

# Diseño de un sistema DAQ mediante Arduino

David Asiain Ansorena

Escuela Universitaria Politécnica de La Almunia (EUPLA),  
centro adscrito a la Universidad de Zaragoza

*Affective Lab*

La Almunia de Doña Godina, Zaragoza, España  
dasiain@unizar.es

Jesús Ponce de León Vázquez

Escuela Universitaria Politécnica de La Almunia (EUPLA),  
centro adscrito a la Universidad de Zaragoza

*Affective Lab*

La Almunia de Doña Godina, Zaragoza, España  
jponce@unizar.es

**Abstract—** Ante la imposibilidad de simular algunos aspectos de la Instrumentación Electrónica durante la época del confinamiento, surgió la idea de improvisar un sistema de adquisición de datos (DAQ) mediante una placa Arduino y algo de software adicional. La idea era que el alumnado dispusiese con seguridad en su casa de todos los componentes necesarios para la elaboración de esta práctica no simulada en tiempos de pandemia, sin por ello renunciar a nada en cuanto al contenido docente de la misma. El resultado fue muy bien recibido por parte de alumnado y profesores.

**Keywords—** Instrumentación Electrónica, práctica, DAQ, LabVIEW, Arduino.

## I. INTRODUCCIÓN

La primavera de 2020 supuso un antes y un después para todo el mundo y en todos los sentidos. El confinamiento y las restricciones de movilidad nos sorprendieron a todos, demostrándonos de la forma más cruda y cruel la fragilidad de nuestro estado de cosas, la añorada “antigua normalidad”. A los docentes nos sorprendió en pleno segundo cuatrimestre y todos recordamos con desasosiego tan extraños días. Toda la enseñanza, a todos los niveles, sufrió un duro golpe del que todavía no se ha repuesto. Pero este golpe fue más importante en aquellas asignaturas con un contenido práctico de mayor peso o relevancia.

Aunque de menor calado que otras especialidades, las enseñanzas sobre electrónica encajan en este patrón. Las prácticas de laboratorio resultan esenciales para el correcto desarrollo de las habilidades de cualquier ingeniero, y la pérdida de las mismas conlleva un deterioro en las capacidades del alumnado cuyo verdadero alcance todavía no conocemos. Durante las diferentes etapas de la pandemia, los profesores hemos cambiado el formato, los medios y a veces incluso el contenido de nuestras asignaturas, y nuestro modo de impartir clases se ha ido adaptando a las cambiantes circunstancias.

En aquella primera etapa, muchos profesionales no tuvimos otra alternativa que sustituir las prácticas de laboratorio habituales por simulaciones (o adaptaciones simuladas) de las mismas, no en una sino en todas nuestras asignaturas. El recurso es valioso, no cabe duda, pero resulta bastante evidente que una simulación, por buena que sea, carece de la profundidad de una sesión de laboratorio y en cualquier caso no sirve para ejercitar las habilidades en el manejo de materiales y aparatos de medida. Además, dado que casi en cada asignatura donde resultó factible se realizaron estas simulaciones, el alumnado enseguida perdió la iniciativa y el afán por aprender, y estas pronto dejaron de aportar beneficios para resultar más bien tediosas e inconvenientes.

En el caso de la asignatura de Instrumentación Electrónica impartida en el grado de Ingeniería Mecatrónica de la Escuela Universitaria Politécnica de La Almunia (UZ), se le concede a las prácticas de laboratorio una importancia muy elevada a nivel curricular, y la simple simulación no resulta satisfactoria en muchos aspectos, lo cual resultó palmario durante la parte final del cuatrimestre. En esta tesitura, de cara a la última práctica de la asignatura, se pensó en la forma de que el alumnado pudiese ejercitar sus habilidades de la forma más parecida a como lo hubiese hecho en el laboratorio.

La práctica original consistía en el manejo de un sistema de adquisición de datos (DAQ) mediante una tarjeta NI USB-6008 y el tratamiento de la información registrada mediante *LabVIEW*. Dado que el alumnado disponía de un Arduino propio (necesario para otras asignaturas del grado) se pensó en la forma de convertirlo en una tarjeta DAQ para que, combinado con una licencia de prueba de *LabVIEW*, se cubriesen los mismos objetivos que en la práctica original desde una perspectiva aplicada, lejos de la simple simulación. Este trabajo presenta el modo en que se resolvió el problema y los resultados de su aplicación.

El trabajo está ordenado como sigue: en la Sección II se describe el sistema DAQ basado en Arduino [1], tanto a nivel de hardware como de software. En la Sección III se explica el funcionamiento del sistema completo tal cual el alumnado debía desarrollarlo y se muestran los resultados más relevantes obtenidos en la práctica. En la Sección IV se presentan los resultados docentes de la experiencia, llevada a cabo durante los cursos 2019-2020 y 2020-2021. Por último, en la Sección V se presentarán las correspondientes conclusiones.

## II. DESCRIPCIÓN DEL SISTEMA

El sistema desarrollado debe responder a dos grandes preguntas: qué datos se van a adquirir y la forma en la que se van a adquirir. Ambos temas se tratarán en las siguientes secciones.

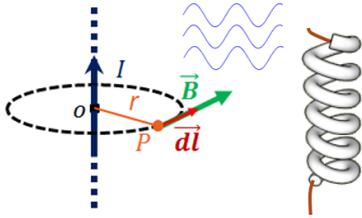
### A. Principio de funcionamiento

La carencia de material práctico (generadores, aparatos de medida, fungible) por parte del alumnado (exceptuando la placa Arduino) resultaba un factor limitador esencial. Para ello se pensó en utilizar una antena inductiva muy básica como sensor de campos eléctricos ambientales. De este modo se elimina la necesidad de un generador y se puede leer una señal variable de una forma muy sencilla y al alcance de todo el mundo.

Sin entrar en excesivos detalles, se puede deducir [2] que el campo magnético de un conductor (supuesto rectilíneo, uniforme e infinito) es radial, y su intensidad es inversamente proporcional a la distancia:

$$B = \frac{\mu_0 I}{2\pi r} \quad (1),$$

donde  $\mu_0$  es la permeabilidad magnética del vacío.



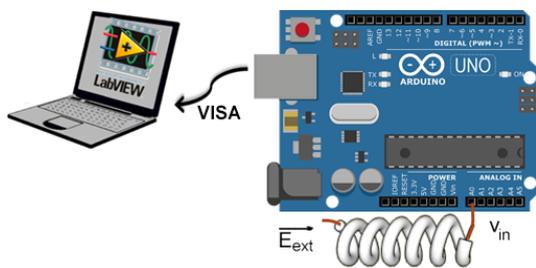
**Figura 1.** Campo magnético de un conductor rectilíneo infinito por el que circula una corriente  $I$ . El módulo de  $B$  viene dado por la Ecuación (1). Su dirección es perpendicular al plano formado por la corriente  $I$  y el segmento  $oP$ . La radiación se propaga por el medio hasta que es recogida por la antena inductiva elemental (cable arrollado).

Este campo (ver Figura 1) se propaga por el medio, de modo que una antena inductiva elemental (en otras palabras, un simple cable arrollado con unas cuantas espiras), que es sensible a este tipo de radiación electromagnética, es capaz de detectarlo. Cuando la corriente es sinusoidal, el campo magnético generado también lo es, y por extensión el campo eléctrico inducido en la antena. Su intensidad decrece con la distancia al conductor. Esta señal electromagnética nos rodea de forma continua en cualquier entorno urbano. Así, se dispone tanto de una señal a detectar como de un detector razonable y de fácil disposición.

A partir de este momento, la cuestión (mucho menos trivial) es cómo convertir una placa Arduino en un DAQ funcional de la forma más sencilla posible. La idea es establecer un método de comunicación entre Arduino y *LabVIEW*, convirtiendo la señal analógica capturada por la antena en una que este software sea capaz a su vez de analizar. Para ello se hace necesario el empleo tanto de una licencia de uso limitado (versión trial de 30 días) de *LabVIEW* [3] como del protocolo de comunicación (*VISA*) de *LabVIEW* con Arduino [4], ambos disponibles en la página web de National Instruments.

### B. La placa Arduino como DAQ

La estructura básica del sistema DAQ propuesto se refleja en la Figura 2. Su aspecto real aparece en la Figura 3.



**Figura 2.** El Arduino como DAQ. Concepto.



**Figura 3.** Aspecto final de la placa y la antena inductiva.

Al alumbrado se le proporcionan dos archivos básicos que convertirán la placa Arduino en un sistema DAQ funcional: un código para volcar en la placa Arduino y un bloque en *LabVIEW* que permite la lectura de la señal capturada para poder trabajar con ella. Ambos se explicarán a continuación.

#### §1. Código para Arduino

El programa completo puede verse en las Figuras 3 y 4. En resumen, en este programa se leen y digitalizan los datos de la entrada analógica en la que se haya conectado la antena. Asimismo, mediante un modulador por ancho de pulsos (PWM), el programa genera un ruido eléctrico a 490Hz que radia al exterior a través de otra salida analógica y que la misma antena registrará, contaminando así la señal de la red con una fuente de ruido independiente. De esta forma se hará necesario introducir el concepto y la necesidad de un filtrado durante la ejecución. Este programa se vuelve en la placa y enviará los datos correspondientes al ordenador a través del puerto serie y el protocolo *VISA*.

La implementación del código en Arduino es un pequeño sketch buscando la mayor simplicidad posible (Figura 3).

```

1 /*
2 ArduinoDataAcquisitionSystem
3 Reads an analog input at pin 0 with a configurable sampling time,
4 prints the result on the serial monitor.
5 Upload this to the Arduino using the Arduino IDE!
6 */
7 const long SamplerTime = 500; // in microsecond 500uS -> 2kHz
8 const int NoisGenerator490HzPin = 9;
9
10 void setup() {
11   Serial.begin(115200);
12   pinMode(LED_BUILTIN, OUTPUT);
13   digitalWrite(LED_BUILTIN, LOW);
14   pinMode(NoisGenerator490HzPin, OUTPUT);
15   //Start Nois Generator 490Hz duty cycle 50%
16   analogWrite(NoisGenerator490HzPin, 127);
17 }
18
19 void loop() {
20   SamplerRateSync(SamplerTime);
21   int val = analogRead(0);
22   Serial.println(val);
23 }
24

```

**Figura 3.** Programa de captura de datos para Arduino. Sketch básico.

La función *setup*, se ejecuta una única vez al inicio de la aplicación y en ella se realiza la inicialización de los módulos del sistema. En la línea 11 se inicializa el puerto serie virtual por el puente USB a una tasa de transferencia de 115200 baudios. Puesto que el flujo de datos se va a transferir por esta conexión, la velocidad de transferencia tiene que estar relacionada con la frecuencia de muestro propuesta.

En las líneas 12 y 13, el pin del procesador que se encuentra conectado con el led interno de la plataforma se configura como salida. Adicionalmente se pone a nivel bajo para que el led se inicie apagado. En la línea 14 se configura como salida el pin 9 del Arduino como soporte de la función de PWM, mientras que en la 16 se inicia esta señal, a una frecuencia de 490Hz y con un ciclo de trabajo al 50%. Dado que la resolución del módulo PWM tiene es de 8 bits, el parámetro se inicia al valor decimal de 127 que equivale al 50% del rango.

La función *loop*, es el bucle principal de la aplicación y se llama cíclicamente. La implementación consta tan solo de tres funciones. *SamplerRateSync* (detallada en la Figura 4) es la encargada que el ciclo de ejecución de la función *loop* sea exactamente el tiempo de muestro. De esta forma se asegura

un muestreo preciso de la señal analógica. Mediante la función *analogRead* (línea 21) se obtiene el valor convertido del módulo ADC de la entrada del canal 0, cuya resolución es de 10 bits. Por último (línea 22) se imprime en consola el valor entero en ASCII con el retorno de carro.

En la Figura 4, puede apreciarse la implementación de la función *SamplerRateSync*. Esta tiene como objetivo retrasar el bucle principal, para que el ciclo de este se ajuste al tiempo de muestreo. Esta solución simplifica (al no emplear interrupciones) y mejora otras implementaciones posibles, basadas exclusivamente en *delays* (que deben ser ajustados a la frecuencia de muestreo, lo cual requiere un reajuste tras cada cambio en el programa principal).

En primer lugar, línea 27, se declara una variable estática (*timer*) en la cual se almacenará el valor del contador de tiempo interno del último ciclo ejecutado, de cara a recalculer con precisión el tiempo exacto a retrasar el bucle principal, línea 29. Si el paso de parámetro *deltaT* en microsegundos supera el valor límite (16383µs), el retraso se implementará con la concatenación de dos funciones: una para el retraso en ms y la siguiente en µs. Si el resultado de la línea 29 es negativo, no hay que aplicar ningún retraso. En tal caso no se estaría cumpliendo el ciclo de muestreo, y como notificación de error (línea 47) se activa un led de aviso y se retorna a la función *loop* inmediatamente.

```

25 void SamplerRateSync(unsigned long deltaT)
26 {
27     static unsigned long timer = 0;
28     unsigned long currTime = micros();
29     long timeToDelay = deltaT - (currTime - timer);
30
31     //the largest value that will produce an accurate delay is 16383
32     if (timeToDelay > 16383)
33     {
34         delay(timeToDelay / 1000);
35         delayMicroseconds(timeToDelay % 1000);
36         digitalWrite(LED_BUILTIN, LOW);
37     }
38     else if (timeToDelay > 0)
39     {
40         delayMicroseconds(timeToDelay);
41         digitalWrite(LED_BUILTIN, LOW);
42     }
43     // timeToDelay is negative so we start immediately
44     else
45     {
46         //reports sampling frequency has not been met in time
47         digitalWrite(LED_BUILTIN, HIGH);
48     }
49     timer = currTime + timeToDelay;
50 }

```

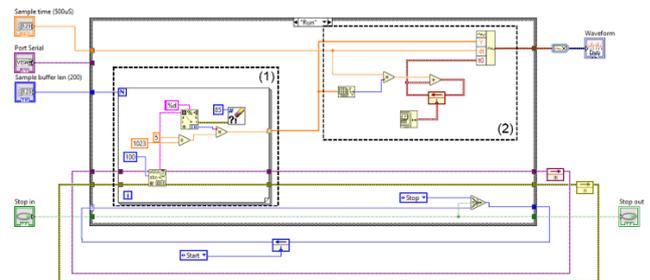
**Figura 4.** Programa de captura de datos para Arduino. Sincronización del bucle *loop* a la frecuencia de muestreo.

## §2. Módulo de comunicación Arduino/LabVIEW (VISA)

En cuanto al bloque básico de *LabVIEW* (véase Figura 5), convierte los datos proporcionados por el anterior programa en una señal legible, con una marca de tiempo y una frecuencia de muestreo asociadas. Este sistema se encarga de abrir y cerrar el puerto COM seleccionado por *VISA* como entrada de datos y de convertir la cadena de caracteres en los correspondientes valores de lectura. Asimismo se genera una marca de tiempos, que combinada con la frecuencia de muestreo nos proporciona una señal variable, muestreada a un ritmo compatible tanto con la frecuencia de red como con el ruido de 490Hz insertado en ella. De este modo no se pierde información a nivel frecuencial y esta señal puede ser representada y tratada por *LabVIEW* como si hubiese sido adquirida a través de cualquier sistema de adquisición de datos, como la citada tarjeta NI USB-6008. La pérdida de prestaciones del sistema propuesto respecto a la NI USB-6008

es perfectamente asumible y no llegará a resultar perceptible en lo sucesivo.

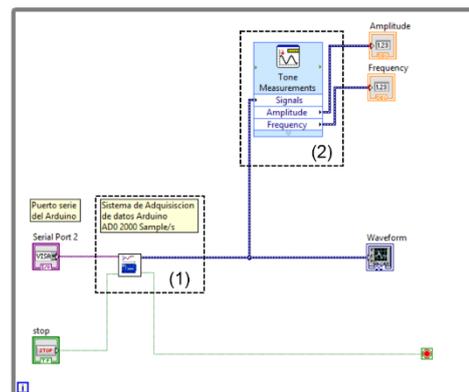
Como puede verse en la Figura 5, la interface del objeto está definida por cuatro variables de entrada y dos de salida. Como variables de entrada se encuentran el puerto serie a utilizar, el tiempo de muestro (con un valor por defecto de 500µs), el tamaño del buffer de muestro (con un valor por defecto de 200 muestras) y por último una variable booleana para indicar la parada de adquisición. Como variables de salida quedan la señal muestreada y un valor booleano que indica el cierre de la tarea. Esta interfaz es compatible con el objeto genérico de las tarjetas de NI soportadas por *LabVIEW*.



**Figura 5.** Bloque de comunicación entre la placa Arduino y *LabVIEW*. El driver que proporciona el protocolo VISA se encarga de abrir y cerrar el puerto de comunicaciones (COM) a la frecuencia adecuada. En la figura se muestra únicamente el bloque de la ejecución del sistema (ver texto). (1) Bloque de lectura de la señal (bucle *while* de conversión de string). (2) Bloque de generación de la señal (marca de tiempo y concatenación).

En el interior del bloque, se ha implementado una máquina de estados a partir de en un *case* con los estados “*Start*”, “*Run*” y “*Stop*” (en la figura aparece únicamente el más importante de ellos, el estado “*Run*”).

En los estados “*Start*” y “*Stop*” se inicia o se cierra el puerto serie. El estado “*Run*”, queda implementado por un bucle acotado con la longitud del buffer de muestreo definido, y ejecuta la lectura de datos del puerto serie, la conversión de estos a un valor entero y la calibración a voltios considerando la tensión de referencia 5V y una resolución de 10 bits. A la salida del bucle se obtiene un vector que se convierte a una señal, indicando el tiempo de muestreo y la marca de tiempos desde el inicio de la adquisición. Esta señal se adapta a un valor dinámico para que pueda ser representada y analizada con facilidad.



**Figura 6.** Empleo básico del bloque DAQ. La adquisición se realiza en un bucle *while*. (1) DAQ proporcionado. Se trata de una caja “cerrada”, con acceso único a entradas (puerto COM y parada) y salidas (señal reconstruida y aviso de parada). (2) Bloque de medidas de tono para la extracción de la amplitud de pico y la frecuencia fundamental.

Los estudiantes reciben este módulo como un objeto “cerrado” (Figura 6). No es necesario que se conozca a la perfección cómo se implementa su funcionamiento, sino tan

sólo que se trata de un sistema con dos entradas externas (el puerto serie del Arduino y la variable booleana de parada, las variables fijas no resultan accesibles) y dos de salida (la función temporal capturada y el aviso de parada). Como se muestra en la Figura 6, tan solo es necesario asignar un puerto de entrada compatible y en la salida del bloque se obtiene de forma directa la señal cuantificada y muestreada, lista para su posterior tratamiento. El resto de las funciones y bloques necesarios (ver Sección III) se generan a partir de este.

### §3. Material necesario

En resumen, el alumnado dispone de la placa Arduino y su correspondiente entorno de programación, y construye su propia antena inductiva con un cable. Tras la descarga del software apropiado, (*LabVIEW* demo y *VISA*), puede emplear los dos programas suministrados por el profesorado para convertir los datos registrados por la antena en una señal equivalente a la que se habría obtenido mediante un DAQ tradicional. El material necesario para la realización de la práctica queda reflejado en la Tabla I.

TABLE I. MATERIAL NECESARIO (HARDWARE Y SOFTWARE).

MATERIAL NECESARIO
Placa Arduino UNO
Cable arrollado (antena)
Entorno de programación Arduino ( <i>Arduino IDE</i> )
Programa básico para Arduino (Lectura)
<i>LabVIEW</i> (demo) [1]
Protocolo de comunicación VISA [2]
Interpretación y lectura de datos en <i>LabVIEW</i> (DAQ)

## III. IMPLEMENTACIÓN

Instalado el software relativo a *LabVIEW* y programado el Arduino, el alumnado debe emplear como base el módulo construido sobre *VISA* para implementar un sistema de medida completo que sirva como detector de la presencia de cables eléctricos. La señal registrada por este módulo presenta una clara distorsión debida al ruido de 490Hz insertado en la misma a través del PWM (ver Figura 7), lo cual nos obliga a emplear filtros, cuya actuación sobre la señal puede apreciarse mejor sobre el espectro de la misma. De este modo se introduce la FFT y, de forma natural, el resto de bloques que se especifican más adelante.

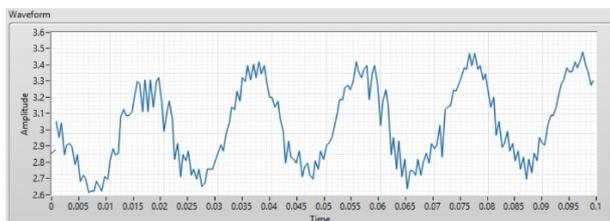


Figura 7. Aspecto de la señal capturada por la antena, una vez registrada por VISA a través del puerto serie y representada en *LabVIEW*. La presencia de ruido resulta evidente y el empleo de filtros para reducirlo surge de forma natural.

En resumen, la idea es registrar la señal obtenida, limpiarla, extraer algunas de sus características y diseñar un ciclo de histéresis que dispare una luz de aviso cuando la señal supere un umbral (cable cercano) y lo apague cuando descienda por debajo de un segundo umbral (distancia de seguridad). Se piden además bloques adicionales que sirven para conocer algunas características adicionales de *LabVIEW*.

### A. Requerimientos prácticos

El alumnado debe implementar un Instrumento Virtual (VI) con las siguientes características:

- Representación gráfica de la señal capturada y su espectro (FFT).
- Implementación de un filtro paso banda (Chevyshev, 40Hz-60Hz de banda de paso y con un nivel de filtrado adecuado).
- Representación de la señal filtrada y su espectro.
- Extracción de frecuencia fundamental ( $f_0$ ) y amplitud de pico de la señal filtrada.
- Representación de la amplitud de pico en un vúmetro u otro tipo de visualizador de nivel.
- Implementación de un ciclo de histéresis para el aviso de la presencia de un cable eléctrico (niveles a regular).
- Escritura de datos a fichero.

Lo cual requiere la inserción y configuración de los siguientes bloques de *LabVIEW*:

- Gráficas (señal y espectro).
- Bloque de medida espectral para cálculo de una FFT.
- Filtro paso banda, adecuadamente diseñado y configurado.
- Bloque de medida para la extracción de  $f_0$  y amplitud de pico de la señal filtrada.
- Vúmetro o similar (para la amplitud de pico).
- Bloques de programación numérica y booleana para el ciclo de histéresis.
- Leds de visualización.
- Bloque de escritura a fichero.

### B. Diagrama de bloques y panel frontal

Una vez conseguido, el diagrama de bloques del VI generado debería contener los subsistemas que implementan todo lo solicitado. En la Figura 8 se muestra uno de tales diagramas de bloque, extraído de los informes de prácticas del curso 2020/2021.

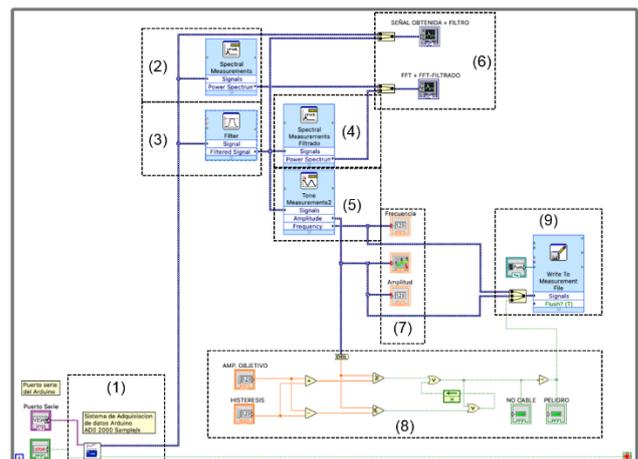
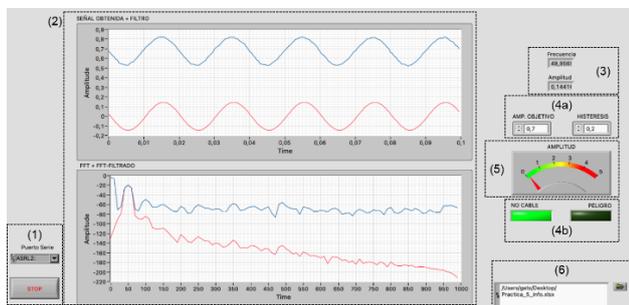


Figura 8. Diagrama de bloques del VI generado. (1) DAQ proporcionado. (2) Bloque de medidas espectrales: FFT. (3) Filtro paso banda (40-60Hz). (4) Espectro de la señal filtrada. Igual que (2). (5) Bloque de medidas de tono: amplitud y  $f_0$ . (6) Gráficas de señales y espectros. (7) Representaciones numéricas de amplitud y  $f_0$  y visualizador (vúmetro) de amplitud. (8) Implementación de un ciclo de histéresis para la alarma de proximidad, incluyendo LEDs de aviso. (9) Bloque de escritura a fichero. Imagen extraída de uno de los informes de prácticas presentados durante el curso 2020/2021.

Por su parte, el panel frontal del VI debería quedar de forma similar a la reflejada en la Figura 9. En ella se pueden

observar las representaciones de la señal y su espectro antes y después de filtrar, el valor numérico de frecuencia fundamental y amplitud de pico, el indicador (vúmetro) de la amplitud, los LED de alarma y situación segura, información relativa a la escritura en fichero y, por supuesto, el botón de parada.



**Figura 9.** Panel frontal del VI generado. Imagen extraída de uno de los informes de prácticas presentados durante el curso 2020/2021. (1) Selección de puerto COM y botón de parada. (2) Arriba, gráficas de las señales original (azul) y filtrada (rojo). Abajo y espectros (azul, original, rojo, filtrado). En este caso el nivel de filtrado no es óptimo, como se puede observar. (3) Visualizadores de amplitud y  $f_0$ . (4a y 4b) Controles numéricos del ciclo para la alarma y LEDs de visualización. (5) Vúmetro para la amplitud. (6) Escritura a fichero.

Una vez desarrollado el VI, el usuario puede poner a prueba su funcionamiento deslizando la placa Arduino a lo largo de una pared en la que haya un cable eléctrico oculto (por ejemplo, por encima del interruptor de la luz). A medida que la antena se aproxima a la fuente, la amplitud de la señal sube hasta que eventualmente se dispara la señal de alarma, indicando una proximidad excesiva al cable. Cuando nos alejamos del mismo la amplitud de la señal decae y, cuando se supera el margen marcado por la amplitud del ciclo de histéresis, nos encontramos a una distancia considerada “segura” del cable. De esta forma, por ejemplo, podríamos evitar sorpresas a la hora de taladrar un agujero en una pared.

La posible aplicación práctica real del sistema así diseñado, montado y verificado, resultó asimismo muy atractiva para el alumnado en general.

#### IV. RESULTADOS DOCENTES

Como apoyo adicional a lo expuesto hasta aquí, el control de la elaboración de esta práctica se llevó a cabo empleando la herramienta *Discord*, muy conocida por el alumnado al tratarse de una plataforma popular en sesiones de *gaming* [5], resultando por otro lado muy útil para las clases en online [6]. Se abrió un aula general, donde el profesorado daba las explicaciones iniciales y permanecía hasta ser reclamado por alguno de los grupos de trabajo, para los cuales se crearon canales adicionales. De esta forma se buscaba fomentar la interacción, a la par que ir solucionando tanto las dudas como los problemas puntuales que pudieran surgir a cada grupo.

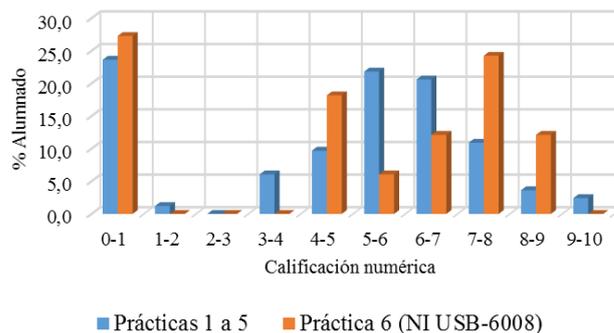
Dadas las limitaciones en la asistencia a los laboratorios debidas a la situación de la pandemia, se decidió repetir la experiencia el año siguiente (curso 20/21). La situación había cambiado sutilmente puesto que el alumnado había acudido de forma regular al laboratorio a lo largo del cuatrimestre, y por lo tanto se perdía la “frescura” que supuso el cambio durante el año anterior.

Los resultados académicos del último curso pre-pandemia (2018/19) se incluyen en la Figura 10. En este curso, los informes de la práctica se entregaban por parejas. En la figura se representa la evolución de las calificaciones (en %) en las primeras 5 prácticas del curso frente a la última, equivalente a

la aquí detallada. Más en detalle, en esta última práctica se empleaba un diodo como sensor de temperatura y una tarjeta de adquisición de datos NI USB-6008, y se trataba la señal de idéntica forma a la presentada en este trabajo (filtrado, representación, ciclo de histéresis con niveles para calefacción, etc.), mediante *LabVIEW*.

En la Figura 10, en el eje horizontal aparecen las calificaciones de 0 a 10 y en el vertical el porcentaje de alumnos que ha obtenido cada nota. Como puede observarse, el número de abandonos (en este caso informes valorados con una nota de 0.0) es muy elevado (23.6% en las primeras 5 prácticas, 27.3% en la sexta). Cabe destacar, además, que las calificaciones se polarizan hacia notas más bajas de lo habitual por parte del alumnado que no lleva bien la asignatura, hasta notas más elevadas por parte de quienes sí lo hacen.

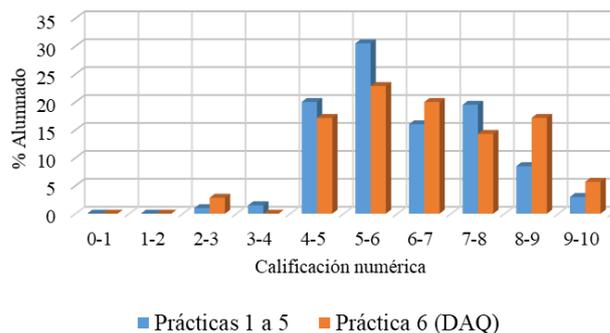
Resultados académicos 2018/2019. Calificaciones



**Figura 10.** Resultados académicos (calificaciones) obtenidos durante el segundo año de implantación de la práctica (2018/2019).

Por otra parte, los resultados académicos de las experiencias que nos ocupan (cursos lectivos 2019/20 y 2020/21), en los que esta práctica se llevó a cabo, aparecen reflejados en las Figuras 11 y 12 respectivamente. En el curso lectivo 19/20 los informes se entregaron de forma individual, mientras que al año siguiente (20/20) volvieron a entregarse por parejas. Ambas figuras se interpretan de la misma forma que la Figura 9, con la salvedad de que en el último curso, por falta de fechas, el número total de prácticas se redujo a 5.

Resultados académicos 2019/2020. Calificaciones



**Figura 11.** Resultados académicos (calificaciones) obtenidos durante el primer año de implantación de la práctica (2019/2020).

Como puede verse en la Figura 11, en primer lugar el porcentaje de abandonos se redujo de forma drástica durante la pandemia, hasta el punto de que no existieron (recordemos, además, que los informes durante ese año fueron individuales, lo cual concede más mérito a este resultado). Analizando los resultados comparativos de las diferentes prácticas, el porcentaje de suspensos de la experiencia aquí presentada decreció con respecto a las primeras 5 prácticas, mientras que

el número de aprobados se redujo de forma significativa a costa de un aumento notable en las calificaciones más altas (8 a 10).

En cuanto a los resultados del curso 2020/21 (Figura 12), los resultados son algo peores y reflejan parcialmente una vuelta a la situación previa al Covid (Figura 10), con la salvedad de que el número de abandonos continuó siendo nulo. Como puede observarse, el número de suspensos volvió a niveles anteriores a la pandemia, mientras que el número de aprobados descendía, siguiendo la misma tónica. A cambio se observa un repunte considerable de los notables y un nuevo descenso en las calificaciones más altas. En este caso se observa de nuevo una clara polarización entre el alumnado, como la observada en la Figura 10. Recordemos que en este caso el curso se llevó a cabo de forma semipresencial y por lo tanto las prácticas de laboratorio volvieron a tener un formato similar al anterior a la irrupción del coronavirus.

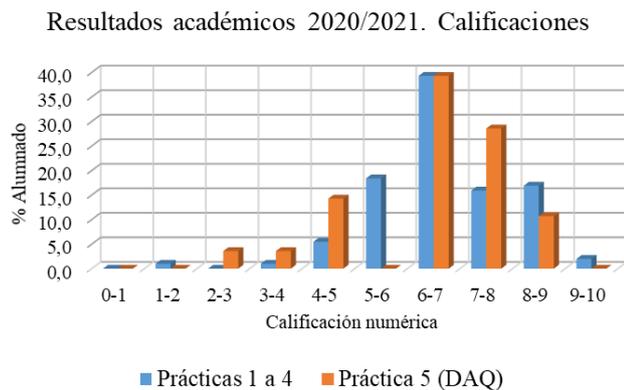


Figura 12. Resultados académicos (calificaciones) obtenidos durante el segundo año de implantación de la práctica (2020/2021).

Durante el curso 2019/20, la opinión de los estudiantes sobre el contenido práctico aquí presentado fue en general muy positiva. Durante las clases online, la mayoría del alumnado valoró muy bien el hecho de poder volver a trabajar con circuitos, por simples que estos fuesen, y el haber construido un sistema con una aplicación práctica tan evidente.

A continuación se reproducen algunos comentarios de estudiantes, extraídos de los informes de esta práctica durante el confinamiento (2019/2020).

“En general, esta práctica ha resultado muy interesante no solo por el hecho de haber podido aprender a desarrollar un circuito con una cierta complejidad en un nuevo y potente programa, sino también a comprobar cómo, efectivamente, pueden llegar a medirse y detectarse todos estos fenómenos físicos de los que siempre hemos oído hablar, pero nunca comprobado manualmente, hasta ahora”.

“Se puede ver como se ha desarrollado un detector de cables, con cosas que un estudiante de este grado tiene acceso en su casa”.

“Creo que en esta práctica se ha aprovechado bien la oportunidad el hecho de estar en casa por el Covid y que se ha ideado una muy buena manera de aprender un programa tan interesante como es *LabVIEW*”.

“Me ha parecido una de las prácticas más interesantes debido a que hemos hecho una práctica en la que hemos podido comprobar el funcionamiento del principio de antenas

de manera física y no solo teórica como alguna de las anteriores debido a la situación actual.”

“Es una práctica que sorprende, debido a ver como con nuestros medios podemos hacer un medidor de corriente eléctrica, con una antena que es un cable enrollado, el Arduino y el programa *LabVIEW*”.

En cuanto a las opiniones del curso siguiente, 2020/2021, se vuelve a la situación pre-pandémica. La mayoría de las quejas están relacionadas con la dificultad intrínseca al empleo de *LabVIEW*, ya que se trata de un software que no se ha empleado con anterioridad y cuya curva de aprendizaje presenta una pendiente que puede resultar excesiva para algunos estudiantes, incluso en un empleo básico como el expuesto. Sin embargo el empleo de *Discord* para las clases online fue bien valorado de forma generalizada.

## V. CONCLUSIONES

En este trabajo se ha detallado el diseño de un sistema de adquisición de datos perfectamente funcional, construido mediante una placa Arduino y un hardware adicional prácticamente inexistente (tan solo cable eléctrico). De esta forma, con mucha imaginación y poco esfuerzo, se ha conseguido mantener de forma exacta el contenido docente tradicional de esta práctica, centrada en la adquisición y el procesamiento de datos con *LabVIEW*. La puesta en marcha del sistema no es demasiado complicada, aunque *LabVIEW* es un entorno completamente nuevo para los estudiantes y puede tener una curva de aprendizaje no del todo compatible con el final de curso (esta práctica se lleva a cabo durante 6 horas lectivas a lo largo de las 3 últimas semanas de mayo).

El primer año en que se implementó, durante lo más duro del confinamiento, buena parte de los alumnos se mostraron muy agradecidos ante el cambio de paradigma frente a lo que se había convertido en el estándar de las prácticas de laboratorio, la simple simulación. Este hecho se ve reflejado en las calificaciones obtenidas, que mejoran de forma sensible los resultados pre-pandémicos, y en la valoración de la práctica expresada por el alumnado. No obstante, al año siguiente, cuando se recuperó en parte la presencialidad, tanto las opiniones como las notas de los estudiantes volvieron a los niveles que solían ser habituales. El empleo de herramientas alternativas a las clases online por *Meet* o *Zoom*, como *Discord*, ha sido muy bien recibido en general, al tratarse de un medio en el que el alumnado se desenvuelve con soltura.

## REFERENCIAS

- [1] *Arduino* web page. Available: <https://www.arduino.cc/>
- [2] Sears, Zemansky, Young, Freedman: “*Física Universitaria*”, Vol. II, 12 Ed., pág. 731. Pearson (2009).
- [3] NI *LabVIEW* 2019. [Online]. Available: [https://www.ni.com/es-es/support/downloads/software-products/download\\_LabVIEW.html#345656](https://www.ni.com/es-es/support/downloads/software-products/download_LabVIEW.html#345656)
- [4] Protocolo de comunicación puerto serie (COM), VISA, incorporado a *LabVIEW* 2019 [Online]. Available: [https://www.ni.com/es-es/support/downloads/drivers/download\\_ni-visa.html#346210](https://www.ni.com/es-es/support/downloads/drivers/download_ni-visa.html#346210)
- [5] *Discord* web page. Available: <https://discord.com/>
- [6] Wulanjani, Arum Nisma. “*DISCORD APPLICATION. Turning a Voice Chat Application for Gamers into a Virtual Listening Class*”. 2nd English Language and Literature International Conference, ELLiC Proceedings Vol. 2, 2018, Electronic ISSN: 2579-7263.