



GRADO EN INGENIERÍA INFORMÁTICA

TRABAJO DE FINAL DE GRADO

**Análisis y tratamiento de datos y diseño e
implementación de dashboards para el
seguimiento de líneas de producción**

Autor:
Carlos ALCALÁ YEPES

Supervisor:
Carlos CABANES HIDALGO
Tutor académico:
María José ARAMBURU CABO

Fecha de lectura: 13 de Julio de 2021
Curso académico 2020/2021

Resumen

En este documento se va a explicar el desarrollo del proyecto dentro de la empresa NSG Pilkington con la intención de realizar un trabajo de análisis, tratamiento y posterior visualización de los datos de tres líneas de producción en diferentes *dashboards*, primero en Excel para una versión preliminar y finalmente en Power BI. Se va a explicar la planificación inicial, la secuencia de tareas realizadas, los problemas que han ido surgiendo, su solución y el resultado final obtenido con sus limitaciones, finalizado con una presentación a la empresa. Además, se añade el desarrollo de diseño completo de base de datos, para las tres líneas, listo para ser implementado cuando se solucionen los problemas técnicos existentes.

Palabras clave

PLC (controlador lógico programable), KPI(indicador clave de rendimiento), Controles de proceso estático, *dashboards* y monitorización

Keywords

PLC(*Programmable Logic Controller*), KPI(*Key Performance Indicators*), *statically process control, dashboard and monitoring*

Índice general

1. Introducción	11
1.1. Contexto y motivación del proyecto	11
1.2. Objetivos del proyecto	12
1.3. Descripción del proyecto	13
1.4. Estructura de la memoria	13
2. Planificación del proyecto	15
2.1. Metodología	15
2.2. Planificación inicial	18
2.3. Seguimiento del proyecto	18
2.4. Estimación de recursos y costes del proyecto	20
3. Análisis y diseño del sistema	23
3.1. Análisis del sistema	23
3.1.1. Análisis de Línea_1	23
3.1.2. Análisis de Línea_4	24
3.1.3. Análisis de Soldadura	28
3.1.4. Descripción de requisitos de las bases de datos óptimas	32
3.2. Diseño de la arquitectura del sistema	33

3.2.1.	Diseño de Linea_4	33
3.2.2.	Diseño de Soldadura	35
3.2.3.	Diseño de las bases de datos óptimas	36
3.3.	Diseño de la interfaz	44
3.3.1.	<i>Dashboard</i> de Linea_4	44
3.3.2.	<i>Dashboard</i> de Soldadura	46
4.	Implementación y test de usuario	49
4.1.	Diseños de los <i>dashboards</i> e informes	49
4.1.1.	<i>Dashboards</i> de Linea_4	49
4.1.2.	Dasboards de Soldadura	53
4.2.	Detalles de implementación	59
4.2.1.	Linea_4	59
4.2.2.	Soldadura	61
4.3.	Limitaciones de la implementación	62
4.3.1.	Limitaciones de acceso a los datos	62
4.3.2.	Limitaciones en Excel	63
4.3.3.	Limitaciones en Power BI	63
4.3.4.	Limitaciones de calendario	64
4.4.	Test de usuario	64
4.5.	Resultados del proyecto	64
5.	Conclusiones	67
	Bibliografía	69
	Anexo A. Presentación	71

Índice de figuras

2.1. Diagrama de Gantt inicial	16
2.2. Diagrama de Gantt final	17
3.1. Diseño conceptual Linea_4	36
3.2. Diseño lógico Linea_4	36
3.3. Diseño conceptual de Soldadura	41
3.4. Diseño lógico de Soldadura	42
3.5. Boceto del <i>dashboard</i> para Linea_4	45
3.6. Boceto del <i>dashboard</i> para Soldadura	47
4.1. <i>Dashboard</i> de Linea_4 en Excel (Turno 1)	50
4.2. <i>Dashboard</i> de Linea_4 en Excel (Turno 2)	51
4.3. <i>Dashboard</i> de Linea_4 en Power BI	52
4.4. <i>Dashboard</i> de Linea_4 de un día completo en Power BI	53
4.5. <i>Dashboard</i> de Soldadura en Excel	54
4.6. <i>Dashboard</i> de selección de fecha y hora para Soldadura	55
4.7. <i>Dashboard</i> de temperatura para Soldadura	56
4.8. <i>Dashboard</i> de temperatura inicial para Soldadura	56
4.9. <i>Dashboard</i> de temperatura final para Soldadura	57
4.10. <i>Dashboard</i> de caudal para Soldadura	57

4.11. <i>Dashboard</i> resumen con detalle para Soldadura	58
4.12. <i>Dashboard</i> resumen sin detalle para Soldadura	58
A.1. Diapositiva 1 de la presentación	71
A.2. Diapositiva 2 de la presentación	72
A.3. Diapositiva 3 de la presentación	72
A.4. Diapositiva 4 de la presentación	73
A.5. Diapositiva 5 de la presentación	73

Índice de tablas

3.1. Tabla Areas	24
3.2. Tabla Models	25
3.3. Tabla Pieces	25
3.4. Tabla ScrapsAuto	26
3.5. Tabla ScrapsButtons	26
3.6. Tabla Areas	27
3.7. Tabla Models	28
3.8. Tabla Pieces	28
3.9. Tabla ScrapPiece	28
3.10. Tabla FloatTable	29
3.11. Tabla LineaFinalFloat	29
3.12. Tabla LineaFinalTag	29
3.13. Tabla TagTable	30
3.14. Tabla final Soldadura	31
3.15. Tabla <i>schema matching</i> de Linea_4	34
3.16. Tabla <i>schema matching</i> de Soldadura	35

Capítulo 1

Introducción

El proyecto que se desarrolla en este documento ha sido propuesto por NSG Pilkington, en su planta en España. Este documento es una memoria técnica del desarrollo del proyecto de tratamiento y análisis de datos para líneas de producción. El primer capítulo describe brevemente el proyecto y explica tanto su contexto como su objetivo y su motivación.

1.1. Contexto y motivación del proyecto

Pilkington Group Limited nació en 1826 y desde 2006 es parte de NSG GROUP (Nippon Sheet Glass), la cual se caracteriza por actuar en el mercado automotor, de arquitectura y de vidrios especiales, empleando a más de 26.000 personas en todo el mundo. Su sede en España está en Sagunto (Valencia) y se encarga de suministrar vidrio a los principales fabricantes de automóviles de este país como Ford, Seat, Mercedes, Peugeot, Citroën, Renault y Suzuki. Algunas de las operaciones que se realizan son laminación, templado y encapsulación de vidrio [1].

La motivación principal de este proyecto es comenzar a trabajar la gran cantidad de datos de los PLC(*Programmable Logic Controller*) de las líneas de producción para analizarlos y transmitir de forma desarrollada *feedback* a los empleados e informes a los supervisores de planta. Mediante una herramienta se proporcionarán informes de turno al supervisor, con los indicadores de rendimiento, paradas, tiempo de ciclo y OEE (*Overall Equipment Effectiveness*), además de almacenarlos, para futuras revisiones, compararlos entre ellos y analizar las variables de proceso en el caso que corresponda.

El proyecto surge de la necesidad de información creciente en la empresa, sobre todo en los últimos años, orientado a los datos de procesos y producción de las distintas líneas. Se busca solucionar el problema de la falta de conocimiento del histórico de los datos de producción y la necesidad de estar presencialmente en la línea para conocer el funcionamiento en cada momento, lo que ralentiza la corrección de errores y el trabajo de los jefes de línea.

1.2. Objetivos del proyecto

En este proyecto se va a trabajar con los datos enviados desde los diferentes PLC al servidor. Un PLC es un controlador lógico programable, que se usa en ingeniería automática por su resistencia. En este proyecto van a ser los que captan las señales de las líneas de producción y las envían al servidor donde son almacenadas. Estas señales son muy diversas, ya que captan tiempos, temperaturas, caudales, la existencia o no de pieza y muchas otras.

El principal objetivo de este proyecto es proporcionar ayuda a los operadores y supervisores de las líneas de producción para corregir errores y visualizar fallos o retrasos. Asimismo podemos diferenciar entre objetivos tecnológicos y objetivos empresariales del proyecto.

Objetivos tecnológicos:

- Conectar el servidor que contiene los datos del PLC con la intranet de la empresa.
- Estructurar los datos recibidos en el servidor.
- Conectar esos datos con los ordenadores mediante Excel o Power BI.
- Realizar gráficas dinámicas que proporcionen una visualización rápida y sencilla de los datos.
- Implementar herramientas para analizar esos datos.
- Generar y presentar en inglés un prototipo que anime al grupo a realizar una inversión mayor para extenderlo a toda la planta de una manera más sólida.

Objetivos empresariales:

- Mejorar la productividad en las líneas.
- Mejorar el *feedback* de los operarios de línea y supervisores respecto a cómo se está realizando el trabajo.
- Analizar potenciales mejoras de proceso tras el análisis de datos por parte de los ingenieros.
- Desarrollar una prueba de concepto para tratar de convencer al grupo de la inversión necesaria para extenderlo a toda la planta.

El alcance organizativo de este proyecto va destinado por un lado a los operadores y supervisores de líneas de producción, quienes se encargan de los informes e indicadores, y por otro lado a los ingenieros de procesos, quienes se ocupan de la explotación y análisis de datos. Además, cabe señalar que no se requieren conocimientos de informática previos para la parte de informes, supervisores y operadores. Sin embargo, la parte de análisis y explotación de datos sí que requiere un nivel medio del lenguaje SQL y de las herramientas Excel y Power BI.

Por lo que respecta al alcance informático, este proyecto necesita los datos proporcionados por los PLC de las líneas y conexión con los servidores donde se almacena esta información. Por

una parte tenemos los datos limitantes que marcan algún tipo de error en la pieza de producción; por otra parte, también encontramos los datos requeridos por la empresa para poder analizar mejor el funcionamiento de las líneas divididas en horas y turnos.

Finalmente, cabe destacar que en los objetivos del proyecto no se incluye ningún tipo de análisis de datos con el fin de anticipar errores. Aunque en un principio no se ha considerado, si que podría tenerse en cuenta como una posible mejora en un futuro. Tampoco se incluyen los datos de documentación escritos manualmente por los operadores de línea.

1.3. Descripción del proyecto

La situación inicial del proyecto es la de unos datos en muy mal estado de organización y formato. Uno de los problemas es la falta de una estructura en una base de datos relacional. Igualmente, se almacenan en unas tablas creadas con el software Transaction Manager que conectan los PLC con el servidor y le envían señales periódicas o cuando se produce algún evento. Por tanto, estos datos no tienen una alta fiabilidad, ya que algunas lecturas no aparecen sobre las salidas correctas del PLC. Además, no se ha comprobado la fiabilidad de las mediciones y los servidores solo son accesibles por cable dentro de la empresa.

A petición de la empresa, este proyecto se va a realizar con las herramientas de tratamiento y visualización de datos Excel, con el fin de realizar una primera versión, y posteriormente con Power BI para una versión definitiva. Los servidores en los que se encuentran los datos usan SQL Server por lo que el lenguaje de consulta será Transact-SQL, una implementación del estándar SQL que no lo implementa por completo y que tiene limitaciones. El programa usado para la conexión de los servidores con las bases de datos es Microsoft SQL Server Management Studio.

El proyecto se va a realizar para las líneas de producción 1 y 4 además de en una parte de la producción llamada “Robots de soldadura”, ya que son las líneas en las que la visualización de los datos de producción es más necesaria. Se espera conseguir una visualización de los datos de producción pasados y en tiempo real para poder así monitorizar la línea desde cualquier punto de la planta y observar cómo ha funcionado en los diferentes turnos, días, semanas y meses.

1.4. Estructura de la memoria

La estructura de la memoria se dividirá en cinco capítulos. En el siguiente capítulo se tratará la planificación del proyecto, la estimación de costes y recursos, y un seguimiento de cómo se han ido desarrollando las tareas previamente previstas. En el capítulo 3 se van a explicar las condiciones iniciales del proyecto, el tratamiento de esos datos y cómo se han utilizado para ser visualizados en los *dashboards* diseñados. En el capítulo 4 se profundizará en cómo se ha realizado esa transformación de datos y en la realización de los *dashboards*, así como en la verificación y validación con la empresa y con los usuarios finales. Para finalizar en el capítulo 5 se tratarán las conclusiones del proyecto en cuanto a su realización y lo aprendido a nivel personal para los siguientes proyectos, pero también las posibles extensiones del proyecto respecto a la empresa y su desarrollo futuro.

Capítulo 2

Planificación del proyecto

Este segundo capítulo se va a centrar en explicar la planificación inicial estimada según la metodología elegida, el desarrollo final que ha tenido el proyecto así como sus costes y recursos estimados inicialmente.

2.1. Metodología

Para el desarrollo de ese proyecto se ha seguido una metodología de trabajo basada en PMBOK, orientado al proceso, con una gestión de costes y riesgos clara y con unos objetivos fijos en un determinado tiempo. Se ha elegido esta metodología debido a que es una metodología invariable, con unos objetivos, costes y riesgos previamente definidos, por lo que permite dejar claro desde el inicio el fin del proyecto. Además se adapta mejor a la forma de trabajo de la empresa. Esto se escenifica tanto en el diagrama de Gantt inicial (figura 2.1) de la propuesta técnica del proyecto, como en el diagrama de Gantt final (figura 2.2) que escenifica el desarrollo real del proyecto.

La forma de trabajo de la empresa y la importancia que tenía el proyecto ha permitido una comunicación rápida, aunque la escasa formación base de partida y las restricciones de software y permisos del grupo han dificultado y ralentizado algunas de las tareas previamente previstas.

	Modo de	Nombre de tarea	Duración	Comienzo	Fin	Pre
1		PROYECTO PRÁCTICAS	49,58 días	lun 22/02/21	lun 07/06/21	
2		Puesta en marcha	2,67 días	lun 22/02/21	mar 23/02/21	
3		Integración en la empresa	2,67 días	lun 22/02/21	mar 23/02/21	
4		Entrevista usuarios clave	1,33 días	lun 22/02/21	mar 23/02/21	
5		Especificación de requisitos	2,67 días	lun 22/02/21	mar 23/02/21	
6		Definición de los objetivos y el alcance	2,67 días	lun 22/02/21	mar 23/02/21	
7		Definición de riesgos	2,67 días	lun 22/02/21	mar 23/02/21	
8		Planificación del proyecto	2,67 días	lun 01/03/21	mar 02/03/21	7
9		Gestión del alcance	2,67 días	lun 01/03/21	mar 02/03/21	7
10		Gestión de riesgos	2,67 días	lun 01/03/21	mar 02/03/21	7
11		Definición de riesgos	2,67 días	lun 01/03/21	mar 02/03/21	7
12		Propuesta Técnica	1,33 días	mié 03/03/21	lun 08/03/21	
13		Formación en el software específico	1,33 días	lun 08/03/21	mar 09/03/21	
14		Formación en SQL Server	1,33 días	lun 08/03/21	mar 09/03/21	12
15		Formación en la carga de datos de excel	1,33 días	lun 08/03/21	mar 09/03/21	12
16		Formación en la realización de graficas excel	1,33 días	lun 08/03/21	mar 09/03/21	12
17		Implementación Línea 4	23 días	mar 09/03/21	lun 03/05/21	
18		Formación en el proceso de producción	1 día	mar 09/03/21	vie 12/03/21	16
19		Conexión a la base de datos	1 día	vie 12/03/21	lun 15/03/21	18
20		Realización de consultas	1 día	lun 15/03/21	mar 16/03/21	19
21		Realización de gráficas en excel	1 día	mar 16/03/21	lun 22/03/21	20
22		Pruebas	6 días	lun 22/03/21	mar 30/03/21	
23		Comprobación de datos	1 día	lun 22/03/21	lun 22/03/21	21
24		Reuniones y feedback de los usuarios	1 día	lun 22/03/21	mar 23/03/21	23
25		Modificaciones	4 días	mar 23/03/21	mar 30/03/21	24
26		Power BI	13 días	mar 30/03/21	lun 03/05/21	
27		Formación	6 días	mar 30/03/21	lun 19/04/21	25
28		Realización de gráficas	3 días	lun 19/04/21	lun 26/04/21	27
29		Análisis de los datos	2 días	lun 26/04/21	mar 27/04/21	28
30		Pruebas	2 días	mar 27/04/21	lun 03/05/21	
31		Test de usuario	1 día	mar 27/04/21	vie 30/04/21	29
32		Modificaciones/Mejoras	1 día	vie 30/04/21	lun 03/05/21	31
33		Implementación Robots Soldadura de terminales, horno templado	15 días	lun 03/05/21	lun 31/05/21	
34		Formación en el proceso de producción	1 día	lun 03/05/21	mar 04/05/21	32
35		Conexión a la base de datos	1 día	mar 04/05/21	vie 07/05/21	34
36		Realización de consultas	2 días	vie 07/05/21	lun 10/05/21	35
37		Realización de gráficas en excel	2 días	lun 10/05/21	vie 14/05/21	36
38		Pruebas	4 días	vie 14/05/21	vie 21/05/21	
39		Comprobación de datos	1 día	vie 14/05/21	lun 17/05/21	37
40		Reuniones y feedback de los usuarios	1 día	lun 17/05/21	mar 18/05/21	39
41		Modificaciones	2 días	mar 18/05/21	vie 21/05/21	40
42		Power BI	5 días	vie 21/05/21	lun 31/05/21	
43		Realización de gráficas	2 días	vie 21/05/21	mar 25/05/21	41
44		Análisis de los datos	1 día	mar 25/05/21	vie 28/05/21	43
45		Pruebas	2 días	vie 28/05/21	lun 31/05/21	
46		Test de usuario	1 día	vie 28/05/21	lun 31/05/21	44
47		Modificaciones/Mejoras	1 día	lun 31/05/21	lun 31/05/21	46
48		Presentaciones a la empresa	1 día	lun 31/05/21	mar 01/06/21	47
49		Finalización del proyecto	2 días	mar 01/06/21	lun 07/06/21	
50		Documentación	2 días	mar 01/06/21	lun 07/06/21	48

Figura 2.1: Diagrama de Gantt inicial

Modo de tarea	Nombre de tarea	Duración	Comienzo	Fin	Predec
	PROYECTO PRÁCTICAS	42,5 días	lun 22/02/21	mar 18/05/21	
	Puesta en marcha	2 días	lun 22/02/21	mar 23/02/21	
	Integración en la empresa	1 día	lun 22/02/21	lun 22/02/21	
	Especificación de requisitos	1 día	lun 22/02/21	lun 22/02/21	
	Definición de objetivos y alcance	1 día	mar 23/02/21	mar 23/02/21	3;4
	Definición de riesgos	1 día	mar 23/02/21	mar 23/02/21	3;4
	Planificación del proyecto	1 día	vie 26/02/21	lun 01/03/21	
	Gestión del alcance	1 día	vie 26/02/21	lun 01/03/21	6
	Gestión de riesgos	1 día	vie 26/02/21	lun 01/03/21	6
	Definición de riesgos	1 día	vie 26/02/21	lun 01/03/21	6
	Propuesta Técnica	0 días	lun 01/03/21	lun 01/03/21	10
	Adaptación	4 días	lun 01/03/21	mar 09/03/21	
	Solicitud de software/permisos/licencias	1 día	lun 01/03/21	mar 02/03/21	11
	Formación	3 días	mar 02/03/21	mar 09/03/21	
	Lineas de producción	3 días	mar 02/03/21	mar 09/03/21	13
	Formación en SQL Server	3 días	mar 02/03/21	mar 09/03/21	13
	Formación en Excel	3 días	mar 02/03/21	mar 09/03/21	13
	Implementación	26,25 días	mar 09/03/21	lun 03/05/21	
	Linea 1	1 día	mar 09/03/21	vie 12/03/21	
	Análisis de datos	1 día	mar 09/03/21	vie 12/03/21	14
	Linea 4	13,25 días	vie 12/03/21	mar 13/04/21	
	Análisis de datos	1 día	vie 12/03/21	lun 15/03/21	20
	Transformación de datos	2 días	lun 15/03/21	lun 22/03/21	22
	Visualización de datos	3 días	lun 22/03/21	vie 26/03/21	23
	Formación Power BI	8 días	lun 22/03/21	mié 07/04/21	23
	Power BI	4 días	mar 06/04/21	mar 13/04/21	
	Soldadura	12 días	mar 13/04/21	lun 03/05/21	
	Análisis de datos	1 día	mar 13/04/21	mié 14/04/21	26
	Transformación de datos	1 día	jue 15/04/21	vie 16/04/21	28
	Visualización de datos	3 días	vie 16/04/21	mié 21/04/21	29
	Power BI	2 días	jue 29/04/21	lun 03/05/21	38
	Diseño bases de datos	5 días	mié 21/04/21	jue 29/04/21	
	Linea 1 y 4	3 días	mié 21/04/21	mar 27/04/21	
	Captación de requisitos	1 día	mié 21/04/21	jue 22/04/21	30
	Conceptual, lógico, físico	2 días	vie 23/04/21	mar 27/04/21	34
	Soldadura	5 días	mié 21/04/21	jue 29/04/21	
	Captación de requisitos	1 día	mié 21/04/21	jue 22/04/21	30
	Conceptual, lógico, físico	2 días	mar 27/04/21	jue 29/04/21	35
	Presentación	4,25 días	mar 04/05/21	lun 10/05/21	
	Preparación	3 días	mar 04/05/21	vie 07/05/21	31
	Presentación	1 día	lun 10/05/21	lun 10/05/21	40
	Pruebas y optimización	5,25 días	vie 07/05/21	lun 17/05/21	
	Pruebas	2,25 días	vie 07/05/21	mar 11/05/21	
	Medición de tiempos	1 día	vie 07/05/21	lun 10/05/21	40
	test de usuario	1 día	mar 11/05/21	mar 11/05/21	41
	Optimización	3 días	mié 12/05/21	lun 17/05/21	
	Optimización de consultas	2 días	mié 12/05/21	vie 14/05/21	45
	Mejoras de interfaz	1 día	vie 14/05/21	lun 17/05/21	47
	Finalización del proyecto	1 día	lun 17/05/21	mar 18/05/21	48

Figura 2.2: Diagrama de Gantt final

2.2. Planificación inicial

La planificación inicial del proyecto se basó en una primera toma de contacto, que permitió la adaptación y formación en la empresa, sus métodos de trabajo y el software específico para posteriormente diseñar una secuencia de tareas preprogramadas. Estas tareas tenían como fin alcanzar los objetivos de desarrollo del proyecto y proporcionar un beneficio a la empresa de manera continua, no solo al finalizar el proyecto.

Se planificó una primera toma de contacto progresiva, toda presencial, en la que se enseñó la planta, se definirían los requisitos y se comenzaría con la formación en el software además de conseguir el material necesario para trabajar, como eran el ordenador propio de la empresa, el usuario y contraseñas y el software. Cabe señalar que es imprescindible solicitar la instalación al grupo para la mayoría de programas, aunque sean gratuitos.

A continuación, se comenzaría con el desarrollo, con una primera parte de formación en los procesos de producción de las líneas con el fin de entender mejor su propósito, la información más relevante y las limitaciones que existían.

Con esta parte finalizada se desarrollaría la propuesta técnica, definiendo el alcance del proyecto y una gestión de riesgos, priorizando los que se veían más probables y unas posibles soluciones a estos. En la propuesta técnica se incluyó la planificación inicial del proyecto de la figura 2.1.

Para el desarrollo del proyecto se eligió Excel, para realizar las primeras versiones ya que es el programa más usado en la empresa, el que mejor manejaban y pese a que no está pensado para este desempeño, es lo que buscaba la empresa a modo de prototipo. La elección de Power BI fue para realizar la demostración al grupo de la mejora respecto a Excel con una herramienta de BI(*business intelligence*) orientada a la realización de *dashboards* que es lo que realmente necesitaban. Sin embargo, como ellos no conocían este software, no tenían las licencias actualizadas y por ello se realizó primero en Excel y finalmente en Power BI.

2.3. Seguimiento del proyecto

En este apartado se va a explicar el desarrollo final del proyecto con respecto a la planificación inicial, las desviaciones por los problemas ocurridos y las medidas llevadas a cabo para solucionarlas.

La primera parte de “Puesta en marcha” se desarrolló sin problemas. Las entrevistas clave no se realizaron como tarea ya que no fueron unas entrevistas formales sino una presentación al personal de la empresa.

La sección de tareas referentes a la planificación del proyecto se realizó más rápido de lo esperado porque el proyecto estaba previamente definido en muchos aspectos en la oferta pública de la bolsa de la UJI(Universitat Jaume I) y se comentaron la mayoría de los riesgos y posibles medidas de manera fluida y sin problemas. Además, la propuesta técnica se realizó en horas no

laborables por lo que se ahorró mucho tiempo.

En este punto se ha añadido una nueva sección de adaptación a la empresa con la formación para englobar la solicitud de software que no estaba prevista inicialmente, ya que en la mayoría de los casos había que solicitarlo a su servicio de informática, para que lo aprobaran y se conectaran de manera remota para la instalación. La formación en las líneas, SQL Server y Excel fue más lenta de lo previsto debido a la complejidad logística de conexión a los servidores y el desconocimiento que tenía sobre los KPI (*Key Performance Indicators*) y procesos de producción de la empresa. No obstante, con el tiempo ahorrado en la sección anterior cumplíamos el día estimado de finalización.

A continuación hubo un cambio en la planificación. La empresa pidió empezar por la línea 1 y no por la 4, puesto que los datos de esta última no estaban en un estado óptimo de análisis y se decidió parar. La implementación de la línea 4 empezó más tarde por ese motivo. Su desarrollo fue correcto hasta el día 22 de marzo en el cual se produjeron problemas internos de la empresa que dificultaron el trabajo. Se optó por realizar un cambio en el calendario y trabajar también de manera telemática, aumentando así las horas semanales.

El aprendizaje de Power BI fue autodidacta con los módulos proporcionados por Microsoft [2] porque en la empresa no se tenía conocimiento sobre esta herramienta y no se había cursado todavía la asignatura EI-1045, que contiene prácticas relacionadas con dicho software.

La sección de pruebas fue eliminada porque el *feedback* con el supervisor era constante y las reuniones con los demás usuarios se sustituyeron por una final con todo el proyecto acabado. La implementación de la línea 4 fue más rápida de lo esperado y finalizó el 9 de abril y no el 3 de mayo como estaba planificado inicialmente.

A continuación, comenzó la implementación de los robots de soldadura. La formación se realizó de manera conjunta previamente por lo que no se ha incluido como tarea en esta sección. La conexión a la base de datos y la tarea de consulta, debido al pobre tratamiento de los datos, se actualizaron a un análisis y transformación de datos. La visualización se hizo primero en Excel y más tarde en Power BI. Se finalizó la implementación el día 24 de abril debido a la eliminación de las tareas de pruebas tanto del Excel como del *dashboard* en Power BI y la carga de horas extra que se añadieron de manera telemática. Ésto facilitó el aprendizaje de Power BI y su desarrollo, así como también, permitió reducir los costes de transporte y posibles problemas en la empresa.

Debido al adelanto en el tiempo de las tareas y el mal estado de los datos almacenados se optó por realizar un diseño de la implementación ideal de las bases de datos, tanto de las líneas 1 y 4 por su similitud como de los robots de soldadura. Estas tareas se desarrollaron con un día presencial de entrevistas y captación de requisitos, además de los que ya se conocían al haber estado trabajando con las líneas tanto tiempo, lo que facilitó su comprensión y después un diseño conceptual, lógico y físico realizado mediante teletrabajo.

La tarea de Power BI perteneciente a los robots de soldadura se realizó después de este diseño ya que hubo un problema con los datos que impedían su realización y por eso se pospuso.

Una vez concluidas las tareas de implementación, se comenzó con la preparación de la presentación programada para el día 3 de mayo, pero a petición de la empresa se aplazó al día

10 de mayo por lo que se extendió su preparación con el fin de desarrollarla mejor. Se pasó también a la sección de tareas de pruebas para empezar a realizar mediciones de tiempos.

Esta presentación estaba prevista para el día 31 de mayo, como aparece en el diagrama de Gantt inicial, no obstante como se ha comentado previamente, el aumento de la carga de trabajo semanal y el retraso de la realización de pruebas hasta después de la presentación, ha desencadenado estos cambios en los días.

La presentación(Anexo A) se realizó el día 10 de mayo con un resultado muy positivo. El personal y los jefes de planta quedaron muy satisfechos con el trabajo realizado, mostraron interés por las posibilidades que tenía y la aplicación útil en la empresa. No se decidió en el momento si se seguiría adelante con el proyecto, ya que se requiere de aprobación por parte del grupo, pero los usuarios principales del proyecto quedaron satisfechos con el trabajo y con la exposición sobre el abanico de posibilidades que aún tenía por desarrollar.

En los días posteriores se realizó un trabajo de formalización en el que se entregaron los ficheros Excel, Power BI y diseños de base de datos para la empresa, así como algunas modificaciones que surgieron del *feedback* de los usuarios. Con esto finalizó el proyecto.

2.4. Estimación de recursos y costes del proyecto

La estimación inicial de los recursos del proyecto se realizó en las primeras semanas de planificación.

Por un lado, tenemos los recursos humanos, centrados en el alumno en prácticas, Carlos Alcalá y su supervisor Carlos Cabanes que actuaría de soporte y guía sobre el funcionamiento de la empresa y posibles problemas. Por tanto, parte de sus horas de trabajo se dedicarían a este proyecto y sería en coste de recursos humanos de la empresa.

Por lo que respecta a recursos tecnológicos tenemos los recursos hardware que se basarían en un equipo portátil individual para Carlos Alcalá que se entregará al finalizar y material de seguridad de la planta que incluiría: botas de seguridad, gafas de protección y chaleco reflectante. Estos son necesarios por norma interna de seguridad de la empresa con el fin de evitar accidentes. Se estima que este material de seguridad le supondrá a la empresa un total de 30€ de las botas de seguridad más 5€ en gafas y chaleco, mientras que para el ordenador personal se estima el coste de alquiler en 50€.

En referencia a la parte software, el proyecto necesitaba principalmente licencias de Microsoft Excel, Microsoft SQL Server Managment Studio y Microsoft Power BI. Se estima unos 10€ al mes en la licencia de Microsoft 365 [3], la licencia de Microsoft SQL Server está pagada previamente por lo que no se tiene en cuenta para este proyecto. También habría que incluir un editor de código de software libre, Notepad++, 0€. Por lo que el coste de todo el software se estima en 10€ durante 4 meses, en total 40€.

El coste de hardware de este proyecto es 0€ ya que los servidores son propiedad de la empresa donde están los datos a tratar y explotar y su conexión vía cable o VPN proporcionada por la

empresa.

Además, la empresa proporciona de manera gratuita su comedor durante la estancia, por lo que si se estima que se realizarán 35 días presenciales a 6'50€ el menú, que es lo que pagan el resto de los trabajadores, 227'5€. En resumen, el coste económico para la empresa se estima en un total de 352'5€.

Por otro lado, los costes indirectos se estimó el coste de transportes a la empresa a 52Km del domicilio del alumno, con el trabajo semipresencial, 35 días presenciales (3.640km) teniendo en cuenta que en España el kilometraje en 2021 esta de media en los 0'20€/Km suma un total de 728€ que correrán a cargo del alumno, Carlos Alcalá.

Capítulo 3

Análisis y diseño del sistema

3.1. Análisis del sistema

El análisis de este proyecto es algo distinto a lo común en un proyecto de desarrollo de software. Se va a diferenciar en dos partes por un lado, la extracción, transformación y visualización de los datos que se están almacenando actualmente en los servidores, y por otro lado, la realización de un diseño de base de datos óptimo para las líneas. Éste último se implantará en un futuro cuando se solucionen los problemas mecánicos a nivel de PLC.

En primer lugar, se explicará el análisis de la extracción, transformación y visualización de los datos. Para ello lo primero es analizar el estado de la base de los datos origen y el estado final que se quiere conseguir. Las bases de datos a analizar se llaman "Linea.1", "Linea.4" y "Soldadura" y se hará referencia a ellas con esos nombres para hablar de las bases de datos o a las líneas de producción en concreto.

3.1.1. Análisis de Linea_1

El análisis de la Linea 1 duró únicamente un día ya que se realizó un pequeño análisis preliminar del estado de sus datos y se observó que los datos se almacenaban periódicamente por tiempo y no por evento de pieza. Se demandó el cambio del PLC para que estos se recogieran de la forma deseada pero, después de esperar el trabajo de la técnica de la línea, se me comunicó que no era posible o que no se iba a poder realizar por el momento.

Después de una reunión con mi supervisor se tomó la decisión de o bien posponer este análisis hasta más adelante, si se conseguían implementar los cambios, o bien descartar la realización del trabajo de transformación y de visualización de esta línea al estar los datos en demasiado mal estado y no ser útiles para la empresa. Al concluir la estancia en la empresa no se habían implementado los cambios, ya sea por falta de tiempo o de conocimientos, y por ese motivo no se ha continuado con este análisis ni posterior diseño e implementación.

Finalmente esta línea sí se tuvo en cuenta en el diseño de base de datos objetivo ya que

los requisitos pactados con el supervisor, en rol de cliente, coincidían con los de la Linea_4, la siguiente en analizar, por lo que el diseño sería válido para ambas por igual.

3.1.2. Análisis de Linea_4

En el esquema inicial de la base de datos Linea_4 tenemos seis tablas: “Areas”, ”Models”, “Pieces”, “PLCEvents”, “ScrapsAuto” y “ScrapsButtons”.

La columna descripción que se encuentra en las tablas siguientes no está en la base de datos. Se ha añadido a modo de explicación después de las reuniones con los técnicos de la línea y las pruebas realizadas en ellas para corroborar los datos.

Tabla 3.1: Tabla Areas

Nombre	Tipo	Descripción
Id	int	Identificador
Description	varchar	Nombre del área: Chemicals, Terolan, Finisher, Soldering y FinalLine.
Cod	int	Código idéntico al Id
Plc	varchar	Desconocido, siempre a NULL

De la tabla Areas (tabla 3.1) solo nos podemos quedar con el nombre del área y su Id.

Tabla 3.2: Tabla Models

Nombre	Tipo	Descripción
Id	int	Identificador, no consecutivo
Description	varchar	Nombre del modelo de vehículo
Customer	varchar	Desconocido, siempre a NULL
CarModel	varchar	Desconocido, siempre a NULL
ProcessTimeMin	int	Tiempo de procesado mínimo
SapCode	varchar	Desconocido, siempre a NULL
Cod	int	Réplica del campo Id
ProcessTimeMax	int	Tiempo de procesado máximo
ProcessTime	int	Tiempo de procesado medio
TransportationTime	int	Tiempo de transporte entre las áreas
BreaksType	int	Tipo de descanso (0/1)

De la tabla Models (tabla 3.2) solo nos interesa el Id y la descripción de cada modelo. Los demás datos son referencias de tiempos de proceso que no han sido probados.

Tabla 3.3: Tabla Pieces

Nombre	Tipo	Descripción
Id	int	Identificador
AreaId	int	Clave ajena a “Area”
ModelId	int	Clave ajena a “Models”
EnterDate	datetime	Fecha de entrada de la pieza en el área correspondiente
ProcessDate	datetime	Fecha de inicio de procesado de la pieza en el área correspondiente
ExitDate	datetime	Fecha de finalización de procesado de la pieza en el área correspondiente

La tabla Pieces (tabla 3.3) es la tabla más importante. De ella vamos a extraer el Id, área, modelo, tiempos de entrada, de proceso y de salida de cada pieza para clasificarla y aplicar fórmulas sobre sus tres tiempos para poder tratarlos, analizarlos y visualizarlos.

De la tabla PlcEvents se desconocen la mayoría de los atributos y como no se usará en el objetivo final se ha omitido su explicación.

Tabla 3.4: Tabla ScrapsAuto

Nombre	Tipo	Descripción
Id	int	Identificador
Date	datetime	Fecha de la pieza detectada de manera automática
ModelId	int	Modelo de la pieza
A_00	int	Número básico
A_01	int	Número básico
...
A_31	int	Número básico

En la tabla ScrapsAuto (tabla 3.4) se almacenan las piezas malas que ha detectado el robot pero también las que se detectan por botonera con la diferencia de que las de botonera no tienen cambios entre A_00 y A_31 de una fila a la otra. Los Id no tiene relación con la tabla piezas por lo que no se corresponde este Id con el Id de la tabla “Pieces”. El ModelId si tiene relación con la tabla “Models”, pero los siguientes 32 campos son códigos de errores que se almacenan de la siguiente forma. De una fila a otra, fila Id a Id+1, todos los campos A_XX tienen el mismo valor numérico que su predecesor, excepto uno que se habrá incrementado, eso quiere decir que ese campo es el que corresponde a su código de error. Por ejemplo:

Id	Date	ModelId	A_00	A_01	A_02	...	A_30	A_31
28	2021-03-23 14:05:06.147	638	1024	1708	1228	...	1031	1032
29	2021-03-23 14:07:36.180	638	1024	1708	1229	...	1031	1032

Estas dos filas representan que la pieza errónea 29 ha sido por el error A_02. Si quisiéramos ver el error de la pieza 28 tendríamos que compararla con la 27.

Tabla 3.5: Tabla ScrapsButtons

Nombre	Tipo	Descripción
Id	int	Identificador
Date	datetime	Fecha de la pieza errónea de manera manual
ModelId	int	Modelo de la pieza
B_00	int	Número básico
B_01	int	Número básico
...
B_31	int	Número básico

En la tabla ScrapsButtons (tabla 3.5) se almacenan las piezas malas que ha detectado el operador pero también las que se detectan por el robot. Sin embargo, la diferencia es que las del robot no tienen cambios entre B_00 y B_31 de una fila a la otra. Los Id no tienen relación con la tabla “Pieces” por lo que no se corresponde este Id con el Id de la tabla “Pieces”.En

cambio, el ModelId si tiene relación con la tabla “Models”, pero los siguientes 32 campos son códigos de errores que se almacenan de la siguiente forma. De una fila a otra, fila ID a ID+1, todos los campos B.XX tienen el mismo valor numérico que su predecesor, excepto uno que se habrá incrementado, eso quiere decir que ese campo es el que corresponde a su código de error. Por ejemplo:

Id	Date	ModelId	B_00	B_01	B_02	...	B_30	B_31
28	2021-03-23 14:05:06.147	638	1024	1708	1228	...	1031	1032
29	2021-03-23 14:07:36.180	638	1024	1708	1229	...	1031	1032

Estas dos filas representan que la pieza errónea 29 ha sido por el error B.02. Si quisiéramos ver el error de la pieza 28 tendríamos que compararla con la 27.

Con esto finaliza la explicación del estado base de la base de datos Linea.4. El esquema objetivo para analizar los datos se realizó en base a la descripción de requisitos demandada por la empresa para mostrar en los *dashboard* y finalizó con la realización de unos prototipos a papel para asegurar tener la misma visión.

Para comenzar, en las reuniones se explicó toda la información de la que se disponía en la base de datos y se comenzó a descartar la que la empresa no consideraba útil o necesaria. De la tabla “Areas” se desechó los atributos “cod” y “PLC” ya que no aportaban información. De la tabla “Models” se eliminó “Customer”, “CarModel”, “ProcessTimeMin”, “SapCode”, “Cod”, “ProcessTimeMax”, “ProcessTime”, “TransportationTime” y “BreaksType” puesto que esta información no se quería visualizar ni se necesitaba para realizar las transformaciones. De la tabla Pieces, que es la más relevante, se decidió extraer todos los atributos ya que no se quieren visualizar directamente, en cambio sí eran necesarios para los cálculos de los KPI que se deben mostrar. Por último de las tablas “ScrapAuto” y “ScrapButton” se propuso juntarlas y no cargar todos los atributos A.00, A.01, etc. Por lo que se creará una tabla nueva respetando el Id y fecha (Date) pero cargando solo el código de error y de qué tabla proviene.

Durante estas reuniones no se tuvo en cuenta las características de las piezas erróneas que son recicladas por desconocimiento de los códigos de error que le correspondían y el funcionamiento real en la línea. Después de realizar gran parte del diseño e implementación se decidió añadir la característica de las piezas erróneas, pero realizando previamente pruebas en la línea y preguntando a los encargados de ésta. Ésto se debe a que por ejemplo, la pieza sale al detectar el error, se repara y se vuelve a introducir pero no se respeta ningún orden ni el número de la pieza por lo que aparecerá duplicada en la tabla “Pieces”.

Como resultado final del análisis, éste es el esquema final con el que se trabajó para diseñar e implementar los *dashboard*, tanto en Excel como en Power BI.

Tabla 3.6: Tabla Areas

Nombre	Tipo	Descripción
Id	int	Identificador
Name	varchar	Nombre del área

Tabla 3.7: Tabla Models

Nombre	Tipo	Descripción
Id	int	Identificador
Name	varchar	Nombre del modelo de vehículo

Tabla 3.8: Tabla Pieces

Nombre	Tipo	Descripción
Id	int	Identificador
AreaId	int	Clave ajena a “Area”
ModelId	int	Clave ajena a “Models”
EnterDate	datetime	Fecha de entrada de la pieza en el área correspondiente
ProcessDate	datetime	Fecha de inicio de procesado de la pieza en el área correspondiente
ExitDate	datetime	Fecha de finalización de procesado de la pieza en el área correspondiente

Tabla 3.9: Tabla ScrapPiece

Nombre	Tipo	Descripción
Id	int	Identificador
Date	datetime	Fecha de la pieza
Error	varchar	Código del error
Auto/Button	boolean	0 en caso de detección automática 1 en caso de detección manual
Recycle	boolean	0 en caso de que no sea reciclable 1 en caso de que lo sea

3.1.3. Análisis de Soldadura

En el esquema inicial de la base de datos “Soldadura” tenemos cinco tablas: “FloatTable”, “LineaFinalFloat”, “LineaFinalTag”, “StringTable” y “TagTable”.

La columna descripción no está en la base de datos, se ha añadido a modo de explicación después de las reuniones con los técnicos de la línea y las pruebas realizadas en ellas para analizar los datos. Esta base de datos estaba en un peor estado que Linea_4. Las mediciones se realizan cada X segundos y no cada X pieza por lo que solo se pueden analizar los KPI por tiempo y no por pieza, que no era lo deseado. Además el almacenamiento de los valores es algo complejo e ineficiente. A continuación, se describen las tablas de la base de datos.

Tabla 3.10: Tabla FloatTable

Nombre	Tipo	Descripción
DateAndTime	datetime	Fecha y hora de la medición.
Millitm	int	Milisegundos
TagIndex	int	Referencia a la tabla “LineaFinaTag” pero no como clave ajena
Val	int	Valor de la fila
Status	char	Desconocido
Marker	char	Desconocido

En la tabla FloatTable (tabla 3.10) se almacenan todos los datos cada 10 segundos, se almacenan 10 filas con el mismo “DateAndTime” ya que cada 10 segundos se almacena 1 fila por cada TagIndex distinto.

Tabla 3.11: Tabla LineaFinalFloat

Nombre	Tipo	Descripción
DateAndTime	datetime	Fecha y hora de la medición.
Millitm	int	Milisegundos
TagIndex	int	Referencia a la tabla “LineaFinaTag” pero no como clave ajena
Val	int	Valor de la fila
Status	char	Desconocido
Marker	char	Desconocido

En la tabla LineaFinalFloat (tabla 3.11) se almacena la misma información que en “Float-Table” pero solo de la última parte de la línea, que es la que se va a procesar y analizar, donde están los dos robots de soldadura trabajando.

Tabla 3.12: Tabla LineaFinalTag

Nombre	Tipo	Descripción
TagName	varchar	Nombre del tag ej: [T01_Salida_Horno]PvMaq15[26]
TagIndex	int	Numeración del 0 al 9, equivalente a un ID
TaxType	int	Desconocido, todos tienen valor “2”
TagDataType	int	Desconocido, todos tienen valor “0”

En la tabla LineaFinalTag (tabla 3.12) es donde se encuentran los nombres de las mediciones que tienen su valor en “LineaFinalFloat”, pero los nombres reales se extrajeron de reuniones con los encargados.

Tabla 3.13: Tabla TagTable

Nombre	Tipo	Descripción
TagName	varchar	Nombre del tag, no explicativo, similar al de "LineaFinalTag"
TagIndex	int	Numeración del 0 al 9, equivalente a un ID
TaxType	int	Desconocido, todos tienen valor "2"
TagDataType	int	Desconocido, todos tienen valor "0"

La tabla TagTable (tabla 3.13) no se usará por el desconocimiento de la fiabilidad de sus datos porque desde los técnicos se comunicó que no se tenía constancia de a que hacían referencia esos datos ni si habían sido probados.

Con esto finaliza la explicación del estado de la base de datos Soldadura. El esquema objetivo para analizar los datos se realizó en base a la descripción de requisitos demandada por la empresa para mostrar en los *dashboard* y finalizó con la realización de unos prototipos a papel para asegurar tener la misma visión.

Para comenzar, en las reuniones se explicó toda la información de la que se disponía en la base de datos y se comenzó a descartar la que la empresa no consideraba útil o necesaria. Lo primero que se planteó fue mantener cada tabla por separado y cargar solo la información necesaria, pero dado el estado de los datos y que la información era menor que en la Linea_4, se decidió hacer solo una tabla en la que se almacenara toda la información. De la tabla "FloatTable" se descartaron los campos "Status" y "Marker", ya que no se conocía a qué hacían referencia, junto con "Millitm", puesto que el campo "DateAndTime" ya contenía los milisegundos. Más adelante en las reuniones se determinó usar la tabla "LineaFinalFloat" y no "FloatTable" porque es donde se iban a encontrar los datos más recientes y del final de la línea, que era el objetivo a graficar. La tabla "LineaFinalTag" sirvió de ayuda para solicitar de una manera más técnica los nombres reales, ya que los codificados no se podían entender, pero el jefe de la línea sí sabía a qué hacían referencia. Por último, la tabla "StringTable" y "TagTable" no se utilizaron y se resolvió desecharlas.

Como resultado del análisis de datos, este es el esquema final de información con el que se trabajó para diseñar e implementar los *dashboards*, tanto en Excel como en Power BI.

Tabla 3.14: Tabla final Soldadura

Nombre	Tipo	Descripción
DateAndTime	datetime	Fecha y hora de la medición
T ^a Soldador Derecho	double	Temperatura básica del robot derecho
T ^a Soldador Izquierdo	double	Temperatura básica del robot izquierdo
Caudal Soldador Derecho	int	Caudal de aire que expulsa el robot derecho
Caudal Soldador Izquierdo	int	Caudal de aire que expulsa el robot izquierdo
T ^a ini Soldadura Derecha	double	Temperatura inicial del robot derecho en una soldadura
T ^a ini Soldadura Izquierda	double	Temperatura inicial del robot izquierdo en una soldadura
T ^a fin Soldadura Derecha	double	Temperatura final del robot derecho en una soldadura
T ^a fin Soldadura Izquierda	double	Temperatura final del robot izquierdo en una soldadura
Tiempo ciclo soldadura derecha	int	Tiempo de ciclo del robot derecho
Tiempo ciclo soldadura izquierda	int	Tiempo de ciclo del robot izquierdo

3.1.4. Descripción de requisitos de las bases de datos óptimas

En este apartado se van a especificar los requisitos ideales que se han usado para realizar el diseño conceptual, lógico y físico de las bases de datos correspondientes a Línea_4, Línea_1 y Soldadura. Estos requisitos han sido redactados después de varias reuniones con los encargados de las diferentes líneas y su posterior consenso. Fueron posteriores a los requisitos previamente mencionados ya que esos requisitos partían de la base actual de los datos y la imposibilidad de modificarlos y mejorarlos. Los que ahora se presentan son unos requisitos redactados de manera idónea, para ser atendidos cuando se solventen los problemas mecánicos a nivel de PLC y algunas mejoras técnicas.

Descripción de requisitos de Línea_4

Se va a diseñar un sistema de información para una de las líneas de producción de la planta de NSG Pilkington en Sagunto, con el objetivo de cubrir las necesidades de almacenamiento de información de los PLC y sensores de la línea 4 y de la línea 1. Los requisitos que debe cumplir el sistema de información se definen a continuación.

El sistema debe almacenar la información de cada pieza que se introduzca en la línea junto con su identificador. Cada pieza pertenece a un modelo del cual queremos conocer su identificador, nombre y descripción.

Las líneas se componen de un número de áreas por las que viaja la pieza y en las que se le realizan diferentes operaciones en cada una de ellas. Solo se pretende almacenar la información referente al número de área, nombre y descripción de cada área. Cada pieza viaja por todas las áreas y se quiere almacenar la información de la fecha de entrada en el área, fecha en la que se empezó a procesar y fecha en la que la pieza salió de esa área.

La pieza viaja por todas las áreas, pero puede ser una pieza mala. La pieza puede ser marcada como mala por defecto de robot (Auto) o que un usuario lo marque manualmente (Button). Queremos también el identificador del error con su nombre. La pieza puede tener varios errores y marcarse de manera automática o manual, pero solo queremos la información del primer error. También queremos la fecha en la que la pieza ha sido mala, dependiendo del código de error que tenga, podrá ser reciclada o no.

Del mismo modo, se necesita saber cuándo ha parado la línea y el porqué. Puede ser por un área en concreto o una parada tecnológica porque hay que cambiar el modelo. De cada parada queremos almacenar, qué área ha sido la culpable o, en caso de tecnológica, todas las áreas, la fecha de inicio de parada y la fecha de reanudación.

Las líneas 1 y 4 se diferencian en las operaciones que realizan sobre las piezas, el nombre de las áreas y en cómo serán los datos registrados de tiempos. Por lo tanto, estos requisitos y posterior diseño también son válidos para la Línea_1.

Descripción de requisitos de Soldadura

Se va a diseñar un sistema de información para una de las líneas de producción de la planta de NSG Pilkington en Sagunto, para cubrir las necesidades de almacenamiento de información de los PLC y sensores de dos robots de soldadura. Los requisitos que debe cumplir el sistema de información se definen a continuación.

El sistema debe poder almacenar la información de cada pieza, con su modelo, que pase por la estación para que le suelden los terminales. Del proceso de soldadura se quiere almacenar dos tipos de información, la ideal *SetPoint* que es la que se introduce al robot dependiendo del modelo, y la información real que es la leída por el PLC. Se desea almacenar la información del *SetPoint* porque se quiere saber, para el histórico, si se modificó ese valor o no.

Los valores que se desean guardar de cada pieza son la temperatura de soldadura, que es la temperatura a la que se ha soldado la pieza, la temperatura de inicio de soldadura, cuándo se empieza a soldar, la temperatura final de soldadura, cuándo se acaba de soldar la pieza, el caudal de mantenimiento, caudal de aire caliente, caudal de aire frío, tiempo de soldadura y tiempo de enfriamiento. Estos valores se guardarán por cada robot izquierdo y derecho con sus respectivos *SetPoint*(valor ideal). Cada variable estará dos veces por cada robot, ideal y real. Asimismo, la pieza tendrá un tiempo de ciclo propio.

3.2. Diseño de la arquitectura del sistema

El diseño del sistema se va a estructurar de la misma forma que el análisis. Primero se tratará el diseño realizado para la transformación de datos de la base de datos Línea_4. Luego el diseño realizado para la transformación de datos de la base de datos Línea_1 y, finalmente, el diseño de las dos bases de datos óptimas.

Primeramente, se ha de mencionar el hecho de que se ha realizado el diseño de esta forma debido a que no se podían implementar los diseños finales por problemas mecánicos y tecnológicos. Estos diseños son un prototipo para poder extraer los datos de las bases de datos y visualizarlos con el fin de que sean analizados por los operadores y jefes de línea, pero solo a modo de prototipo. Cuando se solucionen los problemas tecnológicos, se deberá hacer una integración completa y remodelación de las bases de datos actuales según el diseño final propuesto.

3.2.1. Diseño de Línea_4

Para realizar el diseño de esta base de datos y su transformación de datos se ha realizado una tarea de *schema matching* (tabla 3.15) para determinar las correspondencias entre los elementos de los esquemas actuales y los objetivo que se han definido con los requisitos en la sección de análisis.

Tabla 3.15: Tabla *schema matching* de Linea_4

Esquema Objetivo	Esquema Inicial
Areas.Id	Areas.Id
Areas.Name	Areas.Description
Models.Id	Models.Id
Models.Name	Models.Description
Pieces.Id	Pieces.Id
Pieces.AreaId	Pieces.AreaId
Pieces.ModelId	Pieces.ModelId
Pieces.EnterDate	Pieces.EnterDate
Pieces.ProcessDate	Pieces.ProcessDate
Pieces.ExitDate	Pieces.ExitDate
ScrapPiece.Id	ScrapsAuto.Id AND ScrapsButtons.Id
ScrapPiece.Date	ScrapsAuto.Date AND ScrapsButtons.ID
ScrapPiece.Error	Resta de cada código de error con su fila anterior, si es 1, ese es su código de error.
ScrapPiece.Auto/Button	IF (SUM(ScrapsAuto.filaX)-SUM(ScrapsAuto.filaX+1)) = 1 THEN 0 ELSE 1
ScrapPiece.Recycle	IF ScrapPiece.Error IN(A_02, A_04,A_07, A_08) THEN 1 ELSE 0

Los códigos de error que corresponden a que la pieza ha sido reciclada se extrajeron de reuniones con los encargados y de pruebas en la propia línea.

El esquema Objetivo no se puede implementar en una base de datos ya que no se han tenido permisos para crearla en el servidor ni en la nube. Se han cargado los datos directamente en Excel y posteriormente en Power BI para ser tratados y más tarde visualizados.

3.2.2. Diseño de Soldadura

Para realizar el diseño de esta base de datos y su transformación de datos se ha realizado una tarea de *schema matching* (tabla 3.16) con la finalidad de determinar las correspondencias entre los elementos de los esquemas actuales y los objetivo que se han definido con los requisitos en la sección de análisis.

Tabla 3.16: Tabla *schema matching* de Soldadura

Esquema Objetivo	Esquema Inicial
DateAndTime	LineaFinalFloat.DateAndTime
TªSoldador Derecho	(LineaFinalFloat.Val/10) Where LineaFinalFloat.TagIndex=0
TªSoldador Izquierdo	(LineaFinalFloat.Val/10) Where LineaFinalFloat.TagIndex=5
Caudal Soldador Derecho	(LineaFinalFloat.Val/10) Where LineaFinalFloat.TagIndex=1
Caudal Soldador Izquierdo	(LineaFinalFloat.Val/10) Where LineaFinalFloat.TagIndex=6
Tª ini Soldadura Derecha	(LineaFinalFloat.Val/10) Where LineaFinalFloat.TagIndex=2
Tª ini Soldadora Izquierda	(LineaFinalFloat.Val/10) Where LineaFinalFloat.TagIndex=7
Tª fin Soldadura Derecha	(LineaFinalFloat.Val/10) Where LineaFinalFloat.TagIndex=3
Tª fin Soldadora Izquierda	(LineaFinalFloat.Val/10) Where LineaFinalFloat.TagIndex=8
Tiempo ciclo soldadura derecha	(LineaFinalFloat.Val/10) Where LineaFinalFloat.TagIndex=4
Tiempo ciclo soldadura izquierda	(LineaFinalFloat.Val/10) Where LineaFinalFloat.TagIndex=9

La correspondencia entre los “TagIndex” y el nombre en el esquema objetivo se extrajeron de reuniones con los encargados y pruebas en la propia línea.

El esquema objetivo no se puede implementar en una base de datos ya que no se han tenido permisos para crearla en el servidor ni en la nube. Se han cargado los datos directamente en Excel y posteriormente en Power BI para ser tratados y posteriormente visualizados.

3.2.3. Diseño de las bases de datos óptimas

Linea_4

Siguiendo la especificación de requisitos mencionada en el análisis se ha realizado el siguiente diseño conceptual (figura 3.1):

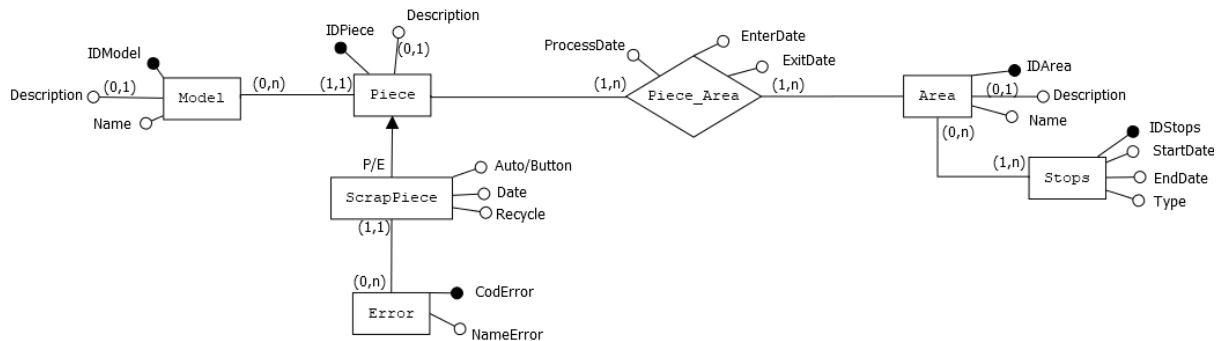


Figura 3.1: Diseño conceptual Linea_4

Utilizando este diseño conceptual se ha realizado el siguiente diseño lógico (figura 3.2):

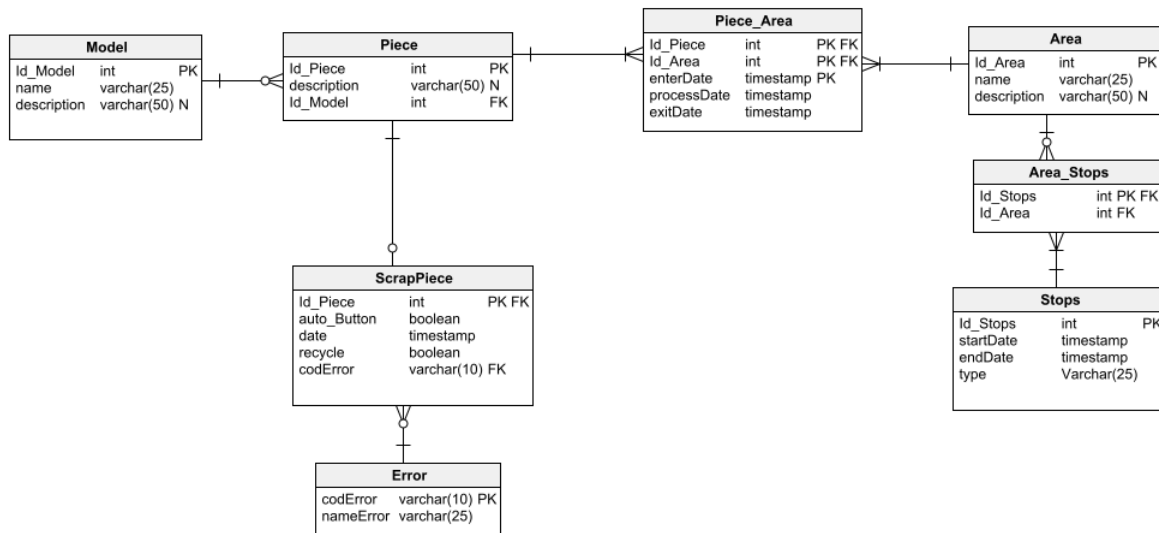


Figura 3.2: Diseño lógico Linea_4

También se ha realizado el diseño físico según el lenguaje PostgreSQL ya que al no estar decidido el lenguaje en el que se va a implementar, se ha elegido este por ser a coste cero y soportar gran parte del estándar SQL [4].

Las sentencias de creación de la base de datos son las siguientes:

```
CREATE TABLE Model (  
  Id_Model int NOT NULL,  
  name varchar(25) NOT NULL,  
  description varchar(50) NULL,  
  CONSTRAINT Model_pk PRIMARY KEY (Id_Model)  
);
```

```
CREATE TABLE Piece (  
  Id_Piece int NOT NULL,  
  description varchar(50) NULL,  
  Id_Model int NOT NULL,  
  CONSTRAINT Piece_pk PRIMARY KEY (Id_Piece)  
);
```

```
CREATE TABLE ScrapPiece (  
  Id_Piece int NOT NULL,  
  auto_Button boolean NOT NULL,  
  date timestamp NOT NULL,  
  recycle boolean NOT NULL,  
  codError varchar(10) NOT NULL,  
  CONSTRAINT ScrapPiece_pk PRIMARY KEY (Id_Piece)  
);
```

```
CREATE TABLE Error (  
  codError varchar(10) NOT NULL,  
  nameError varchar(25) NOT NULL,  
  CONSTRAINT Error_pk PRIMARY KEY (codError)  
);
```

```
CREATE TABLE Area (  
  Id_Area int NOT NULL,  
  name varchar(25) NOT NULL,  
  description varchar(50) NULL,  
  CONSTRAINT Area_pk PRIMARY KEY (Id_Area)  
);
```

```
CREATE TABLE Piece_Area (  
  Id_Piece int NOT NULL,  
  Id_Area int NOT NULL,  
  enterDate timestamp NOT NULL,  
  processDate timestamp NOT NULL,  
  exitDate timestamp NOT NULL,  
  CONSTRAINT Piece_Area_pk PRIMARY KEY (Id_Piece,Id_Area,enterDate)  
);
```

```
CREATE TABLE Area_Stops (  
  Id_Stops int NOT NULL,  
  Id_Area int NOT NULL,  
  CONSTRAINT Area_Stops_pk PRIMARY KEY (Id_Stops)
```

);

```
CREATE TABLE Stops (  
  Id_Stops int NOT NULL,  
  startDate timestamp NOT NULL,  
  endDate timestamp NOT NULL,  
  type Varchar(25) NOT NULL,  
  CONSTRAINT Stops_pk PRIMARY KEY (Id_Stops)  
);
```

```
ALTER TABLE Piece ADD CONSTRAINT Piece_Model  
FOREIGN KEY (Id_Model)  
REFERENCES Model (Id_Model)  
ON DELETE RESTRICT  
ON UPDATE CASCADE  
NOT DEFERRABLE  
INITIALLY IMMEDIATE  
;
```

```
ALTER TABLE ScrapPiece ADD CONSTRAINT ScrapPiece_Piece  
FOREIGN KEY (Id_Piece)  
REFERENCES Piece (Id_Piece)  
ON DELETE RESTRICT  
ON UPDATE CASCADE  
NOT DEFERRABLE  
INITIALLY IMMEDIATE  
;
```

```
ALTER TABLE ScrapPiece ADD CONSTRAINT ScrapPiece_Error  
FOREIGN KEY (codError)  
REFERENCES Error (codError)  
NOT DEFERRABLE  
INITIALLY IMMEDIATE  
;
```

```
ALTER TABLE Piece_Area ADD CONSTRAINT Piece_Area_Piece  
FOREIGN KEY (Id_Piece)  
REFERENCES Piece (Id_Piece)  
ON DELETE RESTRICT  
ON UPDATE CASCADE  
NOT DEFERRABLE  
INITIALLY IMMEDIATE  
;
```

```
ALTER TABLE Piece_Area ADD CONSTRAINT Piece_Area_Area  
FOREIGN KEY (Id_Area)  
REFERENCES Area (Id_Area)  
ON DELETE RESTRICT  
ON UPDATE CASCADE
```

```
NOT DEFERRABLE
INITIALLY IMMEDIATE
;
```

```
ALTER TABLE Area_Stops ADD CONSTRAINT Area_Stops_Area
FOREIGN KEY (Id_Area)
REFERENCES Area (Id_Area)
ON DELETE RESTRICT
ON UPDATE CASCADE
NOT DEFERRABLE
INITIALLY IMMEDIATE
;
```

```
ALTER TABLE Area_Stops ADD CONSTRAINT Area_Stops_Stops
FOREIGN KEY (Id_Stops)
REFERENCES Stops (Id_Stops)
ON DELETE RESTRICT
ON UPDATE CASCADE
NOT DEFERRABLE
INITIALLY IMMEDIATE
;
```

En el apartado de índices de este sistema se deduce que se han creado automáticamente los índices de las claves primarias de todas las tablas, pero se establecerá alguno más para optimizar las consultas.

Se va a elaborar en la tabla Piece_Area un índice para Id_Area ya que en la mayoría de las consultas se buscarán los datos de cada área por separado. Se creará con la instrucción: CREATE INDEX Area ON Piece_Area (Id_Area);

Actualmente se funcionará con índices, pero se recomienda en un futuro realizar cinco particiones, una para cada identificador del área, puesto que es la parte que más crecerá y los datos se consultarán de manera independiente en cada área en la mayoría de los casos.

En cuanto a los disparadores en este sistema, solo se considera necesario un disparador para controlar las piezas malas que serán recicladas, porque al tratarse de datos de proceso, con las condiciones de claves primarias, ajenas, actualización y borrado se mantendrá una consistencia suficiente de los datos.

Si una pieza es mala de manera automática y su código de error es A_02, A_04, A_07 o A_08 será considerada una pieza mala de reciclaje.

Predicado en SQL:

```
NOT EXISTS(
SELECT auto_Button, codError, recycle
FROM ScrapPiece
WHERE auto_Button = TRUE and
codError IN ('A_02', 'A_04', 'A_07', 'A_08') and recycle = FALSE)
```

No debe existir una pieza mala de manera automática, con código de error A.02, A.04, A.07 o A.08 y que no esté marcada como reciclaje

Eventos que pueden violar la regla:

UPDATE O INSERT en la tabla ScrapPiece

Acciones que se deberían hacer:

Cuando se introduce o se modifica una pieza mala, comprobar si cumple las condiciones para ser marcada como reciclaje.

Disparador:

TRIGGER SOBRE LA TABLA ScrapPiece

```
CREATE TRIGGER Reciclo
before INSERT or UPDATE ON ScrapPiece
FOR EACH ROW
EXECUTE PROCEDURE reciclaje();
CREATE OR REPLACE FUNCTION reciclaje() RETURNS TRIGGER AS '
BEGIN
IF (new.auto=TRUE and new.codError IN ('A.02', 'A.04', 'A.07', 'A.08'))
THEN
UPDATE ScrapPiece SET recycle=TRUE when Id_Piece=new.Id_Piece;
END IF;
RETURN new;
END;
' LANGUAGE 'plpgsql';
```

La parte de roles y vistas de este sistema de información no está definida, ya que la integración del mismo dentro de la empresa no será inmediata. Por tanto, cuando se definan los sistemas de información completos se crearán los perfiles, roles y vistas que accederán a una parte de cada sistema. Por el momento el personal encargado de realizar las pruebas y modificaciones tendrá acceso total. En el caso de que se implemente y cuando ésta finalice la implementación se dividirán los roles y vistas para guardar la integridad de este sistema de información.

Asimismo, para este diseño se quiere remarcar unas cuestiones adicionales.

El diseño de este sistema de información está basado en una definición de requisitos de un ideal de la línea. En la actualidad no se dispone de toda la información requerida para la completitud de este sistema. A continuación, se enumeran los puntos fundamentales que no están implementados a nivel físico en la línea, pero que sí se tienen como objetivo futuro como complemento a este sistema de información.

- No se tiene trazabilidad de las piezas, es decir, no podemos seguir una pieza a través de las distintas áreas y recuperar la información de todas las áreas para dicha pieza. Tampoco se tiene trazabilidad en las piezas que son malas, por lo que no sabemos si una pieza será buena o mala. Esto hace que las relaciones entre las tablas “Piece” y “ScrapPiece” y en-

tre “Piece” y “Area” se rompan y se tengan que implementar con identificadores distintos.

- Los diferentes tipos de errores y cuales corresponden a reciclaje no están completamente delimitados, por lo que cuando se estandaricen en la implementación será necesario modificar el disparador “Reciclo”.
- La información que se debe almacenar en la tabla “Stops” actualmente se realiza de manera manual en papel o en Excel, por lo que se deberá implementar una aplicación para introducir estos datos por el encargado de la línea.

Para finalizar aclarar un último punto. Este diseño de base de datos se deberá replicar en dos bases de datos independientes. Para Linea_1 y Linea_4, la información es idéntica pero los datos se quieren tener por separado. Debido a la gran cantidad de datos que se van a generar de los PLC, se decide que es mejor tenerlas por separado.

Soldadura

Siguiendo la especificación de requisitos mencionada en el análisis se ha realizado el siguiente diseño conceptual(figura 3.3):

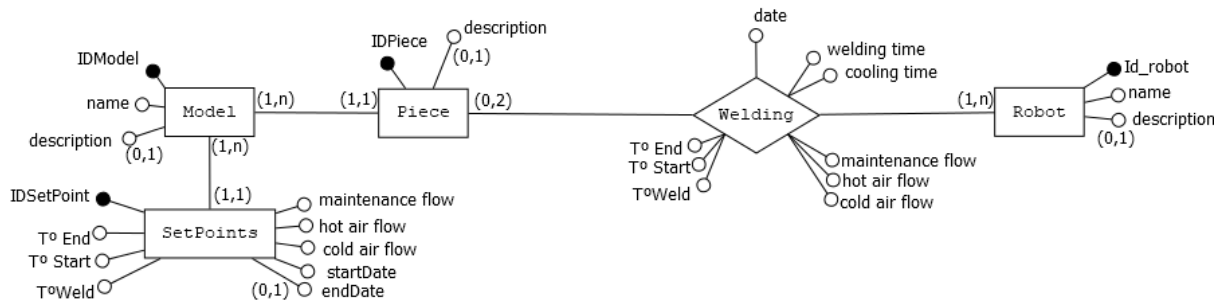


Figura 3.3: Diseño conceptual de Soldadura

Utilizando este diseño conceptual se ha realizado el siguiente diseño lógico(figura 3.4):

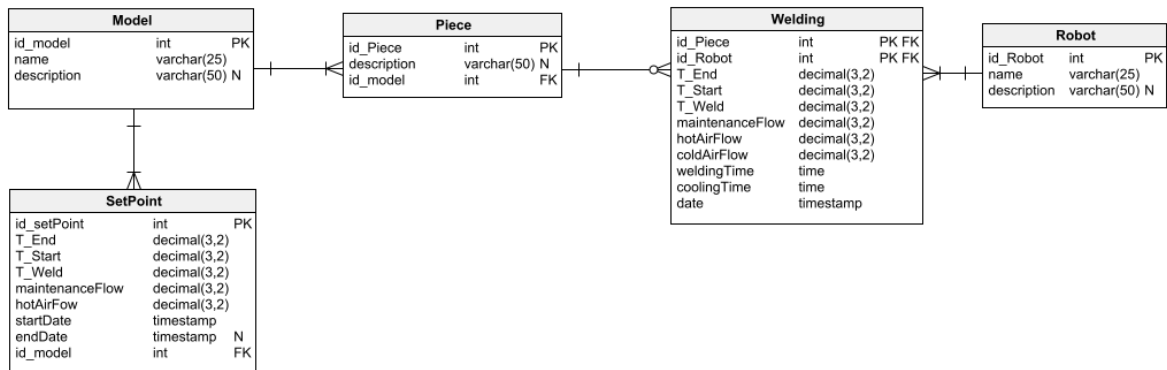


Figura 3.4: Diseño lógico de Soldadura

También se ha realizado el diseño físico según el lenguaje PostgreSQL ya que al no estar decidido el lenguaje en el que se va a implementar se ha elegido este por ser a coste cero e implementar gran parte del estándar SQL.

Las sentencias de creación de la base de datos son las siguientes:

```

CREATE TABLE Model (
id_model int NOT NULL,
name varchar(25) NOT NULL,
description varchar(50) NULL,
CONSTRAINT Model_pk PRIMARY KEY (id_model)
);
  
```

```

CREATE TABLE Piece (
id_Piece int NOT NULL,
description varchar(50) NULL,
id_model int NOT NULL,
CONSTRAINT Piece_pk PRIMARY KEY (id_Piece)
);
  
```

```

CREATE TABLE SetPoint (
id_setPoint int NOT NULL,
T_End decimal(3,2) NOT NULL,
T_Start decimal(3,2) NOT NULL,
T_Weld decimal(3,2) NOT NULL,
maintenanceFlow decimal(3,2) NOT NULL,
hotAirFow decimal(3,2) NOT NULL,
startDate timestamp NOT NULL,
endDate timestamp NULL,
id_model int NOT NULL,
CONSTRAINT SetPoint_pk PRIMARY KEY (id_setPoint)
);
  
```

```
CREATE TABLE Robot (  
id_Robot int NOT NULL,  
name varchar(25) NOT NULL,  
description varchar(50) NULL,  
CONSTRAINT Robot_pk PRIMARY KEY (id_Robot)  
);
```

```
CREATE TABLE Welding (  
id_Piece int NOT NULL,  
id_Robot int NOT NULL,  
T_End decimal(3,2) NOT NULL,  
T_Start decimal(3,2) NOT NULL,  
T_Weld decimal(3,2) NOT NULL,  
maintenanceFlow decimal(3,2) NOT NULL,  
hotAirFlow decimal(3,2) NOT NULL,  
coldAirFlow decimal(3,2) NOT NULL,  
weldingTime time NOT NULL,  
coolingTime time NOT NULL,  
date timestamp NOT NULL,  
CONSTRAINT Welding_pk PRIMARY KEY (id_Piece,id_Robot)  
);
```

```
ALTER TABLE SetPoint ADD CONSTRAINT SetPoint_Model  
FOREIGN KEY (id_model)  
REFERENCES Model (id_model)  
ON DELETE RESTRICT  
ON UPDATE CASCADE  
NOT DEFERRABLE  
INITIALLY IMMEDIATE  
;
```

```
ALTER TABLE Piece ADD CONSTRAINT Piece_Model  
FOREIGN KEY (id_model)  
REFERENCES Model (id_model)  
ON DELETE RESTRICT  
ON UPDATE CASCADE  
NOT DEFERRABLE  
INITIALLY IMMEDIATE  
;
```

```
ALTER TABLE Welding ADD CONSTRAINT Welding_Piece  
FOREIGN KEY (id_Piece)  
REFERENCES Piece (id_Piece)  
ON DELETE RESTRICT  
ON UPDATE CASCADE  
NOT DEFERRABLE  
INITIALLY IMMEDIATE  
;
```

```
ALTER TABLE Welding ADD CONSTRAINT Welding_Robot
FOREIGN KEY (id_Robot)
REFERENCES Robot (id_Robot)
ON DELETE RESTRICT
ON UPDATE CASCADE
NOT DEFERRABLE
INITIALLY IMMEDIATE
;
```

En el apartado de índices de este sistema se deduce que se han creado automáticamente los índices de las claves primarias de todas las tablas. Aparte de estos, no se considera necesario implementar más.

En cuanto a los disparadores se ha considerado que no son necesarios disparadores ya que al tratarse de datos de proceso, los datos serán siempre consistentes.

Además, para este diseño, se quieren remarcar unas cuestiones adicionales. El diseño de este sistema de información está basado en una definición de requisitos de un ideal de la línea. En la actualidad, no se dispone de toda la información requerida para la completitud de este sistema, seguidamente se menciona el punto fundamental que no está implementado a nivel físico en la línea, pero que sí se tiene como objetivo futuro como complemento a este sistema de información.

- Actualmente no se dispone de toda la información de los *Set points* cuando los modifica un jefe de línea. Tampoco hay ningún software encargado de registrarlo en la base de datos.

3.3. Diseño de la interfaz

El diseño de las interfaces, en este caso *dashboards*, se han basado en requisitos propios de los usuarios finales, adecuando un estándar de la empresa a los principios de diseño, limitándonos a una sola pantalla, con dos archivos diferentes para la monitorización y para el análisis del histórico. Se ha tratado de elegir la representación que más satisficiera las necesidades de los usuarios finales además de sus requisitos específicos a la hora de visualizar la información.

Los diseños se separaron para Linea_4 y Soldadura como en los otros apartados. Los diseños en Excel son algo distintos de los de Power BI ya que en Excel se buscaba algo más manejable para los supervisores y con Power BI, una herramienta desconocida para ellos, se diseñó algo más estándar.

3.3.1. *Dashboard* de Linea_4

Durante las reuniones para elegir la representación visual de los datos se definió como parte principal, una gráfica de barras que contuviera las piezas buenas, malas y de reciclaje. Los colores elegidos por la empresa fueron verde para las piezas buenas, rojo para las piezas malas y gris

para las piezas de reciclo. Del mismo modo hicieron énfasis en que el fondo fuera oscuro para mejorar la visibilidad en las pantallas en la planta.

Como KPIs principales, los puntos importantes, la empresa eligió ver los KPIs resumen de todo el turno OEE, Uptime, Yield y YieldReciclo además de un código de colores claro marcado por unos objetivos previamente definidos. Como detalle de relevancia también se solicitó una tabla desglosando los KPI anteriores por cada hora.

El diseño del boceto final (figura 3.5) fue aprobado por ambas partes para realizar su implementación.

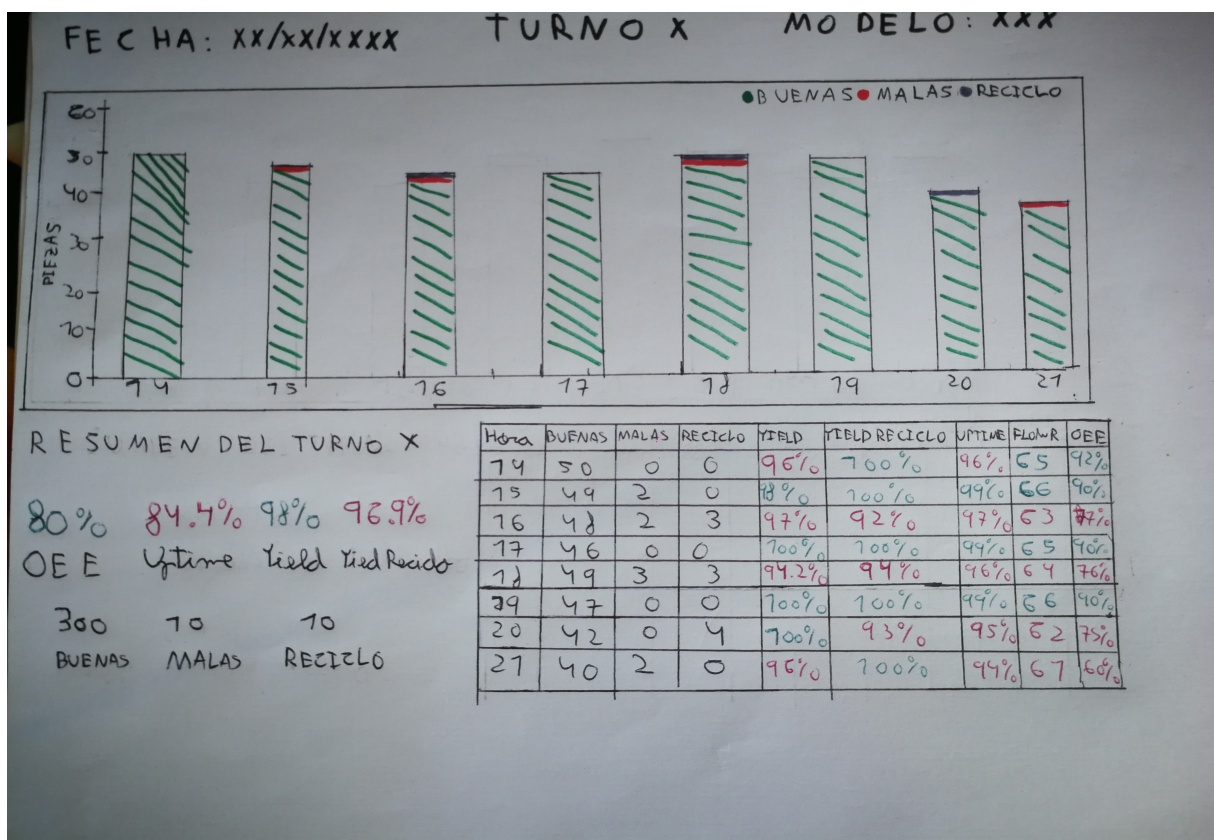


Figura 3.5: Boceto del *dashboard* para Linea_4

Los cuatro KPI principales que aparecen en la parte inferior izquierda son:

Yield es el porcentaje de piezas buenas con respecto a las piezas totales que han fabricado en esa hora.

Uptime es el porcentaje de tiempo real que se ha trabajado durante esa hora.

Flowrate es el número de piezas que se fabrican al minuto. Se calcula utilizando el tiempo de ciclo promedio de esa hora.

Yield Reciclo es el porcentaje de piezas que se han reciclado.

OEE es un valor porcentual que sirve para medir el aprovechamiento integral de la maquinaria industrial se calcula utilizando los anteriores KPI.

Además de los valores totales de piezas buenas, malas y recicladas.

3.3.2. *Dashboard* de Soldadura

Durante las reuniones para elegir la representación de los datos de Soldadura hubo varias limitaciones puesto que los datos deseados no estaban en la base de datos. Después de basarnos solo en la información que teníamos, se decidió realizar unas gráficas de líneas de colores en las cuales se representara la evolución de cada uno de los robots pero con varias líneas horizontales fijas para representar el *Set Point* en verde y los máximos y mínimos en rojo. El objetivo era visualizar de manera muy clara si los valores se salen de los límites. Se propusieron los colores naranja y azul para representar los robots. Por último, la empresa volvió a hacer hincapié en que se quería la gráfica en fondo negro con letras blancas para poder visualizarla de mejor forma en las pantallas de la línea.

Con estos requisitos pactados por ambas partes se diseño el prototipo a papel(figura 3.6) para basarse en él cuando se implementara, esto también fue aprobado por ambas partes antes de continuar con la implementación.

Estos diseños son una primera versión que a posteriori, en la implementación, se podrían ajustar dependiendo de donde se vayan a monitorizar, si lo van a usar los jefes de línea para conocer el estado actual o el histórico o incluso si se busca generar informes para revisarlos en las reuniones de rendimiento.

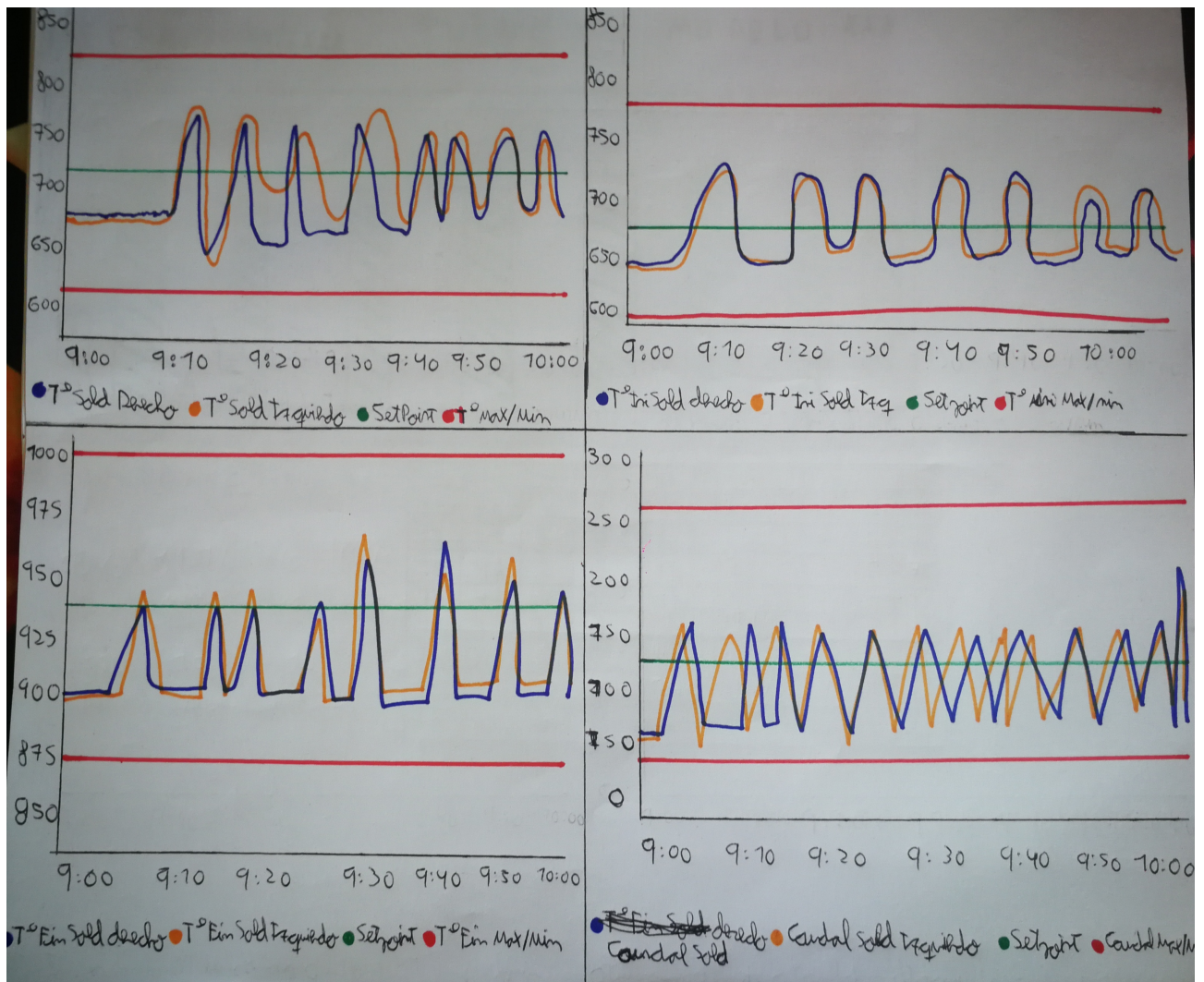


Figura 3.6: Boceto del *dashboard* para Soldadura

En la figura 3.6 cada gráfica representa un indicador diferente. En la superior izquierda se visualiza la temperatura del soldador. En la superior derecha se visualiza la temperatura inicial, la temperatura con la que el robot va a realizar la soldadura. En la inferior izquierda se visualiza la temperatura final, la temperatura con la que el robot acaba cuando se ha realizado la soldadura. En la inferior derecha se visualiza el caudal, el caudal de aire frío que se utiliza para enfriar la soldadura una vez realizada. Además se utiliza el *Set Point*, máximos y mínimos para mejorar la visualización.

Capítulo 4

Implementación y test de usuario

4.1. Diseños de los *dashboards* e informes

En este apartado se exponen los *dashboards* e informes finales, explicando las interfaces, que representan, por qué se eligieron esas representaciones para la información y su navegación entre ellos, además de explicar el informe en global.

4.1.1. *Dashboards* de Linea_4

Los *dashboards* más complejos se realizaron para la Linea_4 ya que es donde mejor estado tenían los datos, donde más información se tenía y donde más cálculos de medidas se debían realizar. Para empezar, éste es el diseño de los *dashboard* en Excel.



Figura 4.1: *Dashboard* de Linea_4 en Excel (Turno 1)

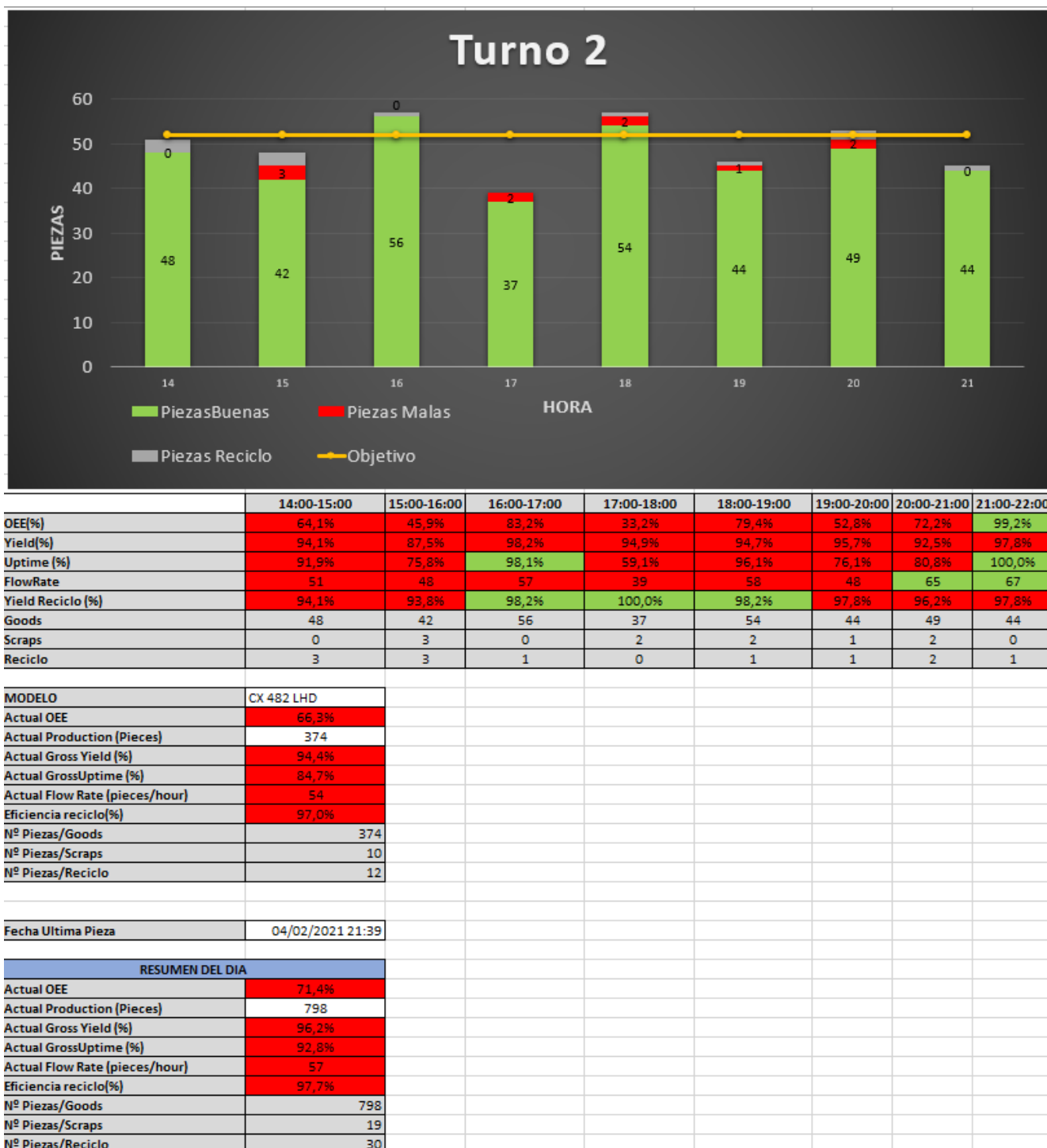


Figura 4.2: *Dashboard* de Linea.4 en Excel (Turno 2)

Estos *dashboards* no siguen exactamente todos los principios de diseño por demandas de la empresa. Del mismo modo, la implementación en Excel no se centra en una sola pantalla y la posición de las representaciones gráficas debería ser con la parte superior con los principales KPI. No obstante para la empresa, la parte más importante era la gráfica. Por lo tanto, se realizó de esa manera. Se realizó de manera principal una gráfica de barras de piezas por hora

limitada en un turno, ocho horas, con cuatro colores distintos: verde para las piezas buenas, gris para las piezas de reciclaje, rojo para las piezas malas y una línea horizontal amarilla para tener rápidamente un indicador de piezas objetivo por hora. En detalle, en la gráfica se pueden ver los valores de cada barra. A continuación encontramos una tabla más específica con las medidas principales por cada hora, con unos códigos de colores básicos de rojo y verde que sirven para indicar si están por encima o por debajo del objetivo. Todo esto se explicará de forma más exhaustiva en el apartado 4.2.1. A continuación, se muestra una tabla para obtener un resumen general de ese turno. Esta tabla incluye el modelo principal que se ha producido, las piezas producidas y las medidas anteriores, con los mismos códigos de colores. Este diseño se replicó para cada turno puesto que la empresa pidió tener los tres turnos visibles en el informe. En el caso del Excel, los datos se disponen en tiempo real al irse cargando según avanzaba el día ya que tampoco se quería perder la información. La forma más alargada del diseño se pidió así debido a que la idea era mostrar la gráfica y la primera tabla en unas pantallas en la línea de producción y tener el Excel a modo de informe con más detalle para los supervisores de línea. Por eso podemos ver en la primera imagen la tabla con los objetivos, al ser valores variables, y en la segunda imagen, la tabla resumen del día con los tres turnos

En segundo lugar, se explicará el *dashboard* en Power BI (figura 4.3).

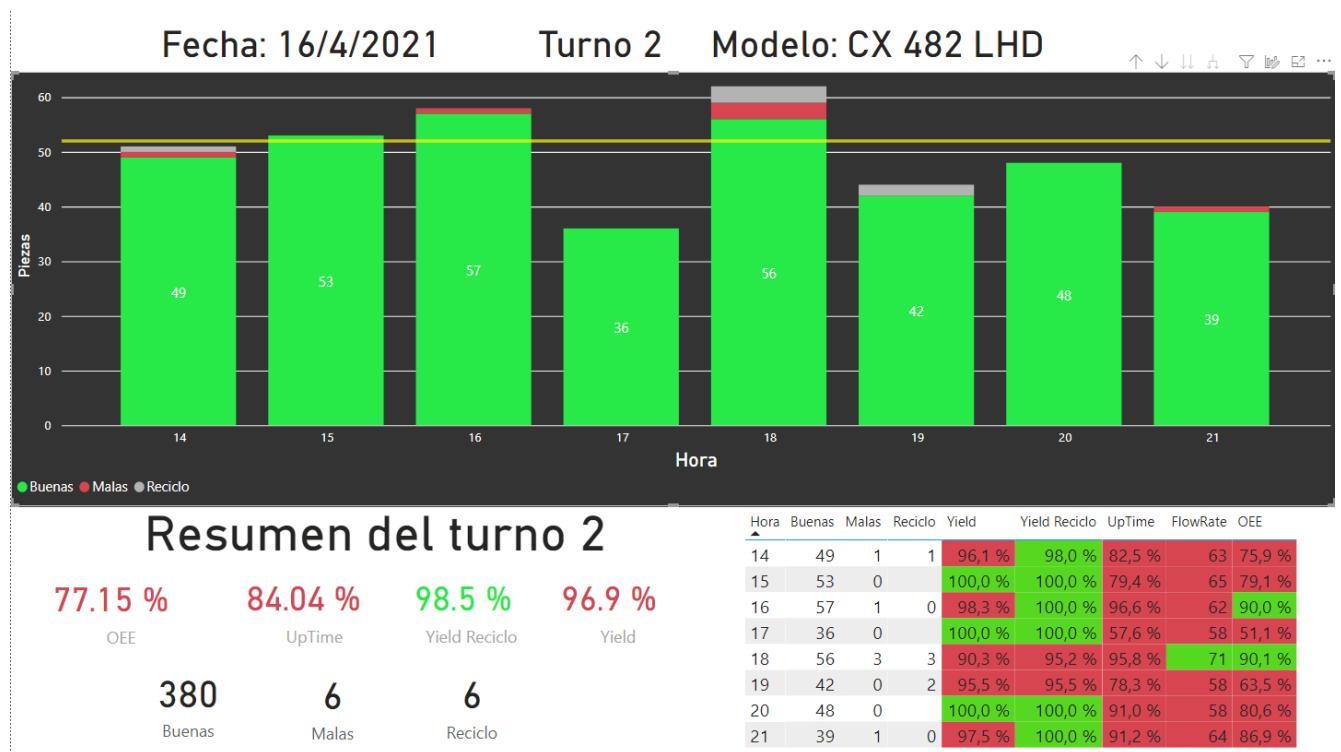


Figura 4.3: *Dashboard* de Linea_4 en Power BI

El *dashboard* de la figura 4.3 sigue la plantilla de colores del anterior (figuras 4.1 y 4.2) para mantener un estándar de diseño, pero con ligeros cambios en la disposición de los objetos visuales. Podemos ver una primera línea a modo de título en la que se muestran el día, el turno y el modelo al que hace referencia esa información. Lo que observamos a continuación es un gráfico de barras similar al creado en Excel para poder ver la producción de piezas buenas,

malas y de reciclo por horas. Justo debajo podemos observar una tabla resumen del turno con los KPI para analizarlos rápidamente y ver si se llega a los objetivos, así como una tabla resumen más en detalle con todos los KPI desglosados por horas. En este caso no es necesario un compendio general del día debido a que, gracias a la navegación en profundidad con operaciones OLAP (*On-Line Analytical Processing*), usando el interfaz gráfico podemos agrupar los datos por días (figura 4.4) o bajar en la jerarquía hasta el turno, por tanto que veríamos todo el día o lo que lleve producido del mismo. En este caso el informe se genera al tener tres hojas con el mismo diseño pero en los tres turnos diferentes. Con esto se obtiene el informe completo del día en detalle, como demandaba la empresa.

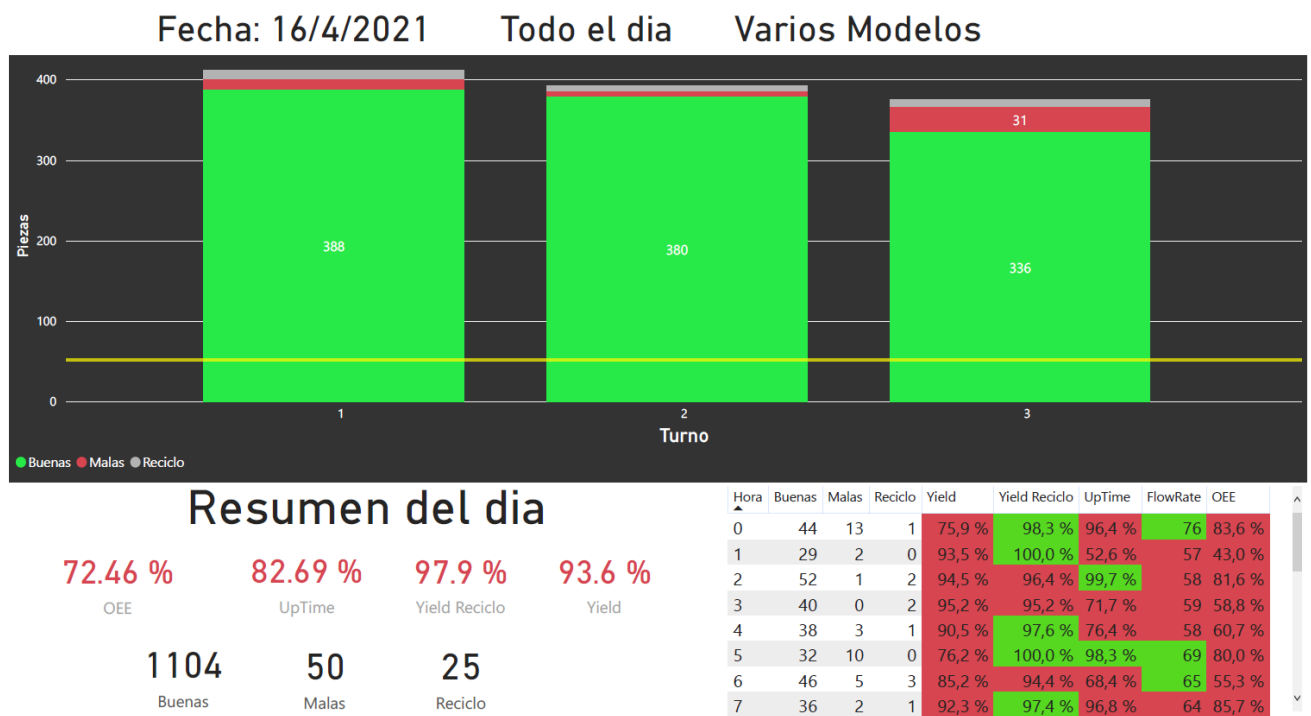


Figura 4.4: *Dashboard* de Linea.4 de un día completo en Power BI

4.1.2. Dashboards de Soldadura

Estos *dashboard* no forman parte de unos diseños estandarizados ni completos debido a que, para obtener los KPI más completos como en Linea.4, se requerían los datos por pieza y no temporales. Sin embargo, la empresa no tuvo tiempo de implementarlo a tiempo durante la estancia en ella. Por esta razón se realizaron únicamente unas gráficas para poder monitorizar y comprobar si los valores de los robots eran los deseados. Para empezar, en la figura 4.5 se muestra el diseño del *dashboard* en Excel.

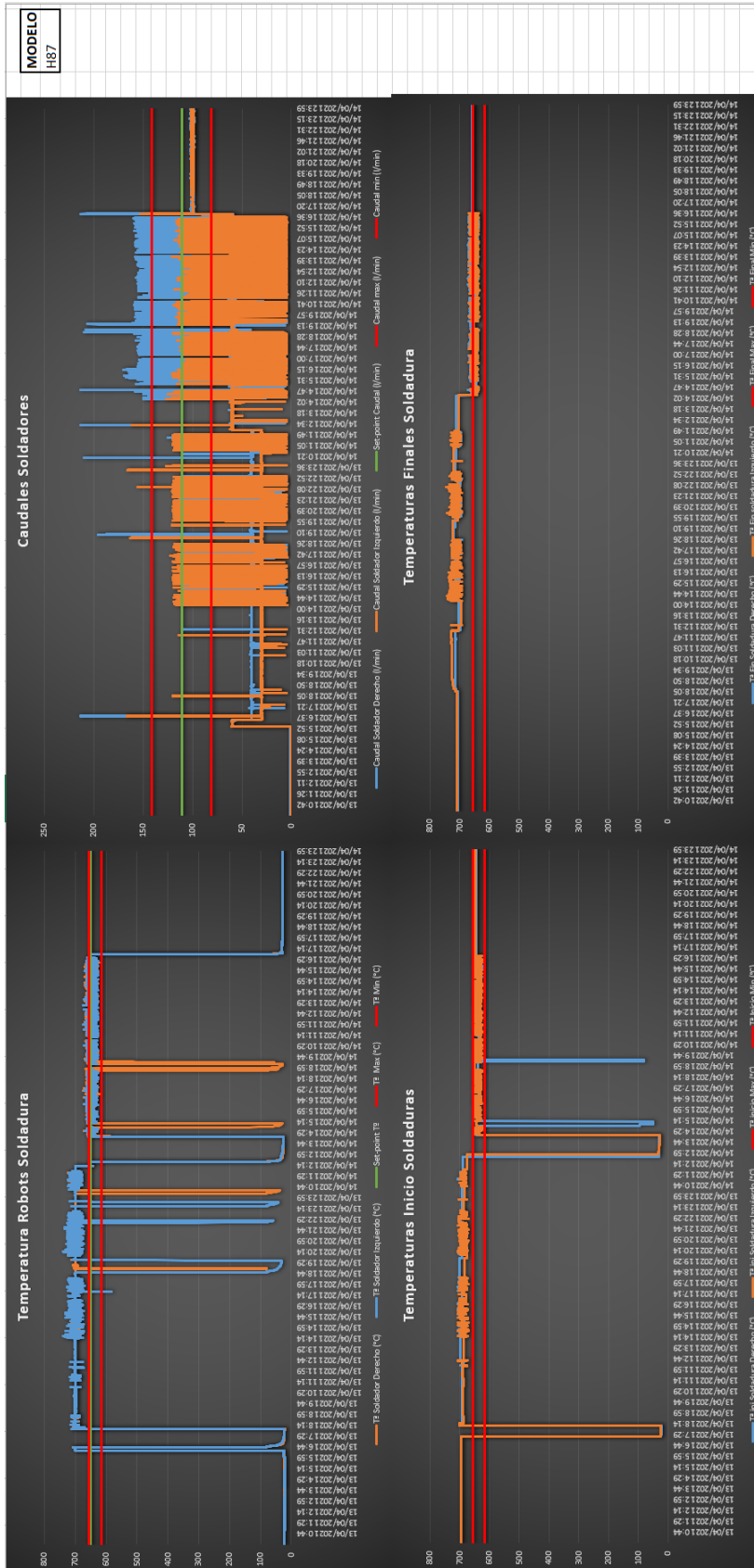


Figura 4.5: Dashboard de Soldadura en Excel

En la figura 4.5 se pueden apreciar las cuatro gráficas: temperatura, caudal, temperatura de inicio y de fin. La gráfica "Tiempo de ciclo" se decidió no implementar ya que los valores no eran los que deberían y no se solucionó ese problema. Al llegar la información de manera incorrecta, se decidió dejarla en otra pantalla aparte para más adelante cuando se solucionara. En estas gráficas podemos ver la evolución de los valores a lo largo de una hora. Los valores tienen los códigos de colores naranja y azul para los robots y dos o tres líneas horizontales para los diferentes límites esperados según el modelo que aparece a la derecha, previamente seleccionado en una hoja aparte para poder variar los límites de las gráficas y sus *Set Point*. El *Set Point* se muestra en una línea horizontal verde y los límites máximos y mínimos en líneas horizontales rojas.

A continuación, se muestran los *dashboards* a modo de informe en Power BI.

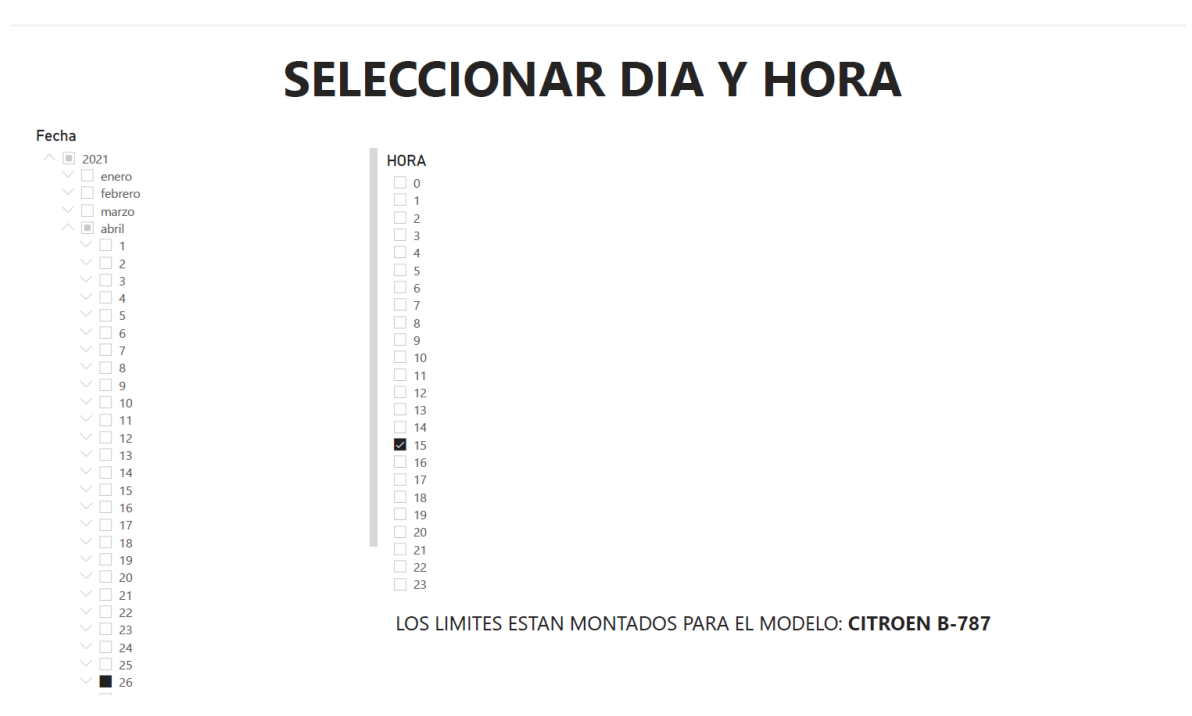


Figura 4.6: *Dashboard* de selección de fecha y hora para Soldadura

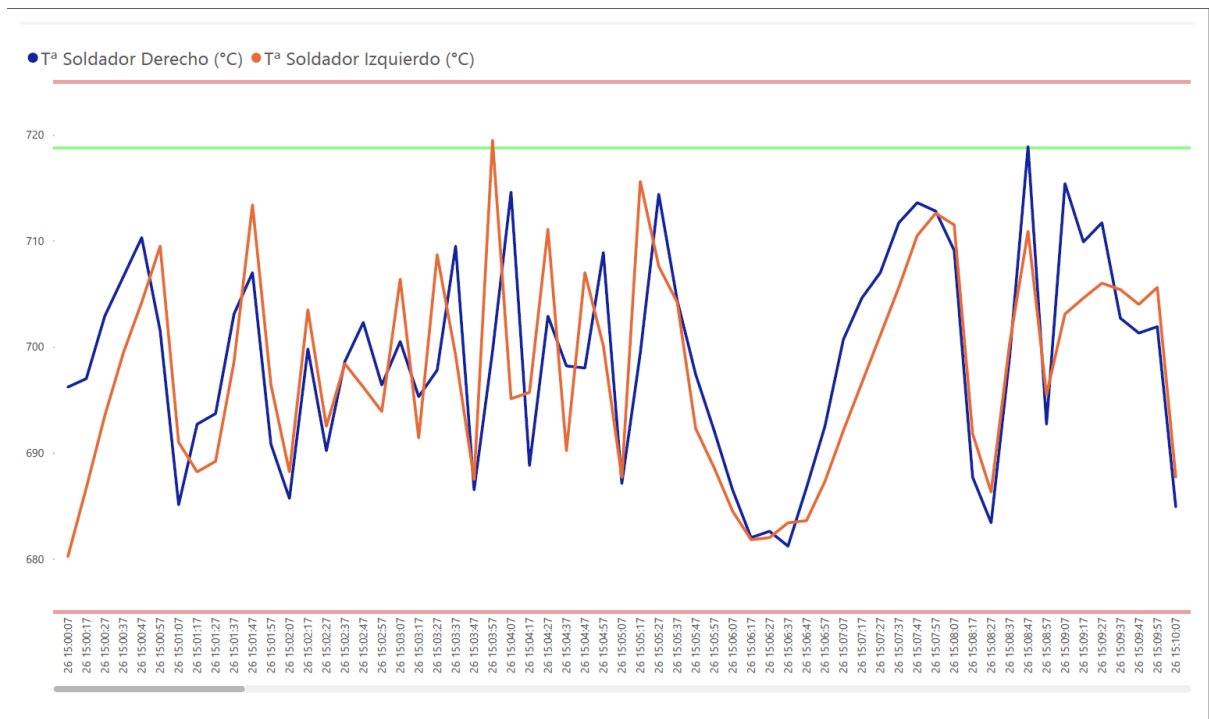


Figura 4.7: *Dashboard* de temperatura para Soldadura

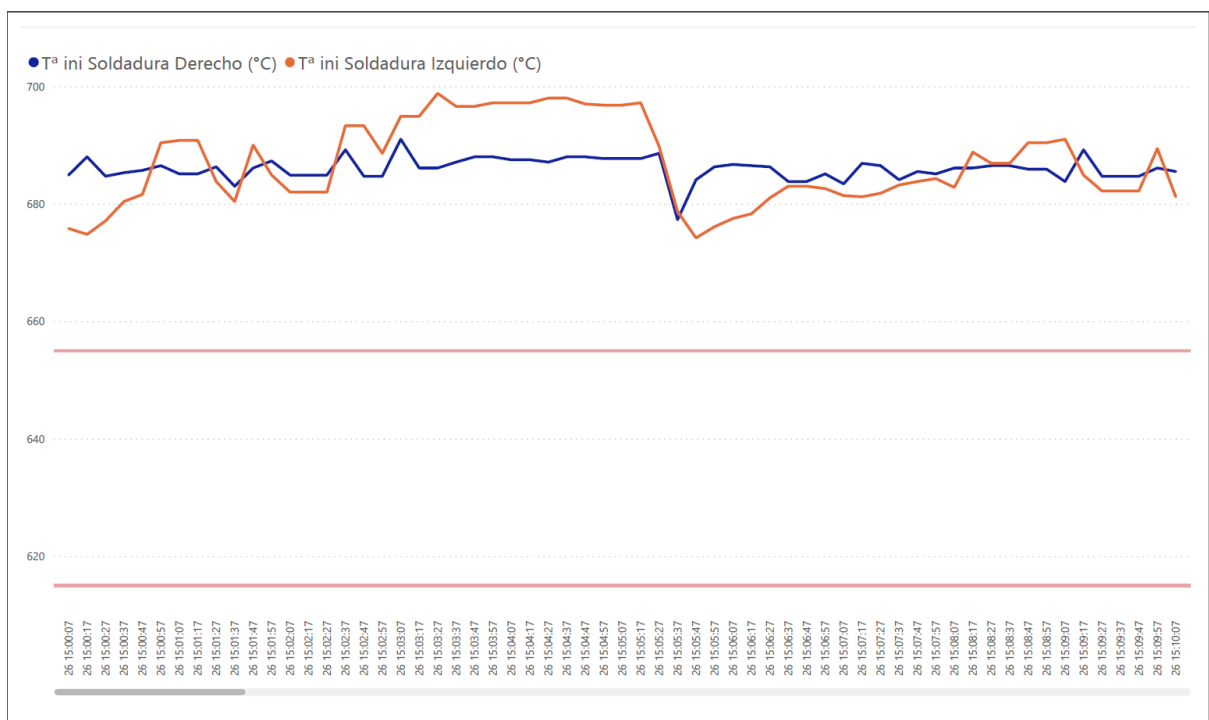


Figura 4.8: *Dashboard* de temperatura inicial para Soldadura

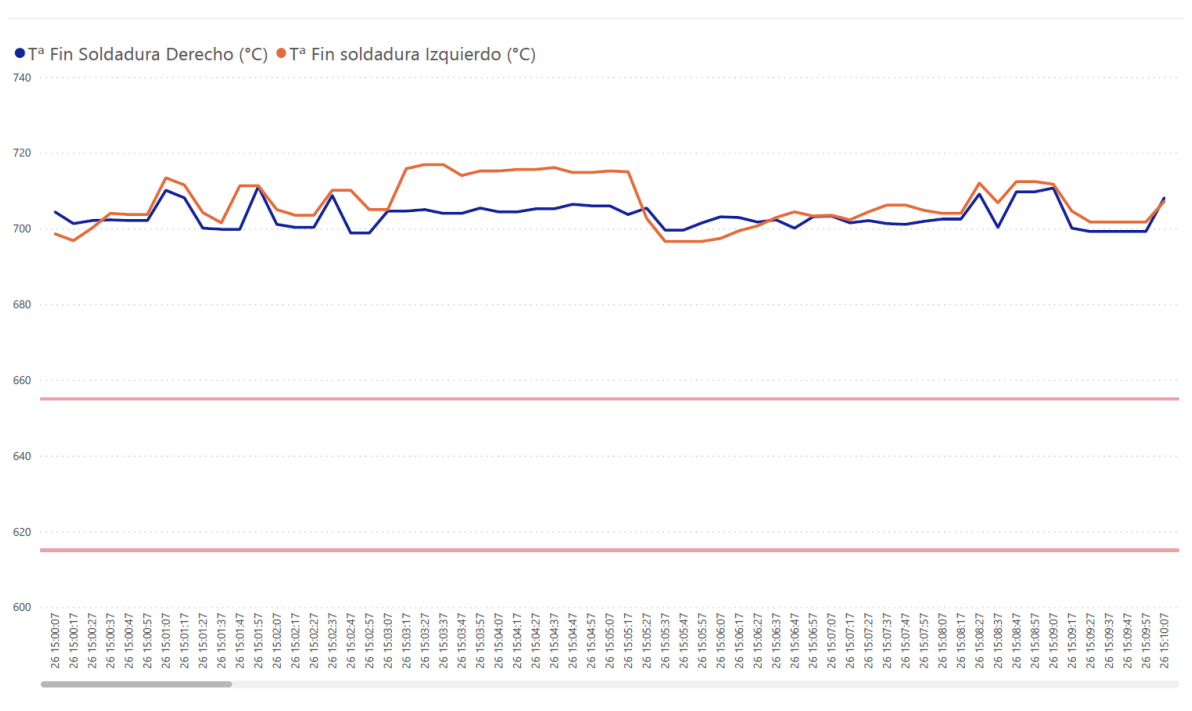


Figura 4.9: *Dashboard* de temperatura final para Soldadura

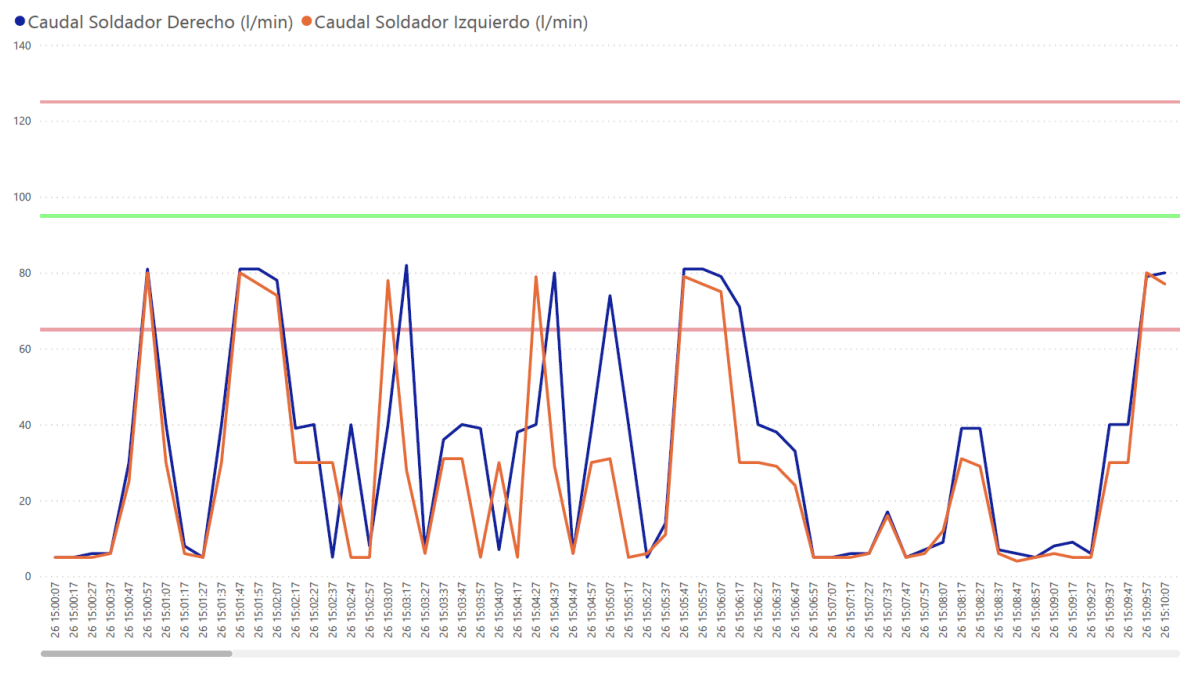


Figura 4.10: *Dashboard* de caudal para Soldadura

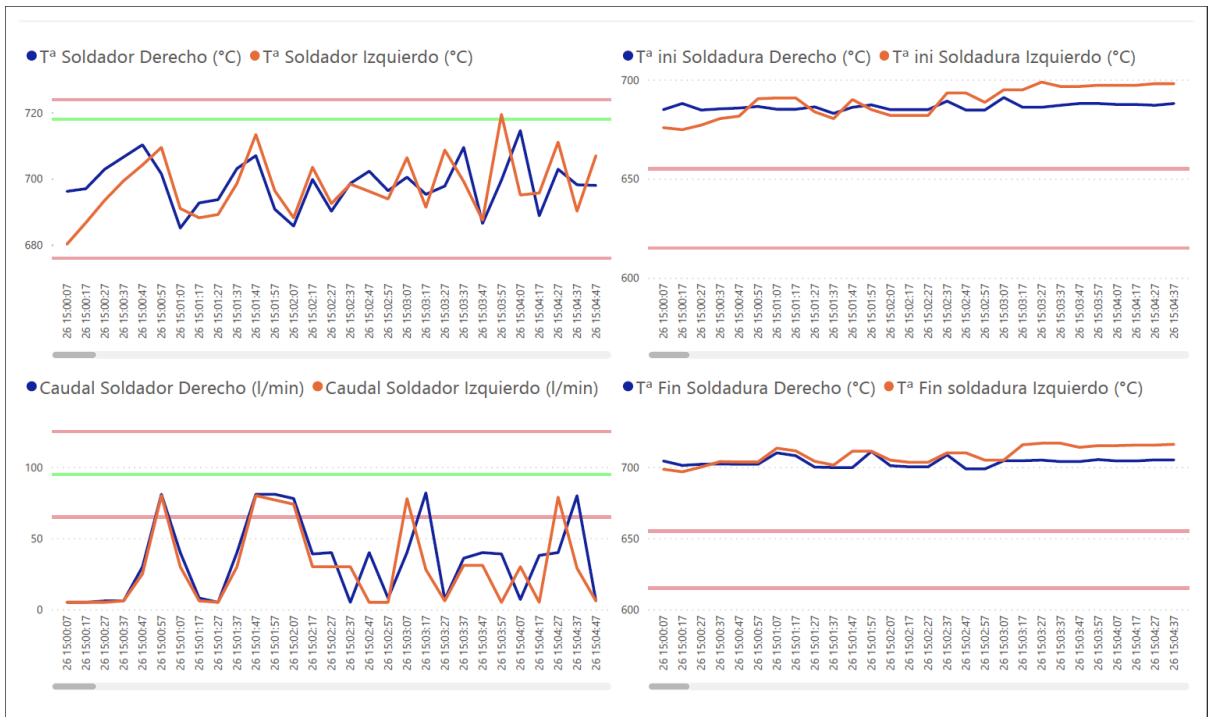


Figura 4.11: *Dashboard* resumen con detalle para Soldadura

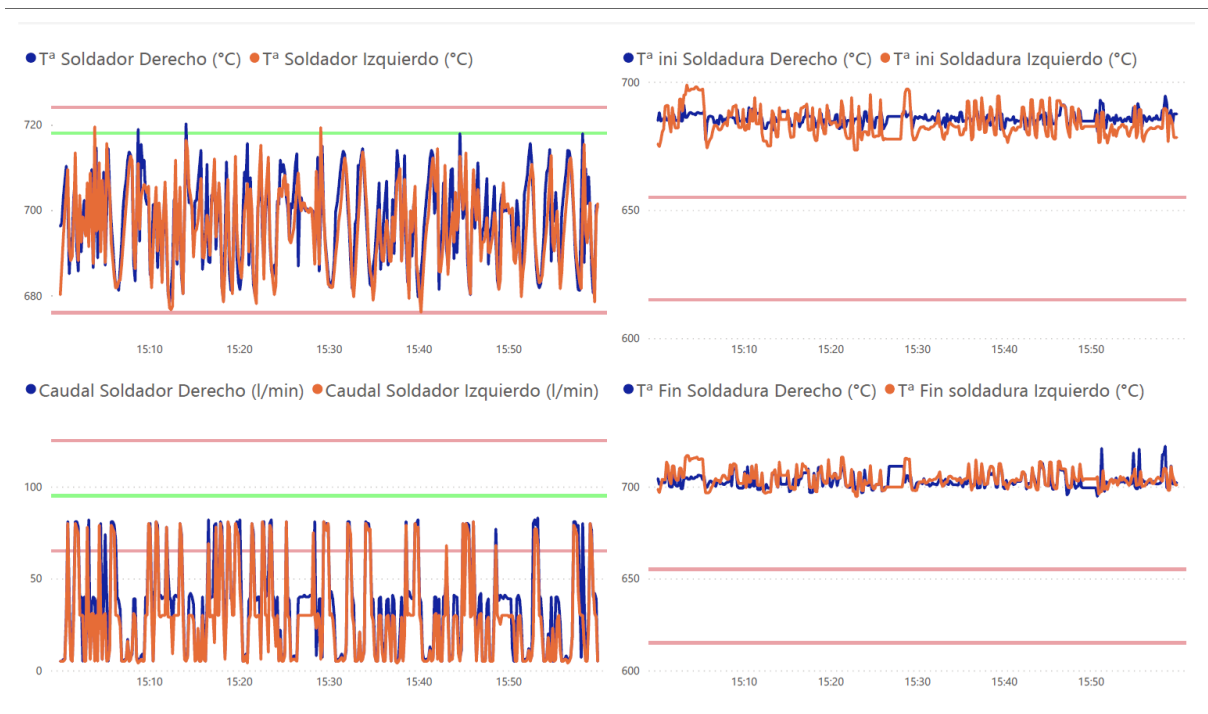


Figura 4.12: *Dashboard* resumen sin detalle para Soldadura

En la figura 4.5 podemos ver una primera pantalla de selección de fecha y hora para graficar los datos. Seguidamente, se observan cuatro pantallas similares con las gráficas en detalle de

cada medición durante la hora seleccionada. Se pueden seleccionar varias, con un *scroll* para poder ver los datos realmente en detalle. Después dos pantallas a modo de resumen de las anteriores, una con *scroll* y otra sin *scroll*, en la que se ve la información más concentrada. Para todas estas gráficas se han mantenido los mismos códigos de colores, azul y naranja para los valores de los robots, verde para los *Set Point* y rojo para los máximos y mínimos.

Este diseño no sigue los estándares de un *dashboards* al no estar concentrado en una sola pantalla ni tener los KPI principales. No obstante, al no tener la completitud de los datos ni la forma esperada no se podían extraer los KPI necesarios para diseñar uno completo por lo que se optó por estas “pantallas” a modo de informe para exportarlo y almacenarlo en formato PDF.

Con esto finaliza la explicación de los *dashboard* e interfaces creadas, en la siguiente sección se explicarán en detalle la implementación de cada uno.

4.2. Detalles de implementación

En este apartado se detallará cómo se ha realizado la implementación de los diseños previamente mencionados, exceptuando los de las bases de datos óptimas ya que no se pudieron realizar. En global, ambas partes Linea_4 y Soldadura, se realizaron de forma parecida en la carga de datos, pero fue variando y optimizándose la forma en la que se hacía conforme mejor se adaptaba a las necesidades, dificultades y diferenciando entre Excel y Power BI.

4.2.1. Linea_4

La implementación de la Linea_4 comenzó con la carga directa mediante consultas SQL a un fichero Excel. Se realizaron consultas separadas para las tablas “Areas” y ”Models” puesto que contienen valores fijos que raramente deben ser actualizados. Después se dividió la carga de la tabla “Pieces”, separando en 3 consultas, una para cada turno. De modo que, de 6 a 14 horas corresponde al turno 1, de 14 a 21 horas corresponde al turno 2 y de 21 a 6 horas corresponde al turno 3.

Para la tabla “ScrapPiece” hubo mayores dificultades ya que no estaban bien definidos los códigos de errores, había repetición de datos y los códigos que correspondían a piezas de reciclaje no se conocían con seguridad. Se comenzó con una primera versión que solo contabilizaba las piezas erróneas, utilizando solo una tabla del esquema inicial al estar duplicados sus valores. Más adelante se mejoró para implementar también la diferenciación entre piezas malas por botonera y de manera automática.

Las consultas SQL debían ir directamente al servidor. En una primera versión se cargan los datos con la interfaz básica de Excel, pero más tarde se mejoró para realizar algunas transformaciones y optimizaciones a nivel de consulta y así reducir el volumen de datos cargado.

Después de la carga de datos, se necesitaba realizar una clasificación horaria de éstos. Esto se realizó mediante columnas y fórmulas auxiliares en Excel y dos operaciones sobre los datos “Tiempo de procesado”, “Tiempo de ciclo” y “Tiempo de parada técnica”. Mientras que, el

tiempo de procesado de una pieza se calcula restando la fecha de procesado a la fecha de entrada. El tiempo de ciclo de una pieza se calcula restando la fecha de entrada de la siguiente pieza menos la fecha de entrada propia. Este cálculo no fue fácil ya que el lenguaje Transact-SQL no implementa correctamente todo el estándar, además de tener que conocer la entrada siguiente en una tabla para poder completar la anterior. Asimismo, se introdujo un limitador para que no introdujera en los cálculos aquellos tiempos de ciclo que superaran un umbral por considerarse que eran valores por errores en la línea o paradas y que no se debían tener en cuenta. Esta limitación y el valor del umbral se realizaron con la aprobación del encargado de la línea.

El tiempo de parada técnica se calcula restando el tiempo de ciclo de una pieza a un tiempo límite que marca si en esa pieza se ha realizado una parada o no. Este limitador y el umbral se realizaron con la aprobación del encargado de la línea.

Una vez realizadas las transformaciones y los cálculos sobre los datos se procedió a elaborar las gráficas para visualizar y monitorizar los datos. Se siguieron los modelos mostrados a papel, pero con ligeras modificaciones. Se escogió un fondo negro para mejorar la visibilidad en las pantallas de la línea y a preferencia de los usuarios finales. Los KPI de las tablas se calcularon de la siguiente forma:

Yield: piezas buenas/piezas totales.

Uptime: tiempo disponible-tiempo parada/tiempo disponible.

Flourate: una hora/tiempo de ciclo medio en cada hora.

Yield Reciclo: piezas de reciclo/piezas totales.

OEE: Yield * Uptime * Flowrate * Yield *(Yield reciclo/referencia de OEE).

Luego se agrupan los valores totales de piezas buenas, malas y recicladas para cada hora.

Esto se realiza para cada uno de los turnos. De este modo se pueden visualizar cada turno de manera individual o las 3 tablas para ver una visión del día de manera global.

Para facilitar este resumen, se realizaron también unas tablas de resumen de cada turno añadiendo el modelo que se trabajó durante ese turno y otra agrupando los resúmenes de los turnos y un resumen del día completo.

Después de esto se realizaron modificaciones en las consultas y diferenciación de archivos con el fin de crear dos diferentes para cada objetivo. Uno para monitorizar, el cual solo cargaría los datos del día en concreto en el que se abre el archivo, actualizándose cada minuto en tiempo real de la línea. Y otro llamado “flexible” para poder obtener el resumen de un día concreto con el fin de analizarlo en las reuniones de rendimiento o revisar posibles malos funcionamientos.

Para la realización de este último se documentó, mediante un PDF insertado en el archivo Excel, un tutorial explicativo de cómo modificar las consultas para poder seleccionar el día en concreto que se quería cargar. Este documento se realizó por petición del supervisor debido al desconocimiento del lenguaje SQL y la necesidad de trabajar con él cuándo se finalice las

prácticas. Este tutorial fue probado por él y varios usuarios para comprobar que se podía seguir sin problema. Se realizó un documento aparte para mostrar cómo se modificaba el código en SQL ya que se desconocía el lenguaje de Power Query de Excel. Asimismo la empresa no estaba tampoco familiarizada con este lenguaje, y como también había una parte de formación en Power BI se decidió optar por esta vía, más rápida pero siempre con la aprobación del supervisor, realizando directamente consultas SQL con el lenguaje Transact-SQL a través de Power Query. De una manera óptima se debería haber realizado la consulta con la interfaz y lenguaje de Power Query añadiendo un parámetro dentro de las propias celdas de Excel para que fueran modificadas.

Para los *dashboards* de Power BI hubo varios problemas con las licencias y actualizaciones que dificultaron y ralentizaron su desarrollo, pero como la formación se realizó de manera autodidacta con las colecciones de Microsoft[2] se pudo compaginar. Cabe señalar que los módulos concretos se encuentran listados en el anexo B.

Para la carga y transformación de datos había dos maneras distintas de hacerlo. Por una parte, cargar todos los datos en una sola operación y dentro del Power BI transformarlos, realizar los cálculos y luego generar los *dashboards* y paneles. Y, por otra parte, para que se pudieran actualizar presionando el botón, ya que las actualizaciones periódicas y por lotes necesitan de la versión premium y una API o Microsoft Azure. Por tanto, la única manera viable de realizar esa actualización era mediante una consulta que realizara todas las transformaciones y cálculos directamente y cargara los datos listos para ser visualizados. Cabe destacar que como primera solución se probó con una carga de datos desde un fichero Excel, que sería el que se actualizaría de manera periódica y realizaría las transformaciones y cálculos. Esta opción se desechó por la complejidad e ineficiencia que podría darse al tener dos archivos abiertos cargando los mismos datos. La consulta es algo compleja y pesada para el servidor, pero como actualmente tiene capacidad suficiente y no sería un cuello de botella, se optó por ello como decisión final.

De esta manera se cargaban los datos en Power BI y los *dashboards* se podían actualizar en tiempo real además de consultar días pasados. Gracias a la agrupación entre atributos jerárquicos de las dimensiones podemos visualizar los resúmenes de turnos, días, semanas e incluso meses naturales.

Asimismo, en las gráficas se realizaron varias medidas auxiliares para mejorar la visualización y que los títulos se autoajustaran según el nivel de agrupación que se realice, además de un objeto visual importado para poder visualizar las tareas de KPI con formatos condicionales de colores.

Por último, se llevó a cabo un trabajo de investigación para mejorar la usabilidad compartiendo el panel con personal de la empresa e incluso con la versión móvil. Pero solo se consiguió realizarlo en la cuenta del supervisor por problemas con las licencias y conexión con el servidor dentro de la empresa.

4.2.2. Soldadura

La implementación de Soldadura se comenzó con la carga directa mediante SQL a un fichero Excel. En este caso se trata de una única consulta, pero mucho más costosa de ejecutar ya que

por cada entrada de datos, una fila en nuestro diseño final, hay que transponer las diez que hay en la base de datos, una por cada TagIndex diferente, asociando a cada TagIndex el nombre real que le corresponde. La consulta SQL final tiene diez *JOIN-ON* y diez *WHERE* para juntar esos valores para una misma fecha y tiempo. En la versión de Excel, la consulta carga los datos directamente del servidor al archivo, pero se ha delimitado debido al coste computacional de la misma.

Para este caso se han realizado dos Excel distintos, como en Linea_4 , uno para monitorizar el estado actual/reciente, cargando los datos de la última hora y otro para cargar los datos de un intervalo de tres días, que es el tiempo medio de producción de un modelo. Para este último también se realizó un documento PDF insertado en el archivo Excel, con un tutorial de cómo modificar la consulta para seleccionar el intervalo de días deseado. Ésto se realizó así por los mismos motivos que en Linea_4.

Las gráficas tenían delimitadores de *SET-Point*, máximo y mínimo dependiendo del modelo a visualizar, pero estos valores no estaban en la base de datos por lo que se introdujeron de manera manual en Excel además de una casilla donde seleccionar el modelo para que se actualizaran las líneas de referencia.

Para los *dashboards* de Power BI se optó por otra vía. Ya que había un mayor número de gráficas. Se ha optado por realizar diferentes *dashboards* a modo de informe en el cual hay uno principal con las 4 gráficas en las que se muestra toda la hora, otro en el que se puede navegar para ver en detalle esa hora y las 4 gráficas por separado por si se quiere analizar o comprobar algún valor en concreto. Este Power BI no cumple con las reglas de diseño ya que no es un *dashboard* al uso con toda la información, pero se decidió hacerlo así ya que era lo que buscaba la empresa y lo que les resultaba más útil. También se optó por separar dos documentos como en Linea_4, dejando uno con solo la última hora, ya que es mucho más rápido, y otro que carga todos los datos, ya que power BI tiene la capacidad de almacenarlos, no como Excel. Esta consulta es algo más lenta, entorno a los 20 segundos, pero no es algo que se vaya a realizar todos los días a todas por lo que es un coste asumible.

4.3. Limitaciones de la implementación

En este apartado se van a recapitular y profundizar en las limitaciones que ha habido en la implementación de este proyecto ya sea de Linea_4, Soldadura, Linea_1 tanto en Excel como en Power BI

4.3.1. Limitaciones de acceso a los datos

Lo primero es tratar las limitaciones que ha habido en el acceso a los datos o a la estructura óptima de los datos. Desde el primer día ha habido problemas con la conexión a los servidores ya que no estaban en la red de la empresa. Como consecuencia, se tenía que estar presencialmente conectado por cable puesto que se podían acceder desde la VPN (*Virtual Private Network*). Como se ha comentado en el análisis, el estado de los datos inicial era muy pobre, casi sin ningún tratamiento previo. Sin embargo, la limitación más importante viene al no poder cambiar

ese estado, aunque se ha hecho el diseño de base de datos “óptimo” bien sea a nivel técnico o bien por falta de recursos no se podía crear las base de datos que hubiesen sido necesarias para almacenar de manera eficiente la información y procesarla de una manera óptima. Esto en algunos casos ha llegado a suponer que no se pudiera trabajar, como es el caso de la Línea_1, o que se trabajara de una manera subóptima como en Soldadura.

Por último, añadir que las fases de exposición final para presentar el trabajo, recibir el *feedback* y realizar los cambios en consecuencia de los test de usuario y la validación se realizaron una semana más tarde de lo previsto, por lo que no se pudo hacer por falta de tiempo una validación más adecuada. Remarcar el complejo análisis de las bases de datos al no tener ningún tipo de documentación sobre ellas, tener unos nombres nada explicativos y algunos códigos no documentados de los que solo disponían información gente experta en la línea, cosa que dificultó y ralentizó esta primera parte del trabajo. Todo esto ocasionó que la base con la que trabajar no fuera estable, pero por demandas de la empresa se continuara hacia adelante ya que ellos buscaban un prototipo y no un proyecto completo, que se podría haber hecho sin dificultad con los recursos adecuados o con una base previa de conocimiento sobre los datos.

4.3.2. Limitaciones en Excel

Las limitaciones en la carga de datos en Excel no han sido excesivas debido a que la complejidad venía en la propia transformación de los datos. No obstante, el diseño de *dashboard* no es un diseño que siga los estándares a petición de la empresa puesto que buscan algo más flexible y manejable. El editor Power Query era desconocido y no hubo ningún tipo de formación ni de corrección de errores en su utilización.

4.3.3. Limitaciones en Power BI

Las limitaciones en Power BI comenzaron desde el primer día que se quiso trabajar con él ya que no había nadie conocedor de la herramienta en la empresa. Las horas de aprendizaje autodidacta fueron más de las deseadas. A todo eso había que sumarle el hecho que no se tenía una licencia oficial actualizada. La versión que se instalaba al solicitarla a su servicio técnico era del año 2018 lo que imposibilitaba la realización del trabajo ya que toda la formación estaba actualizada a versiones más recientes. Finalmente se consiguió compaginar con la licencia propia al ser estudiante de la Universitat Jaume Primer.

Puesto que no había un plan de formación definido, se eligió realizar los módulos de Microsoft para poder profundizar más, pero no estaban claras las limitaciones y hubo mucho trabajo de investigación para realizar las tareas de actualización periódica entre otras. Muchas de las peticiones se descartaron al estar todas dentro del paquete Power BI premium y al no tener la licencia la gran mayoría de la planta.

4.3.4. Limitaciones de calendario

Las limitaciones de calendario fueron a raíz de que las fórmulas de cálculo de los KPI varían en función del calendario de la empresa y no de un calendario natural, dependiendo de los días que se trabaje, de los descansos y de los turnos que se trabaje, ya que normalmente en el tercer turno no se trabaja. Por estos motivos las operaciones OLAP de agrupación por mes y año no funcionan correctamente.

4.4. Test de usuario

En cuanto a las pruebas, validación y test de usuario. La mayoría de las pruebas y validación se realizaban al acabar junto con el supervisor, asimismo al no haber código de programación, no se podía ser más exhaustivo.

Para los test de usuario, al tratarse de un proyecto centrado en la visualización final del usuario, se realizaron los test de los PDF a modo de tutorial en los Excel. Con el fin de corroborar que eran usables las gráficas y paneles finales, se presentaron en una reunión con los principales jefes de la planta, en la cual se explicó el trabajo realizado, las necesidades que subsanaban y los paneles. Aquí se obtuvo muy buen *feedback* por parte de la totalidad de la empresa tanto en la usabilidad, como en la estética de estos. Como este era un proyecto prototipo de la empresa, también se quiere destacar el buen recibimiento que tuvo, la gran valoración que hicieron de él y, al margen de lo que suceda al final, se fue muy convincente a la hora de explicar el coste del proyecto real y sus ventajas con los diseños óptimos implementados. Después de esto, se realizó algún pequeño ajuste con el *feedback* recibido y se dio por finalizado el proyecto.

4.5. Resultados del proyecto

En esta sección se van a recapitular todos los resultados que se han conseguido y explicado en detalle anteriormente, a pesar de las limitaciones.

Primero, se ha conseguido solucionar los problemas de conexión con los servidores, analizar y transformar satisfactoriamente todos los datos, a pesar de su pobre estado inicial. Con esto se ha conseguido realizar unos archivos Excel dinámicos para poder monitorizar y analizar los datos históricos de Linea_4 y Soldadura. Aunque no sea la mejor opción para realizar *dashboards*, se ha conseguido unas gráficas dinámicas, que monitorizan las líneas de producción y que resultan útiles tanto a los operadores de línea como a los jefes de línea.

En segundo lugar, se ha conseguido aprender a usar una herramienta desconocida, pero muy útil, como es Power BI a pesar de que tuvo que ser de manera autodidacta. Con esto se consiguió desarrollar unos *dashboard* de calidad con los que se pueden analizar todos los datos de Linea_4 y Soldadura, sobretodo el dashboard de Linea_4 por su mejor estado de los datos. Ésta es la mejor de las dos soluciones, sin ninguna duda, pero era necesaria la contrapuesta de Excel para poder escenificar mejor las limitaciones de Excel en estas tareas y el buen desempeño que puede

proporcionar Power BI.

En tercer lugar, se ha diseñado de manera completa las bases de datos óptimas para Línea_4, Línea_1, ya que comparten requisitos y Soldadura. Aunque por motivos técnicos no se hayan implementado, sí le serán útiles en un futuro a la empresa.

En conclusión, con los *dashboards* en Excel y Power BI, los diseños de las bases de datos y la presentación final (Anexo A) se ha conseguido aportar un valor real a la empresa en este momento, preparar las futuras mejoras, así como demostrar el valor real que se puede extraer del tratamiento de datos y su análisis.

Capítulo 5

Conclusiones

En este apartado se desarrollarán las conclusiones sobre el proyecto. En primer lugar, me gustaría reflexionar sobre mis conclusiones personales. La estancia en la empresa ha sido mucho más difícil de lo esperado por los múltiples problemas y dificultades que ésta tenía a nivel informático. Me he dado cuenta de la gran cantidad de trabajo que tenemos los informáticos en el mundo laboral para actualizar a la mayoría de las empresas al ritmo que evolucionan las tecnologías y por ende las demandas de los clientes. Me ha sorprendido el cambio de mentalidad de solo centrarse en hacer bien tu trabajo, lo más rápido y eficaz posible, a una mentalidad centrada en la petición de recursos, en que te escuchen y que simplemente te entiendan para que te dejen trabajar como mejor sabes.

En cuanto al trabajo que he realizado, he de confesar que no estoy muy contento con el resultado final aunque soy capaz de llevar a cabo lo el trabajo que demandaba y necesitaba la empresa, no he podido realizarlo en su totalidad. Aun así, estoy muy contento con el desarrollo, mi evolución dentro de la empresa y sobre todo por la gran cantidad de problemas que se han tenido que solventar para poder finalizar con un trabajo que aporta valor a la empresa inmediatamente y que, sobre todo esto es lo que más valoro, les ha ayudado a estar un poco más cerca de una buena informatización de sus líneas de producción, despertando esa curiosidad y demostrándoles el gran abanico de posibilidades que tienen sin explotar.

Las conclusiones técnicas del trabajo han sido muy positivas ya que se han completado en su totalidad los objetivos tecnológicos excepto el de estructurar los datos recibidos en el servidor. Cabe destacar que éste se ha imposibilitado por causas externas al tener problemas de configuración en el envío de los datos por los PLC a los servidores y la limitación a la hora de crear bases de datos externas en el servidor. En cuanto a los objetivos empresariales se considera un éxito total puesto que los jefes de planta quedaron muy satisfechos con el trabajo efectuado, los *dashboard* ya se están utilizando como guía y hay un interés real en pasar del prototipo a una implementación total del proyecto.

En resumen, no creo que haya realizado una estancia en prácticas sin más, sino una buena dosis de realidad de la incorporación a un mundo laboral muy complejo, en un ámbito diferente al común, al no ser una empresa de base tecnológica, pero con la satisfacción al menos de que la empresa ha estado satisfecha con el resultado final. Al fin y al cabo, el cliente es el que manda.

Bibliografía

- [1] Pagina web de Pilkington. <https://www.pilkington.com/es-es/es/acerca-de-pilkington> [Consulta: 3 de Mayo de 2021]
- [2] Pagina web de los modulos de aprendizaje de Power BI. <https://docs.microsoft.com/es-es/learn/browse/?products=power-bi> [Consulta: 3 de Mayo de 2021]
- [3] Licencias Microsoft 365. <https://www.microsoft.com/es-es/microsoft-365/buy/compare-all-microsoft-365-products> [Consulta: 4 de Mayo de 2021]
- [4] Pagina web de Vertabelo. <https://vertabelo.com/> [Consulta: 23 de Abril de 2021]

Anexo A

Presentación

Definición

Éstas son las diapositivas de la presentación que se realizó a los jefes de la fábrica al finalizar el proyecto. En la diapositiva número 3 se navegó por los diferentes archivos e interfaces para explicarlo mejor y de una manera más interactiva con el fin de llamar su atención.

Diapositivas



Figura A.1: Diapositiva 1 de la presentación

Data processing and analysis

Problem description:

Now a days, we have the reports and graphics in excel, we need to ask the operators how the line is working. The reports and graphics in Excel are more limited to store and can not analyze the data, for this reasons:

- We don't have real time production monitoring.
- We can't see how the plant is working
- We can't see how the plant worked in the past
- We don't have reports or information about one line in case of claim

Proposal:

Our proposal to solve this is use power BI, Power Bi is a business intelligence tool of Microsoft and witch it we can solve all the problems and develop more this plant for the future.

To be able to use it we need to create a base to structure data in a databases to store all the data

How:

- Create prototype with excel, with this data and servers
- Create a simple dashboard
- Implement databases in the servers and demonstrate improvement
- Prototype done with:
 - Hot air Soldering process T01 (Excel & Power BI)
 - L4 Tertiary (Excel & Power BI)

Figura A.2: Diapositiva 2 de la presentación

Dashboard prototype

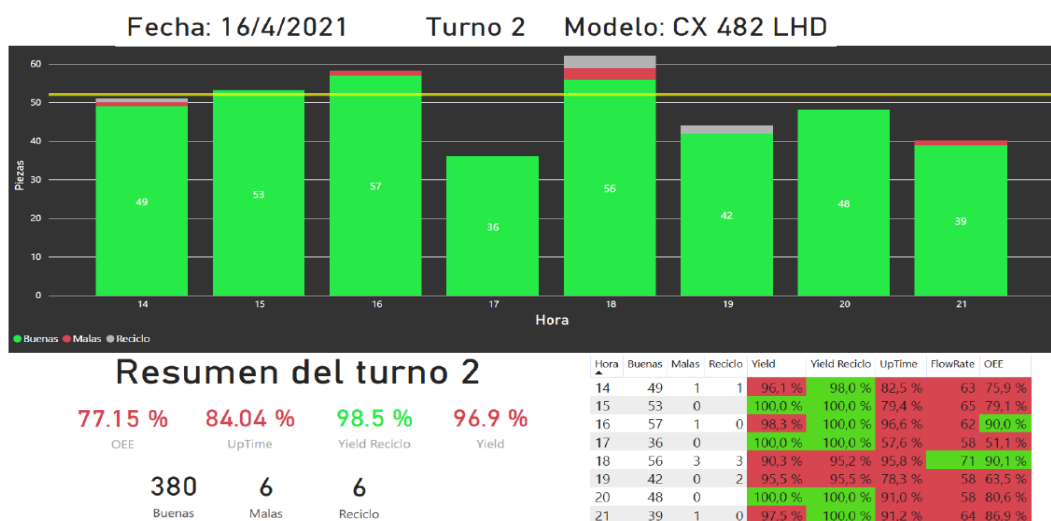


Figura A.3: Diapositiva 3 de la presentación

SUMMARY



- **Benefits**
 - Real time production monitoring (KPIs & Statically process control)
 - Fast and flexible reports
 - Quick scalation for management (KPIs)
 - Scalable, when we have the base, the possibilities are unlimited
 - data mining
 - trend analysis
 - anticipation of errors
 - and more
- **Next steps**
 - Cost
 - On the order of 15 k€ for new servers and licenses
 - Resources
 - Computer engineer profile who specializes in SQL, BI and data processing
 - Automation engineer in plant who specializes in PLC to connect it to databases
 - Timing
 - Estimation one year for the complete plant (Preprocess, Secondary & Tertiary)

4

Figura A.4: Diapositiva 4 de la presentación



5

Figura A.5: Diapositiva 5 de la presentación

Anexo B

Módulos de Microsoft realizados

Nombre y duración

1. Introducción a Power BI. 46 minutos.
2. Introducción a la compilación con Power BI. 40 minutos.
3. Creación y uso de informes de análisis con Power BI. 6 horas y 24 minutos.
4. Explore lo que Power BI puede hacer por usted. 1 hora y 9 minutos.
5. Análisis de datos con Power BI. 1 hora.
6. Descripción del análisis de datos. 23 minutos.
7. Obtención de datos con Power BI Desktop. 1 hora y 15 minutos.
8. Limpieza, transformación y carga de datos en Power BI. 2 horas y 1 minuto.
9. Modelado de datos en Power BI. 1 hora y 7 minutos.
10. Obtención de datos en Power BI. 1 hora y 25 minutos.
11. Preparación de datos para análisis. 3 horas y 26 minutos.
12. Uso de objetos visuales en Power BI. 1 hora y 25 minutos.
13. Exploración de datos en Power BI. 53 minutos.
14. Optimización de un modelo para rendimiento en Power BI. 43 minutos.
15. Modelado de datos en Power BI. 5 horas y 31 minutos.
16. Introducción a la creación de medidas con DAX en Power BI. 2 horas y 40 minutos.
17. Publicación y uso compartido en Power BI. 1 hora y 4 minutos.
18. Diseño de un modelo de datos en Power BI. 2 horas y 8 minutos.
19. Creación de una historia basada en datos con informes de Power BI. 1 hora y 44 minutos.

20. Creación de paneles en Power BI. 1 hora y 8 minutos.
21. Creación de informes paginados. 1 hora
22. Uso de objetos visuales de Power BI. 1 hora y 13 minutos