



Unión Europea

Fondo Europeo
de Desarrollo Regional



GOBIERNO DEL
PRINCIPADO DE ASTURIAS

FICYT
FUNDACION PARA EL FOMENTO EN ASTURIAS
DE LA INVESTIGACION CIENTIFICA APLICADA
Y LA TECNOLOGIA

PCTI
ASTURIAS
Plan de Ciencia, Tecnología e Innovación de Asturias

INFORME TÉCNICO 2017 PROGRAMA ASTURIAS – 2016-2017

Título actuación: Sistema de monitorización
inteligente en procesos de mecanizado SMART
MACHINING (SMASCHING4)

Nº Expediente: IDI/2016/000239

Entidad Beneficiaria:

Fundación PRODINTEC (CIF: G33914557)



ÍNDICE

A INTRODUCCIÓN	3
B CUMPLIMIENTO DE OBJETIVOS	4
C METODOLOGÍA	6
C.1 ACTIVIDADES REALIZADAS	6
D RESULTADOS ALCANZADOS.....	8
D.1 HITO 2: MONITORIZACIÓN DE CONSUMOS	8
D.1.2. Programación de microcontroladores.....	8
D.2 HITO 3: EQUIPO DE MONITORIZACIÓN DE VIBRACIONES Y ULTRASONIDOS	15
D.2.1. Programación de microcontroladores.....	15
D.3 HITO 5: GESTIÓN DE LA INFORMACIÓN	21
D.3.1. Desarrollo del proceso ETL (Extracción, Transformación y Carga de Información)	21
D.3.2. Implementación de cubos de decisión	29
D.3.3. Diseño de experimentos	32
D.3.4. Realización de experimentos. Integración de tablas de decisión y funcionalidades de visualización de información	33
D.3.5. Generación dinámica de informes.....	36
D.4 PRINCIPALES CONCLUSIONES.....	37

A INTRODUCCIÓN

El proyecto SMASCHING4 tenía como objetivo primordial el desarrollo de un **sistema de monitorización inteligente de consumos eléctricos, vibraciones y ultrasonidos de bajo coste** que permita a los **equipos de mecanizado** adaptarse a la cuarta revolución industrial, la “Industria 4.0”, y aumentar su vida útil, su eficiencia energética y su conexión al mundo virtual de las redes.

Según el informe AVIVA, informe que mostraba el estado del sector del mecanizado en España en 2015, las empresas de mecanizado poseen un parque de maquinaria que en su gran mayoría tiene una media de edad superior a 2 años. Esto quiere decir que la mayoría de estos equipos no podrán ser actualizados a la nueva era digital o, en su defecto, que la opción de actualización del fabricante del equipo será demasiado costosa. El proyecto SMASCHING4 trata de cubrir ese hueco mediante la aplicación de electrónica de bajo coste para conectar y monitorizar los equipos actuales e introducirlos en la era del Internet de las cosas (IoT) para optimizar su eficiencia energética, monitorizando los consumos de los diferentes procesos; así como mejorar su vida útil monitorizando las vibraciones y ultrasonidos producidos por las partes móviles y aplicando mantenimientos predictivos para no romper la cadena de fabricación.

El conjunto de sistemas desarrollado dentro del presente proyecto posee las siguientes características, definidas al inicio del mismo:

- Sistema de bajo coste, asequible para los talleres de mecanizado sea cual sea su tamaño y número de máquinas.
- Sistema versátil para adaptarse a los diferentes procesos de mecanizado (torneado, fresado, taladrado, mandrinado, rectificando, cepillado, brochado, etc.).
- Sistema modular que permite la adaptación a las diferentes estructuras y configuraciones de los equipos de mecanizado (número de ejes, husillos, equipos auxiliares, etc.).
- Sistema automático y con capacidad para trabajar sin la supervisión de un operario para evitar generar costes añadidos.
- Sistema con bajo consumo eléctrico.
- Sistema adaptable y reconfigurable para poder ser montado en nuevos equipos o con nuevas capacidades (implementar nuevos ejes en la máquina, aumentar los equipos auxiliares, multiplicadores de revoluciones para el husillo, etc.).
- Posibilidad de volcar datos a la intranet de la empresa o a la nube.

B CUMPLIMIENTO DE OBJETIVOS

Los objetivos técnicos específicos del proyecto eran los siguientes:

- O1. Desarrollo, montaje e implementación de un sistema de **control de consumos eléctricos** para equipos de mecanizado de amplio espectro (tornos, fresadoras, rectificadoras, mandrinadoras, etc.).
- O2. Desarrollo, montaje e implementación de un sistema de **detección de vibraciones y ultrasonidos** para equipos de mecanizado de amplio espectro (tornos, fresadoras, rectificadoras, mandrinadoras, etc.).
- O3. Creación de una **aplicación base web y otra base app** para control y visualización de los sistemas implementados en un parque de maquinaria tipo de un taller de mecanizado.
- O4. Implementación de **algoritmos avanzados** (cubos de decisión y tablas dinámicas) para la obtención y generación de informes que permitan mantener criterios de mantenimiento predictivo de la combinación de los desarrollos del O1 y O2.

Para asegurar una gestión óptima del proyecto y una correcta monitorización del grado de avance se establecieron una serie de indicadores globales del proyecto. En la siguiente tabla se muestran los indicadores definidos para el proyecto, así como el nivel de cumplimiento alcanzado:

Nº	INDICADOR	CUMPLIMIENTO
1	Redacción de los informes técnicos del estado del arte en sistemas de monitorización de consumos, vibraciones y ultrasonidos y listado de normativa aplicable a la seguridad de los equipos de mecanizado	CUMPLIDO. Se han elaborado los distintos entregables indicados en la memoria de solicitud, por una parte el estado del arte de consumo y vibraciones (E1.1) y por otra el listado de normativa aplicable a ambos sistemas (E1.2).
2	Montaje físico del sistema para monitorización de consumos energéticos y su programación intrínseca	CUMPLIDO. Se han adquirido las placas de desarrollo y las pinzas amperimétricas y se han montado en ambos centros de mecanizado los sistemas de monitorización de consumos, los cuales registran y vuelcan los datos al servidor web para su posterior análisis. Durante las tareas de depuración del microcontrolador se detectó un error puntual de recepción de información por parte de la base de datos, que se solucionó con la sustitución de la placa de desarrollo y una nueva programación del microcontrolador y con la integración de una placa de acondicionamiento de señal, dando así por finalizado el desarrollo del equipo de monitorización de consumos energéticos (E2.1).

Nº	INDICADOR	CUMPLIMIENTO
3	Montaje físico del sistema para monitorización de vibraciones y ultrasonidos y su programación intrínseca	CUMPLIDO. Se han adquirido las placas de desarrollo y los grupos de sensores (acelerómetros, galgas extensiométricas, láser y micrófonos) y se han montado en ambos centros de mecanizado los sistemas de monitorización de vibraciones y ultrasonidos, los cuales registran y vuelcan los datos al servidor web para su posterior análisis. Durante las tareas de depuración de los microcontroladores se detectó un error en la medición de ultrasonidos, que se solucionó con la sustitución de la placa de desarrollo y una nueva programación del microcontrolador, dando así por finalizado el desarrollo de los equipos de monitorización de vibraciones y ultrasonidos (E3.1).
4	Implementación del sistema web en servidor (interno o externo)	CUMPLIDO. Por motivos de seguridad, finalmente se optó por implementar el sistema web en un servidor interno.
5	Implementación de la aplicación para dispositivos móviles	CUMPLIDO. En el servidor web se ha implementado una aplicación web para la visualización y configuración de los equipos de monitorización (E4.1), diseñada para su uso en cualquier dispositivo móvil, incluyendo smartphones y tabletas Android (E4.2).
6	Redacción de los informes técnicos sobre estructuración de bases de datos, cargas de cálculo, algoritmos OLAP y tablas dinámicas	CUMPLIDO. Se ha configurado, implementado y validado una base de datos para el registro de los datos obtenidos de los sistemas de monitorización basada en el informe llevado a cabo sobre distintas estructuras de la base de datos y cargas de cálculo (E5.1) y se ha estudiado y seleccionado el software a utilizar para la programación de los cubos de decisión y de las tablas dinámicas (E5.2).
7	Implementación de los algoritmos cubos de decisión y tablas dinámicas para la generación dinámica de informes	CUMPLIDO. Se han programado cubos OLAP y tablas dinámicas para el análisis y visualización de la información recogida en la base de datos y se ha desarrollado el proceso ETL (extracción, transformación y carga de información) correspondiente, así como un script para la generación y envío automáticos de informes (E5.3).

C METODOLOGÍA

Para el desarrollo del proyecto se definieron hitos y tareas clave para la correcta consecución de los objetivos planteados, siguiendo dos vías de trabajo definidas al inicio del proyecto: el desarrollo e implementación del sistema de monitorización de consumos energéticos, colocado en el armario eléctrico de los centros de mecanizado, y el del sistema de monitorización de vibraciones y ultrasonidos, situado dentro de la zona de trabajo de los centros de mecanizado.

Paralelamente a las tareas de desarrollo de los sistemas de monitorización se ha creado e implementado una aplicación de software que permite la visualización de los datos captados y enviados por los sistemas de monitorización. Finalmente, se han implementado algoritmos basados en cubos de decisión y tablas dinámicas de Excel que permiten procesar los datos “en bruto” y conseguir que los mismos entren a formar parte de un almacén de datos de una manera normalizada, excluyendo aquellos valores que pudieran ser debidos a acciones no propias del sistema, así como tramitar las peticiones de información del usuario y generar y mostrar los informes correspondientes.

A continuación, se resumen las acciones realizadas durante la anualidad 2017 agrupadas por tareas.

C.1 Actividades realizadas

HITO 2. Monitorización de consumos

Tarea 2.4. Programación de microcontroladores. Finalizada.

Dentro de esta tarea, durante la anualidad 2016 se llevó a cabo la programación en C++ de los microcontroladores del sistema de monitorización de consumos.

Durante la anualidad 2017 se ha llevado a cabo la depuración de errores para optimizar el rendimiento del sistema. Durante el proceso de depuración se observó que en algunos momentos la base de datos dejaba de recibir información debido a que el excesivo “ruido” ambiental (perturbaciones de tipo electromagnético) propio de un entorno industrial generaba niveles de interferencias demasiado elevados que provocaban fallos puntuales en el software, así como a la mala cobertura, provocada también por el excesivo “ruido” ambiental y por la ubicación del sistema de monitorización en la parte trasera del centro de mecanizado, por lo que entre los sistemas de envío y captación de datos se interponía un armario metálico de gran tamaño que disminuía la potencia de la señal.

Para solucionar ambos problemas, se desarrolló una nueva placa para el control del proceso de toma de datos y almacenaje y envío de los mismos a la base de datos, incluyendo la programación de un nuevo microcontrolador, así como una placa de acondicionamiento de señal para aumentar la potencia de la misma.

HITO 3. Monitorización de vibraciones y ultrasonidos

Tarea 3.4. Programación de microcontroladores. Finalizada.

Dentro de esta tarea, durante la anualidad 2016 se desarrolló la programación tanto del sistema de monitorización de vibraciones como del de ultrasonidos, que se volcó en la memoria de las placas de desarrollo para su ejecución, registro de los parámetros necesarios y envío de dichos parámetros por WiFi al clúster para su almacenamiento en la base de datos.

Durante la anualidad 2017 se ha llevado a cabo la depuración de errores para optimizar el rendimiento del sistema. Durante el proceso de depuración se observó que, si bien el sistema de monitorización de vibraciones funcionaba correctamente, al comparar las mediciones de ultrasonidos realizadas con distintas herramientas y distintos materiales no había apenas variación, es decir, que esas mediciones no se estaban realizando correctamente. Esto era debido a que la frecuencia máxima que podía capturar la placa desarrollada estaba muy por debajo de 5 KHz, que es la frecuencia predominante de la señal captada por los micrófonos ubicados en el centro de mecanizado. Para solucionarlo se llevó a cabo un rediseño del sistema utilizando una plataforma más potente, que se validó mediante la realización de pruebas de captura de ultrasonidos en diferentes situaciones, antes de proceder al desarrollo del programa para la conexión de dicha plataforma con la base de datos.

HITO 5. Gestión de la información

Tarea 5.2. Algoritmos cubos de decisión. Finalizada.

Esta tarea comenzó en la anualidad 2016 con la realización de un estudio teórico sobre el estado actual de desarrollo de los algoritmos OLAP y la selección del software a utilizar para programar los cubos OLAP o de decisión (Mondrian Schema Workbench en este caso).

El desarrollo posterior de un proceso de extracción, transformación y carga de información durante la anualidad 2017 permitió la implementación y configuración de los cubos en las herramientas de software seleccionadas, proporcionando las conexiones necesarias entre la base de datos y la tabla dinámica. Estos cubos constan de tres dimensiones, una para cada tipo de dato recogido durante el proceso de fabricación (consumo, aceleraciones y sonido), que a su vez cuentan con diferentes niveles, lo que permite ordenar estructuradamente la información recogida en la base de datos y facilitar su análisis.

Tarea 5.3. Algoritmos tablas dinámicas. Finalizada.

Esta tarea comenzó en la anualidad 2016 con la realización de un estudio teórico sobre el estado actual del proceso de creación de tablas dinámicas y la selección del software a utilizar su programación (MS Excel en este caso).

Del mismo modo que en la tarea anterior, los trabajos realizados durante la anualidad 2017 orientados al desarrollo de un proceso de extracción, transformación y carga de

información permitió la programación de la tabla dinámica a utilizar dentro de este proyecto para tramitar las peticiones de información del usuario y generar y mostrar los informes correspondientes. La tabla dinámica permite la visualización de los datos de forma sencilla e intuitiva para facilitar la toma de decisiones, sobre todo de aquellas relacionadas con el mantenimiento predictivo por rotura de la herramienta de mecanizado.

Tarea 5.4. Generación dinámica de informes. Finalizada.

Esta tarea se ha ejecutado en su totalidad durante 2017 y ha supuesto la posibilidad de generar automáticamente informes en tiempo real basados en la tabla dinámica desarrollada en la tarea anterior y su envío por correo electrónico.

D RESULTADOS ALCANZADOS

A continuación, se describen los resultados obtenidos en la anualidad 2017 del proyecto en mayor profundidad, indicando los retos tecnológicos alcanzados más relevantes.

D.1 Hito 2: Monitorización de consumos

D.1.2. Programación de microcontroladores

Durante la anualidad 2016 se definieron las variables a monitorizar para determinar el consumo energético del centro de mecanizado. En este sentido, se decidió utilizar pinzas amperimétricas como sensores para la captura de datos, puesto que tienen un coste más bajo que la otra opción considerada (vatímetros), existe un amplio abanico de marcas y rangos de corriente o voltaje y son muy fáciles de instalar y poner en marcha en los centros de mecanizada sin necesidad de parar su funcionamiento (ya que la medición se realiza de manera indirecta). Unos vez calibrados los sensores se abordó el desarrollo del sistema de monitorización de consumos, que, además de las pinzas amperimétricas estaba compuesto por un conjunto de placas: una placa de sensor 4-20 mA para la adquisición de datos de las pinzas amperimétricas y su conversión a un valor numérico, una placa de desarrollo Raspberry Pi con conexión ethernet, WiFi y bluetooth 4.0 y un microcontrolador para el proceso de toma de datos y su almacenaje local y envío a la base de datos. Este sistema se colocó en el interior de los armarios de control de los centros de mecanizado, para lo que se montó en una caja industrial estanca.

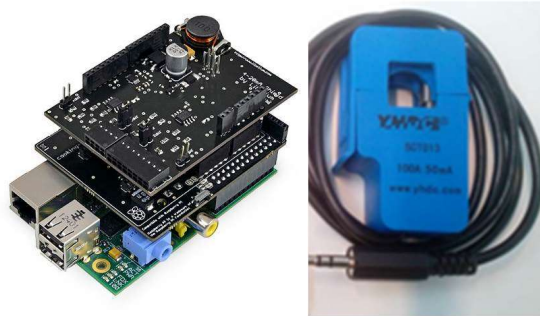


Figura 1. Izquierda: placa de desarrollo Raspberry Pi + placa de sensor 4-20 mA; Derecha: sensor de corriente

Ya en la anualidad 2017 se llevó a cabo la depuración de errores del microcontrolador para validar el sistema completo de monitorización de consumos, durante la que se observó que en ciertos momentos la base de datos dejaba de recibir información. La investigación de este hecho detectó que esto se debía a **dos problemas: fallos puntuales del software y mala cobertura**.

Con respecto al **primer problema**, los **fallos puntuales de software**, sobre todo de los sistemas experimentales, pueden tener relación con el “ruido” ambiente (perturbaciones electromagnéticas). En el entorno industrial, como es este caso, hay que tener en cuenta que los ruidos e interferencias son mucho mayores que en el entorno residencial. Las interferencias entre dispositivos electrónicos son materia de estudio desde hace décadas y han dado lugar a la normativa actual de compatibilidad electromagnética que deben cumplir todos los productos comerciales. En el caso concreto del presente proyecto hay que tener en cuenta que el sistema funciona en las proximidades de una máquina que dispone de seis variadores de velocidad de motores, una de las mayores fuentes de interferencias que se conocen. Además, el sistema en sí, al estar equipado con un procesador ARM, resulta bastante susceptible a este tipo de interferencias, puesto que este tipo de procesadores no están diseñados para trabajar en un ambiente con gran cantidad de perturbaciones electromagnéticas. Al observar las tensiones de alimentación del equipo con un osciloscopio efectivamente aparecieron niveles de interferencias demasiado elevados, que pueden originar fallos puntuales del software.

Se plantearon tres posibles soluciones para resolver estos problemas de compatibilidad electromagnética: modificación de la placa Raspberry, modificación del firmware (código interno del microcontrolador) y utilización de componentes externos de filtrado. Para modificar el circuito de la placa Raspberry se necesitarían los esquemas electrónicos y los archivos de diseño de la placa, a partir de los que se diseñaría un nuevo circuito. Teniendo en cuenta que se trata de un ordenador embebido que funciona con componentes muy pequeños y que emplea señales de alta frecuencia, se estimó que sólo el desarrollo de esta nueva placa duraría más de 9 meses, lo que hacía inviable esta solución. La segunda opción tampoco es posible, ya que no se dispone del firmware, por la que la **solución de añadir componentes externos de filtrado** era en principio la única posible, siempre y cuando el espacio disponible dentro de la carcasa fuera suficiente.

Con respecto al **segundo problema** detectado, el de la **mala cobertura**, se observó que se debía a que el módulo WiFi de la placa Raspberry también es sensible a las interferencias anteriormente mencionadas y a la ubicación del sistema en la parte trasera de la máquina, que hace que entre este y el cluster donde se ubica la base de datos se interponga un armario metálico de gran tamaño que atenúa la señal.

Para resolver este problema se plantearon dos posibles soluciones: colocar el sistema en el interior del armario de la máquina o en la parte superior del armario. En ambos casos, la **solución** pasaría por **añadir nuevo hardware para subir la potencia de la señal**, especialmente en el primer caso, ya que de lo contrario al cerrar la puerta del armario se produciría un efecto tipo “jaula de Faraday”, aislando el equipo del exterior y provocando la pérdida de la señal por completo.

El sistema desarrollado en la anualidad 2016 basado en una placa Raspberry, aunque a priori resulta muy cómodo porque se trabaja con componentes comerciales, presenta el inconveniente de que no se pueden realizar modificaciones de hardware con libertad, ni tampoco actuar sobre el firmware. Por ello, **para solucionar los dos problemas descritos anteriormente** (problemas puntuales de software y mala cobertura) se optó por **desarrollar una nueva placa de desarrollo Arduino y una placa de acondicionamiento de señal diseñada a medida**.

Desarrollo de placa Arduino

La placa de Arduino es un elemento comercial que sí permite modificaciones en el firmware (de hecho, todo el firmware implementado es diseño propio de PRODINTEC). Además, tiene un coste similar a la placa Raspberry desarrollada inicialmente, por lo que la premisa de que el sistema de monitorización tenga un coste lo más bajo posible se sigue cumpliendo. Esta placa se alimenta a 5 V, lo que permite utilizar la misma fuente de alimentación del sistema anterior. Dispone de una serie de pines de entrada y salida digitales mediante los cuales se puede controlar hardware externo y también dispone de pines analógicos con los que se pueden leer señales de tensión procedentes de otro sistema o periférico.

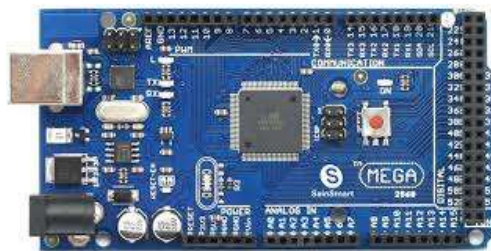


Figura 2. Placa de desarrollo Arduino

Al cambiar a un nuevo microcontrolador es necesario volver a hacer la programación del mismo, si bien se han seguido las mismas premisas que en la programación del microcontrolador realizada en la anualidad 2016. El nuevo programa se ha estructurado según el flujograma de la figura 3, donde cada cuadro representa una función de programa. En él existen dos bloques principales: la función de configuración y el bucle

infinito. La función de configuración se ejecuta una sola vez al comienzo del programa y no se vuelve a ejecutar a menos que se realice un reiniciado del sistema. El bucle infinito se ejecuta a continuación de la función Setup, de manera continua hasta que se apaga el sistema. En la función Setup es donde se realiza la configuración de los distintos elementos del sistema, el más relevante de los cuales es el módulo WiFi, que requiere comunicación al inicio del proceso para ajustar diferentes parámetros de la comunicación. En esta función también se inicializan algunas variables del sistema.

Después de la función Setup se pasa al bucle infinito, el cual realiza de continuo la ejecución de las distintas funciones de lectura de los canales A, B y C y el envío de datos por puerto Serie y por WiFi.

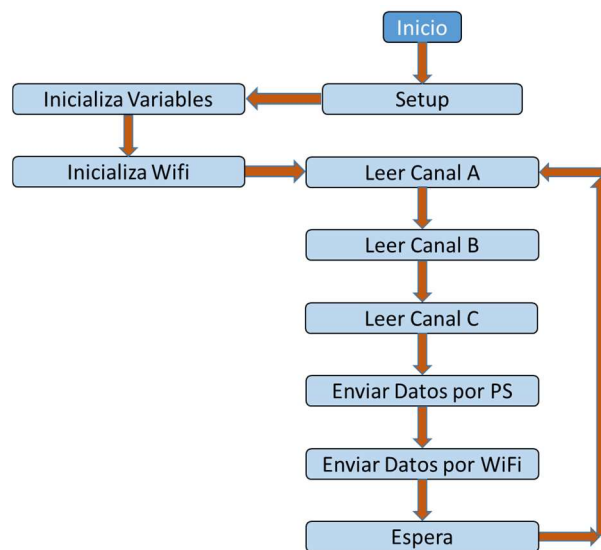


Figura 3. Flujograma del programa de monitorización de consumos

No hay tres funciones diferentes encargadas de la lectura de los canales A, B y C, sino que en realidad existe una única función común para los tres canales, sólo hay que seleccionar el canal que se quiere leer cuando se hace la llamada a la función y esta se encarga de almacenar los datos en tres variables diferentes, una para cada canal.

La complejidad de la lectura de señales analógicas de este proyecto radica en que se está intentando medir una señal que tiene una forma similar a la corriente de red, es decir, se trata de medir una señal senoidal sólo con parte positiva, tal y como se muestra en la figura 4. Esta señal presenta valores muy distintos según el instante en el que se mida, en este caso el objetivo es que el programa devuelva el valor del punto más alto de la señal. Esto se realiza dentro de la función mencionada anteriormente, tomando una serie de medidas periódicas y comparando unas con otras hasta encontrar el valor máximo. Una vez que se ha encontrado, este se almacena en una variable. Este proceso se repite 10 veces y finalmente se hace la media de todos los valores máximos.

La realización de esta media es una forma de suavizar la medida, o dicho de otro, de filtrar el ruido presente en el sistema. Como puede verse en la figura 4, además de una forma senoidal (parte positiva) aparecen una serie de picos que corresponden con el

ruido presente, por lo que, de no filtrarse, ocasionarían la aparición de pequeños errores de medida.

Finalmente, una vez que se tienen los datos de consumo almacenados en las variables de programa correspondientes, se procede a su envío por WiFi ejecutando la función del mismo nombre mostrada en la figura 5. En esta función hay dos instrucciones importantes que conviene diferenciar: *serial1.println* y *serial.println*. La primera sirve para comunicar con el módulo WiFi y la segunda para enviar un dato por el puerto serie que se recibiría en un PC que estuviese conectado al Arduino. Los datos que se envían al PC son únicamente para el diagnóstico del sistema.

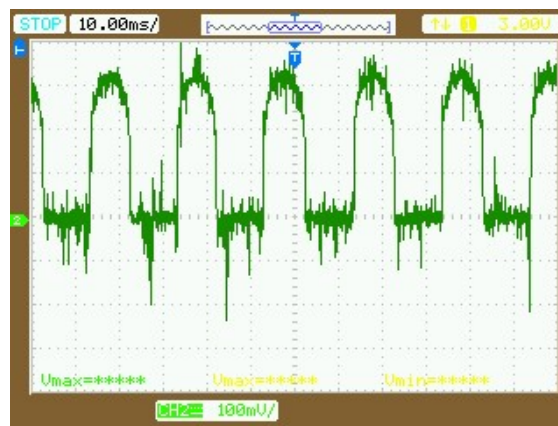


Figura 4. Señal analógica correspondiente a la medida de corriente

```

1. void Enviar()
2. {
3.   Serial.println("Enviando...");
4.   Serial1.println("AT+CIPSTART=0,\"TCP\", \"192.168.30.203\", 80
5. );
6.   delay(500);
7.   Serial1.println("AT+CIPSEND=0,115");
8.   delay(500);
9.   Serial1.print("GET /consumos_deckel.php?&canal1=");
10.  Serial1.print(P_0);
11.  Serial1.print("&canal2=");
12.  Serial1.print(P_1);
13.  Serial1.print("&canal3=");
14.  Serial1.print(P_2);
15.  Serial1.print("&canal4=");
16.  Serial1.print(P_3);
17.  Serial1.println("
18. HTTP/1.0\r\nHost:192.168.30.203\r\n\r\n");
19.  delay(500);
20.  Serial.println("Datos recibidos.");
  
```

Figura 5. Código de programación de la función enviar por WiFi

Desarrollo de placa de acondicionamiento de señal

Para aprovechar los sensores de corriente de la solución desarrollada en 2016 se diseñó una placa de acondicionamiento de señal. Estos sensores proporcionan una señal, proporcional a la corriente que se está midiendo pero mucho más pequeña. Puesto que el Arduino sólo puede leer señales de tensión analógicas, la etapa de acondicionamiento debe primeramente traducir las señales de los sensores a valores de tensión, para lo que se emplea una resistencia (R25) en paralelo con el sensor de corriente. Seguidamente esta señal ha de rectificarse, es decir quedarse sólo con su parte positiva. Para realizar el rectificado no puede emplearse un simple diodo, como ocurre en los rectificadores de los sistemas de alimentación, puesto que al tratarse de una señal muy pequeña en corriente y considerando que los diodos no son componentes ideales y tienen una pequeña caída de tensión, no es posible asegurar que al utilizarlos el sistema funcione. Al ser estar tratando con una señal tan pequeña se debe utilizar el rectificador de precisión de la figura 6 el cual se complementa con un amplificador para subir el nivel de la señal de salida.

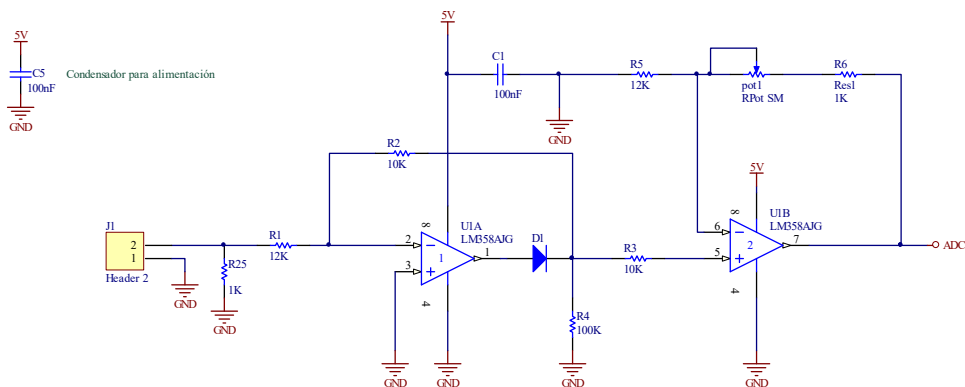


Figura 6. Esquema de la etapa de acondicionamiento de señal

En la placa desarrollada se han añadido otros tres sistemas como el del esquema anterior, para poder medir en las tres fases de la máquina.

Para evitar volver a tener problemas de compatibilidad electromagnética, se han seguido una serie de reglas o buenas prácticas en el diseño de esta nueva placa: primero se han añadido condensadores de 100 nF en las alimentaciones de los amplificadores operacionales y después se han añadido planos de masa formados por el cobre sobrante que no se utiliza para conectar ningún componente, que aporta una mejor inmunidad frente al ruido. También se han considerado los puntos de conexión con la placa Arduino, puesto que aparte de los pines analógicos es necesario añadir alimentación y masa, los cuales se han conectado a un único punto para el caso de alimentación 5 V y a dos puntos en el caso de masa, buscando con esto también la mejora de la inmunidad.

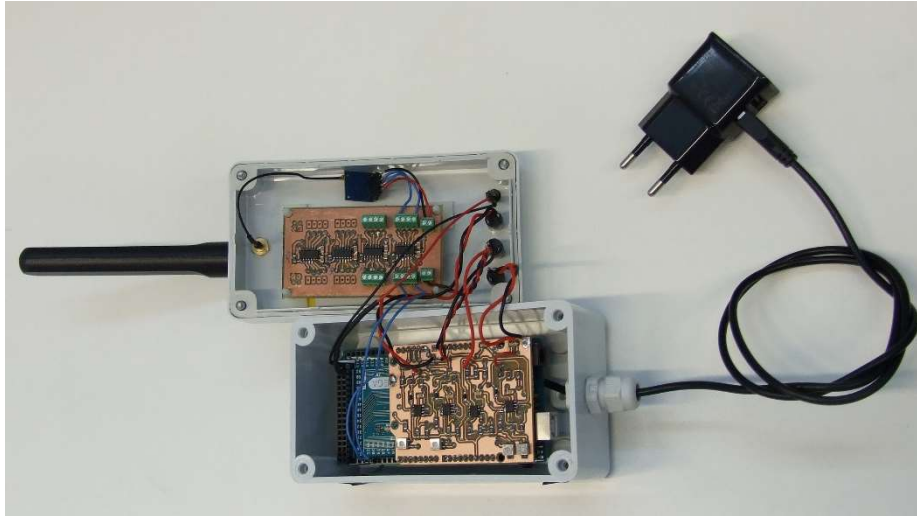


Figura 7. Montaje final del sistema de medida de consumos para máquina Deckel

El montaje final del sistema completo puede verse en la figura 8, donde se observa que se ha aprovechado la tapa de la carcasa para colocar más elementos como son el módulo WiFi, una placa adicional para traducir niveles de tensión, la antena y los conectores de los sensores. En este nuevo diseño se ha incluido un módulo WiFi externo que se comunica con la placa Arduino mediante los pines digitales. Se trata además de una placa comercial independiente del circuito, por lo que puede ser sustituida en caso de que presente algún problema. Este módulo WiFi es el modelo ESP, el cual dispone de conector para colocar una antena externa, aspecto que lo hace idóneo para ganar flexibilidad en el montaje. Un pequeño inconveniente que presentan todos los módulos WiFi comerciales es que trabajan a una tensión de 3,3 V, a diferencia de los 5 V del Arduino y de la placa de acondicionamiento de señal, por lo que ha sido necesario añadir un circuito que actúa de traductor de niveles de 5 V a 3,3 V, que es la placa que ocupa casi toda la superficie de la tapa. Esta placa se basa en un integrado TXB de Texas Instruments. Finalmente, a esta carcasa se le han añadido varios imanes para que pueda ser colocada en la superficie de la puerta del armario metálico de la máquina Deckel, así como un enchufe para alimentar el sistema.

El nuevo sistema de monitorización de consumos así desarrollado se validó mediante la realización de mediciones durante una jornada laboral completa. Una vez verificado que la toma de datos era correcta y que el software no tenía fallos puntuales, se dio por validado el correcto funcionamiento del nuevo sistema de monitorización de consumos y por solucionados los problemas detectados en la versión anterior, por lo que se llevó a cabo el desarrollo de un sistema semejante para la máquina Kern. Este sistema es prácticamente idéntico que el de la máquina Deckel, con la premisa adicional de reducir el cableado y simplificar el montaje para optimizar así el diseño del sistema. Para conseguir esto se procedió a la eliminación de los componentes de la tapa de la carcasa, salvo los conectores de los sensores.

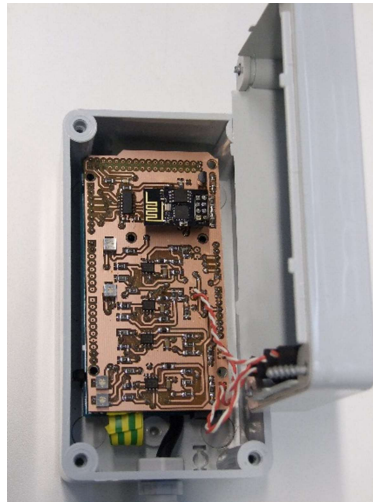


Figura 8. Montaje final del sistema de medida de consumos para máquina Kern

D.2 Hito 3: Equipo de monitorización de vibraciones y ultrasonidos

D.2.1. Programación de microcontroladores

Durante la anualidad 2016 se realizó un estudio de las zonas de las máquinas que generan vibraciones. Estas vibraciones son normales dentro de un rango establecido pero toman valores altos a medida que las partes móviles de la máquina adquieren holguras, desalineamientos o excentricidad antes de un posible fallo de la máquina o antes de la rotura de la herramienta por desgaste o la aparición de chatter (vibración autoexcitada que se genera entre los filos de la herramienta y la superficie del material ya mecanizado). También se seleccionaron los sensores, acelerómetros de tres ejes para la detección de vibraciones y micrófonos para la detección de ultrasonidos, buscando siempre las mejores prestaciones con el menor coste posible. Además, se realizó un primer mapa de sonido de los equipos Deckel y Kern para poder compararlos con futuras mediciones y conocer cómo evolucionan a medida que las máquinas se van deteriorando por el uso. A continuación se desarrollaron los sistemas. El sistema de monitorización de vibraciones está integrado en la propia herramienta de mecanizado y esá compuesto por una placa Wemos la cual se ha programado en entorno Arduino para enviar al servidor periódicamente los valores recogidos por el acelerómetro en sus tres ejes, que se almacenan en la base de datos y se analizan para determinar la presencia o no de frecuencias autoexcitadas que puedan indicar la presencia de chatter. El sistema de monitorización de ultrasonidos se ha desarrollado sobre la plataforma Wasmote de la casa Libelium con un entorno de programación similar a Arduino. El microcontrolador de la placa obtiene el espectro de sonido discretizado en una serie de frecuencias y amplitudes, forma un paquete de unas 500 muestras y la envía al servidor, que es el encargado de analizar las frecuencias presentes y la amplitud de cada una de ellas, almacenarlas en la base de datos y compararlas con los patrones ya existentes.

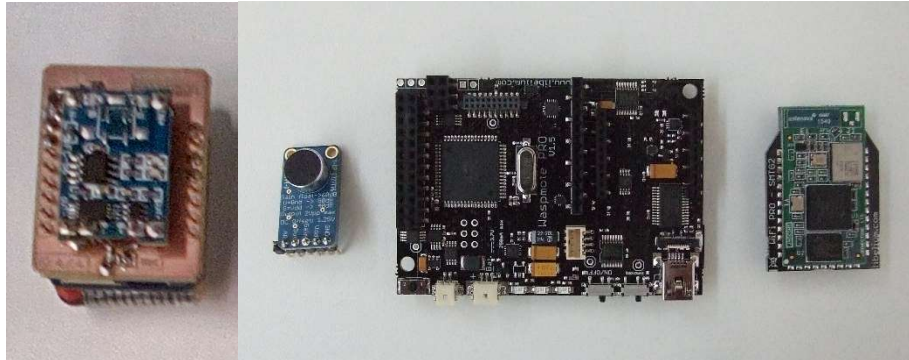


Figura 9. Izquierda: sistema de monitorización de vibraciones; Derecha: sistema de medición de ultrasonidos

Ya en la anualidad 2017 se llevó a cabo la depuración de errores de los microcontroladores para validar los sistemas completos de vibraciones y ultrasonidos. Si bien el primero superó las pruebas de validación sin necesidad de realizar modificaciones, en el sistema de medición de ultrasonidos se detectó que apenas existía variación al comparar las mediciones realizadas con distintas herramientas y distinto material, lo que indicaba un problema de funcionamiento.

Para detectar la causa del problema, en primer lugar se analizaron las señales procedentes del micrófono. En la figura 10 se observa una medida de la forma de onda (amarillo) y de la transformada rápida de Fourier (FTT, rosa). Esta última es una herramienta de software de la que disponen muchos osciloscopios digitales y que permite obtener el espectro en frecuencia de la señal que se está midiendo. En este caso, la FTT indicaba que la frecuencia predominante de esta señal se encontraba ligeramente por encima de los 5,5 KHz, valor que a priori parece algo elevado.

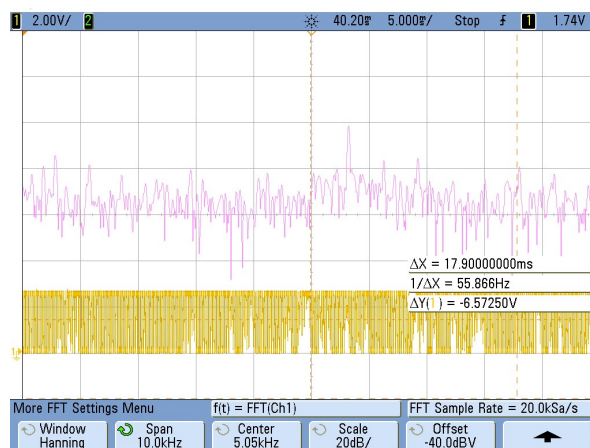


Figura 7. Cálculo de la FFT del micrófono mediante osciloscopio

Puesto que no se contaba con información de la frecuencia máxima que puede medir la placa Wasp mote, se diseñó y realizó un experimento para obtener este dato.

El experimento consistió en capturar señales de diferente frecuencia con la Wasp mote y guardar los datos de estas en la tarjeta de memoria, para a continuación extraerlos y

representarlos en una hoja Excel. Estas señales de frecuencia variable se consiguen con un generador de funciones, un equipo que puede generar diferentes señales (senoidales, triangulares, cuadradas, etc.) con frecuencia mínima de 1 Hz y máxima (para el modelo disponible en PRODINTEC) de 10 Mhz. Con una señal de entrada senoidal de 100 Hz se obtuvo la forma de onda de la figura 11, es decir, una reconstrucción aceptable. La siguiente señal que se probó fue de 1 KHz y el resultado obtenido fue el de la figura 12, donde puede verse que la forma ya no es senoidal, sino que está deformada, por lo que es una mala reconstrucción.

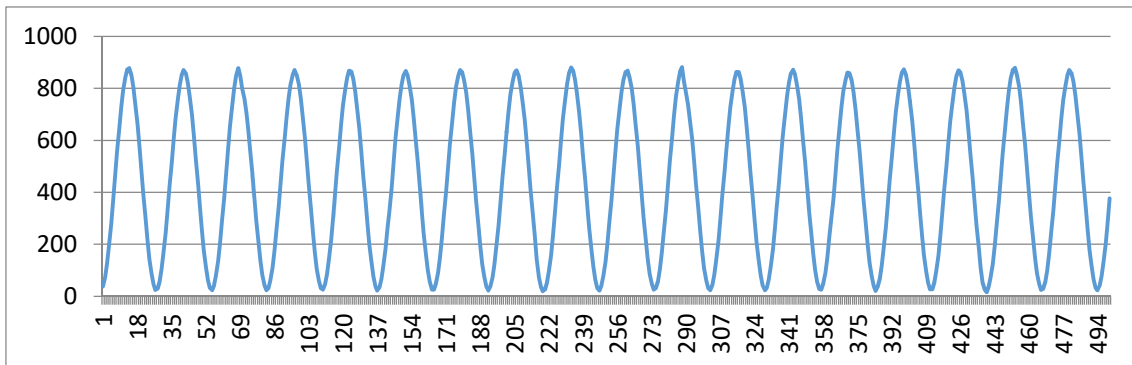


Figura 81. Reconstrucción de la señal de 100 Hz

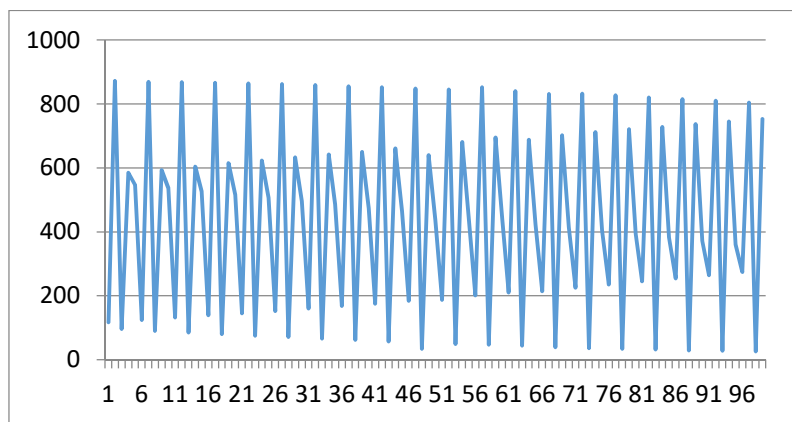


Figura 9. Reconstrucción de la señal de 1 KHz

La conclusión obtenida del experimento realizado es que la frecuencia máxima que puede capturar la placa Waspnote está muy por debajo de los 5 KHz de la figura 10, puesto que ya presentaba problemas para la captación correcta de frecuencias de 1 KHz, por lo que para continuar con la medida de ultrasonidos era necesario realizar un rediseño del sistema utilizando una plataforma más potente. La plataforma escogida fue el módulo DAQ multifunción para USB del fabricante National Instruments, puesto que, según sus especificaciones técnicas, es capaz de capturar señales con una frecuencia de hasta 450 KHz. Además, es un sistema comercial con marcado CE, es decir, con las pruebas de compatibilidad electromagnética superadas y por tanto con una robustez apta para el sector industrial. Cuenta con una serie de entradas y salidas digitales y

analógicas que se pueden programar desde el software Labview. Una vez que la programación está lista se genera un archivo ejecutable que puede funcionar en cualquier PC sin necesidad de tener instalado el Labview. El programa desarrollado cuenta con una pantalla inicial en la que aparecen tres opciones, capturar señales, comparar señales o salir de la aplicación, tal y como se muestra en la figura 13.



Figura 10. Izquierda: módulo DAQ; Derecha: pantalla inicial

El código escrito en Labview se basa en un lenguaje gráfico con un estilo de programación intuitivo basado en la relación entre bloques, en el que para escribir un determinado programa únicamente se deben seleccionar los bloques necesarios y conectarlos entre sí. Para el programa de captura de ultrasonidos se han seleccionado tres bloques de programa principales, cuyos esquemas se muestran en la figura 14, que realizan diferentes tratamientos de la señal:

- **Spectral Measurement:** se encarga de la transformada de Fourier de los datos que provienen de la plataforma DAQ NI.
- **Sample Compression:** realiza la compresión de la señal de audio de forma similar a los formatos comerciales tipo MP3 para reducir el tamaño de los datos sin afectar de manera significativa a la calidad del audio.
- **Amplitude and Level Measurements:** realiza diversas medidas relacionadas con la amplitud de la señal, en este caso devuelve el valor pico a pico, es decir desde el extremo más bajo al extremo más alto.

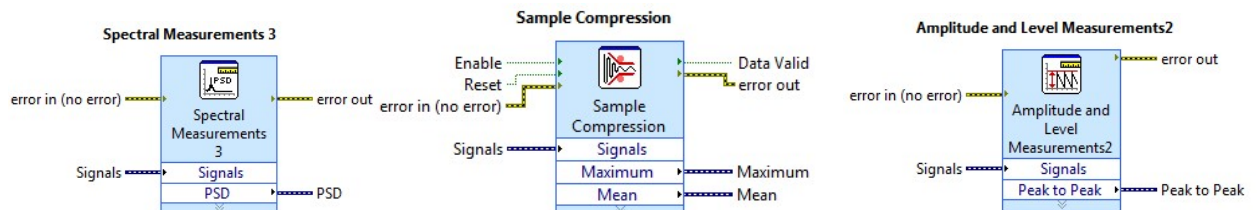


Figura 11. Bloques del programa Labview

Con el programa obtenido al conectar estos bloques se realizaron varias capturas de audio durante el mecanizado utilizando diferentes herramientas de corte, obteniendo,

esta vez sí, espectros de audio diferentes, a diferencia de lo que ocurría con el sistema desarrollado en 2016. En primer lugar se realizó una captura del ruido con la máquina parada, es decir, ruido en vacío (figura 15). Seguidamente se hicieron capturas durante el mecanizado con una herramienta plana de 4 haciendo agujeros y ranuras. Al analizar estas gráficas (figuras 16 y 17) se observaron diferencias notables en el espectro, aunque la forma global tiene cierta similitud al tratarse de la misma herramienta. En el caso de la herramienta plana de 10 sí que tiene una curva diferente al resto (figura 18).

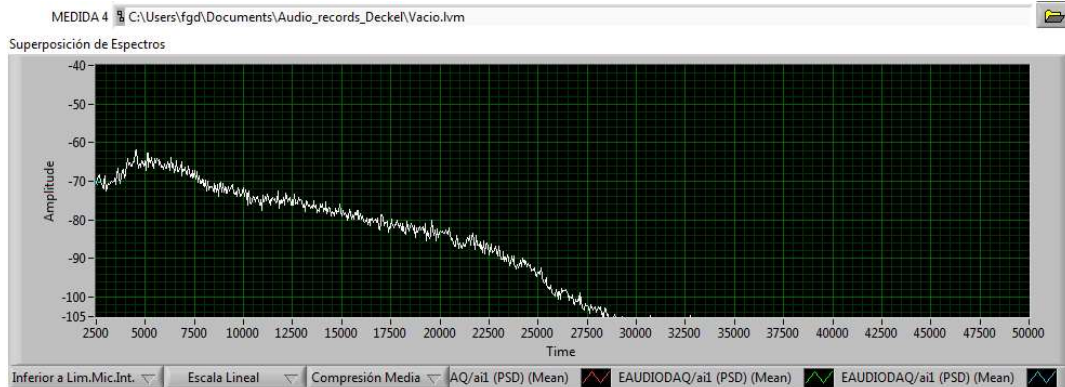


Figura 12. Ruido en vacío

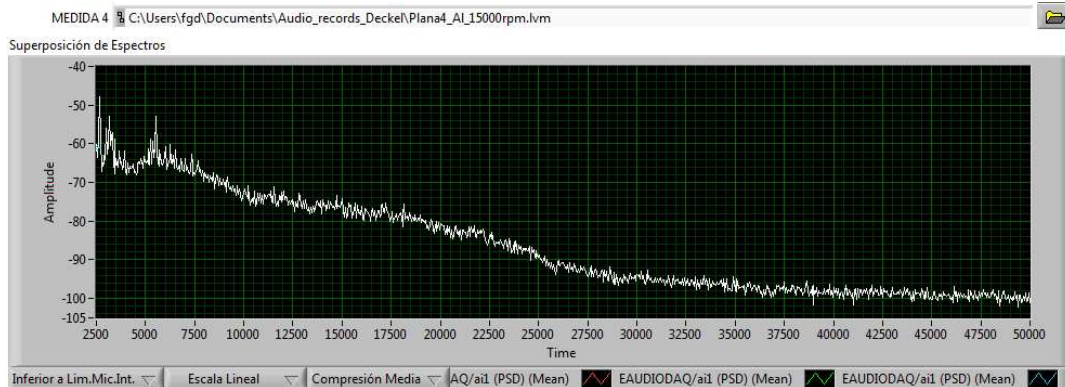


Figura 13. Herramienta plana de 4 haciendo ranuras

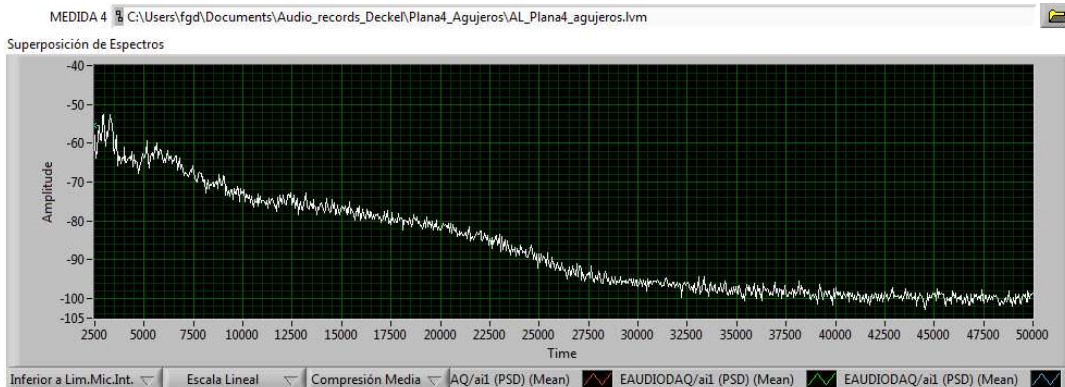


Figura 14. Herramienta plana de 4 haciendo agujeros

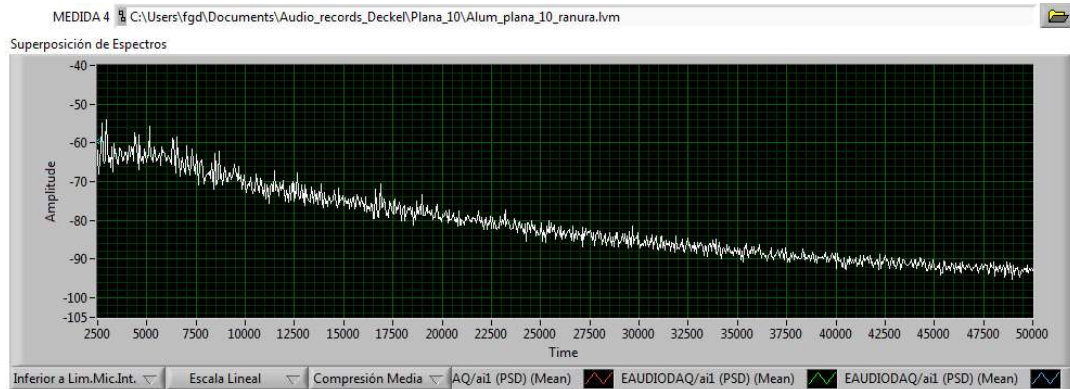


Figura 15. Herramienta plana de 10 haciendo ranuras

Una vez verificada la correcta medición de los ultrasonidos, se instaló un servidor ODBC para conectar Labview con la base de datos existente, configurando en el servidor el origen de datos y la conexión (figura 19).

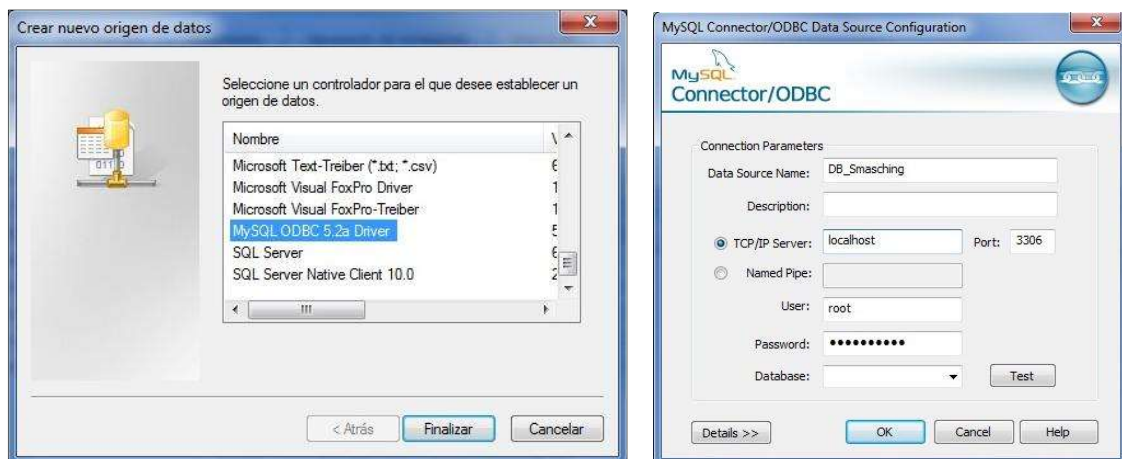


Figura 16. Izquierda: origen de datos; Derecha: conexión con MySQL

Una vez realizada la configuración de la conexión, se desarrolló el programa para la conexión con la base de datos, también utilizando Labview, para lo que se seleccionaron cuatro bloques de programa principales, que a continuación se conectaron, y se establecieron dos elementos de control en los que se especifica a qué base de datos se debe conectar el programa y los datos que debe introducir en ella, con lo que se dio por finalizado el rediseño del sistema de monitorización de ultrasonidos.

D.3 Hito 5: Gestión de la información

D.3.1. Desarrollo del proceso ETL (Extracción, Transformación y Carga de Información)

Diseño básico del proceso ETL

De manera complementaria al desarrollo del esquema de la base de datos que almacena la información gestionada por el hardware desarrollado, durante la anualidad 2016 se llevó a cabo un estudio teórico sobre los cubos OLAP y las tablas dinámicas, que consistió en el análisis del tipo de software existente en el mercado y en la localización de software gratuito Open Source para poder implementarlo en el cluster de placas Raspberry donde se ha configurado la base de datos y manejar los datos contenidos en la misma. Los cubos OLAP también requerían de un motor para hacer la base de datos objeto-relacional (y así almacenar objetos más complejos en una sola tabla con referencias a otras relaciones), otro motor que recibiera las solicitudes de información, realizara las consultas contra la base de datos y devolviera la información en formato multidimensional, así como una interfaz para la solicitud de información por parte del usuario y la generación y visualización de los informes (figura 20).



Figura 17 Diagrama de bloques de conexión a MySQL

Tomando lo anteriormente descrito como base para la continuación del proyecto, durante la anualidad 2017 se trabajó de manera fundamental en la implementación de la base de datos, los cubos y las tablas dinámicas en un sistema que permitiera gestionar toda la información de manera eficiente. Para ello se acometió una labor de diseño y desarrollo de una arquitectura tipo BI (Business Intelligence), capaz de llevar a cabo todo el proceso de extracción, transformación y carga de información, comúnmente denominado proceso ETL (siglas en inglés de Extraction, Transformation and Loading).

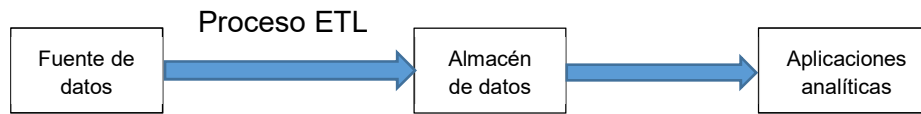


Figura 18. Funcionamiento de un sistema BI

Con el fin de resumir el proceso desarrollado, indicar que en primer lugar se dispone de las fuentes de datos, que es donde se encuentra la información en un estado más “primitivo”, procedente del hardware de adquisición o de medida, que la vuelca directamente en una base de datos sin realizar ningún tipo de tratamiento. En los sistemas de BI convencionales esta información inicial puede proceder de encuestas u otros sistemas de captación de datos, donde el error humano puede ser un factor a considerar, por lo que es necesario incluir un bloque de tratamiento de la información. Dentro de las casuísticas contempladas en el presente proyecto, se dispone de un sistema de medida semiautomatizado, que depende por una parte del factor humano pero también de diversos imprevistos que puedan surgir (como por ejemplo que el sistema de medida no puede controlar en qué momento el usuario detuvo la maquinaria para realizar un control sobre la pieza ni tampoco la desconexión accidental de algún sensor). Este tipo de imprevistos deben ser excluidos del análisis para evitar llegar a conclusiones erróneas. Esta exclusión se incluye dentro del proceso ETL desarrollado que, de manera básica, supone plantear las operaciones a realizar sobre los datos captados, a fin de lograr su adecuada extracción, transformación y carga en el almacén de datos. Esto es, el proceso ETL lo que busca es procesar los datos “en bruto” y conseguir que los mismos entren a formar parte del almacén de datos de una manera que podríamos denominar “normalizada” y con capacidad de excluir aquellos valores que pudieran ser debidos a acciones no propias del sistema.

La diferencia entre la fuente y el almacén de datos, aparte de que ahora la información está tratada, es que es posible acceder a ella de una manera más ágil. Es por ello que la fuente de datos se asocia a un sistema operacional que podría ser un servidor remoto y el almacén de datos a un servidor local. En el caso de este proyecto, el sistema operacional donde se almacena la base de datos sería el clúster ubicado en la zona de las máquinas de mecanizado y el servidor local sería un PC en la zona de oficinas. El sistema operacional realiza continuas escrituras en la base de datos según esté disponible la información, a diferencia del servidor local, que realiza consultas puntuales a la base de datos en momentos en los que se necesita actualizar la información. Aparte de lo comentado, la transformación de los datos también resulta relevante cuando intervienen en el proceso diferentes usuarios, facilitando la comprensión de aquello que se está analizando. Por ejemplo, un usuario experto en mecanizado sabrá lo que es una plana de 10 pero un experto en BI posiblemente no, por lo que al tratar estos datos se pueden incluir comentarios en los campos de datos que identifiquen este objeto como *Fresa Plana de 10*, o *datos sobre Fresas*.

En cuanto la arquitectura del almacén de datos existen dos modelos teóricos llamados aproximación de Kimball¹ y aproximación de Immon². Ambos coinciden en establecer el almacén de datos como el repositorio de información de la empresa, pero cómo organizar la información dentro del almacén de datos y cómo la información entra en este, es donde estriban las diferencias entre las dos aproximaciones. Para Immon, los sistemas operacionales escriben toda la información en el almacén de datos, este subdivide la información en pequeños mercados de datos, que son bloques de datos que realizan un determinado análisis (Finanzas, Departamento de Compras, Inventario, etc.) y son las aplicaciones analíticas las que se comunican directamente con los mercados de datos y no con el almacén de datos. En el caso de Kimball, la información cuando entra en el almacén de datos ya viene dividida en distintos bloques y es dentro del almacén de datos donde se crean los mercados de datos. Si bien para el usuario final no hay apenas diferencia, la elección entre una aproximación y otra influye en la complejidad de la implementación y en las velocidades de acceso. La aproximación más extendida es la de Kimball; teniendo en cuenta además que en nuestro caso contamos con tres sistemas independientes que se encargan de obtener la información, se ha optado por esta por considerarla idónea.

La estructura dimensional dentro del almacén de datos comprende el uso de tablas de hechos y tablas de dimensión, relacionadas también con la arquitectura empleada. Las tablas de hechos contienen información de las áreas de análisis (típicamente productos, clientes, zonas geográficas...), pero en nuestro caso están compuestas por Herramienta, Número de Prueba y Usuario. Las tablas de dimensión contienen las mediciones, que son los valores que se quiere analizar.

¹ Ralph Kimball PhD, Stanford University, Kimball Group, "The Data Warehouse Toolkit"

² William Immon, Yale University, "Bulding the Data Warehouse"

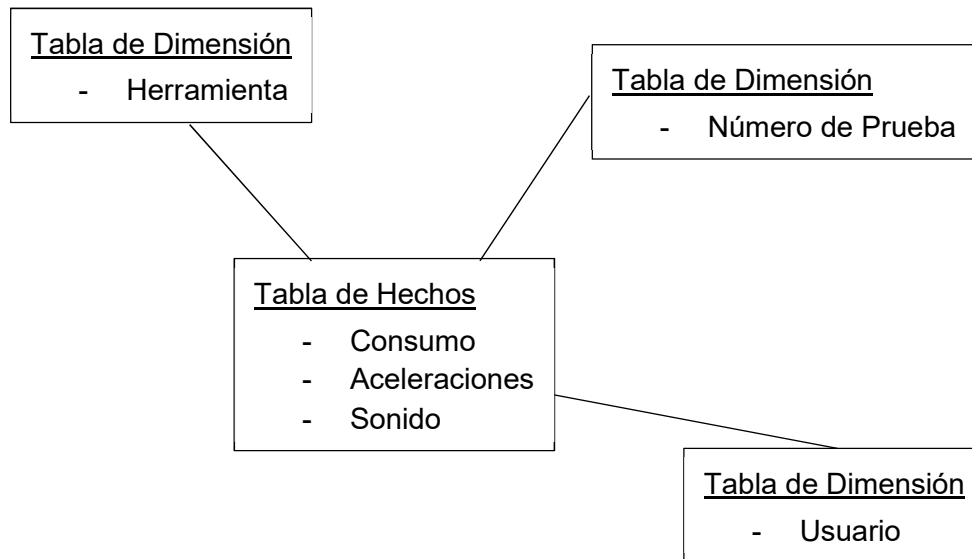


Figura 19. Estructura dimensional

La estructura dimensional empleada corresponde al tipo estrella donde una única dimensión cubre todas las áreas de interés de un tipo: una tabla donde están todos los usuarios, una para todas las herramientas y una para el número de prueba. Esta estructura se utiliza, como veremos posteriormente, a la hora de realizar la transformación de la información, pero también se tuvo en cuenta cuando se realizó la creación de los cubos OLAP.

Selección de software para el tratamiento de la información.

En lo que respecta al grupo de las aplicaciones analíticas referidas en la Figura 18, tenemos diversas herramientas como son los cubos OLAP, las tablas dinámicas y los informes.

Durante la anualidad 2016 se llevó a cabo un estudio teórico sobre los cubos OLAP y las tablas dinámicas, que consistió en el análisis del tipo de software existente en el mercado y en la localización de software adecuando para poder implementarlo en el cluster de placas Raspberry donde se ha configurado la base de datos y manejar los datos contenidos en la misma.

Las tablas dinámicas son otro tipo de herramientas disponibles para poder visualizar información recogida en el almacén de datos, es decir, harían la función de interfaz con el usuario final junto con los informes.

A la hora de implementar los desarrollos anteriores se llevó a cabo una labor de análisis de softwares de BI, lo que nos llevó a decantarnos por la herramienta Pentaho BI, puesto que agrupa una serie de aplicaciones que dan respuesta a las necesidades identificadas para el presente proyecto:

- Data Integration (PDI): preparación e integración de datos
- Mondrian Schema Workbench: motor de gestión de sistemas OLAP
- Report Designer: generación de informes.

La primera aplicación es la más potente, ya que permite realizar todas las funciones ETL, así como el análisis de datos, la toma de decisiones automatizada y la generación de tablas dinámicas y gráficos. Mondrian Schema Workbench permite el diseño de los cubos OLAP y Report Designer permite la creación de una plantilla de informe y la posterior escritura de los datos en esa plantilla. Desde PDI es posible lanzar también la generación de informes como parte del algoritmo ETL pero en este caso se ha realizado directamente con Report Designer con la utilización de una plantilla desarrollada a medida para tal fin.

Diseño detallado del proceso ETL

Tal y como se ha mencionado anteriormente, la implementación del proceso ETL descrito se apoyó en la herramienta Pentaho BI. Siguiendo el esquema de la Figura 18, esta herramienta se encargó primeramente de conectar con la base de datos ubicada en el clúster, tras lo cual se realizaron las transformaciones pertinentes y al final se volcó la información en una hoja de Excel. Las aplicaciones analíticas para este proyecto han sido cuatro: Microsoft Excel, Cubos OLAP, Tablas Dinámicas e Informes. En lo que respecta al proceso ETL en sí, se definieron los flujogramas para la carga de la tabla de hechos compuesta por los tipos de herramientas, el número de prueba y los usuarios y la tablas de dimensiones formadas por las medidas que se han realizado (Figura 20 y Figura 21, respectivamente), todo ello acorde con lo recogido en la Figura 19. Estos flujogramas se denominan en Pentaho BI como trabajos, y dentro de cada trabajo hay varios bloques que vienen a representar las transformaciones comentadas anteriormente.



Figura 20. Carga de la tabla de hechos



Figura 21. Carga de las tablas de dimensiones

Si abrimos las transformaciones de la tabla de hechos, veremos en cada una un flujograma similar al de la Figura 22, con la diferencia de que cada uno lee posiciones distintas de la base de datos. Cada flujograma se compone de 3 bloques: el primero, entrada tabla, es el que realiza la lectura de la información. Introduciendo la dirección, usuario y contraseña de la Raspberry que almacena la base de datos en el clúster, es posible extraer los datos de un determinado campo almacenados hasta el momento. El segundo bloque se utiliza para renombrar los campos existentes de forma que sean más entendibles, para el caso de las herramientas se cambia el texto *Herr* por *Herramientas de fresado*. El tercer bloque conecta la información con las tablas de Excel.

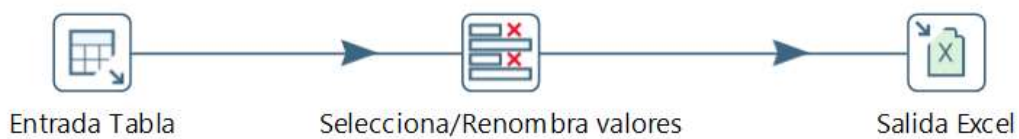


Figura 22. Flujograma de las transformaciones “Cargar herramienta”, “Cargar número de prueba” y “Cargar usuario”

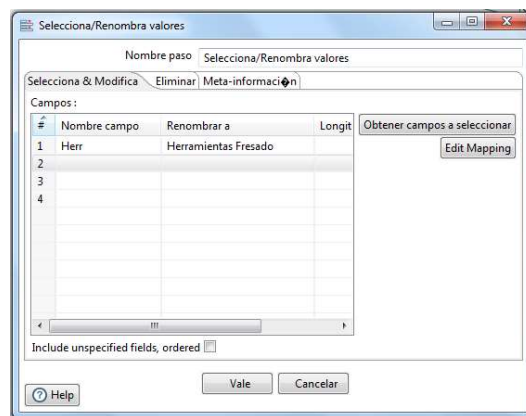


Figura 23. Renombrar campo herramientas

Las transformaciones realizadas sobre las tablas de dimensiones son un poco más complejas ya que, en este caso, sí están presentes varios algoritmos de toma de decisiones. En el caso de consumos, estos algoritmos descartan todos aquellos valores que estén por debajo de 100 W, ya que este es el consumo mínimo que se ha medido en la máquina, por lo que cualquier valor inferior es un dato erróneo. Para ello se emplea la instrucción *number range entre 100 y 500*, que son los valores típicos de consumo de cualquiera de las herramientas analizadas, identificados a partir de las mediciones realizadas. Los consumos comprendidos dentro de este rango pasan al siguiente bloque de información, las tablas de Excel. En el caso de detectar un consumo anómalo, este será reflejado en la misma tabla de Excel previo paso por la instrucción *Seleccionar Renombrar Error Consumo*, con la que se indicará que este valor resulta demasiado alto.

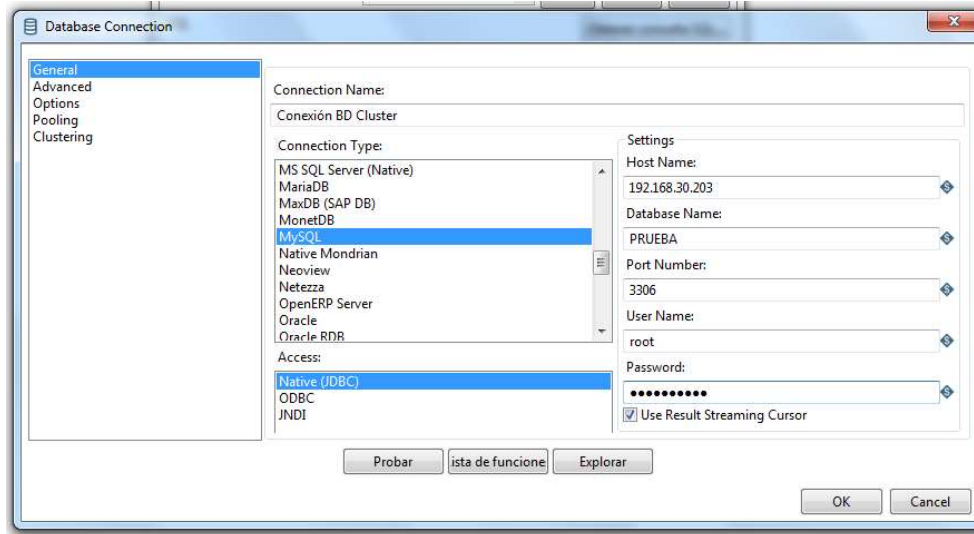


Figura 24. Conexión con la BD

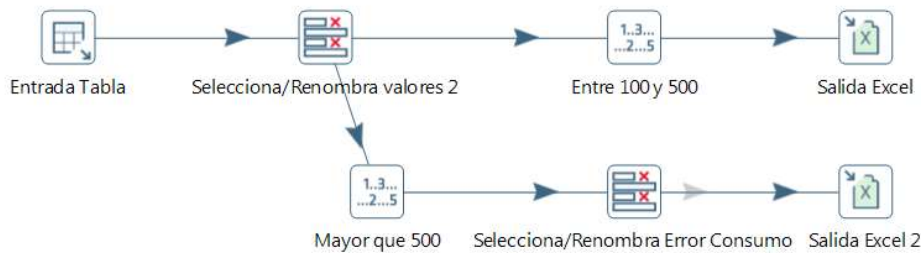


Figura 25. Transformación Consumos

Para la transformación de sonido en primer lugar se filtran aquellos datos que estén fuera del rango 1.000-10.000 Hz, al cual pertenecen los ruidos procedentes de las herramientas. A continuación se separa la información de las distintas herramientas, ya que cada una tiene un pico de sonido a una frecuencia característica: si este pico se desplaza fuera de un rango, sabemos que esta herramienta presenta problemas. Los rangos predefinidos de funcionamiento normal para cada herramienta, identificados en las mediciones realizadas, son:

- Herramienta plana de 2: entre 5.500 y 5.700 Hz
- Herramienta plana de 4: entre 5.200 y 5.500 Hz
- Herramienta plana de 10: entre 5.000 y 5.200Hz
- Herramienta plana 40: entre 4.000 y 5.000 Hz

Si el sonido procedente de una de estas herramientas queda fuera del rango mencionado quedará reflejado en el archivo Excel de salida.

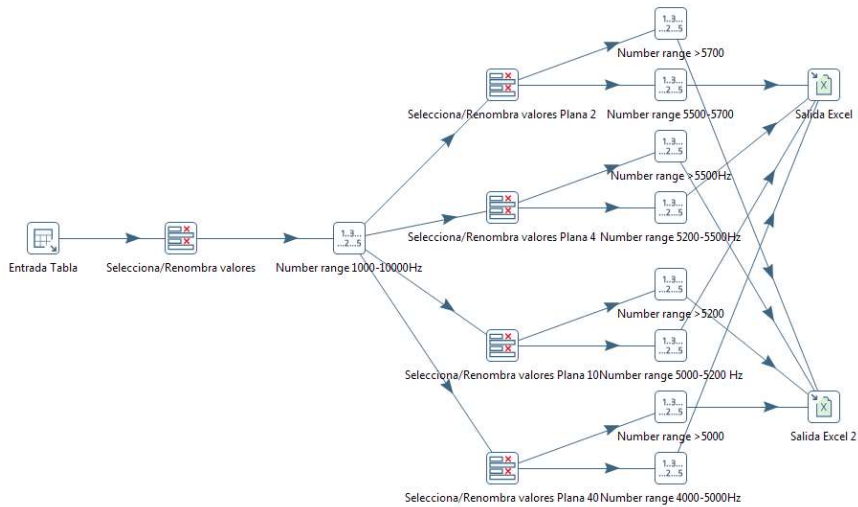


Figura 26. Transformación sonido

Finalmente, la conexión entre el almacén de datos y las aplicaciones analíticas se realizó mediante Excel por diversos motivos: por una parte hay que considerar que se tienen varios bloques y componentes de software que no pueden verse como un todo, puesto que, siguiendo la filosofía de modularidad que se emplea en programación, resulta muy cómodo poder observar lo que están realizando los bloques por separado para que en caso de que se produzca algún error poder entender en cuál de los bloques está el problema.

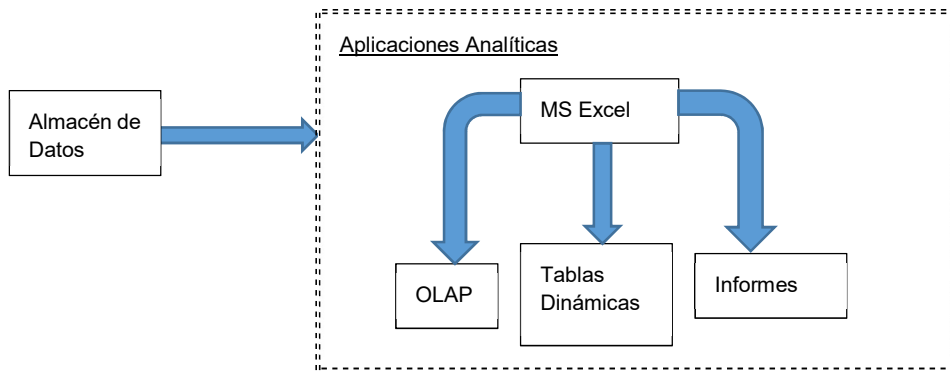


Figura 27. Escritura en las aplicaciones analíticas

En este caso, la herramienta nueva es el Pentaho BI, así que para saber con exactitud qué está haciendo esta, la salida de datos se realiza mediante Excel. Además esto hace posible que las personas que no están familiarizadas con este software puedan ver igualmente los datos tratados en Excel.

D.3.2. Implementación de cubos de decisión

Basándose en el estudio teórico sobre los cubos de decisión realizado en 2016, durante la anualidad 2017 en primer lugar se definió la información a analizar procedente de la base de datos sobre la que trabajarán dichos cubos de decisión. En el caso del proyecto SMASCHING4 esa información consiste en datos relativos a consumos eléctricos, aceleraciones y sonidos registrados para una máquina específica y en unas fechas determinadas.

En base a lo anterior, se realizó una primera configuración de un cubo de decisión utilizando la herramienta Mondrian Schema Workbench, que es un motor de Java orientado a generación de cubos OLAP que proporciona las conexiones necesarias con las base de datos SQL y atiende las consultas de petición de datos (es decir, hace la función de motor OLAP relacional descrita en la figura 17). De manera resumida, se ilustran a continuación de manera gráfica las secuencias de operaciones desarrolladas para lograr la integración de los cubos de decisión en el software:

- En primer lugar, se estableció la conexión con la base de datos, usando el driver específico para ello, siguiendo la ruta Tools/Connection y rellenando los campos que aparecen en la ventana:



Figura 28. Ruta para establecer la conexión

- Así, se especificó la dirección del servidor donde se aloja la base de datos (en este caso, 192.168.30.203, que corresponde con la IP de la Raspberry Pi del cluster donde se aloja la base de datos), el nombre de la base de datos (en este caso, PRUEBA), el nombre de usuario y la contraseña de acceso a la base de datos.

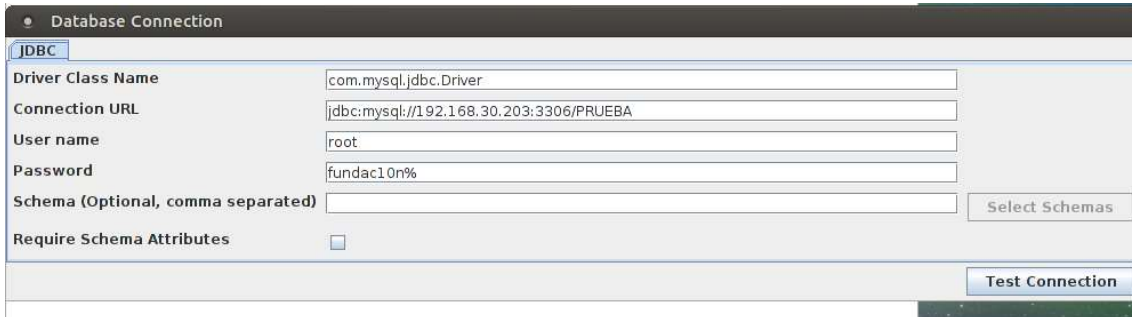


Figura 29. Establecimiento de la conexión

- Una vez verificada que la conexión se había establecido correctamente, se comenzó con el desarrollo del cubo, para lo que se creó un esquema nuevo siguiendo la ruta File/New/Schema. Este esquema es el objeto que contiene todos los cubos de un diseño y está asociado a un fichero XML.

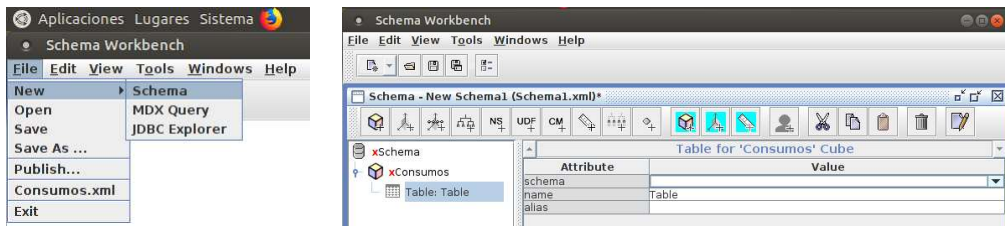


Figura 30. Izquierda: creación de un esquema nuevo; Derecha: ventana de características del esquema

- Una vez creado el esquema, se creó un cubo nuevo y se definió la tabla de hechos, que es una tabla en el modelo multidimensional que contiene la información que se quiere analizar, en este caso, los datos de consumos energéticos, aceleraciones y sonidos de las máquinas Kern y Deckel.

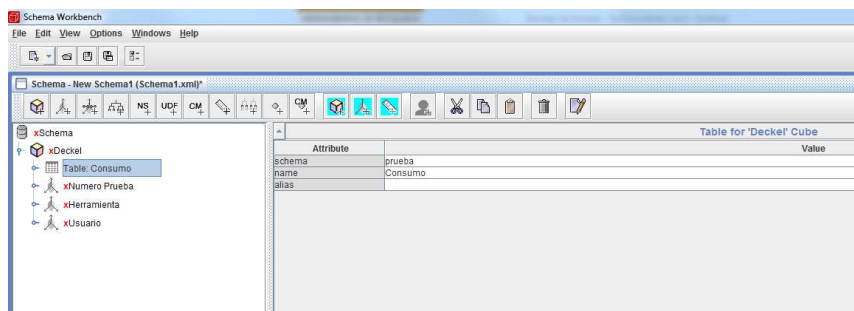


Figura 31. Tabla de hechos

- Tras la definición de la tabla de hechos, se añadieron tres dimensiones al cubo de decisión creado anteriormente. Cada dimensión hace referencia a un elemento de análisis, en el caso del proyecto SMASCHING4 esas dimensiones son herramienta, número de prueba y usuario.

- En el caso de la tabla de hechos consumos en primer lugar se configuró el elemento table, asignándole en el campo name la tabla con la que se quiere trabajar. A continuación se configuró el elemento nivel y se crearon tres, correspondientes a los tres canales de medida (puesto que para la medición de los consumos se ha utilizado un sistema trifásico).

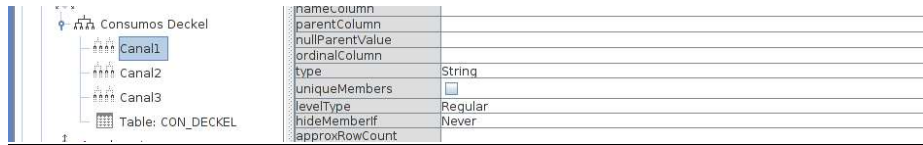


Figura 32. Configuración del elemento nivel dentro de la dimensión consumos

- De la misma manera, se configuró la dimensión aceleraciones, con tres niveles correspondientes a las aceleraciones registradas en los ejes, X, Y, Z y enlazados con las tablas existentes en la base de datos que corresponden a los tres canales de medida de aceleración.



Figura 33. Configuración de nivel dentro de la dimensión aceleraciones

- Finalmente se configuró la dimensión sonido, en este caso con un único nivel ya que únicamente hay un canal de medida.

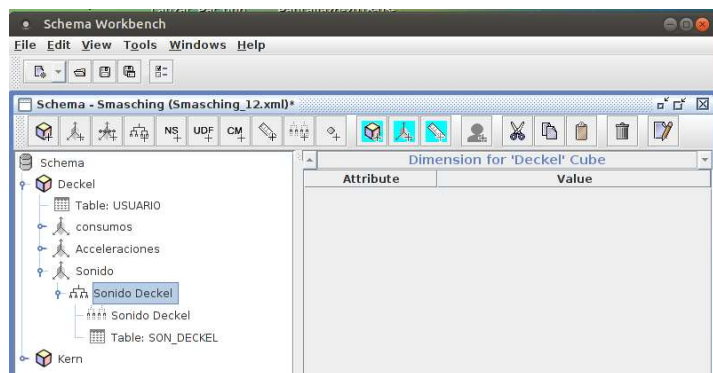


Figura 34. Configuración de nivel dentro de la dimensión sonido

- Una vez diseñado y configurado el cubo, se guardó y se publicó en el servidor de la herramienta Pentaho para poder visualizarlo.

D.3.3. Diseño de experimentos

A fin de evaluar las capacidades del sistema hardware/software desarrollado, se diseñaron una serie de experimentos, consistentes de manera básica en la realización de varios trabajos de mecanizado con diversas herramientas sobre un mismo tipo de material, a la vez que se guardaban datos de consumos, aceleraciones y sonido.

En la Figura 35 pueden verse las cuatro herramientas utilizadas durante las pruebas: se trata de fresas fabricadas en carburo de tungsteno donde la principal diferencia entre ellas es su diámetro, pudiendo realizar operaciones de planeado, ranurado o perfilado de diferentes dimensiones.

Se han llevado a cabo un total de 361 pruebas que han consistido en realizar mediciones de consumos, aceleraciones y sonido durante 10 minutos. Para cada medición se ha obtenido un único valor que representa la media de la variable durante un periodo de 10 minutos; dicho de otro modo: las 361 mediciones son en realidad 361 medias. En el caso de los consumos, el sistema realizó la medida de las potencias de las 3 fases con las que se obtenía la potencia trifásica total en cada instante de medida, que está programado para que sea de 10 segundos.

Para el caso de las aceleraciones se ha seguido el mismo proceso, sólo que no existe una medida total de aceleración, sino que se han almacenado los datos de los tres ejes de medida.

En el caso de las medidas de sonido, recordemos que se trabaja con las transformadas de Fourier, por lo que no tiene sentido hablar de medidas en el tiempo, sino que se realiza un barrido en frecuencia entre 0 y 10 KHz y se obtiene el valor predominante, es decir, tendremos un pico a una frecuencia. Cada barrido dura 3 segundos y se repite periódicamente durante 10 minutos, transcurridos los cuales se realiza la media de todos los picos obtenidos.



Figura 35. De izquierda a derecha: herramienta plana de 4, plana de 10, plana de 2 y plana de 40

Herramienta	Material de la herramienta	Número de filos	Diámetro	Material de la pieza	Tiempo de ensayo
Plana 4	Carburo de Tungsteno	4	4mm	Aluminio	10 min
Plana 10	Carburo de tungsteno	4	10 mm	Aluminio	10'
Plana 2	Carburo de tungsteno	4	2 mm	Aluminio	10'
Plana 40	Carburo de tungsteno	4	40 mm	Aluminio	10'

Tabla 1. Herramientas

D.3.4. Realización de experimentos. Integración de tablas de decisión y funcionalidades de visualización de información

La generación de tablas dinámicas y gráficos se puede realizar de manera casi inmediata con el software PDI dentro de la herramienta Pentaho BI. Para ello basta con accionar el botón de Play presente en el archivo, y configurar lo que queremos ver arrastrando los datos que queremos ver al campo Measures, obteniendo un resultado como el de la figura 36.

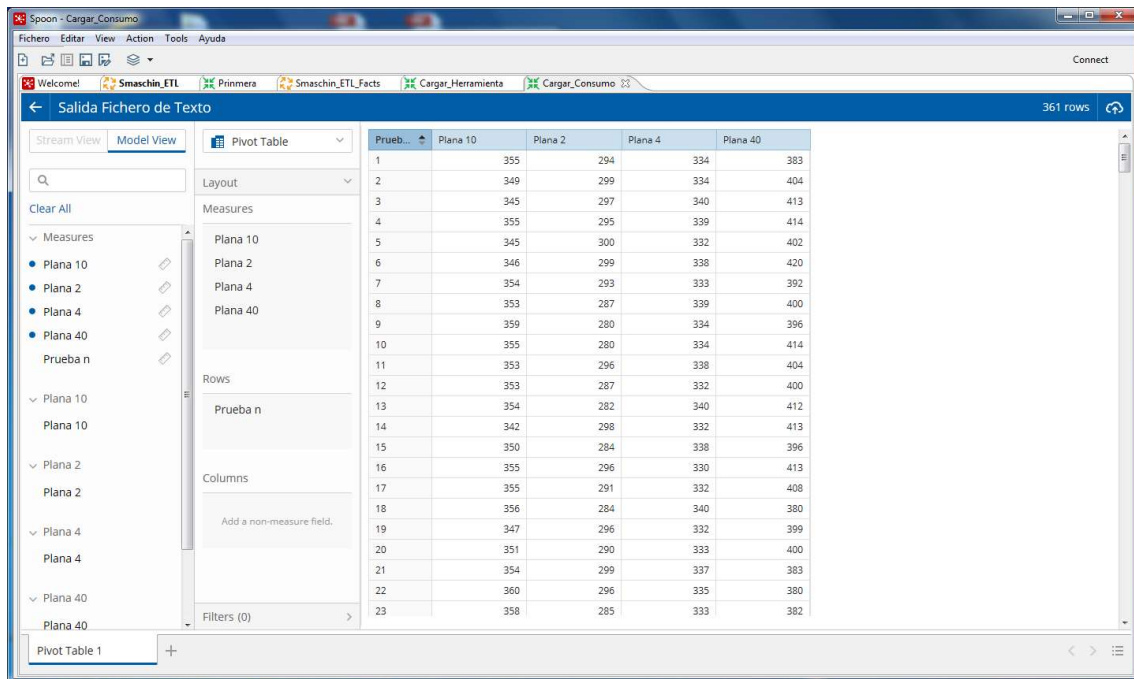


Figura 36. Extracto de la Tabla Dinamica de PDI

Para generar un gráfico que contenga los datos de la tabla anterior, basta con cambiar la opción *Pivot Table* por la de *Line*, con lo que aparece un gráfico como el de la Figura 37.

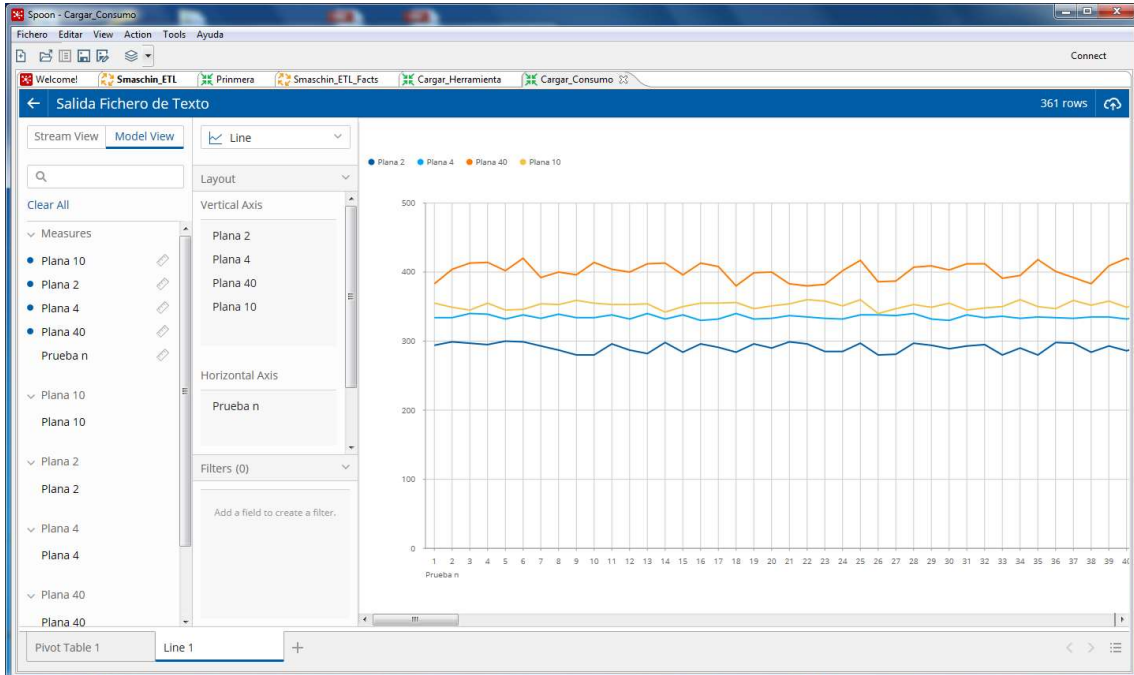


Figura 37. Gráfico dinámico en PDI

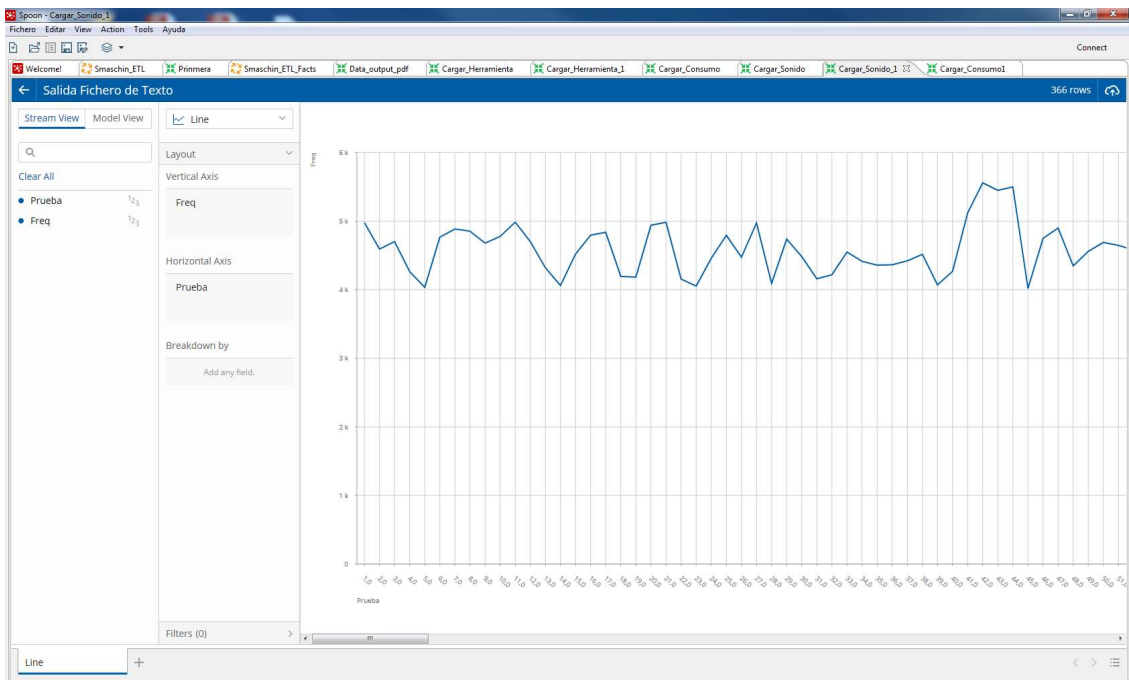


Figura 38. Gráfico dinámico con filtro aplicado para mostrar los datos de la herramienta plana de 40

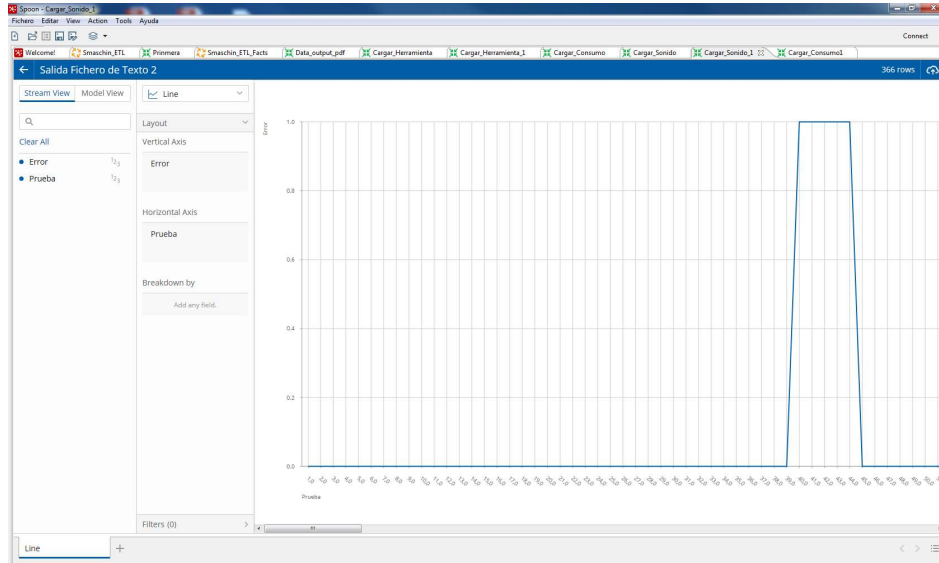


Figura 39. Gráfico mostrando Error en Herramienta (frecuencia demasiado alta)

Se muestran a continuación las gráficas generadas en Excel para los consumos y aceleraciones:

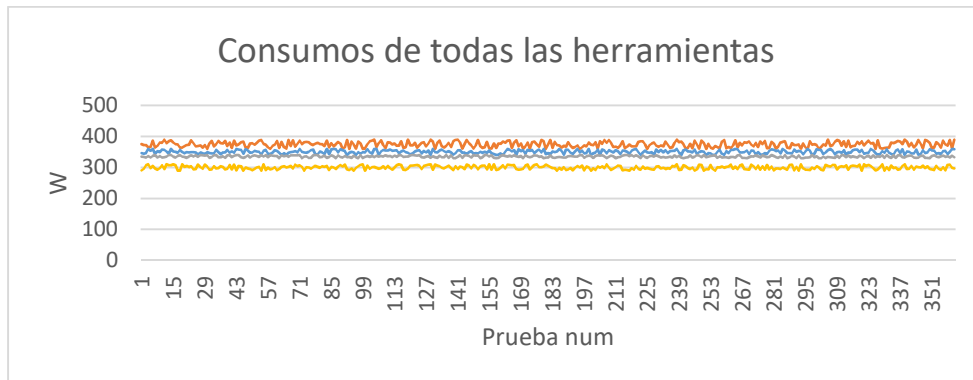


Figura 40. Representación gráfico de los datos de consumo

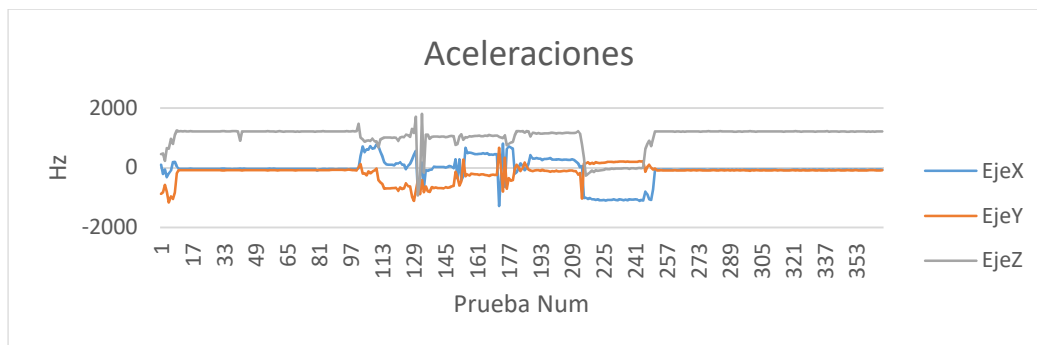


Figura 41. Representación gráfica de los datos de aceleraciones

D.3.5. Generación dinámica de informes

Para la generación de informes se utiliza la aplicación Report Designer, con la que es posible aprovechar los entornos de trabajo generados de manera previa para generar informes a petición. Primeramente se debe generar una plantilla donde se definen los campos a rellenar, en este caso las tablas y los gráficos. También es posible añadir otra información como el logotipo de la empresa, la fecha, etc. Además, debemos definir las conexiones con los datos, para que el programa pueda actualizar el informe cada vez que lo necesitemos de manera más ágil. Todo esto se hace sólo una vez, al inicio, después únicamente es necesario hacer clic en el botón de play y el informe se genera con los datos actualizados.

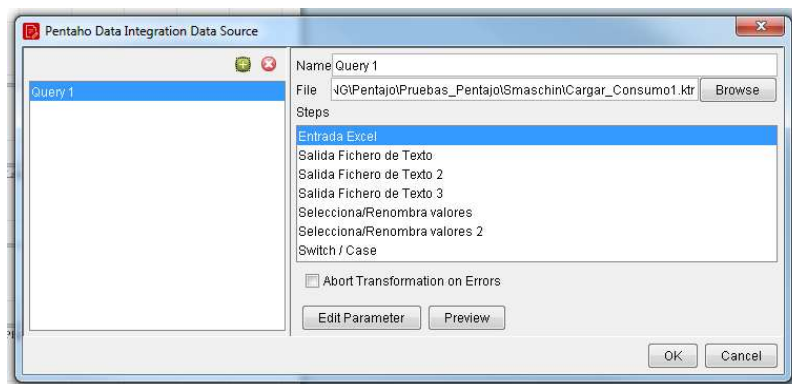


Figura 42. Conexión con los datos a través de Excel

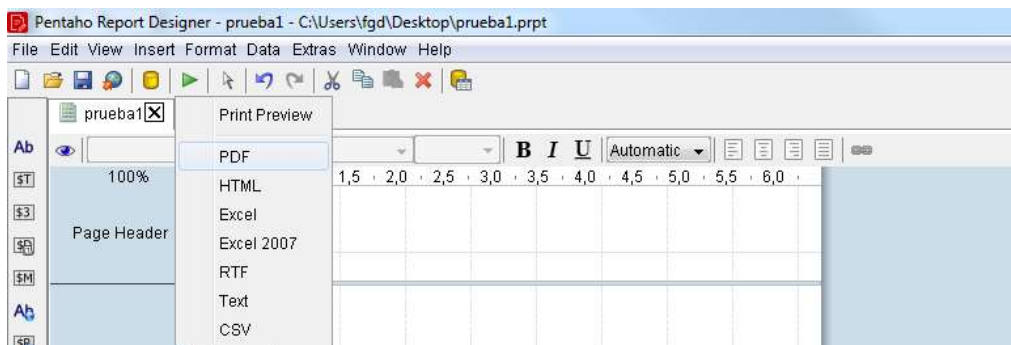


Figura 43. Generar informe

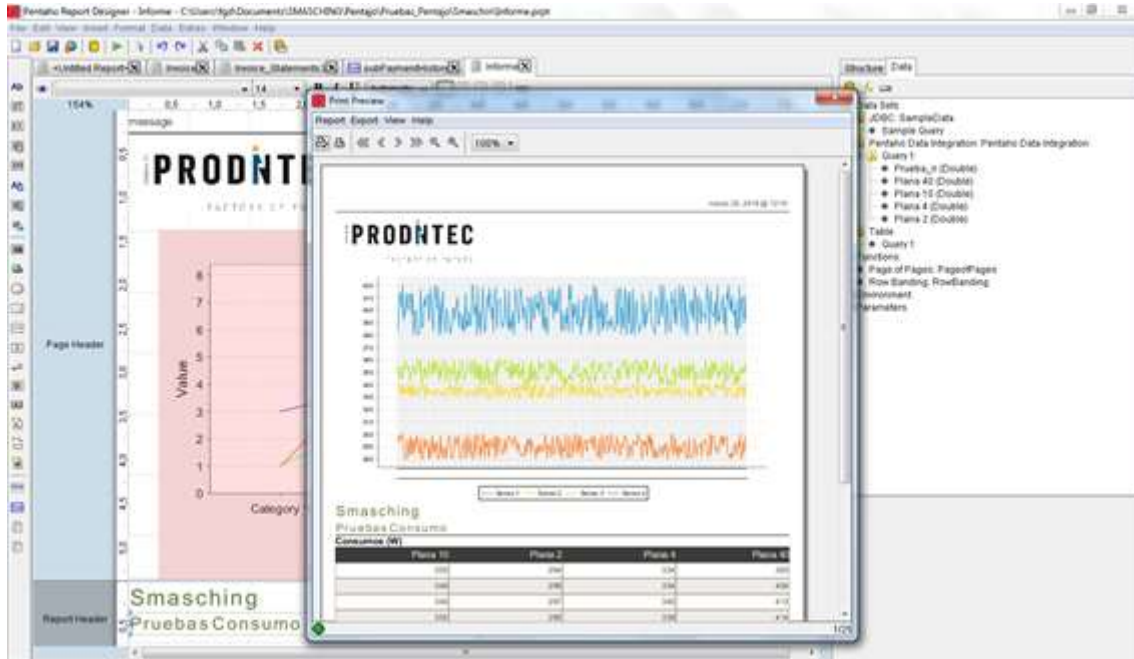


Figura 44. PDF generado

D.4 Principales conclusiones

A modo de resumen de los resultados alcanzados con la ejecución del proyecto SMASCHING4, recogidos tanto en la presente memoria de justificación como en la referente a la anualidad 2016, mencionar que se ha desarrollado un sistema de business intelligence compuesto por:

- Sistemas de medición de datos (consumo energético, aceleraciones y sonido).
- Base de datos.
- Proceso ETL (extracción, transformación y carga de información) + almacén de datos + aplicaciones analíticas.
- Aplicación web (diseñada para su uso en cualquier dispositivo móvil) para la visualización y configuración de los sistemas de medición.

Como trabajo futuro se podría plantear el incremento de la frecuencia de muestreo del sistema de sonido para analizar posibles señales de hasta 100 Khz. También podría plantearse la integración de la generación del informe dentro del PDI implementado en el diagrama de trabajo en lugar de utilizar el software Report Designer. También podría incluirse el envío de correos electrónicos de manera automática conteniendo el informe generado.