesta de la API.

En el servicio, para poder comunicar con la API, se usará Axios³⁹, que es un cliente HTTP que se basa en promesas. En Axios se tiene que indicar el método de la solicitud que se va a realizar, junto con la url de la API y los datos que se quieran pasar, en caso de que se quiera pasar algo.

```
// Send a POST request
axios({
  method: 'post',
  url: '/user/12345',
  data: {
    firstName: 'Fred',
    lastName: 'Flintstone'
  }
});
```

Figura 5.18: Ejemplo de solicitud *POST* con Axios

³⁹ Más información sobre Axios: <https://github.com/axios/axios>.

De esta forma se esperará a que la API devuelva los datos usados en la página deseada, en este caso en formato JSON. A diferencia del WebSocket, una vez devueltos los datos, la conexión se cierra y, en caso de querer comunicarte de nuevo con la API, tendrás que volver a lanzar la acción correspondiente.

5.4. Implantación de servicios LMS

Para llevar a cabo esta transformación digital, se necesitan herramientas que monitoricen en tiempo real el proceso de producción que se esté llevando a cabo. Es por esto por lo que implantar un servicio LMS es necesario a la hora de tener una visión global de todo el proceso.

Estos LMS son un módulo del MES con el que se pueden realizar cálculos de KPIs de las líneas de producción. Gracias a este módulo se puede obtener el OEE de una planta de producción y, de esta manera, aumentar la eficiencia.

5.4.1. Tecnología Siemens LMS

Entre todas las herramientas que Siemens ofrece se pueden encontrar SIMATIC IT *Line Monitoring System* (LMS) que, como su nombre indica, implementa un *software* de monitorización a sus productos.

Para crear un LMS se han de seguir una serie de pasos: primero se tendrá que modelar el equipamiento o maquinaria que se va a tener en la planta, y la estructura de esta. Una vez que se haya hecho esto, es el momento de ir a la configuración de la planta, donde se debe de realizar un modelo de la planta. Por último, una vez que se tenga configurada la planta, se tendrán que sincronizar los datos con los módulos LMS para obtener por ejemplo el OEE de la planta. Para esta sincronización se utiliza el protocolo RTDE⁴⁰, que permite la sincronización de aplicaciones externas con el LMS en tiempo real.

⁴⁰ *Real-Time Data Exchange (RTDE) Guide*, 10 de octubre de 2019. Disponible en: <https://www.universal-robots.com/articles/ur/real-time-data-exchange-rtde-guide/>

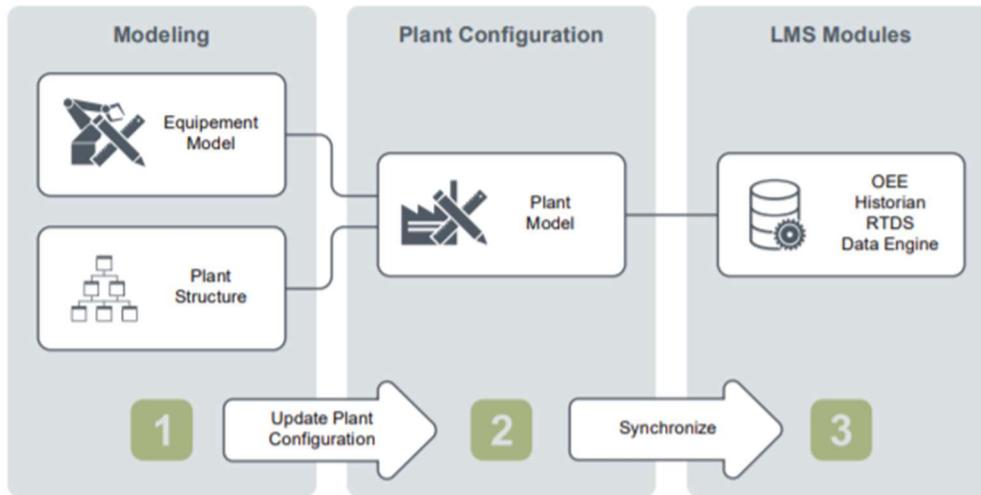


Figura 5.19: Pasos para la creación de un proyecto SIMATIC IT LMS

El LMS necesita una estructura para integrarse con la línea de producción, pues es necesario para por ejemplo poder detectar qué máquina está causando un cuello de botella en la producción (es decir, la máquina más lenta de la línea).

Otra de las funcionalidades del *software* LMS de Siemens es la configuración del OEE. Para calcular el OEE es necesario un modelo del tiempo de funcionamiento de la línea. Este modelo define varios estados de una máquina, como puede ser el tiempo planeado de producción, el tiempo operativo de esta, y el tiempo que está produciendo a máximo rendimiento o cuando hay una pérdida de velocidad causada por un cuello de botella en la producción.

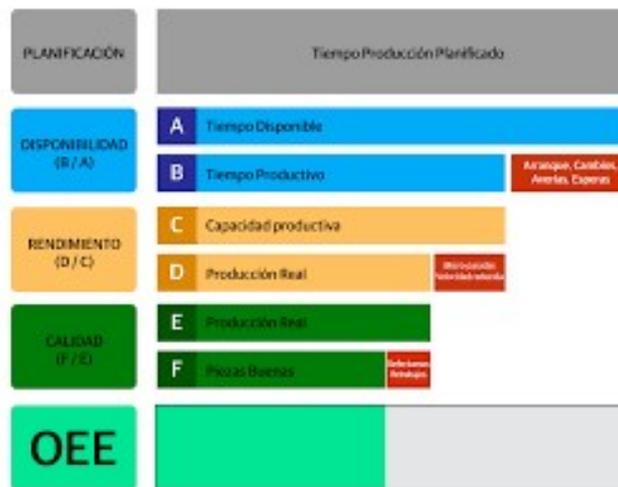


Figura 5.20: Ejemplo de cálculo del OEE

SIMATIC IT LMS usa diferentes algoritmos para monitorizar las operaciones de la línea de producción. Con estas definiciones se podrá obtener una vista general de los algoritmos utilizados:

- **Tiempo planeado de producción:** es el tiempo programado que debe de estar la máquina en producción, incluidas las detenciones planeadas.
- **Tiempo operativo:** es el tiempo en que la máquina está en producción, ya sea a máximo rendimiento o con pérdida de velocidad.
- **Disponibilidad:** Es el indicador que mide las pérdidas causadas por las detenciones no planificadas de la máquina.
- **Utilización:** Este KPI es igual que el de disponibilidad, pero también tiene en cuenta las detenciones planificadas, es decir, que mide el tiempo en que la máquina se encuentra en marcha en comparación con el tiempo total de producción.
- **Ratio de calidad:** Este indicador mide los productos producidos en buen estado con respecto al total de los productos producidos.
- **Ratio de rendimiento:** Este KPI mide las pérdidas que se han producido al estar ejecutando la producción con una velocidad que no es la óptima, es decir, cuando se producen pérdidas de velocidad en la máquina.
- **OEE:** El OEE de los equipos de producción se obtiene al sumar la disponibilidad, la calidad y el rendimiento de las máquinas. Sin embargo, su definición puede variar dependiendo de lo que busque el cliente, siendo el OEE la media de estos tres parámetros, o el menor de ellos, en caso de que busquemos cuellos de botella en el proceso de producción.
- **Tiempo medio entre fallo:** También llamado MTBF por sus siglas en inglés *Mean Time Between Failure*, es un KPI que compara el tiempo operativo de la producción con el número de fallos internos que se han producido durante esta.
- **Tiempo medio de reparación:** También llamado MTTR por sus siglas en inglés *Mean Time To Repair*, compara el tiempo que han durado estos fallos internos antes mencionados con el número total de fallos internos.

Algorithm	Calculation
Availability	Availability = Operating Time / Planned Production Time
Quality Rate	Quality = Good Pieces / Total Pieces
Performance Rate\Production Loss	Performance = Total Pieces / (Operating Time * Σ Design Speed)
OEE	OEE = Availability * Quality * Performance
Utilization	Utilization = Operating Time / Total Time
MTBF	MTBF = Operating Time / Number of internal faults
MTTR	MTTR = Internal Faults (time) / Number of internal faults

Tabla 5.1: Cálculos utilizados para la ejecución de los algoritmos

Para la comunicación entre la planta y el LMS se usan estándares de comunicación como Weihenstephan (WS)⁴¹, con los que se traduce el estado en que se encuentra una máquina (en marcha o parada), que se utilizan luego en los algoritmos vistos anteriormente. Para la obtención de la calidad de los productos se utiliza la función de RTDE “CalcGoodCounters”, para obtener los productos cuya calidad es buena, y posteriormente actualizar el algoritmo de ratio de calidad. Existen otras funciones RTDE como pueden ser la de “Line_EnergyCounter”, para obtener los valores de energía de la línea, o “Machine_EnergyCounter”, que tiene el mismo funcionamiento, pero a nivel de máquina.

Gracias a este *software* se pueden generar reportes o informes que sirvan para tener una visión general del proceso de producción y, de esta manera, ver qué es lo que está fallando y qué se puede mejorar. Estos reportes pueden ser tanto de la línea de producción como de las propias máquinas que se encuentran en ella, y también incluye reportes energéticos.

5.5. Diseño de la interfaz de usuario del LMS

Para implantar este servicio LMS que Siemens ofrece, se ha diseñado una interfaz para el cliente del que se ha hablado anteriormente, Acram, que permita obtener una visión general de su producción. Para ello, se deben de mostrar los datos de rendimiento, disponibilidad o calidad tanto de las áreas de la planta, como de las líneas y las máquinas que se encuentran en cada una, entre otras cosas que se verán a continuación.

5.5.1. Inicio de sesión

⁴¹ WS Communication interface. Disponible en: <https://www.weihenstephaner-standards.de/index.php?id=19&L=1>

Para entrar a la página web se deberá iniciar sesión con un usuario que esté registrado en la base de datos de la página. Dependiendo del usuario con el que se entre, se dispondrán de unas opciones u otras. En este caso, se va a entrar con un usuario que pueda ver y acceder a cualquier página.

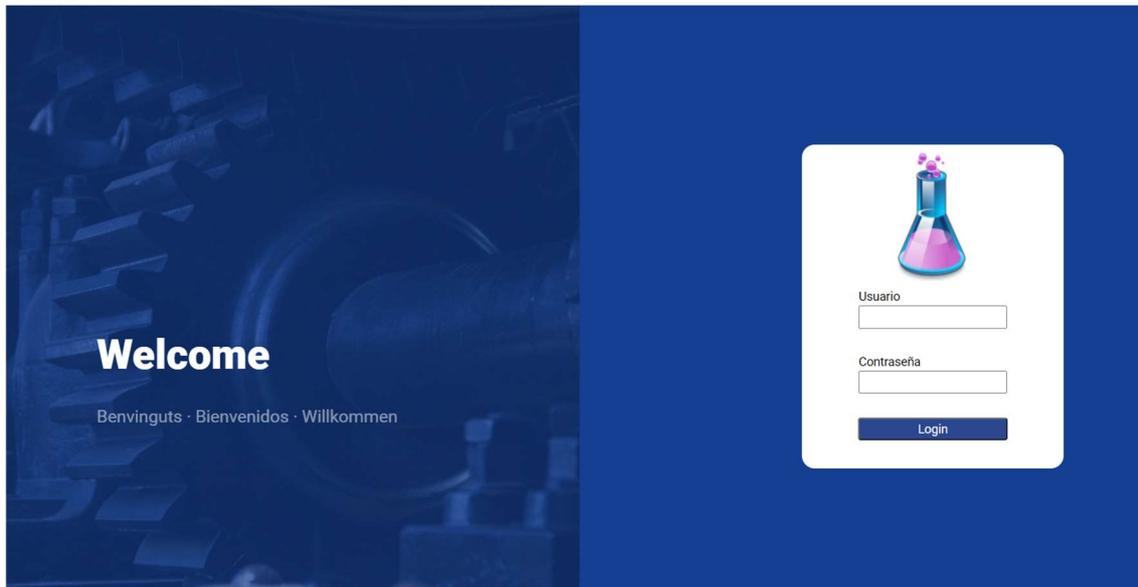


Figura 5.21: Página de inicio de sesión del portal

En esta primera pantalla se puede ver un mensaje de bienvenida, junto con una imagen de fondo proporcionada por el cliente. En la parte derecha se encuentra el logo del cliente, y los formularios, donde se tendrá que introducir el usuario y la contraseña, junto con un botón para lanzar el evento que comprueba si los datos introducidos son correctos.

En esta página, que se encuentra en el [Apéndice B.1](#), se puede ver claramente la utilidad que ofrece Quasar. Gracias a esta herramienta, se pueden ver dos “col-6” claramente diferenciados, el de la izquierda donde se encuentra el mensaje de bienvenida, y el de la derecha donde se encuentra el formulario para iniciar sesión. Gracias a las clases *row* se podrán añadir propiedades para alinear las columnas, como puede ser el formulario, que está justificado dentro de la fila con “justify-center”, y con “items-center” se consigue que el formulario se sitúe en el centro de la columna.

Al hacer clic en el botón *Login*, se ejecuta un evento al que se le ha llamado *userExists*, donde se comprobará en la base de datos si ese usuario existe y si se corresponde con la contraseña indicada.

Dentro de este evento se encuentra un plugin de Quasar que sirve para iniciar una animación de carga. Este plugin, además de toda la configuración de Quasar, se encuentra en un archivo llamado “quasar.conf.js”, que se puede ver en el [Apéndice A.2](#).

A continuación, se llamará a la acción *doLogin*, pasándole el usuario y la contraseña con las variables definidas en *data*, *username* y *password* respectivamente.

En este proyecto, Vuex se ejecuta dentro de módulos, donde cada módulo se corresponde con una parte de la página web. Uno de estos módulos es *login*, dentro del cual se encuentran las acciones relacionadas con el inicio de sesión y el manejo de sesiones dentro de la página. Este módulo se podrá ver en detalle en el [Apéndice B.2](#).

Para comprobar si el usuario y la contraseña son correctos, se llamará a un servicio de forma asíncrona al que hemos llamado *login*, pasándole el usuario y la contraseña introducidos. Por cada módulo se ha creado un servicio, desde donde se consumirá la API necesaria para la obtención de los datos de nuestra página. En el [Apéndice B.3](#) se puede ver cómo se realizan las llamadas a esta API.

```
import * as ApiMesService from './ApiMesService';  
  
export const login = payload => ApiMesService.postGenericMethod('AdministracionWeb', 'GetUsers', payload);
```

Figura 5.22: Servicio que comprueba si existe un usuario y una contraseña

Dentro del *payload* se encontrarán los parámetros a enviar a la API (en este caso el usuario y la contraseña que se han introducido), junto con la clase y el método que se van a ejecutar en la API. Dentro del servidor en el que se encuentra la API disponemos de varias clases, dentro de las cuales hay métodos donde están los datos a recuperar. En este caso, la clase en la que se encuentra nuestro método es “AdministracionWeb”, y el método “GetUsers”.

Este evento devolverá *true* si existe, redirigiéndose a la página principal de la interfaz, o *false* en caso contrario, lanzando una alerta para que se vuelvan a introducir los datos.

5.5.2. Página principal

Una vez que hayamos iniciado sesión, entraremos a una página conformada por tres componentes. Una barra lateral que se mantendrá en cualquier página que se visite, un menú con las plantas que tiene el cliente, acompañadas de las máquinas que funcionan y las que tienen algún fallo, y un mapa con los países donde se encuentran las plantas.



Figura 5.23: Página principal de la interfaz.

Menú lateral principal

En el menú lateral se pueden ver varios iconos, que tienen distintas funcionalidades. El primer icono es el logo del cliente, cuya función es la de volver a la página principal en caso de hacer clic sobre el en cualquier página en la que se esté.

Debajo nos encontramos con otro icono que al pulsarlo se abre un menú con los departamentos que existen en la empresa Acram.



Figura 5.24: Pop-up que muestra los departamentos

Como se puede ver, al pulsar en este icono aparece una ventana que se superpone por encima del resto de la página. Para ello se ha usado una utilidad de Quasar llamada “q-dialog”, que aparecerá cuando la variable *sidebarOpened* se ponga a *true*. Esta variable por defecto está a *false*, y para ponerla a *true* se lanzará un evento en el componente de la barra lateral. El código necesario para ello se puede ver en el [Apéndice B.4](#).

Los departamentos se obtendrán haciendo una llamada a la API al pulsar en el icono, usando la función *mounted*, que sirve para manejar el ciclo de vida de un componente. En este caso, sirve para ejecutar *scripts* al montar el componente, es decir, al aparecer el componente en la página.

```
const mounted = function mounted() {  
  this.getOptions();  
};
```

```
const getOptions = async function _getOptions() {  
  this.options = await this.getOptionInfo();  
};
```

Figura 5.25: Obtener departamentos al montar el componente.

Para insertar los departamentos en el menú se usará la utilidad de Quasar “q-list”, dentro de la cual se encontrarán los departamentos, que son un “q-expansion-item”, de insertar un submenú dentro de un menú, con un icono (en este caso una flecha) que permite esconder y mostrar los submenús. El código de estos elementos se encuentra en el [Apéndice B.5](#).



Figura 5.26: Menús expandibles

Al expandir el menú, se pueden encontrar botones que llevan a una parte distinta de un departamento. Como se puede ver, aquellas partes que no estén implementadas aparecerán deshabilitadas para impedir el acceso.

Al pulsar en alguno de los botones que estén habilitados, ejecutará una función a la que se le pasará como argumentos el nombre del menú y el del botón que se ha pulsado.

```
const goToPage = function _goToPage(item, page) {
  this.$emit('hide-menus');
  const realPage = page.normalize('NFD').replace(/[\u0300-\u036f]/g, '');
  this.$router.push(`/${item.replace(' ', '')}/${realPage.replace(' ', '')}`);
};
```

Figura 5.27: Función para moverse por el menú

En esta función se emitirá un evento al componente padre para oculte el *pop-up*, y abrirá la página que se corresponda con los argumentos que se han pasado, y que se encuentre en la tabla de enrutamiento definida con anterioridad.

En la barra lateral hay otros botones a destacar, como puede ser el de las notificaciones. Este botón tendrá el número de avisos que hay en las plantas que tengamos implementadas, y donde se encuentran.

Planta	Área	Línea	Inicio	Categoría	Estado	Descripción	Acciones
SANT LLORENÇ D'HORTONS	ÁREA VA	LV1A	11/03/2021 08:52	⚠	Activa	Fallo Desconocido	🔗
SANT LLORENÇ D'HORTONS	ÁREA VB	LV1B	10/03/2021 10:23	✅	Activa	Calidad Deficiente	🔗

Figura 5.28: Pop-up de avisos

Se puede ver en detalle cada aviso, con un botón que al pulsarlo se puede ir a la máquina en cuestión que está dando un fallo.

Más abajo tenemos un botón para cambiar la configuración de la página, el avatar de usuario que se haya seleccionado, y un botón que sirve para cerrar sesión, y que redirigirse a la página de inicio de sesión.

Menú de plantas industriales

En el menú de plantas industriales se encuentran todas las plantas que tiene el cliente junto con el estado en el que se encuentran sus máquinas, todas ellas dentro del país en el que se encuentran. En este menú, al pulsar en las plantas se puede abrir el menú de los departamentos de cada planta. En caso de que la planta pulsada no esté implementada aun, saldrá un mensaje de alerta advirtiéndolo de esto.

A la derecha de las plantas industriales se puede ver el estado en el que se encuentran las distintas máquinas. En verde se puede ver las máquinas que tienen un funcionamiento correcto, y en rojo aquellas que tienen algún tipo de fallo. Al pulsar en este botón se podrá acceder a la línea en cuestión en la que se encuentran las máquinas que están fallando.

Mapa de las plantas industriales

En cuanto al mapa que se encuentra a la derecha de la página, se pueden ver remarcados aquellos países en los que se encuentran plantas industriales del cliente. Este mapa es una imagen svg, que sirve para poder acceder a distintos puntos de la imagen.

Gracias a este svg se pueden asignar identificadores a distintos puntos de una imagen, de manera que se puede acceder a distintos puntos de la imagen. Al poder asignar identificadores, se pueden añadir eventos al mapa, como puede ser la aparición de un *pop-up* al situar al puntero sobre unas de estas plantas. En el [Apéndice B.6](#) se puede ver cómo se maneja este svg.



Figura 5.29: *Pop-up* de una planta

Al pulsar en uno de los botones te llevará a los indicadores de la planta que se corresponda con ese botón.

5.5.3. Indicadores operacionales

Existen dos formas de acceder a esta página: la primera es a través de los departamentos, en el menú “Indicadores” y pulsado en el botón “Operacional”. La segunda es pulsando en el *pop-up* que aparece en el mapa de la página principal.



Figura 5.30: Página de los indicadores

En esta página se pueden ver los distintos indicadores o KPIs que existen en cada planta de producción. Esta página está definida por dos componentes claramente diferenciados.

Menú lateral

Como en todas las páginas de la interfaz, existe un menú lateral por el que puedes navegar por todos los indicadores que hay en cada planta. Dentro de la planta de Sant Llorenç D'Hortons, hay tres áreas distintas, dependiendo del tipo de producto que se produce: *Home Care* (cuyo nombre para identificarla es VA, y donde se producen productos para el cuidado del hogar), *Personal Care* (VB en este caso, con productos para el cuidado personal) y Fabricación de Plásticos (llamada VC, como su nombre indica sirve para fabricar plásticos que luego serán usados en sus productos).

Si desplegamos cada una de las áreas aparecerán las líneas que existen en cada una de ellas, de la misma forma que si desplegamos las líneas aparecerán todas las máquinas que hay en esa línea.

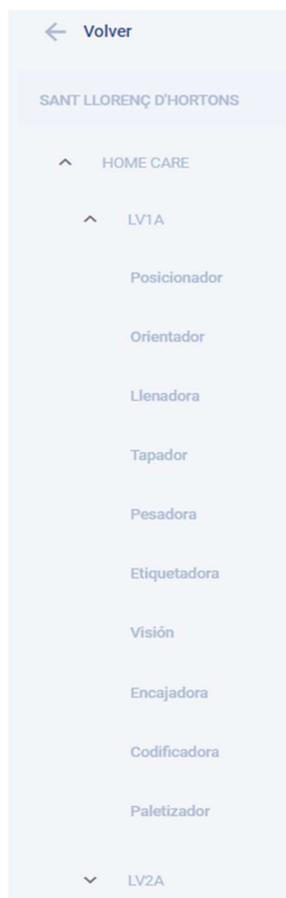


Figura 5.31: Máquinas y líneas de cada área

Como se ha dicho anteriormente, este menú va a acompañar siempre al usuario en todas las páginas de los departamentos, por lo que será necesario reutilizar este componente en sus distintas páginas. En el [apartado 5.3.1](#) se ha hablado de una funcionalidad muy útil para la reutilización de componentes, los *slots*. En el [Apéndice B.7](#) se puede ver cómo se usan en este menú lateral.

Al hacer clic en un área o en una línea se ha definido un evento para que en el estado se incluyan aquellas propiedades de la línea o del área pulsada. De esta manera, al navegar a donde se quiera, se deberán de recuperar los datos que se encuentren en el estado.

Si por ejemplo se pulsara en la línea “LV1A”, se ejecutaría la *mutation* “setLine(línea)”, cuyo argumento “línea” contiene todas las propiedades de la línea seleccionada. En el siguiente apartado se verá cómo se obtienen dichas propiedades.

```

methods: {
  ...mapMutations('area', ['setArea']),
  ...mapMutations('line', ['setLine']),
  getOperacional
},
export function setLine(state, line) {
  state.actualLine = line;
}

```

Figura 5.32: Actualización del estado en el módulo *line*

Por último, cabe destacar en este menú lateral un botón “Volver” que permite volver a la página principal. Para ello se utiliza una directiva de Quasar llamada “v-go-back”, y la dirección a la que se quiera navegar.

Indicadores del proceso de producción

Esta es una de las páginas más importantes del LMS, ya que es la que da visibilidad a los KPIs que existen en un proceso de producción. Estos indicadores se pueden ver tanto a nivel de planta, como a nivel de área, como a nivel de línea y, por último, a nivel de máquina. Esta página consta de dos componentes: un componente donde se puede ver el resumen de la parte del proceso que estemos visualizando (en este caso la planta), y otro donde se pueden ver los indicadores de dicha planta. Por último, la página consta de otro componente, que es una cabecera común para todos los departamentos de la interfaz, y que, al igual que la barra lateral, también usa *slots*.

En la cabecera de la página, en la parte izquierda podemos distinguir una ruta que indica dónde se encuentra el usuario. En la parte derecha se pueden ver dos elementos *select*, cada uno con su función.

En el *select* intervalo, se puede elegir la fecha en la que se quieren obtener los datos y, dependiendo de lo que se elija, los datos vendrán de una forma o de otra.



Figura 5.33: Intervalos a elegir

En este *select* se pueden elegir tres opciones. Por defecto el intervalo se encuentra en “Hoy”, con lo que la fecha de inicio se corresponderá con la fecha de hoy, y la hora será la hora marcada por el cliente, que en este caso son las 7:00. En caso de marcar en intervalo “Ayer”, la fecha de inicio será la de ayer, y la hora la misma. La otra opción es la de “Seleccionar Fechas”, que permite obtener los indicadores en el intervalo de tiempo que se desee, abriendo un *pop-up* en el que se pueda elegir tanto la fecha de inicio como la fecha de fin.

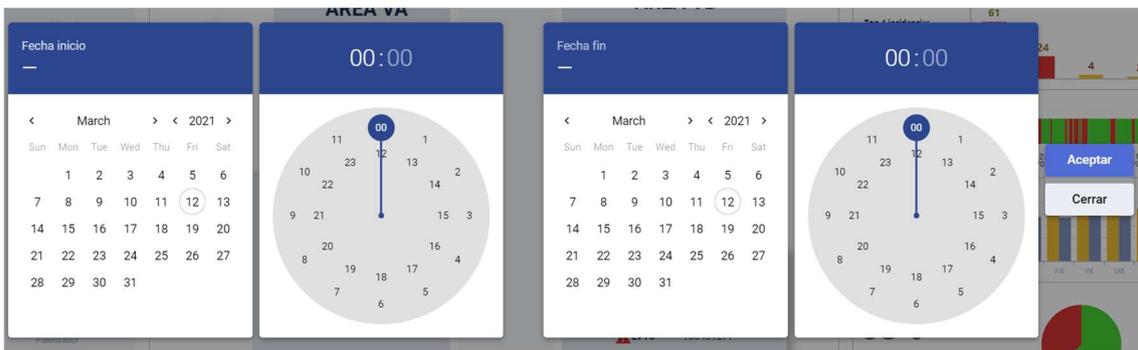


Figura 5.34: *Pop-up* para seleccionar fechas

El manejo que se realiza del *select* intervalo se puede ver en el [Apéndice B.8](#), donde se puede ver también cómo se eligen la fecha de inicio y de fin al elegir la opción “Seleccionar fechas”. Una vez elegidas las fechas, al pulsar el botón “Aceptar” se lanzará un evento llamado “sendDates()”. En la Figura 5.34. se puede ver el código de esta función.

```
const sendDates = async function _sendDates() {
  if (this.startDate && this.startHour && this.endDate && this.endHour) {
    this.showResumenData = false;
    const fIni = date.formatDate(`${this.startDate} ${this.startHour}`, '
YYYY/MM/DDTHH:mm:ss.SSSZ');
    const fFin = date.formatDate(`${this.endDate} ${this.endHour}`, 'YYYY
/MM/DDTHH:mm:ss.SSSZ');
    if (this.connection) {
      this.connection.close();
    }
    this.plant = await this.getPlantInfo({ fin: fIni, ffin: fFin });
    this.showIntervalo = true;
    this.showPopup = false;
    this.showResumenData = true;
  }
};
```



Figura 5.35: Función que recupera los datos en un intervalo.

Para manejar correctamente las fechas se usará la utilidad de Quasar Date⁴², que permite dar el formato que se desee a una fecha, en este caso el formato será “YYYY/MM/DD HH:mm:ss”. A continuación, se lanzará una acción que consuma una API donde, pasando como argumentos la fecha de inicio y la fecha de fin, obtenga los datos de los indicadores para esas fechas.

Al realizar la búsqueda aparecerán los datos de esta nueva fecha, con un párrafo que indicará la fecha de los datos que se muestran.

**Figura 5.36:** Información del intervalo de fechas

La obtención de datos es la misma que la que se ha visto ahora; sin embargo, esto cambia cuando en intervalo se selecciona “Hoy” o “Ayer”, ya que entonces se deben de obtener los datos en tiempo real. Como se ha hablado en el apartado 5.3.4.1, si se quieren usar datos que se vayan actualizando en tiempo real, se debe usar WebSocket.

La creación de la conexión WebSocket se podrá ver en el [Apéndice B.9](#). Cuando el intervalo esta en una de estas dos opciones, se ejecutará una función que envíe la fecha de inicio al servidor, dependiendo de si la fecha es la de hoy o la de ayer.

```
const sendMessage = function _sendMessage() {
  this.connection.send(`{PARAMETERS:[{"fecha": "${this.intervalo}"}]}`);
};
```

Figura 5.37: Función que envía un mensaje al servidor

De esta manera, los datos se irán actualizando cada cierto tiempo, obteniendo de esta forma una visibilidad total de los datos.

⁴² Más información sobre la utilidad *Date*: <https://quasar.dev/quasar-utils/date-utils>

En la cabecera se encuentra otro *select* llamado panel, que cambia el contenido de la página. El panel que se va a ver a continuación es el de “Resumen”, que es el que está por defecto.

El componente Resumen consta de tres zonas, una por cada área que existe en la planta. En cada área se encuentra su nombre y una gráfica que indica el OEE. El gráfico es un componente llamado “halfPieGraph.vue”, donde se usan utilidades para gráficas. La definición de esta gráfica se podrá ver en el [Apéndice B.10](#).

Por último, se encuentra las líneas que hay en esa área, junto con el orden de producción que tiene asignada cada línea. También se ha definido un icono junto a las líneas que indique el estado en el que se encuentra.

En el componente que está a la derecha de la página, se encuentran los indicadores de la planta. Por una parte, se encuentra el top 4 de las incidencias que se producen en la planta, cuyo número indica el porcentaje que corresponde a ese fallo respecto al total de los fallos. Al situar el cursor encima de cada barra se puede ver la incidencia a la que hace referencia dicha barra.



Figura 5.38: Ejemplo de incidencia en planta

Debajo de esta gráfica se encuentran los indicadores de disponibilidad, rendimiento y calidad. El indicador de disponibilidad muestra, en el intervalo que se haya elegido, el tiempo que la producción ha estado en marcha, representado en verde, y el tiempo que ha estado parada, representado en rojo. El indicador de rendimiento muestra, en la última semana del intervalo que se ha escogido, el rendimiento que ha tenido la producción en cada día de la semana. La barra amarilla representa el rendimiento que debería de tener la máquina si todo va como se espera, y la barra gris el rendimiento que ha tenido en realidad. Por último, se encuentra la gráfica de la calidad de la producción, que representa el número de piezas que han sido buenas y el número de piezas que han salido defectuosas.

Como se ha dicho en capítulos anteriores, el cálculo del OEE es definido por el cliente. En este caso, el cliente ha decidido que el porcentaje del OEE sea el resultado de la multiplicación de la disponibilidad, el rendimiento y la calidad.

En este capítulo se ha visto el desempeño de la producción de una planta, pero se puede ver a cualquier nivel del proceso.

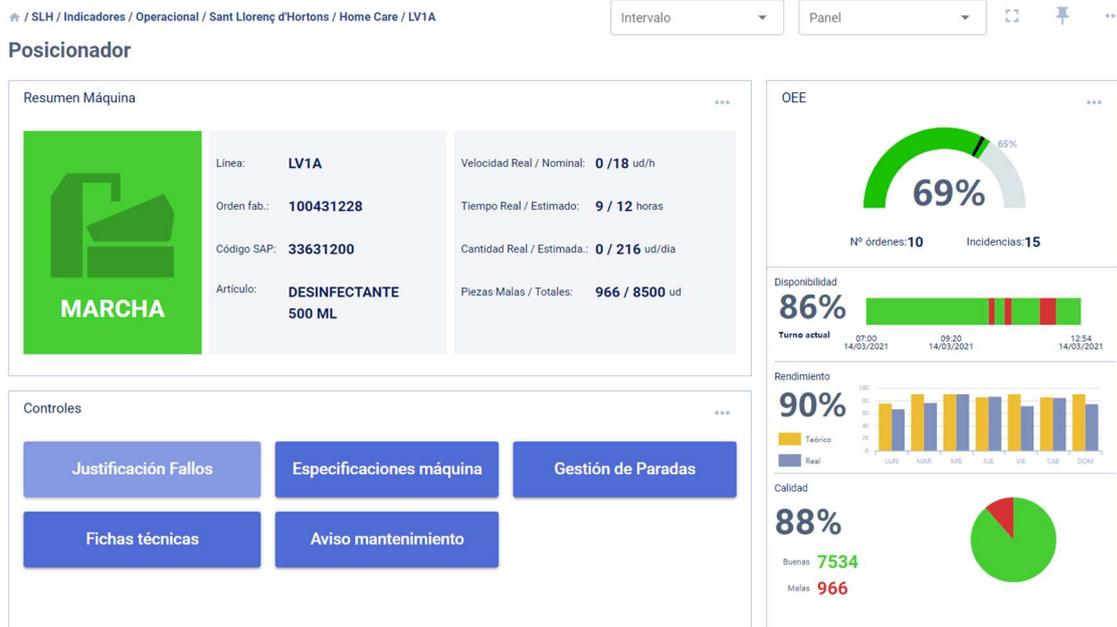


Figura 5.39: Indicadores de máquina

5.5.4. Paneles complementarios

Lo que se ha visto hasta ahora era el panel de “Resumen”, sin embargo, no es el único panel que se ha creado. Otro de los paneles creados para dar más visibilidad a la producción es el de comparativa. Gracias a este panel se puede comparar los indicadores de cada parte del proceso, facilitando de esta manera la detección de la parte del proceso que está ralentizando la producción, entre otras cosas.

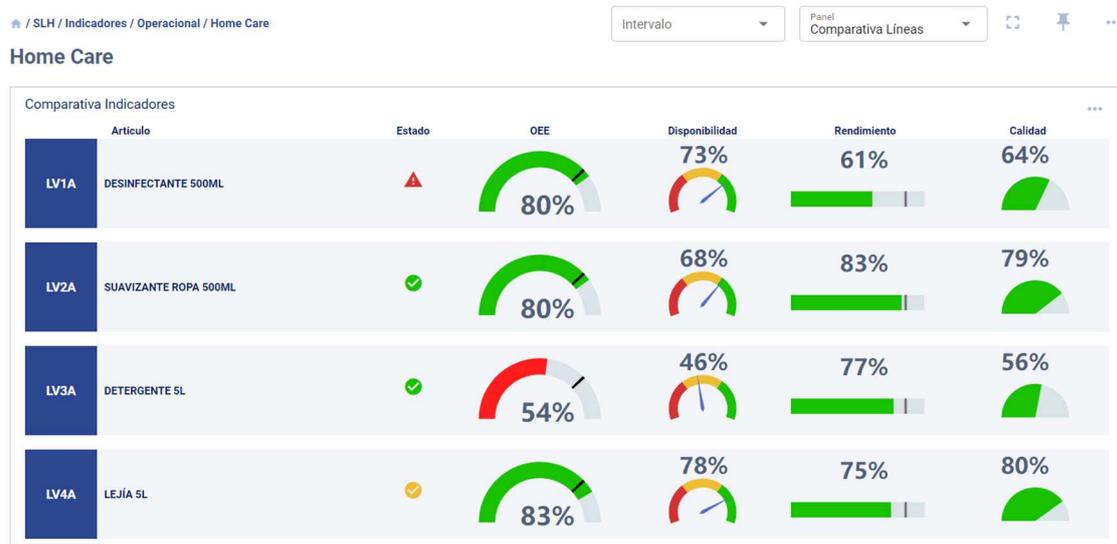


Figura 5.40: Comparativa de los indicadores a nivel de área

Estas comparativas son comunes para todo el proceso, por lo que se reutilizará el mismo componente, añadiendo los datos que correspondan en cada momento.

Otro de los paneles que permiten ver el estado en el que se encuentra la producción es el de “Esquema”. Este panel se ha creado para las áreas y para las líneas, de forma que se muestre el estado de las máquinas que se encuentran en ella.

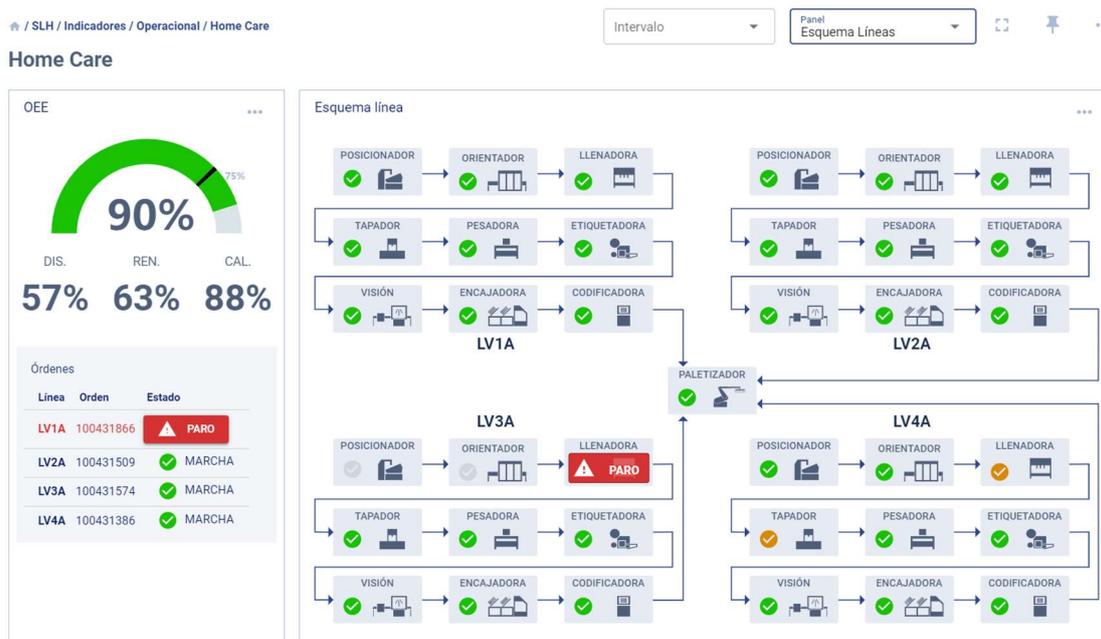


Figura 5.41: Esquema de las líneas de un área

En este panel se pueden ver dos componentes principales, uno que muestra un pequeño resumen del OEE del área, además de una tabla donde se encuentran la orden de producción que está llevando a cabo cada línea. En el otro componente se encuentra un esquema de las máquinas de cada línea, indicando el estado en el que se encuentran.

La zona del esquema de las líneas es un svg al que se le ha pasado la información de las máquinas para actualizar el icono que aparece dependiendo del estado en el que esté. En caso de que alguna máquina esté parada, se habilitará un botón en la máquina para ir directamente a la misma, y verlo en más detalle.

5.5.5. Control de calidad

Otro de los departamentos que se han implementado en este proyecto es el de Calidad. Como se ha visto en el capítulo 5.5.2, en esta página existe una barra lateral que se puede encontrar en cualquier página del sitio web, y que tiene un botón que al pulsarlo

se abren los departamentos existentes en las plantas de producción. Si pulsamos en el menú de “Calidad”, y luego pulsamos en el botón “Control Calidad”, se redirigirá a la página que se ve en la Figura 5.41, en la que se encuentran los documentos de control que han sido creados durante el proceso de producción.

Documentos de Control

Fecha Creación Línea Turno Orden producción Tipo Estado

Exportar en CSV Imprimir en PDF

Filtrar

Id	Creación	Línea	Turno	Orden Prod.	Tipo	Versión	Estado	Fugas extra	Incidencias	Est. botellas	Revisado	Aprobado	Observaciones	Acciones
641120191127	2021-03-02	LV1A	Mañana	100431573	Materias primas	v3	Abierto	0	1	1	✓ Joan Serra	✓ Jaime Castell	1	...
641120191128	2021-03-02	LV1A	Mañana	100431329	CIP	v2	Abierto	0	1	2	✓ Joan Serra	✓ Jaime Castell	0	...
641120191129	2021-03-02	LV1A	Mañana	100431573	Producción	v1	Cerrado	3	0	0	✓ Joan Serra	✓ Jaime Castell	0	...
641120191130	2021-03-01	LV1A	Tarde	100493404	Envasado	v1	Cerrado	0	0	2	✓ Manuel Sanz	✓ Alex Fuster	0	...
641120191131	2021-03-01	LV1A	Tarde	100493404	Materias primas	v3	Cerrado	0	2	0	✓ Manuel Sanz	✓ Alex Fuster	2	...

Records per page: 5 1-5 of 9

Figura 5.42: Documentos de Control de Calidad

En esta página se pueden ver tres componentes: una cabecera, parecida a la que se encontraba en la página de los indicadores, visto en el capítulo 5.5.3. Un componente que contiene un buscador con varios parámetros que permiten buscar en la tabla, y por último una tabla donde se encuentran estos documentos de control.

Buscador Documentos de Control

Para esta página se necesita buscar por las distintas columnas que aparecen en la tabla. La primera columna se corresponde con la fecha en la que se ha creado dicho documento. Al pulsar en el *select* se abre un calendario que permite seleccionar una fecha, tal y como se ha implementado en el *select* de intervalo. El resto de elementos son *select* en los que se encuentran las distintas opciones que ofrece cada columna, excepto el de “Orden producción”, que se trata de un *select* en el que, además, también se puede escribir la orden para facilitar su búsqueda.

Para realizar la búsqueda se ha implementado un botón “Buscar” que, al pulsarlo, ejecute una función que permita realizar la búsqueda. El funcionamiento de este botón se puede ver en el [Apéndice B.11](#).

Documentos de Control

Fecha Creación | Línea | Turno | Orden producción | Tipo Envasado | Estado

Exportar en CSV | Imprimir en PDF

Filtrar

Id	Creación	Línea	Turno	Orden Prod.	Tipo	Versión	Estado	Fugas extra	Incidencias	Est. botellas	Revisado	Aprobado	Observaciones	Acciones
641120191130	2021-03-01	LV1A	Tarde	100493404	Envasado	v1	Cerrado	0	0	2	Manuel Sanz	Alex Fuster	0	...
641120191134	2021-03-01	LV1A	Mañana	100384738	Envasado	v1	Cerrado	0	2	0	Joan Serra	Jaume Castell	0	...

Records per page: 5 | 1-2 of 2

Figura 5.43: Tabla filtrada por el tipo “Envasado”

Tabla Documentos de Control

Este componente consta de una tabla, implementada gracias a un componente de Quasar llamado “q-table”. El código se puede ver en el [Apéndice B.12](#).

En este caso tenemos una paginación para el caso en que haya demasiadas filas en la tabla. Por defecto se ha configurado para que se muestren 5 filas por página, pero puede modificarse en la misma página.

Si se desea se puede exportar el contenido de la tabla en formato CSV.

	A	B	C	D	E	F	G	H	I	J	K	L
1	Id,"Creación"	"Línea","Turno","Orden Prod."	"Tipo","Versión","Estado","Fugas extra","Incidencias","Est. botellas","Revisado","Aprobado","Observaciones","Acciones"									
2	641120191127,"2021-03-02","LV1A","Mañana","100431573","Materias primas","v3","Abierto","0","1","1","Joan Serra","Jaume Castell",""											
3	641120191128,"2021-03-02","LV1A","Mañana","100431329","CIP","v2","Abierto","0","1","2","Joan Serra","Jaume Castell",""											
4	641120191129,"2021-03-02","LV1A","Mañana","100431573","Producción","v1","Cerrado","3","0","0","Joan Serra","Jaume Castell",""											
5	641120191130,"2021-03-01","LV1A","Tarde","100493404","Envasado","v1","Cerrado","0","0","2","Manuel Sanz","Alex Fuster",""											
6	641120191131,"2021-03-01","LV1A","Tarde","100493404","Materias primas","v3","Cerrado","0","2","0","Manuel Sanz","Alex Fuster",""											
7	641120191132,"2021-03-01","LV1A","Tarde","100493404","CIP","v2","Cerrado","0","0","1","Manuel Sanz","Alex Fuster",""											
8	641120191133,"2021-03-01","LV1A","Tarde","100493404","Producción","v1","Cerrado","1","0","0","Manuel Sanz","Alex Fuster",""											
9	641120191134,"2021-03-01","LV1A","Mañana","100384738","Envasado","v1","Cerrado","0","2","0","Joan Serra","Jaume Castell",""											
10	641120191135,"2021-03-01","LV1A","Mañana","100384738","Materias primas","v3","Cerrado","0","2","0","Joan Serra","Jaume Castell",""											

Figura 5.44: Tabla exportada en formato CSV

También se puede imprimir la tabla en formato PDF. Ambas exportaciones se pueden realizar, tanto de la tabla al completo, como de la tabla filtrada por alguna de sus columnas.

Id	Creación	Línea	Turno	Orden Prod.	Tipo	Versión	Estado	Fugas extra	Incidencias	Est. botellas	Revisado	Aprobado	Observaciones
641120191130	2021-03-01	LV1A	Tarde	100493404	Envasado	v1	Cerrado	0	0	2	Manuel Sanz	Alex Fuster	0
641120191134	2021-03-01	LV1A	Mañana	100384738	Envasado	v1	Cerrado	0	2	0	Joan Serra	Jaume Castell	0

Figura 5.45: Tabla filtrada e imprimida en formato PDF



Estas son las páginas principales y más importantes que tiene el proyecto. En el siguiente capítulo, se realizarán distintas pruebas para comprobar si el sitio web está listo para su uso.

6. Pruebas

Una vez terminado el proyecto, deberemos de realizar diversas pruebas que demuestren el correcto funcionamiento de este.

Como es sabido, todo proyecto de desarrollo web debe de ser, en la medida de lo posible, multiplataforma. Para ello, se va a ejecutar la página web en distintos navegadores, y se pondrán diversas capturas que demuestren que funciona y que está visualmente correcto. Las capturas serán de la página principal de la interfaz, ya que de esta manera se demostrará que hace correctamente la comunicación con la API y que tiene una buena presentación visual.

El proyecto ha sido desarrollado en Google Chrome 89.0.4389.90, y todas las capturas de este proyecto pertenecen a dicho navegador.



Figura 6.1: Página principal en Google Chrome

El otro navegador en el que se ha probado su funcionamiento ha sido Microsoft Edge 89.0.774.54.



Figura 6.2: Página principal en Microsoft Edge

Para probar la interoperabilidad del LMS, se comprobará que la comunicación con la API funcione correctamente. Para ello, se ha accedido a la pestaña *Network* en la herramienta para desarrolladores, y se ve que el código de estado devuelto es un 200 OK, que indica que la conexión ha ido correctamente.

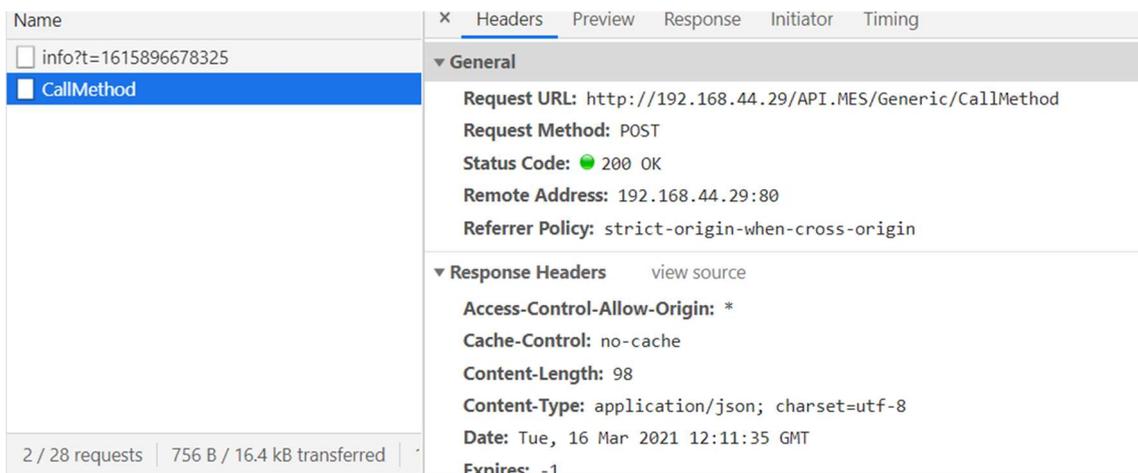


Figura 6.3: Comunicación con la API

Como se puede ver se hace un *POST* a la dirección <http://192.168.44.29/API.MES/Generic/CallMethod>, que es la dirección en la que se encuentra la API. En *Visual Studio Code* se ha realizado un “console.log” para que se muestre lo que esté devolviendo la API con la que se está comunicando la página web, para que se vea en la consola de esta herramienta.

```
Mirage: Passthrough request for POST http://192.168.44.29/API/MES/Generic/CallMethod mirage-esm.js?227b:7061
actions.js?ee27:80
▼ {data: {...}, status: 200, statusText: "OK", headers: {...}, config: {...}, ...} ⓘ
  ► config: {transformRequest: {...}, transformResponse: {...}, timeout: 0, xsrfCookieName: "XSRF-TOKEN", adapter: f, ...}
  ▼ data:
    retCode: 0
    ▼ retData: Array(1)
      ► 0: {...}
        length: 1
        ► __proto__: Array(0)
      retDesc: "Successful"
      ► __proto__: Object
    ► headers: {cache-control: "no-cache", content-length: "1300", content-type: "application/json; charset=utf-8", expires: "-1", pragma: "no-cache"}
    ► request: FakeRequest {eventListeners: {...}, readyState: 4, requestHeaders: {...}, requestBody: '{"Library":"MaterialesWeb","Method":"CheckPlantSta.0+01:00","Ffin":"2021/03/02T09:00:00.000+01:00"}', ...}
    status: 200
    statusText: "OK"
    ► __proto__: Object
```

Figura 6.4: Devolución de datos de la API

En el parámetro *request* se puede ver qué información es la que se está pasando, y en “data:retData” se encuentra el array que contiene la información solicitada.

También se ha realizado pruebas para la conexión y obtención de datos con WebSocket, de forma que, al abrirse la conexión, en el *listener* “onopen” salga por consola un mensaje. Como se puede ver, cada cierto tiempo se va enviando la información para ir actualizándola, sin necesidad de cerrar y volver a abrir la conexión.

```
Successfully connected to the echo websocket server...
▼ {retCode: 0, retDesc: "Successful", retData: Array(1)} ⓘ
  retCode: 0
  ► retData: [{...}]
  retDesc: "Successful"
  ► __proto__: Object
  ► {retCode: 0, retDesc: "Successful", retData: Array(1)}
  ► {retCode: 0, retDesc: "Successful", retData: Array(1)}
  ► {retCode: 0, retDesc: "Successful", retData: Array(1)}
```

Figura 6.5: Conexión por WebSocket

Por último, para probar que el proyecto que se ha creado cumple con las especificaciones marcadas del cliente, se deberán llevar a cabo pruebas de aceptación. Las pruebas FAT son las pruebas de aceptación que se realizan en nuestras instalaciones, mientras que las pruebas SAT son las pruebas que se llevarán a cabo en la ubicación final del sistema que se ha creado. Estas pruebas son el último paso a llevar a cabo antes de comenzar a funcionar en un entorno real. Como se sabe, este trabajo pertenece a un proyecto mucho más grande, por lo que estas pruebas se llevarán a cabo cuando se haya finalizado el proyecto y ya se vaya a poner en producción.



7. Conclusiones

En este último apartado se hará un repaso de todo el proyecto, comprobando si se han cumplido los objetivos marcados al principio de este documento, y de los problemas que se han encontrado. También se añadirá un apartado de líneas futuras, tanto a nivel personal, como de este proyecto y de la empresa.

7.1. Cumplimiento de los objetivos

En este proyecto se ha llevado a cabo una página web para un LMS multiplataforma, que se pueda ver en distintos navegadores y en distintos dispositivos. Se ha probado su funcionamiento en Google Chrome y en Microsoft Edge y no ha presentado ningún problema, ha funcionado de forma similar en ambos.

Los datos se representan de forma correcta en todo momento, con una interfaz muy intuitiva para aquellos usuarios que la usen por primera vez, gracias al uso de tecnologías como “echarts” para representar datos en gráficas, y Quasar para aquellas páginas que deban de usar tablas.

En todo momento se tiene conocimiento de los datos reales del proceso de producción, gracias a las dos formas en las que se obtiene la información: API *Rest* para datos estáticos y WebSocket para aquellos datos que deben ir actualizándose cada cierto tiempo para obtener los últimos datos del proceso.

Por lo tanto, el objetivo principal que debería de tener cualquier interfaz web LMS se ha superado con éxito.

En cuanto a los objetivos secundarios, he conseguido dominar y entender el funcionamiento del *framework* Vue y de las demás tecnologías utilizadas en este proyecto. Todas estas tecnologías eran en su mayoría desconocidas para mí, y he quedado satisfecho al aprender sobre Vue o Quasar. Gracias a estas tecnologías se ha conseguido obtener una buena impresión visual de la página web, además de conseguir que sea muy eficiente, gracias a las facilidades que ofrece herramientas como Vuex, que permite reducir los tiempos de carga.

En relación con los conocimientos adquiridos durante el grado, he conseguido ampliar mis conocimientos en el sector del desarrollo y diseño web, que era algo de lo que más me interesaba durante mis estudios. Gracias a este proyecto, como he comentado anteriormente, he aprendido tecnologías nuevas en este sector que estoy seguro de que me servirán para el futuro.

Por último, a nivel personal, he acabado muy satisfecho con la realización de este proyecto, pues a parte de la experiencia obtenida en el sector del desarrollo y del diseño web, he conseguido que la empresa siga confiando en mí para seguir con mi estancia en la misma.

7.2. Problemas y contratiempos surgidos

Durante el transcurso de este proyecto han surgido distintos problemas y contratiempos que se han tenido que ir resolviendo durante el transcurso de este.

- **Velocidad de comunicación con la API:** este proyecto ha estado enfocado en su gran mayoría en la parte *front* o del cliente de la página. Sin embargo, también se ha tenido que hacer algunos ajustes en la parte del servidor para que la comunicación con el cliente fuera correcta. Pese a que se comunica correctamente, la velocidad de entrega de datos de la API es algo lenta, por lo que se tendrán que incrementar los recursos proporcionados al servidor en el que se encuentra la API.
- **Problemas con el servidor de pruebas:** Antes de que los datos se obtuvieran de forma real, se usó una librería que simulara el funcionamiento de una API llamada Mirage JS⁴³. Sin embargo, al publicar la web con los servicios IIS no se pudo realizar la conexión con esta API simulada, por lo que se perdió bastante tiempo insertando los datos que se obtenían de esta API en el propio Vue.
- **Dificultades para la visualización en algunos dispositivos:** Esta página web contiene mucha información sobre la trazabilidad y el mantenimiento de la producción. El problema reside en que, en algunos dispositivos con una pantalla menos grande, no es capaz de mostrar tanta información, pues hay elementos que son de un tamaño fijo y no se han podido ajustar. Para el futuro queda conseguir cambiar el formato de la página para que muestre información de forma distinta dependiendo del dispositivo que se tenga y del tamaño del que se disponga.

7.3. Líneas futuras

Pese a que el proyecto ha terminado de forma satisfactoria, aún hay ciertos aspectos en los que se puede mejorar con vistas al futuro.

Por el momento, este proyecto solo tiene implementada una de las plantas de producción con la que cuenta el cliente. En el futuro se deberá de continuar con este proyecto añadiendo el resto de plantas, líneas, etc. con las que se cuenta, además del resto de departamentos. De esta manera, se obtendrá una digitalización total del LMS de todas las plantas de producción, lo que sin duda mejorará la eficiencia de estas.

Otro aspecto a mejorar del proyecto es la comunicación con la API ya que, como se ha dicho anteriormente, el tiempo de respuesta es lento. Esto se debe principalmente a

⁴³ Más información sobre Mirage JS: <https://miragejs.com/docs/getting-started/introduction/>

que la máquina en la que se encuentra el servidor carece de los recursos necesarios, por lo tanto, en el futuro se debería buscar incrementar dichos recursos.

Otro de los proyectos que se podrían llevar a cabo en el futuro es la creación de una aplicación móvil para esta página, ya que esta web por el momento solo está pensada para pantallas grandes, que permiten mostrar mucha información. De esta manera, será mucho más accesible y permitirá una mayor movilidad de sus usuarios.

En cuanto a Maval se refiere, se debería de seguir el camino para entrar en la próxima revolución industrial, la Industria 5.0. Para ello, se debería apostar aún más por la unión entre máquinas y humanos, desarrollando la inteligencia artificial para que realice procesos similares a los que realiza el pensamiento humano.

A nivel personal, espero seguir trabajando en proyectos similares a este, que utilicen herramientas y tecnologías para desarrollo web, con especial interés en las nuevas tecnologías que vayan surgiendo, ya que este mundo nunca para de evolucionar. También me gustaría adquirir nuevos conocimientos como puede ser el desarrollo de aplicaciones móviles, inteligencia artificial, etc. Mi idea es seguir progresando y no quedarme encajonado en un único sector, siempre buscando que me guste lo que hago.



8. Bibliografía

- [1] **Garrell, Antoni y Guilera, Llorenç. 2019.** *La Industria 4.0 en la sociedad digital*. València : ICG Marge SL, 2019.
- [2] Assembly Line. *Science Encyclopedia*. [En línea] <https://science.jrank.org/pages/558/Assembly-Line-History.html>.
- [3] **Blasco, Ana Inés, y otros. 2018.** *Industria 4.0: Fabricando el Futuro*. Buenos Aires : s.n., 2018.
- [4] **2019.** Por qué millonarios y economistas quieren que los robots paguen impuestos. *BBC News Mundo*. [En línea] 19 de Junio de 2019. <https://www.bbc.com/mundo/noticias-48685444>.
- [5] ¿Qué es Andon? Sistema de control visual de producción - Geinfor ERP. *Geinfor*. [En línea] <https://geinfor.com/business/que-es-andon-sistema-de-control-visual-de-produccion/>.
- [6] **Robben, Xavier.** *La cadena de valor de Michael Porter*.
- [7] **Rodríguez, Pablo, y otros. 2020.** *Sociedad Digital en España 2019*. Madrid : Taurus; Fundación Telefónica, 2020.
- [8] **Moya, Samuel. 2017.** Estándar ISA 95: Integración de los Sistemas de Control Empresarial. *ISA Sección Central México*. [En línea] 26 de Septiembre de 2017.
- [9] Controladores Lógicos Programables (PLCs). *Departamento de Ingeniería Eléctrica, Electrónica y de Control*. [En línea] http://www.ieec.uned.es/investigacion/Dipseil/PAC/archivos/Informacion_de_referencia_ISE6_1_1.pdf.
- [10] **Barrio Andrés, Moisés. 2018.** *Internet de las Cosas*. Madrid : Editorial Reus S.A., 2018.
- [11] Product Lifecycle Management (PLM) para la industria discreta. *Engineering USA*. [En línea] <https://www.engusa.com/es/solution/product-lifecycle-management-plm-for-discrete-industry>.
- [12] **Jürgen, Kletti. 2007.** *Manufacturing Execution System - MES*. Berlín : Springer, 2007.
- [13] **McClellan, Michael. 1997.** *Applying Manufacturing Execution Systems*. Boca Raton : CRC Press LLC, 1997.
- [14] Mejores estrategias para tu empresa. *Gestion.Org - Formación Empresarial*. [En línea] <https://www.gestion.org/fiestas-de-empresa-por-que-y-como-organizarlas/>.

- [15] **Parmenter, David. 2010.** *Key Performance Indicators (KPI): Developing, Implementing and Using Winning KPIs.* Hoboken : John Wiley & Sons, Inc., 2010.
- [16] Inteligencia de fabricación de arquitectura unificada de SIMATIC IT. *Siemens Digital Industries Software.* [En línea]
<https://www.plm.automation.siemens.com/global/es/products/manufacturing-operations-center/simatic-it-unified-architecture-manufacturing-intelligence.html>.
- [17] **2016.** Thomas Newcomen - el padre de la revolución industrial. *ElPlural.com.* [En línea] 24 de Julio de 2016. https://www.elplural.com/el-telescopio/innovacion/thomas-newcomen-el-padre-de-la-revolucion-industrial_122421102.
- [18] **Blázquez Morales, Luis Fernando.** Grandes Inventores. *Museo Virtual - Oficina Española de Patentes y Marcas.* [En línea]
http://historico.oepm.es/museovirtual/galerias_tematicas.php?tipo=INVENTOR&xml=Watt%2C+James.xml.
- [19] **Martínez Medina, Nuria. 2011.** Cartwright y la evolución de los telares mecánicos. <https://www.rtve.es/noticias/20110401/cartwright-evolucion-telares-mecanicos/421218.shtml>. [En línea] 1 de Abril de 2011.
<https://www.rtve.es/noticias/20110401/cartwright-evolucion-telares-mecanicos/421218.shtml>.
- [20] Henry Ford, historia de un sueño cumplido. *Ford España.* [En línea]
<https://www.ford.es/acerca-de-ford/historia>.
- [21] Tim Bernes-Lee: Biography. *World Wide Web Consortium.* [En línea]
<https://www.w3.org/People/Berners-Lee/>.

Apéndice A. Configuración inicial de la página

El sitio web que se ha definido en este proyecto debe de tener una base a partir de la que se van a ir creando las distintas páginas requeridas.

A.1. Fichero de enrutamiento

Aquí se encuentra la variable *routes*, con todas las rutas que se pueden utilizar en la página. Dentro de cada ruta se pueden ver las plantillas utilizadas, de las que derivan las páginas de cada ruta. En este proyecto se pueden distinguir dos plantillas: “LoginLayout.vue”, que solo tiene una página y que se corresponde con la página de inicio de sesión, y “MainLayout.vue”, donde se accede al iniciar sesión, y de la que derivan el resto de páginas de la interfaz.

```
const routes = [
  {
    path: '/',
    component: () => import('layouts/MainLayout.vue'),
    children: [
      { path: '', component: () => import('pages/Index.vue') },
      {
        path: '/Calidad/ControlCalidad',
        component: () => import('pages/ControlCalidadLayout.vue'),
        children: [
          {
            path: '',
            redirect: 'DocumentosControl'
          },
          {
            path: 'DocumentosControl',
            component: () => import('pages/DocumentosControl.vue')
          },
          {
            path: 'DocumentosControl/:idDocument',
            component: () => import('pages/DocumentInfo.vue')
          },
          {
            path: 'PlantillasControl',
            component: () => import('pages/PlantillasControl.vue')
          }
        ]
      }
    ]
  }
]
```

```

    {
      path: 'PlantillasControl/CreatePlantilla',
      component: () => import('pages/CreatePlantilla.vue')
    }
  ]
},
{
  path: '/Indicadores/Operacional',
  component: () => import('pages/OperacionalLayout.vue'),
  children: [
    {
      path: '',
      redirect: 'sant-lloren'
    },
    {
      path: 'sant-lloren',
      component: () => import('pages/SantLloren.vue')
    },
    {
      path: 'sant-lloren/home-care',
      component: () => import('pages/AreaLayout.vue')
    },
    {
      path: 'sant-lloren/home-care/LV1A',
      component: () => import('pages/LineLayout.vue')
    },
    {
      path: 'sant-lloren/home-care/LV1A/:idMaquina',
      component: () => import('pages/MachineLayout.vue')
    }
  ]
},
{
  path: '/Indicadores',
  component: () => import('pages/IndicadoresLayout.vue'),
  children: [
    {
      path: 'Resultados',
      component: () => import('pages/IndicadoresResultados.vue')
    },
    {
      path: 'Análisis',
      component: () => import('pages/IndicadoresAnálisis.vue')
    }
  ]
},
{
  path: '/Configuracion',
  component: () => import('pages/Configuracion.vue'),

```

```

        children: [
          {
            path: '',
            redirect: 'General'
          },
          {
            path: 'General',
            component: () => import('pages/General.vue')
          }
        ]
      },
      {
        path: '/login',
        component: () => import('layouts/LoginLayout.vue'),
        children: [{ path: '', component: () => import('pages/Login.vue') }]
      },
      {
        path: '*',
        component: () => import('pages/Error404.vue')
      }
    ];

export default routes;

```

A.2. Configuración inicial de Quasar

Dentro de este fichero se pueden encontrar distintas configuraciones que va a usar la página web, como pueden ser los *plugins* o animaciones que se verán en este proyecto.

```

module.exports = function (/* ctx */) {
  return {
    supportTS: false,

    boot: ['mirage', 'i18n', 'axios'],

    css: ['app.scss'],

    extras: [
      'roboto-font', // optional, you are not bound to it
      'material-icons' // optional, you are not bound to it
    ],
  },

```

```
build: {
  vueRouterMode: 'hash', // available values: 'hash', 'history'

  extendWebpack(cfg) {
    cfg.module.rules.push({
      enforce: 'pre',
      test: /\.?(js|vue)$/,
      loader: 'eslint-loader',
      exclude: /node_modules/
    });
  }
},

devServer: {
  https: false,
  port: 8080,
  open: true // opens browser window automatically
},

framework: {
  iconSet: 'material-icons', // Quasar icon set
  lang: 'en-us', // Quasar Language pack
  config: {},
  importStrategy: 'auto',
  plugins: ['Loading', 'Meta']
},

animations: ['fadeIn', 'fadeOut'],

ssr: {
  pwa: false
},

pwa: {
  workboxPluginMode: 'GenerateSW', // 'GenerateSW' or 'InjectManifest'

  workboxOptions: {}, // only for GenerateSW
  manifest: {
    name: 'Quasar App',
    short_name: 'Quasar App',
    description: 'A Quasar Framework app',
    display: 'standalone',
    orientation: 'portrait',
    background_color: '#ffffff',
    theme_color: '#027be3',
    icons: [
      {
        src: 'icons/icon-128x128.png',
        sizes: '128x128',
```

```

        type: 'image/png'
      },
      {
        src: 'icons/icon-192x192.png',
        sizes: '192x192',
        type: 'image/png'
      },
      {
        src: 'icons/icon-256x256.png',
        sizes: '256x256',
        type: 'image/png'
      },
      {
        src: 'icons/icon-384x384.png',
        sizes: '384x384',
        type: 'image/png'
      },
      {
        src: 'icons/icon-512x512.png',
        sizes: '512x512',
        type: 'image/png'
      }
    ]
  }
},

cordova: {
},

capacitor: {
  hideSplashscreen: true
},

electron: {
  bundler: 'packager', // 'packager' or 'builder'

  packager: {
  },

  builder: {

    appId: 'frontacmarca'
  },

  nodeIntegration: true,

  extendWebpack(/* cfg */) {
  }
}
}

```

```
};  
};
```

Apéndice B. Código página web

Dentro de este apéndice se encuentran trozos de código que han sido utilizados para la creación de las distintas páginas que conforman el sitio web.

B.1. Código de la página de inicio de sesión

Dentro de este punto, se encontrará el código utilizado en la página de inicio de sesión, llamada “Login.vue”, que servirá de ejemplo para la estructura que tendrán el resto de las páginas.

```
<template>
  <q-page>
    <div id="login-container" class="row">
      <div id="left-col" class="col-6">
        <div class="row justify-center items-center fit content-end">
          <div id="message">
            <div id="firstMessage">Welcome</div>
            <div id="secondMessage">Benvinguts · Bienvenidos · Willkommen
          </div>
        </div>
      </div>
      <div id="rightCol" class="col-6">
        <div class="row justify-center items-center fit q-pl-xl">
          <div id="loginForm">
            <div class="col-12">
              <div id="logo" class="row q-my-sm justify-center">
                
              </div>
              <div id="usuario" class="row justify-center">
                <div>
                  <div>Usuario</div>
                  <input v-model="username" />
                </div>
              </div>
              <div id="contrasena" class="row justify-center">
                <div>
                  <div>Contraseña</div>
                  <input type="password" v-model="password" />
                </div>
              </div>
            <div class="row justify-center">

```

```

        <button id="loginButton" @click="userExists()">Login</but
ton>
    </div>
</div>
</div>
</div>
</div>
</div>
</q-page>
</template>

<script>
import { mapActions } from 'vuex';

const data = () => ({
  username: '',
  password: ''
});

const userExists = async function _userExists() {
  this.$q.loading.show();
  const isLoggedIn = await this.doLogin({ username: this.username, password
: this.password });
  this.$q.loading.hide();

  if (!isLoggedIn) {
    alert('USUARIO Y CONTRASEÑA ERRÓNEOS');
    this.username = '';
    this.password = '';
  } else {
    this.$router.push('/');
  }
};

export default {
  meta: {
    title: 'Login'
  },
  name: 'Login',
  data,
  methods: {
    ...mapActions('login', ['doLogin']),
    userExists,
    iniciarSesion
  }
};
</script>

<style lang="scss">

```

```

#left-col {
  > div {
    background: rgba(20, 63, 146, 0.66);
  }
}

#rightCol {
  background-color: #143f92;
}

#message {
  height: 50vh;
}

#logo {
  height: 20vh;
}

#usuario {
  height: 10vh;
}

#contrasena {
  height: 10vh;
}

#loginButton {
  width: 23vh;
  border-radius: 3px;
  opacity: 1;
  color: white;
  background: $primary;
}

#loginForm {
  width: 40vh;
  height: 50vh;
  border-radius: 15px;
  background-color: white;
}

#login-container {
  height: 100vh;
  background-image: url('~assets/common/images/welcomebackground.png');
  background-size: 75%;
}

#firstMessage {
  color: #fff;
}

```



```

font: normal normal 900 93px/111px Roboto;
font-size: 3rem;
text-align: left;
}

#secondMessage {
  opacity: 0.5;
  color: #fff;
  font: normal normal normal 20px/34px Roboto;
  text-align: center;
}
</style>

```

B.2. Módulo *login* de Vuex

En este proyecto se utiliza Vuex, que está dividido en módulos para que sea más accesible y fácil de manejar. Dentro de estos módulos se encuentra el módulo *login*, con sus correspondientes ficheros que sirven para dar funcionalidad.

Fichero actions.js

Dentro de este fichero se encuentran las acciones de inicio de sesión, desde donde se llama tanto al servicio que llama a la API, como a las *mutations* que modificarán el estado de Vuex.

```

import { login } from '../services/UserService';

export const setLogged = ({ commit }, user) => {
  commit('setLogged');
  commit('setUser', user);
};

export const doLogin = async ({ dispatch }, { username, password }) => {
  const result = await login({ username, password });
  console.log(result);
  const user = result.data.userData[0];

  if (result.status === 200 && user.isLoggedIn) {
    dispatch('setLogged', user);

    return true;
  }
}

```

```
return false;
};
```

Fichero mutations.js

Dentro de este archivo se encuentran las *mutations* que modifican el estado de este módulo.

```
export function setLogged(state) {
  state.logged = !state.logged;
}

export function setUser(state, user) {
  state.user = user;
}
```

Fichero state.js

En este archivo están las variables utilizadas en la página, a las que se podrán acceder desde cualquier componente, y que sirven para detectar si se ha iniciado sesión, entre otras cosas.

```
export default {
  logged: false,
  user: null
};
```

B.3. Llamadas a API Rest

Como se ha comentado en distintas secciones del proyecto, hay datos que deberán de ser obtenidos de una API. Con este fin, se han definido servicios para cada una de las partes de nuestra página.

Lo primero que se ha hecho es crear un fichero donde se puedan ejecutar funciones para los dos métodos más frecuentes, y los que se usarán durante este proyecto, *get* y *post*. En estas funciones se indicará el método a utilizar, y una variable *data*, donde se encontrarán la clase y el método a ejecutar, dependiendo de los datos que queramos obtener en cada momento, además de los parámetros a enviar, en caso de que se quiera enviar alguno.

```
import { apiMES } from './API';

export const getGenericMethod = (LibraryName, MethodName, DataObject) =>
  apiMES({
```



```

    method: 'get',
    data: { Library: LibraryName, Method: MethodName, Data: DataObject }
  });

export const postGenericMethod = (LibraryName, MethodName, DataObject) =>
  apiMES({
    method: 'post',
    data: { Library: LibraryName, Method: MethodName, Data: DataObject }
  });

```

Dentro de estas funciones se llama a otra función, “apiMES”, donde se ejecutará una promesa *Axios*, que recibirá como argumento la *url* en la que se encuentra la API de la que obtendremos todos los datos.

```

import axios from 'axios';

export const apiMES = axios.create({
  baseURL: 'http://192.168.44.29/API.MES/Generic/CallMethod'
});

```

B.4. Manejo de Pop-ups

En la página hay ocasiones en las que se necesitan utilizar *pop-ups*. En este apéndice se va a mostrar un ejemplo de cómo se manejan en la página principal de la página, aunque se puede extrapolar a las demás páginas que usen estos *pop-ups*.

En la página se definirán los componentes de la barra lateral principal, y la del *pop-up*, con sus respectivos eventos y los parámetros que se deben pasar a cada componente.

```

<main-menu @sidebar-opened="sidebarOpened = true" @pin-
opened="pinOpened = true"></main-menu>
<q-dialog v-model="sidebarOpened" position="left" full-height>
  <left-menu
    class="left-menu"
    :sidebar-opened="sidebarOpened"
    @go-back="openPlants"
    @hide-menus="sidebarOpened = false"
  ></left-menu>
</q-dialog>

```

```

<div class="q-mt-sm">
  <q-btn
    flat
    @click="$emit('sidebar-opened')"
    title="Abrir departamentos"
    dense

```

```

        icon="menu"
        class="menu__button"
    />
</div>

```

En el componente `mainMenu.vue` se deberá llamar al evento “`sidebar-opened`” al hacer clic, que permitirá poner la variable `sidebarOpened` a `true` y, de esta manera, mostrar el componente `leftMenu.vue`, que es en el que se muestran los departamentos.

B.5. Menús expandibles

Dentro del proyecto se van a usar menús que deban de expandirse, mostrando otros menús expandibles, botones, etc. Esta es la estructura típica que usarán estos menús, usando las utilidades que Quasar ofrece.

```

<q-scroll-area class="left-menu">
  <q-list padding>
    <q-item>
      <q-item-section avatar>
        <q-icon name="public" />
      </q-item-section>
      <q-item-section>
        <div class="title-delegation">Sant Llorenç...</div>
      </q-item-section>
    </q-item>
    <q-item clickable v-ripple @click="$emit('go-back')">
      <q-item-section avatar>
        <q-icon name="arrow_back" />
      </q-item-section>

      <q-item-section> Volver </q-item-section>
    </q-item>
    <div v-if="options === null">
      <div class="q-ma-sm" v-
for="(skeleton, index) in 10" :key="index">
        <q-skeleton type="rect"></q-skeleton>
      </div>
    </div>

    <q-expansion-item
      v-else
      class="menu-options bold-weight information-data"
      switch-toggle-side
      :content-inset-level="1"
      v-for="item in options"
      :key="item.id"

```

```

      :label="item.name.toUpperCase()"
    >
    <q-btn
      :disabled="!suboption.implemented"
      align="left"
      class="full-width"
      @click="goToPage(item.name, suboption.name)"
      v-for="suboption in item.suboptions"
      :key="suboption.id"
      flat
      no-caps
    >{{ suboption.name }}</q-btn
  >
</q-expansion-item>
</q-list>
</q-scroll-area>

```

B.6. Manejo del svg del mapa

En esta página web se podrán encontrar distintas imágenes en las que se deberán ejecutar eventos. En el caso del mapa del menú principal, se definirá un *pop-up* cuya opacidad se encuentra a 0 por defecto, y que se mostrará al situar el puntero sobre las plantas. En este trozo de código se han insertado puntos suspensivos para que no se haga demasiado tedioso la codificación de la imagen en *base64*.

```

<g id="DIN_PopUp" data-name="DIN PopUp" style="opacity: 0">
  <g id="fondoPopUp">
    <image
      width="274"
      height="244"
      transform="translate(608 180)"
      xlink:href="data:image/png;base64,iVBORw0KGgoAAAANSUgAAARIAAAD0CAYAAABXXakIAAAACXBIWXMAAAASAAALEgHS3X78AAASQEI EQVR4Xu2d63bjtg5GYS
czbd//YXs6ufj8cBB/gsGLDDmTsfZei4uULMuSSWYBdKY9nE4nAwCocBwdAAA wApEAQJnn0QF
fzeFw0Iy0Adg7p2+2JnG45/UgBYDvwz3ls6lIbhTHLe8BgCWrA31LsWwiko5ARpIYvQ4A84yC
OX19C6GURJII ZLQ92j/70sCemQna7Ji4b7FdEcpNIhkIZKadbc++BrB3RkGrr69t3ySU1SIJE
snao7rXB0dbaGUbs/WivVYmq0TSkIjWM+2sVhALwBy9qcrJcmnE/U2xrJHJtEg6Esmk0SqWtB
0EArCeLKPIxJGVeIyF9rRMPv4gLZFITxzHiX3ZeRSkAtCmlymMBPKebCsnW8be6XA4HGZkMhT
...
    />
    <path
      d="M850,384.5H695.41-11.3,12-10.1-12H626a2.9,2.9,0,0,1,3-
3v-182a2.9,2.9,0,0,1,3-
3H850a2.9,2.9,0,0,1,3,3v182A2.9,2.9,0,0,1,850,384.5Z"
      transform="translate(-1 -2.5)"

```

```

        style="fill: #fff"
      />
      <path
        d="M850,384.5H695.41-11.3,12-10.1-12H626a2.9,2.9,0,0,1,3-
3v-182a2.9,2.9,0,0,1,3-
3H850a2.9,2.9,0,0,1,3,3v182A2.9,2.9,0,0,1,850,384.5Z"
        transform="translate(-1 -2.5)"
        style="fill: none; stroke: #b3b3b3; stroke-miterlimit: 10"
      />
    </g>
    <g id="txt_ESPAÑA" data-name="txt ESPAÑA">
      <text
        transform="translate(647.5 229.8)"
        style="font-family: Roboto-Regular, Roboto; font-
size: 15px; fill: #8b95a6"
      >
        ES
        <tspan x="17.4" y="0" style="letter-spacing: -
0.06744791666666666em">P</tspan>
        <tspan x="25.9" y="0">AÑA</tspan>
      </text>
    </g>
    <g id="DIN_boton_santLlorenc" data-
name="DIN boton santLlorenc" @click="goToPage">
      <rect x="647.5" y="253" width="179" height="41" rx="3" style=
"fill: #336cde" />
      <text
        transform="translate(659.6 278.3)"
        style="font-family: Roboto-Bold, Roboto; font-
size: 15px; font-weight: 700; fill: #fff"
      >
        Sant Llorenç d'Ho
        <tspan x="119.6" y="0" style="letter-
spacing: 0.024348958333333334em">r</tspan>
        <tspan x="125.4" y="0" style="letter-spacing: -
0.014583333333333334em">t</tspan>
        <tspan x="130.2" y="0" style="letter-spacing: -
0.00006510416666666667em">ons</tspan>
      </text>
    </g>
    <g id="DIN_boton_bigues" data-name="DIN boton bigues">
      <rect x="647.5" y="313" width="179" height="41" rx="3" style=
"fill: #336cde" />
      <text
        transform="translate(689.5 338.3)"
        style="font-family: Roboto-Bold, Roboto; font-
size: 15px; font-weight: 700; fill: #fff"
      >
        Bigues i Riells

```

```

    </text>
  </g>
</g>

```

```

const getSvg = function _getSvg() {
  const svg = document.getElementById('DIN_PopUp');

  svg.addEventListener('mousemove', function () {
    svg.setAttributeNS(null, 'style', 'opacity:1');
  });

  svg.addEventListener('mouseout', function () {
    svg.setAttributeNS(null, 'style', 'opacity:0');
  });
};

const mounted = function mounted() {
  this.getSvg();
};

```

B.7. Uso de *slots*

En el proyecto Vue se ha definido un componente llamado `sideMenu.vue`, cuyo código es común para muchas de las páginas de la interfaz. Sin embargo, cada departamento debe de tener sus propias opciones de menú, por lo que se deben usar *slots*. A continuación se verá cómo se han implementado estos *slots* en la página Operacional.

`sideMenu.vue`

```

<template>
  <q-scroll-area class="calidad-menu">
    <q-list class="row items-center fields">
      <q-item class="back q-pl-sm">
        <q-item-section>
          <q-btn flat no-caps class="q-pr-sm" v-go-back="/">
            <q-icon class="arrow q-mr-sm" name="arrow_back" />
            <q-item-label class="back-label label-size">Volver</q-item-
label>
          </q-btn>
        </q-item-section>
      </q-item>

      <q-item>
        <q-item-section class="side-menu-options q-mt-sm">
          <q-item-label><slot name="main"></slot></q-item-label>

```

```

        <q-item-label class="row items-center q-ml-sm">
            <slot name="buttons"></slot>
        </q-item-label>
    </q-item-section>
</q-item>
</q-list>
</q-scroll-area>
</template>

```

Operacionallayout.vue

```

<side-menu>
    <template v-slot:main>
        <q-btn
            class="btn-operation fields q-mr-lg"
            flat
            align="left"
            @click="$router.push(`/Indicadores/Operacional/${plantRoute}`
        )"
        >Sant Llorenç D'Hortons</q-btn>
    <></template>
    <
    <template v-slot:buttons>
        <q-expansion-item
            class="fields btn-operation q-ml-sm q-mr-lg q-mb-sm"
            expand-icon-toggle
            switch-toggle-side
            v-for="operation in operacional"
            @click="setArea(operation)"
            :key="operation.id"
            :label="operation.subname.toUpperCase()"
            :to="`/Indicadores/Operacional/sant-
lloren/${operation.route}`"
        >
            <div v-if="operation.subname === 'Home Care'">
                <q-expansion-item
                    class="menu-lines q-mb-sm"
                    expand-icon-toggle
                    switch-toggle-side
                    :header-inset-level="0.15"
                    :content-inset-level="0.75"
                    v-for="linea in operation.line"
                    :label="linea.name"
                    @click="setLine(linea)"
                    :key="linea.id"
                    :to="`/Indicadores/Operacional/sant-
lloren/${operation.route}/${linea.name}`"
                >
                    <q-btn

```



```

        align="left"
        class="fields q-py-sm btn-operation full-width"
        v-for="lineOption in lineOptions"
        :key="lineOption.id"
        flat
        no-caps
        :to="`/Indicadores/Operacional/sant-
lloren/${operation.route}/${linea.name}/${lineOption.id}`"
        >{{ lineOption.name }}</q-btn
    ></q-expansion-item
    >
    </div>
</q-expansion-item>
</template>
</side-menu>

```

B.8. *Select* intervalo

Al seleccionar un intervalo, el componente *select* de Quasar ofrece una propiedad que permite emitir un evento al componente padre para que, dependiendo de lo que se haya seleccionado, realice unas acciones u otras.

```

<q-select
  dense
  label="Intervalo"
  outlined
  v-model="selectedFecha"
  :options="intervalo"
  @input="$emit('change-intervalo', selectedFecha)"
/>

```

```

const changeIntervalo = function _changeIntervalo(selectedFecha) {
  this.resetSelect = false;
  this.showIntervalo = false;
  if (selectedFecha === 'Seleccionar Fechas') {
    this.resetSelect = true;
    this.showPopup = true;
  } else {
    let dateObject = Date.now();
    if (selectedFecha === 'Ayer') {
      dateObject = date.subtractFromDate(dateObject, { days: 1 });
    }

    this.intervalo = date.formatDate(dateObject, 'DD/MM/YYYY');

    if (!this.connection) {

```

```

    this.webSocketListeners();
  } else {
    this.sendMessage();
  }
}
};

```

Uno de los elementos más útiles que ofrece Quasar, es un calendario y un reloj que permite indicar la fecha exacta en la que se quieren mostrar los indicadores de la parte del proceso de producción que se quiera. Para manejar estos elementos se cuenta con variables definidas en *data* que manejen las fechas por una parte y las horas por otra, tanto de inicio como de fin.

```

<q-dialog class="popup-warnings" v-model="showPopup">
  <q-date class="q-mr-sm" v-
model="startDate" subtitle="Fecha inicio"> </q-date>
  <q-time class="q-mr-xl" v-model="startHour" format24h> </q-time>
  <q-date class="q-mr-sm" v-
model="endDate" subtitle="Fecha fin"> </q-date>
  <q-time class="q-mr-sm" v-model="endHour" format24h> </q-time>
  <div>
    <div class="row q-pb-sm">
      <q-btn
        @click="sendDates()"
        flat
        no-caps
        size="15px"
        label="Aceptar"
        class="standar-button control-buttons full-width"
      ></q-btn>
    </div>
    <div class="row">
      <q-btn no-caps flat class="close-button q-px-md" v-close-
popup>Cerrar</q-btn>
    </div>
  </div>
</q-dialog>

```

B.9. Conexión WebSocket

Para obtener datos en tiempo real se deberá crear una conexión WebSocket. Para ello se necesitará la dirección IP donde se encuentra el servidor, que en este caso es 192.168.44.29, y el puerto en el que se encuentra el WebSocket. En este proyecto se usan varios puertos para obtener distintos tipos de datos. En este ejemplo se usará el 8082.

```

const webSocketListeners = function _webSocketListeners() {
  this.showResumenData = false;

```

```

this.connection = new WebSocket('ws://192.168.44.29:8082');
this.connection.onopen = () => {
  const connectionINTERVAL = setInterval(() => {
    if (this.connection) {
      this.sendMessage();

      clearInterval(connectionINTERVAL);
    }
  }, 1000);
};
this.connection.onmessage = eventMessage => {
  const json = JSON.parse(eventMessage.data);
  const retData = json.retData[0];
  this.plant = retData;
  this.showResumenData = true;
};
this.connection.onerror = error => {
  console.log(error);
};

this.connection.onclose = () => {
  this.connection = null;
};
};

```

Con el *listener* “onopen” se esperará a que se abra la conexión con el WebSocket. Usando el método “setInterval()”, cada segundo esperará a que la variable *connection* deje de ser *null*, y entonces enviará un mensaje al servidor. Una vez que el servidor haya recibido el mensaje, irá enviando constantemente mensajes hasta que se le indique para. El intervalo de tiempo en el que debe de enviar mensajes lo define el cliente, en este caso son 20 segundos. Por lo tanto, los datos de los indicadores se irán actualizando cada 20 segundos.

El *listener* “onmessage” quedará esperando a que reciba un mensaje por parte del servidor. A continuación, se mostrará la forma en la que se reciben los datos.

1. {retCode: 0, retDesc: "Successful", retData: Array(1)}
 1. retCode: 0
 2. retData: Array(1)
 1. 0:
 1. availabilityPercentage: (...)
 2. badPieces: (...)
 3. goodPieces: (...)
 4. id: (...)
 5. name: (...)
 6. percentageOee: (...)
 7. performancePercentage: (...)
 8. plantAvailability: (...)
 9. plantPerformance: (...)
 10. plantQuality: (...)

- 11. `qualityPercentage`: (...)
- 12. `topIncidence`: (...)
- 13. ...
- 2. `length`: 1
- 3. `__proto__`: Array(0)
- 3. `retDesc`: "Successful"
- 4. `__proto__`: Object

Como se puede ver, lo que se recibe es un objeto con distintos parámetros, donde el más reseñable es el parámetro “retData”, donde se encuentra la información, en este caso la información de la planta Sant Llorenç d’Hortons. En este caso se ha optado por manejarlo como un objeto JSON, ya que facilita la representación de los datos.

B.10. Gráfica OEE

Dentro del proyecto se usan muchos tipos de gráficas, pero de entre ellas, la que muestra el OEE es la más importante, pues es la que indica la eficiencia general de los distintos equipos de la producción. A continuación, se verá el código usado para crear dicha gráfica.

```
<template>
  <div class="no-
hover" :id="idDiv" style="width: 100%; height: 100%"></div>
</template>

<script>
import * as echarts from 'echarts/dist/echarts';
import { forEach } from 'lodash';

const data = () => ({
  graphValues: {}
});

const setColour = function setColour() {
  if (this.objetivo !== null) {
    if (this.percentage < this.objetivo) {
      return '#ff1d1d';
    }
  }

  return '#19C200';
};

const barColour = function barColour() {
  if (this.plantAreas) {
    return '#fff';
  }
}
```

```

    return '#DAE4E6';
  });

const placeholderStyle = {
  normal: {
    color: 'rgba(0,0,0,0)',
    borderWidth: 0,
    cursor: 'pointer'
  },
  emphasis: {
    color: 'rgba(0,0,0,0)',
    borderWidth: 0,
    cursor: 'pointer'
  }
};

const onReadyChart = function onReadyChart(instance) {
  this.chart = instance;
};

const setParameters = function _setParameters() {
  const series = [];
  series.push(
    {
      type: 'pie',
      hoverAnimation: false,
      center: ['50%', '65%'],
      startAngle: 180,
      labelline: { normal: { show: false } },
      label: { normal: { position: 'center' } },
      data: [
        {
          value: 0,
          itemStyle: { normal: { color: this.barColour() } }
        },
        {
          value: 0,
          itemStyle: placeholderStyle
        }
      ]
    },
    {
      type: 'pie',
      hoverAnimation: false,
      center: ['50%', '65%'],
      startAngle: 180,
      endAngle: 0,
      labelline: { normal: { show: false } },

```

```

    label: { normal: {} },
    data: [
      {
        value: (this.percentage * 180) / 360,
        itemStyle: { normal: { color: this.setColour() } }
      },
      {
        value: 100 - (this.percentage * 180) / 360,
        itemStyle: placeholderStyle
      }
    ]
  }
);
if (this.objetivo !== null) {
  series.push({
    type: 'pie',
    hoverAnimation: false,
    center: ['50%', '65%'],
    startAngle: 180 - (180 * this.objetivo) / 100,
    labelLine: { normal: { show: false } },
    data: [
      {
        value: 0.75,
        itemStyle: {
          normal: {
            color: '#000'
          }
        },
      },
      {
        label: {
          formatter: ` ${this.objetivo}% `,
          padding: this.labelProps.padding,
          color: '#ACBBD4',
          fontWeight: 'bold',
          fontSize: this.labelProps.fontSize,
          show: this.showLabel
        }
      },
      {
        value: 100,
        itemStyle: placeholderStyle
      }
    ]
  });
}

if (this.radius) {
  forEach(series, serie => {
    if (this.homeCare) {
      serie.center = ['50%', '85%'];
    }
  });
}

```



```

        serie.radius = ['150%', '110%'];
    } else {
        serie.radius = ['100%', '75%'];
    }
    });
} else if (this.homeCare) {
    forEach(series, serie => {
        serie.center = ['55%', '80%'];
    });
}

if (this.percentage !== null) {
    this.graphValues = {
        title: {
            text: `${this.percentage}%`,
            left: `${this.titleProps.left}%`,
            top: `${this.titleProps.top}%`,
            textAlign: 'center',
            textStyle: {
                fontWeight: 'bold',
                fontSize: this.titleProps.fontSize,
                color: '#505f79',
                textAlign: 'center'
            }
        },
        series
    };
}

const getGraphDataInit = function getGraphDataInit(chart) {
    chart.clear();
    chart.setOption(this.graphValues);
};

const mounted = async function mounted() {
    const chartContainer = document.getElementById(this.idDiv);
    const chart = echarts.init(chartContainer, null, { renderer: 'svg', width: 'auto', height: this.height });
    window.addEventListener('resize', () => {
        chart.resize();
    });
    this.setParameters();

    await this.getGraphDataInit(chart);
};

const props = {
    percentage: {

```

```

    type: Number,
    required: true,
    default: 0
  },
  objetivo: {
    type: Number,
    default: null
  },
  height: {
    type: Number,
    required: true
  },
  idDiv: {
    type: String,
    default: 'default'
  },
  titleProps: {
    type: Object,
    default() {
      return { top: 40, left: 49, fontSize: 25 };
    }
  },
  showLabel: {
    type: Boolean,
    default: false
  },
  radius: {
    type: Boolean,
    default: true
  },
  alt: {
    type: Boolean,
    default: false
  },
  plantAreas: {
    type: Boolean,
    default: false
  },
  homeCare: {
    type: Boolean,
    default: false
  },
  labelProps: {
    type: Object,
    default() {
      return { padding: 0, fontSize: 10 };
    }
  }
}
};

```



```

export default {
  name: 'HalfPieGraph',
  props,
  methods: {
    setParameters,
    setColour,
    onReadyChart,
    getGraphDataInit,
    barColour
  },
  data,
  mounted
};
</script>

```

Esta gráfica, y todas las demás que aparecen a lo largo del sitio web, se han creado gracias a una herramienta para JavaScript llamada Apache ECharts⁴⁴. Estas gráficas son realmente imágenes svg, a las que se le pueden pasar como parámetros los datos que se quieran mostrar.

En el método *setParameters* se asignan los datos, tanto como de las barras, como la del objetivo que está marcado en negro. En caso de que el porcentaje del OEE sea menor que el objetivo, la barra se mostrará en rojo para que quede resaltado que hay un problema. En el caso contrario, se mostrará en verde. Como se puede ver, este componente consta de muchas propiedades, ya que dependiendo de si la gráfica sea para una planta, para un área, para una línea o para una máquina, cambiarán distintas cosas, como pueden ser los colores de las gráficas, el objetivo, etc.

B.11. Búsqueda en una tabla

En esta página se ha implementado un buscador que permita buscar en una tabla. Para ello, en el componente “documentsSearch.vue” se ha definido una propiedad en forma de función, a la que se le pasarán los datos que hayamos introducido en cada campo.

```

<q-btn
  no-caps
  class="standar-button calidad-buttons"
  label="Buscar"
  @click="
    filteredField([
      selectedDate,
      selectedLinea,
      selectedTurno,

```

⁴⁴ Más información en: <https://echarts.apache.org/en/feature.html>

```

        selectedOrdenProduccion,
        selectedTipo,
        selectedEstado
    ])
    "
  ></q-btn>
  props: {
    filteredField: {
      type: Function,
      required: true,
      default: null
    }
  },

```

En el componente padre entonces se enviarán los datos al componente de la tabla para que muestre la información que se corresponda con dicha búsqueda.

```

  <documents-search :filtered-field="filteredField"></documents-search>
  <documents-table :filtered-value-updated="filteredValue"></documents-table>

  const filteredField = function _filteredField(value) {
    this.filteredValue = value;
  };

```

Y, por último, en el componente “documentsTable.vue” se ejecutará una función teniendo en cuenta si hay o no hay valores en el array *filteredValue*. Con la utilidad *watch* de Vue se podrá controlar cuándo ha cambiado dicho array, y de esta manera ejecutar la función que actualice la tabla. La variable *initialData* contiene la información inicial de la tabla obtenida al consumir la API, donde se encuentran todos los documentos de control que posteriormente podrán ser filtrados.

```

  watch: {
    filteredValueUpdated() {
      this.getDocuments();
    }
  },
  mounted() {
    this.getDocuments();
  }

  const getDocuments = function _getDocuments() {
    this.showReturnData = false;
    if (this.filteredValueUpdated.length === 0) {
      this.documentsData = this.initialData;
    } else {
      this.documentsData = this.filterDocuments({

```



```

    creacion: this.filteredValueUpdated[0],
    linea: this.filteredValueUpdated[1],
    turno: this.filteredValueUpdated[2],
    ordenProduccion: this.filteredValueUpdated[3],
    tipo: this.filteredValueUpdated[4],
    estado: this.filteredValueUpdated[5]
  });
}
this.pagesNumber = this.documentsData.length / this.pagination.rowsPerPage;

this.showReturnData = true;
});

const filterDocuments = function _filterDocuments(request) {
  const { creacion } = request;
  const { linea } = request;
  const { turno } = request;
  const { ordenProduccion } = request;
  const { tipo } = request;
  const { estado } = request;
  let aux = this.initialData;
  if (creacion) {
    const dates = creacion.split('-');
    const ini = dates[0].split('/');
    const fin = dates[1].split('/');

    const from = new Date(ini[0], ini[1] - 1, ini[2]).getTime(); // -
    1 because months are from 0 to 11
    const to = new Date(fin[0], fin[1] - 1, fin[2]).getTime();
    aux = filter(aux, item => from <= new Date(item.creacion).getTime() &
    & to >= new Date(item.creacion).getTime());
  }

  if (linea) {
    aux = filter(aux, item => item.linea === linea);
  }

  if (turno) {
    aux = filter(aux, item => item.turno === turno);
  }

  if (ordenProduccion) {
    aux = filter(aux, item => item.ordenProduccion === ordenProduccion);
  }

  if (tipo) {
    aux = filter(aux, item => item.tipo === tipo);
  }
}

```

```

if (estado) {
  aux = filter(aux, item => item.estado === estado);
}
return aux;
};

```

Para filtrar los documentos de control se ha usado la utilidad de Lodash *filter*.

B.12. Creación de una tabla

Una de las mejores utilidades que Quasar ofrece es el elemento HTML “q-table”. Este elemento tiene una gran cantidad de propiedades que lo hace muy flexible y fácil de usar, como se verá a continuación

```

<transition appear enter-active-class="animated fadeIn" leave-active-
class="animated fadeOut">
  <div id="tabla-documentos" v-show="showReturnData">
    <q-table
      :data="documentsData"
      dense
      class="table-padding documents-table label-size common-tables"
      @row-click="goToRow"
      :columns="columns"
      row-key="id"
      :pagination="pagination"
      :filter="filter"
      separator="none"
    >
      <template v-slot:top-left>
        <q-btn
          class="q-mr-md"
          color="blue"
          icon-right="archive"
          label="Exportar en CSV"
          no-caps
          @click="exportTable"
        />
        <q-btn color="blue" icon-
right="picture_as_pdf" label="Imprimir en PDF" no-
caps @click="printContent" />
      </template>
      <template v-slot:top-right>

```

```

    <q-input borderless dense debounce="300" v-
model="filter" placeholder="Filtrar">
    <template v-slot:append>
        <q-icon name="search" />
    </template>
</q-input>
</template>

<template v-slot:body-cell-linea="props">
    <q-td :props="props">
        <b>{{ props.row.linea }}</b>
    </q-td>
</template>
<template v-slot:body-cell-ordenProduccion="props">
    <q-td class="numeric-field" :props="props">
        {{ props.row.ordenProduccion }}
    </q-td>
</template>
<template v-slot:body-cell-fugasExtra="props">
    <q-td class="numeric-field text-right" :props="props">
        {{ props.row.fugasExtra }}
    </q-td>
</template>
<template v-slot:body-cell-incidencias="props">
    <q-td class="numeric-field text-right" :props="props">
        {{ props.row.incidencias }}
    </q-td>
</template>
<template v-slot:body-cell-botellas="props">
    <q-td class="numeric-field text-right" :props="props">
        {{ props.row.botellas }}
    </q-td>
</template>
<template v-slot:body-cell-revisado="props">
    <q-td :props="props">
        <q-icon class="done-icon" name="done"></q-icon>
        {{ props.row.revisado }}
    </q-td>
</template>
<template v-slot:body-cell-aprobado="props">
    <q-td :props="props">
        <q-icon class="done-icon" name="done"></q-icon>
        {{ props.row.aprobado }}
    </q-td>
</template>
<template v-slot:body-cell-acciones="props">
    <q-td :props="props">
        <q-btn dense round flat icon="more_horiz"></q-btn>
    </q-td>

```

```

    </template>
    <template v-slot:body-cell-observaciones="props">
      <q-td class="observaciones-
field" :props="props"> {{ props.row.observation }} </q-td>
    </template>
  </q-table>
</div>
</transition>

```

A la propiedad *data* se le debe de asignar la variable *documentsData*, que es en la que se encuentra la información que va a aparecer en la tabla, que en el apéndice anterior se ha visto cómo se asigna. Con la propiedad *columns* se define el nombre de cada columna, y la variable del objeto *documentsData* a la que se asocia dicha columna, entre otras propiedades.

```

columns: [
  {
    name: 'id',
    required: true,
    label: 'Id',
    align: 'left',
    field: 'id',
    sortable: true
  },
  { name: 'creacion', align: 'left', label: 'Creación', field: 'creacion', sortable: true },
  {
    name: 'linea',
    align: 'left',
    label: 'Línea',
    field: 'linea',
    sortable: true
  },
  { name: 'turno', align: 'left', label: 'Turno', field: 'turno', sortable: true },
  { name: 'ordenProduccion', align: 'left', label: 'Orden Prod.', field: 'ordenProduccion', sortable: true },
  { name: 'tipo', align: 'left', label: 'Tipo', field: 'tipo', sortable: true },
  {
    name: 'version',
    label: 'Versión',
    align: 'left',
    field: 'version',
    sortable: true
  },
  {
    name: 'estado',

```



```
    label: 'Estado',
    align: 'left',
    field: 'estado',
    sortable: true
  },
  {
    name: 'fugasExtra',
    label: 'Fugas extra',
    align: 'right',
    field: 'fugasExtra',
    sortable: true
  },
  {
    name: 'incidencias',
    label: 'Incidencias',
    align: 'right',
    field: 'incidencias',
    sortable: true
  },
  {
    name: 'botellas',
    label: 'Est. botellas',
    align: 'right',
    field: 'botellas',
    sortable: true
  },
  {
    name: 'revisado',
    label: 'Revisado',
    align: 'center',
    field: 'revisado',
    sortable: true
  },
  {
    name: 'aprobado',
    label: 'Aprobado',
    align: 'center',
    field: 'aprobado',
    sortable: true
  },
  {
    name: 'observaciones',
    label: 'Observaciones',
    align: 'left',
    field: 'observaciones',
    sortable: true
  },
  {
    name: 'acciones',
```

```

        label: 'Acciones',
        align: 'left',
        field: ''
    }
]

```

Con la propiedad *pagination* se puede indicar cómo será la paginación de la tabla, en casa de que queramos. En este caso hemos definido que se muestren 5 filas por página.

```

pagination: {
    sortBy: 'desc',
    page: 1,
    descending: false,
    rowsPerPage: 5
},

```

En esta tabla también aparecen un cuadro de texto para filtrar por la columna que se quiera, y dos botones para exportar la tabla tanto en formato CSV como en formato PDF.

```

function printContent() {
    const table = document.getElementById('tabla-
documentos').getElementsByTagName('table')[0].outerHTML;
    // create a new div container
    const div = document.createElement('div');

    // assing your HTML to div's innerHTML
    div.innerHTML = table;
    const elements = div.getElementsByTagName('i');
    while (elements[0]) elements[0].parentNode.removeChild(elements[0]);

    const repl = div.innerHTML;
    console.log(repl);
    let style = '<style>';
    style += 'table {width: 100%;}';
    style += 'table, th, td {border: solid 1px #DDD; border-
collapse: collapse;';
    style += 'padding: 2px 3px;text-align: center;font-size:11px;}';
    style += '</style>';

    // CREATE A WINDOW OBJECT.
    const win = window.open('', '', 'height=700,width=700');

    win.document.write('<html><head>');
    win.document.write('<title>Documentos de control</title>'); // <title>
FOR PDF HEADER.
    win.document.write(style); // ADD STYLE INSIDE THE HEAD TAG.
    win.document.write('</head>');
    win.document.write('<body>');

```



```

win.document.write(repl); // THE TABLE CONTENTS INSIDE THE BODY TAG.
win.document.write('</body></html>');

win.document.close(); // CLOSE THE CURRENT WINDOW.

win.print(); // PRINT THE CONTENTS.
}

```

```

function wrapCsvValue(val, formatFn) {
  let formatted = formatFn !== undefined ? formatFn(val) : val;

  formatted = formatted === undefined || formatted === null ? '' : String(
    formatted);

  formatted = formatted.split('').join('');

  return `"${formatted}"`;
}

const exportTable = function _exportTable() {
  // naive encoding to csv format
  const content = [this.columns.map(col => wrapCsvValue(col.label))]
    .concat(
      this.documentsData.map(row =>
        this.columns
          .map(col =>
            wrapCsvValue(
              typeof col.field === 'function' ? col.field(row) : row[col.
                field === undefined ? col.name : col.field],
              col.format
            )
          )
        )
    )
    .join(',')
  )
  .join('\r\n');

  const status = exportFile('control-
documents.csv', content, 'text/csv');

  if (status !== true) {
    this.$q.notify({
      message: 'Browser denied file download...',
      color: 'negative',
      icon: 'warning'
    });
  }
};

```

