

# WINAD: APLICACIÓN EN ENTORNO WINDOWS PARA EL APRENDIZAJE DE TÉCNICAS DE ADQUISICIÓN Y PROCESADO DIGITAL

Pascual Pérez Blasco  
Escuela Universitaria de Informática  
Universidad Politécnica de Valencia  
pperez@disca.upv.es

Ginés Benet Gilabert  
Escuela Universitaria de Informática  
Universidad Politécnica de Valencia  
gbenet@disca.upv.es

## Resumen

*El presente trabajo presenta una aplicación y una serie de experiencias prácticas que se utilizan actualmente en la enseñanza de los métodos de adquisición y procesamiento digital a alumnos de último curso de la escuela y facultad de informática que cursan las asignaturas optativas de Adquisición y Procesado Digital de la Señal y de Instrumentación y Periféricos.*

**Palabras Clave:** Sistemas de adquisición de datos, Procesamiento digital de la señal, Entornos visuales, Windows.

## 1 INTRODUCCIÓN.

Las asignaturas *Adquisición y Procesado Digital de Señales por Computador e Instrumentación y Periféricos* incluidas en los planes de estudios de la Escuela y Facultad de Informática tienen entre sus objetivos la descripción paso a paso de las diferentes etapas de transformación y procesamiento que sufre la señal en su camino hacia el computador, estudiando los diferentes elementos que intervienen en el mismo. El aprendizaje de estos métodos lo realizan conjuntamente estas dos asignaturas en el laboratorio de tecnología computadores acondicionado con computadores personales, fuentes de alimentación, generadores, osciloscopios, etc. En este laboratorio se imparten también asignaturas relacionadas con el diseño lógico y electrónica. Dada la heterogeneidad de las aplicaciones su gestión se hace cada semestre más compleja.

La motivación de desarrollar una aplicación propia es la de disponer de ejemplos reales de programación de sistemas de adquisición de datos. En estos ejemplos los alumnos observan las etapas de inicialización de

un sistema de adquisición de datos (SAD) y realizan la programación para llevar a cabo diferentes modos de sincronización y transferencia de muestras adquiridas con estos SAD's. Otro motivo que determinó el desarrollo de esta aplicación fue la imposibilidad de alcanzar totalmente estos objetivos con herramientas como MatLab y su Toolbox Real Time Windows Target, que siendo herramientas muy potentes y útiles y que permiten un aprendizaje muy gráfico e intuitivo, mantienen ocultas los procedimientos de adquisición y configuración de las tarjetas.

Por otra parte la elección del entorno gráfico Windows se justifica por razones de motivación, es importante que los alumnos trabajen en entornos de desarrollo de última generación, en nuestro caso Visual C++ 6.0, en lugar de herramientas como Borland C++, que aunque a efectos de consecución de objetivos sea totalmente válido, no consigue motivar a los alumnos suficientemente. Además el uso de este entorno de desarrollo facilita la incorporación de diferentes tipos de representación según el tratamiento que se le dé a la señal. Su uso tiene, además, dos efectos laterales, por una parte este tipo de entornos gráficos y en particular las aplicaciones MFC y sobre el núcleo ejecutivo Win32 aseguran una larga vista a estas aplicaciones. Y otro efecto, a priori perjudicial es que el desarrollo en un entorno de desarrollo de propósito general como *Microsoft Visual Studio* en un entorno multitarea imposibilita generalmente el control de cambios de contexto. Esto motiva la aparición de pequeños desfases en las muestras de la señal durante la etapa de adquisición, en particular cuando se realizan adquisiciones mediante sincronización por polling. Estos efectos que no aparecen cuando se trabaja con interrupciones o por modo de adquisición con DMA se prestan a la introducción de aspectos relacionados con los sistemas de tiempo real.

A continuación, en la sección 2 se describen brevemente el desarrollo de unas prácticas comunes a estas dos asignaturas. La sección 3 describe la aplicación, estructura, interfaz de usuario, ventanas de visualización, y el interfaz al que debe ceñirse el alumno durante la realización de las practicas, etc. se continua con la estructura de la aplicación utilizada, el interfaz de usuario. En la sección 4 de describen brevemente algunas prácticas que se desarrollan durante el curso de esta asignatura y en el que se utiliza esta aplicación. En la sección 5 se especifican aspectos de compatibilidad de esta aplicación y aspectos de la compilación y la disponibilidad de la misma. Finalmente se comentan en la sección 6, las conclusiones y algunos aspectos sobre las posibles mejoras de la aplicación.

## 2 DESARROLLO PRÁCTICO.

En las dos asignaturas se aborda el tema de digitalización de la información y en ambas se trata en una única práctica común, y cuyos objetivos se pueden definir como: Conocer y comprender el funcionamiento de una tarjeta de adquisición de datos, conectada al bus de un computador personal compatible PC, aprender a distinguir cuales son las especificaciones clave de una tarjeta de adquisición de datos (SAD) e implementar en un lenguaje de alto nivel, una serie de rutinas de manejo de la tarjeta que nos permitan realizar conversiones tanto de analógico a digital (A/D) como de digital a analógico (D/A).

La instrumentación mínima necesaria para la realización de una práctica básica de adquisición consta de un computador personal con un SAD en tarjeta, y un generador de señal. Este último elemento es aconsejable que pueda generar al menos señales moduladas en amplitud (AM). Con este tipo de señales resulta muy gráfico e intuitivo la explicación de la descomposición de señales en armónicos y el problema del aliasing.

El primer objetivo es un objetivo general de la práctica y su consecución depende del logro de los dos siguientes, en una primera aproximación, al alumno se le desarrollan sobre pizarra y con tarjeta en mano los conceptos básicos de un sistema de adquisición de datos. Estructura general, líneas digitales, líneas analógicas, convertidor A/D y D/A, tensión de referencia, resolución, canales, ganancia, circuito de muestreo y retención, y el concepto de disparo, los alumnos disponen de un boletín donde se desarrollan también estos conceptos y donde pueden acudir en caso de duda. Una vez terminada esta primera lección teórica se le pide que desarrollen una serie de funciones que de una manera progresiva le aproximan a los conceptos teóricos. Esta segunda

fase consta de tres bloques de ejercicios que se describen a continuación

### 2.1 RUTINAS DE CONVERSION A/D.

Se plantean tres ejercicios, utilizando disparo por muestreo y por temporizador, captura de muestras individuales, captura de un vector de muestras y una captura escalando los datos, con este último caso se pretende que el alumno observe e interprete la información adquirida, así como los problemas que puedan surgir, pérdida de datos, aliasing, etc. En lo que respecta a la pérdida de datos identificada con la aparición de cambios de fase sugiere la introducción de conceptos relacionados con los sistemas de tiempo real.

### 2.2 RUTINAS DE CONVERSION D/A.

Este ejercicio supone la conversión de los datos digitales en una señal analógica. Se genera una rutina que genera una señal senoidal sintética en tiempo real. Se introducen aspectos relacionados con la temporización, las tensiones de referencia, y se observan los efectos de los circuitos de retención.

### 2.3 PROCESO GLOBAL.

Como indica el título del apartado se trata de observar el proceso completo de adquisición procesado y calculo de la salida de actuación actuación.

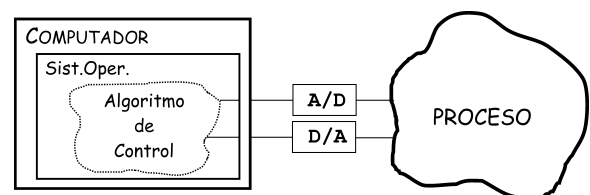


Figura 1: Proceso Global.

Inicialmente se configura el generador para que se genere una señal periódica que el alumno debe adquirir, en tiempo real, el alumno aplica a la señal un filtro de media, el resultado lo aplica al convertidor D/A y compara las dos señales en el osciloscopio. Como este ejercicio se presta a la aplicación de multitud de algoritmos, si se observa un alto grado de comprensión y un desarrollo fluido de la sesión practica se les proporcionan los coeficientes calculados de diferentes filtros y se les pide que implementen una función que aplique una función de transferencia genérica como la de la siguiente ecuación:

$$F(z) = \frac{a_0 + a_1 z^{-1} \dots a_n z^{-n}}{b_0 + b_1 z^{-1} \dots b_m z^{-m}} \quad (1)$$

### 3 ESTRUCTURA DE LA APLICACIÓN.

Sobre el entorno Windows se pueden desarrollar tres estilos de aplicaciones: Aplicaciones de Consola, Aplicaciones de Cuadro de dialogo, y Aplicaciones de Ventana. La arquitectura Documento-Vista es la que predomina sobre estas últimas y es la que se ha empleado para el desarrollo de la aplicación WINAD. Esta arquitectura consiste en la existencia de un contenedor de datos, el documento, que se presenta en pantalla mediante la vista. El tipo de representación depende exclusivamente de la vista y no del documento, es decir, la plantilla con la que se presentan los datos. Siguiendo esta organización la aplicación se divide en las siguientes clases:

- ❑ **Aplicación:** Esta clase une los diferentes elementos que forman la aplicación y al ser un objeto global facilita el acceso al resto de componentes de la aplicación, en particular, al documento y a la vista.
- ❑ **Documento:** El documento es un contenedor que almacena tanto la señal muestreada en cada ejercicio como los parámetros con los que se ha llevado a cabo la adquisición. Estos parámetros permiten escalar correctamente las muestras en los diferentes tipos de representación.
- ❑ **Vista:** El objeto vista permite la visualización de los datos por pantalla. Para cada tipo de representación se dispone de un objeto diferente, en particular, en la aplicación se dispone de tres vistas:

En primer lugar *Vista Fuente* en la que se representan los datos en bruto (figura 2), es decir, un listado numérico de los datos y a la que es posible acceder mediante las funciones estándar del lenguaje C (printf, putchar, scanf, etc).

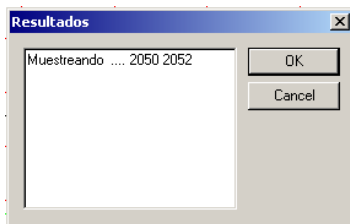


Figura 2: Vista Fuente.

El segundo tipo de vista: *Vista dominio del tiempo* permite mostrar mediante una gráfica, los datos en el dominio del tiempo, es necesario establecer la base de tiempos y los voltios por división además hay que reescalar estos datos con la ganancia y la frecuencia a la que se han tomado las muestras de la señal.

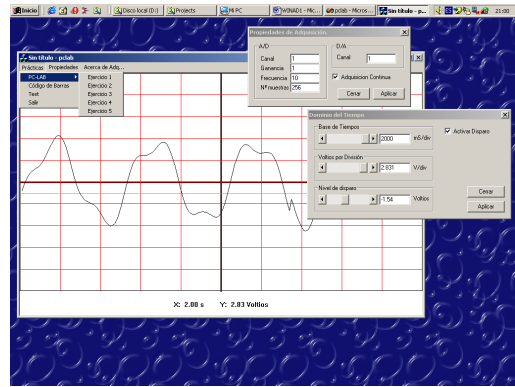


Figura 3: Vista dominio del tiempo.

Por último, la *vista dominio de la frecuencia* que calcula la FFT de las señal adquirida. Esta vista utiliza los datos de escalado de la anterior, es decir, calcula la FFT de tamaño equivalente al número de muestras que caben en la *vista dominio del tiempo*<sup>1</sup>.

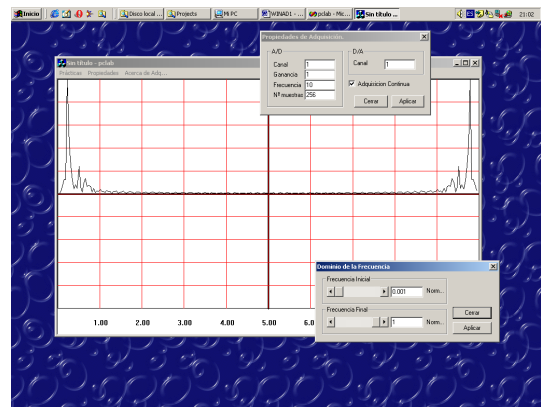


Figura 4: Vista dominio de la frecuencia.

- ❑ **Interfaz de Usuario:** De los cuatro elementos quizás este es, desde el punto de vista docente, el que menos importancia tiene, ya que no se contempla en ninguno de los objetivos de las prácticas. Es tarea del profesor establecer un esqueleto de aplicación donde ya está articulado e implementado este interfaz de usuario. En cualquier caso, excepto la ejecución de un determinado ejercicio, todos los factores que intervienen en la configuración del sistema de adquisición de datos deben de ser configurados tanto desde el interfaz de usuario como desde el propio programa a desarrollar. No cabe duda de que este interfaz de usuario irá perdiendo protagonismo conforme el alumno profundice en los aspectos claves de los sistemas de adquisición de datos.

<sup>1</sup> En realidad, si el número de datos no coincide con una potencia de 2 se añaden ceros.

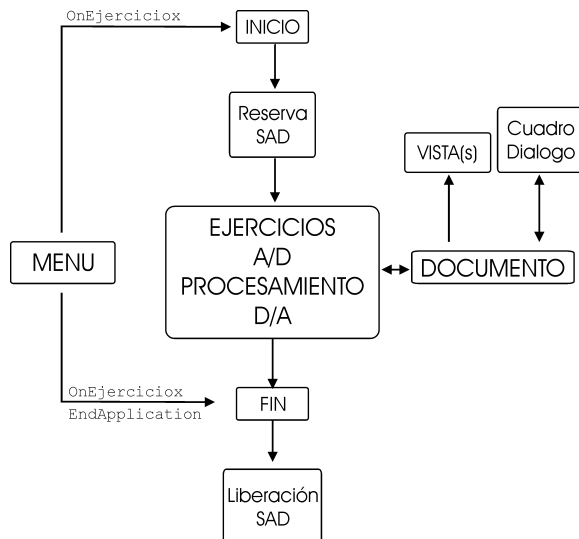


Figura 5: Estructura de la aplicación.

Además de estos cuatro elementos estándar, existe un quinto elemento, el fichero de esqueletos que permite el desarrollo de los ejercicios sin tener en cuenta los aspectos particulares de la jerarquía de clases MFC. En la tabla 1 se muestra el interfaz entre la aplicación y el esqueleto de código para prácticas. Este consiste en un fragmento de código que genera automáticamente el entorno Visual y que conecta el menú de la aplicación con un determinado ejercicio, en este caso el primero ejercicio de la práctica de adquisición. El segundo elemento del interfaz es el esqueleto del ejercicio que debe generar el profesor y que debe completar el alumno. El primer paso que debe realizar el esqueleto es llamar a la función **EntraSecCritica(frame)** que permite establecer el uso en exclusión del SAD e inicializa los canales necesarios para acceder al documento y a la vista de forma transparente. De la misma forma, cuando el usuario decide terminar la ejecución de ejercicio se debe liberar el SAD mediante la llamada a la función **SaleSecCritica()**.

```
void CMainFrame::OnMenuitemej1()
{
    if (!TrabajaMenu())
        return;

    m_iEjercicio_en_ejecucion=1;
    CreateThread(NULL, 0, Ejercicio1, this, 0, &aux_pt);
}

DWORD WINAPI Ejercicio1( LPVOID frame)
{
    EntraSecCritica(frame);

    Do
    {
    }
    while(SalirEjercicio(1));

    SaleSecCritica();
    return 0;
};
```

Tabla 1: Esqueleto de ejercicio

Esta estructura permite trabajar de dos formas diferentes: (1) Se puede adquirir señales, representarlas y utilizar el bucle para mantener la representación de los datos hasta que se decida terminar la aplicación o se pruebe otro ejercicio. (2) Se pueden realizar adquisiciones y representaciones continuas y cíclicas y observar, por ejemplo, el efecto del aliasing.

En el siguiente ejemplo se presenta un ejercicio resuelto que muestra la adquisición y representación de una señal. En este caso, los parámetros de frecuencia de adquisición los recoge de unas variables que establece un cuadro de dialogo de la aplicación y que ya esta programado.

```
DWORD WINAPI Ejercicio3( LPVOID frame)
{
    int count;
    int valor_en_cuentas;
    int muestrasacapturar;

    EntraSecCritica(frame);

    /* Bucle para realizar multiples conversiones */
    do
    {
        /* Selección del canal de conversiön */
        _outp(MULTIPLEXOR,0); // Canal 0
        /* Configurar el rango de entrada */
        _outp(GANANCIA,0); // Rango de entrada a

        Programar_Frecuencia(pDoc->m_iFrecuencia);

        /* Configurar el modo de disparo */
        _outp(MODO,0x06); // Disp TEMPOR., tr. encuesta

        muestrasacapturar=TiempoRepresentacion();

        for(count=0;count<=muestrasacapturar;count++)
        {
            while((_inp(CONVAD_BA)&0x10)==0x10);
            valor_en_cuentas=((unsigned
            int)_inp(CONVAD_BA)&0x000f)<<8+_inp(CONVAD_BB);

            vector[count]=valor_en_cuentas;
            Dibuja(vector,count);
        }

        //Dibuja(vector,count);
        while(SalirEjercicio(3) && AdquisicionContinua());

        SaleSecCritica();
        Return 0;
    }
}
```

Tabla 2: Esqueleto de ejercicio

## 4 DISPONIBILIDAD.

El código fuente de la aplicación puede recogerse en la dirección web: [www.disca.upv.es/pperez/herramie.htm](http://www.disca.upv.es/pperez/herramie.htm). Se trata de un proyecto SDI utilizando la jerarquía de clases Microsoft Foundation Classes MFC. El nombre del proyecto es *PCLAB.DSW* y solo es compatible con Windows 9x y Millenium.

La aplicación funciona tanto con la tarjeta PC-LAB como con su tarjeta clónica NuDAQ. Para su adaptación a otras tarjetas solo hace falta conocer su mapa de puertos y tener la precaución de codificar modos, ganancias y canales que correspondan a la

tarjeta sobre la que se realizan las prácticas. Si bien los SAD's para PC son muy similares a la PC-LAB.

## 5 TRABAJO FUTURO.

Entre el trabajo futuro a realizar está el desarrollo de un driver que permita realizar un acceso simple a bajo nivel mediante operaciones IN-OUT sobre un rango determinado de direcciones en un entorno protegido como WINNT. De este modo no serán necesarias las librerías del fabricante que impiden obtener una imagen real de un sistema de adquisición de datos.

## 6 CONCLUSIONES

En el artículo se ha presentado una aplicación que permite la realización de prácticas para el aprendizaje de técnicas de adquisición de datos y procesado digital en el entorno Windows y utilizando una de las herramientas más avanzadas de programación gráfica sin que el alumno tenga que preocuparse de aspectos del control de la visualización y control de la aplicación. Además, la organización de la aplicación y la potente herramienta de gestión de clases y ficheros que posee el entorno permite de una manera muy sencilla la inserción de nuevas prácticas. Un ejemplo de ello es la progresiva centralización en la aplicación de todas las prácticas de las asignaturas *AYPD* e *INYP* que recurren a la PC-LAB para tratamiento de la señal, en particular, ha resultado muy sencillo la incorporación de dos prácticas:

1. Adquisición de la presión sanguínea, en la que se utiliza una barrera óptica de infrarrojos habitualmente empleada como fin de carrera, pero que en esta práctica se utiliza para medir las variaciones en la transparencia del lóbulo de la oreja debidas al latido cardíaco, utilizando la zona lineal de respuesta del fototransistor, lo que indirectamente permite mostrar en función del tiempo la presión sanguínea. Utilizando la vista *dominio de la frecuencia* permite mediante la observación del armónico principal el cálculo de la frecuencia cardíaca y,
2. Utilizando un circuito similar al de la práctica anterior, pero en conmutación, podemos leer, analizar y visualizar un código de barras utilizando un sensor óptico de infrarrojos trabajando por reflexión. En este caso, gracias a la vista *fuentes* se utiliza el entorno como si de una aplicación C clásica se tratara, para el procesamiento de la señal y obtención del valor de diferentes códigos de barras.

## Referencias

- [1] Benet, G. (1990). "Ejercicios de Instrumentación", Servicio de Publicaciones UPV 90-174.
- [2] Benlloch, J.V., Valera, A. (1993) "Experiencias Prácticas de Instrumentación". Servicio de Publicaciones UPV 93-690.
- [3] Gregory, K. (1998) "Microsoft Visual C++ 6", Prentice Hall.
- [4] Jones, L.D., Chin, A.F. (1991) "Electronic Instruments and Measurements", Prentice Hall.
- [5] Manual de usuario de tarjeta de adquisición NuDAQ.
- [6] Manual de usuario de tarjeta de adquisición PC-LAB.
- [7] Pallás, R. (1993) "Adquisición y Distribución de Señales". Ed. Marcombo.

\*\*\* Comentarle a Gin'es lo del enlace con el  
parrafo anterior