

# APRENDIZAJE INTEGRADO EN SISTEMAS DE ADQUISICIÓN DE DATOS MÁS ALLÁ DEL LABORATORIO

*Ángel Perles, José Albaladejo, Juan Vcte. Capella,  
Juan Miguel Martínez, Houcine Hassan, Carlos Domínguez*

*Universidad Politécnica de Valencia.  
aperles@disca.upv.es*

## RESUMEN

Este artículo presenta una metodología docente que integra satisfactoriamente la lección magistral, los problemas de aula, las prácticas de laboratorio, la evaluación y el trabajo del alumno fuera del horario lectivo.

La lección magistral, los problemas de aula y las prácticas de laboratorio se plantean proponiendo al alumno un proyecto informático real simplificado al inicio del curso cuya resolución es el hilo conductor de toda la docencia.

La evaluación de la asignatura está formada por pruebas distintas en las que cabe destacar el "miniproyecto", un trabajo en grupo que obliga a poner en práctica las aptitudes deseadas.

EL "miniproyecto" implica, además, poder orientar el aprendizaje del alumno fuera del horario lectivo. Para conseguirlo ha sido necesario desarrollar un simulador de sistemas de adquisición de datos que pueden emplear los alumnos en sus casas para desarrollar el trabajo y afianzar los conocimientos.

## 1. INTRODUCCIÓN

La fuerte necesidad de las técnicas informáticas en la industria moderna ha propiciado la introducción de asignaturas como Informática Industrial de 2º curso de la titulación de Ingeniero Técnico Industrial, especialidad Electrónica Industrial impartida en la Escuela Técnica Superior de Ingeniería del Diseño de la Universidad Politécnica de Valencia.

En esta asignatura se vienen realizando una serie de experiencias docentes [1] cuyos objetivos son, entre otros: garantizar unos conocimientos básicos generales, fomentar el esfuerzo del alumno en las actividades prácticas, fomentar la interacción grupal y fomentar la capacidad de adaptación a nuevas metas. Así, los alumnos podrán afrontar los retos competitivos dentro del sector industrial, teniendo conocimiento de la situación tecnológica actual y la capacidad para adaptarse a los constantes cambios que se producen en la aplicación de la informática a la industria.

Actualmente, la docencia se realiza en aulas-laboratorio que facilitan la conexión entre teoría y práctica, siguiendo un criterio de dar la clase en un solo día durante 4 horas: teoría + problemas + práctica. Ello disminuye la dispersión de materias en una jornada y facilita la programación de la asignatura.

Tras varios años de experiencias hemos logrado una metodología que integra satisfactoriamente la lección magistral, los problemas de aula, las prácticas de laboratorio, la evaluación y el trabajo del alumno fuera del horario lectivo. La Figura 1 representa el reparto temporal de cada uno de los aspectos del aprendizaje que se pretenden controlar.

TIEMPO LECTIVO			TIEMPO NO LECTIVO	
Lección magistral	Problemas de aula	Prácticas de laboratorio	Tutorías	Trabajo/estudio

Figura 1. Reparto temporal de los distintos aspectos del aprendizaje.

En la lección magistral, los problemas de aula y las prácticas de laboratorio se le plantea al alumno un proyecto informático real simplificado [2] al inicio del curso. La resolución de dicho problema es el hilo conductor de toda la docencia durante el curso.

La evaluación de la asignatura es otro pilar de la metodología, y está formada por pruebas distintas en las que cabe destacar el 'miniproyecto', un trabajo en grupo que obliga a poner en práctica las aptitudes deseadas.

El miniproyecto facilita la comprensión de la materia y el saber aplicarla prácticamente. Sin embargo, este trabajo requiere de laboratorios adecuadamente equipados y disponibles para los alumnos, lo cual no es factible en asignaturas masificadas como la nuestra.

Para obtener los máximos beneficios del miniproyecto hemos creado un simulador [3,4] de sistemas de adquisición de datos que permite al alumno trabajar el miniproyecto fuera de nuestro laboratorio, sólo necesita un computador personal, disponible en la mayor parte de hogares del alumno. Con esta iniciativa se consigue orientar también el trabajo fuera del horario lectivo de la asignatura.

El simulador y su código fuente en Borland C++ Builder se pueden obtener en la dirección <http://www.disca.upv.es/aperles/simseny>.

El resto del artículo se estructura de la siguiente forma: primero se presenta el enfoque de la docencia de la asignatura desde el punto de vista del proyecto, a continuación se describe el *miniproyecto* como parte de la evaluación de la asignatura, el desarrollo del *miniproyecto* es problemático, pero se ha superado la problemática con el desarrollo de un simulador de sistemas de adquisición de datos que permite a los alumnos trabajar en sus casas sin necesidad de recurrir al laboratorio de la asignatura, este simulador se describe en el apartado 4. Finalmente se sintetizan las ventajas e inconvenientes de esta propuesta en el apartado de conclusiones.

## 2. ENFOQUE DE LA DOCENCIA

La asignatura Informática Industrial se plantea de formas muy diferentes en las distintas Universidades españolas. En nuestro caso particular, uno de los objetivos principales es que el alumno pueda afrontar el diseño y desarrollo de entornos sencillos del estilo de la Figura 2, donde un sistema informático debe ser capaz de interactuar con un proceso mediante el uso de sensores y actuadores (interfaz con el proceso) y con un operador (interfaz de operador).

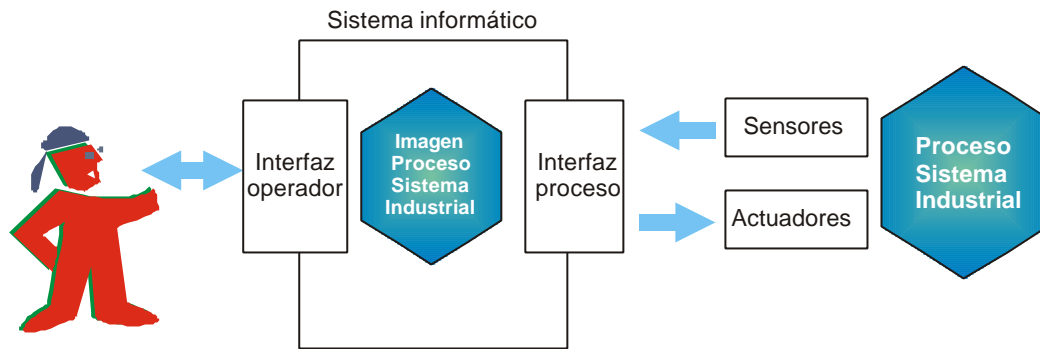


Figura 2. Esquema del sistema informático industrial

El enfoque de la docencia para conseguir este objetivo está fuertemente limitado por los conocimientos previos del alumno, restringiendo, en la mayoría de los casos, las técnicas y herramientas que se pueden emplear; pues podría llegar a confundirse los propósitos de la asignatura con la explicación de una determinada herramienta o técnica formal de diseño de sistemas informáticos industriales. En nuestro caso particular, el alumno tiene los suficientes conocimientos de electrónica analógica y digital, y conocimientos muy básicos de programación en C practicados en el entorno Borland Turbo C.

Para afrontar con éxito las limitaciones del alumnado y obtener el máximo provecho, la idea central ha sido emplear un *proyecto real simplificado* que permite plantear al alumno un problema general básico al inicio del curso, y usar su resolución como base para los conocimientos teóricos y para fomentar/potenciar las aptitudes deseadas.

Para que la complejidad del proyecto no oscurezca los verdaderos propósitos, éste debe ser sencillo y abarcar todas las posibles necesidades. Nosotros hemos elegido un sencillo control de nivel y temperatura de un depósito cuyo diagrama se muestra en la Figura 3.

El proceso consiste en un depósito de líquido que debe estar a una determinada temperatura para poder disponer del líquido a través de una válvula. Una termorresistencia se encarga del calentamiento del líquido y una bomba de suministrar líquido al depósito. Sensores de nivel y temperatura permiten conocer la cantidad de líquido en el depósito y su temperatura.

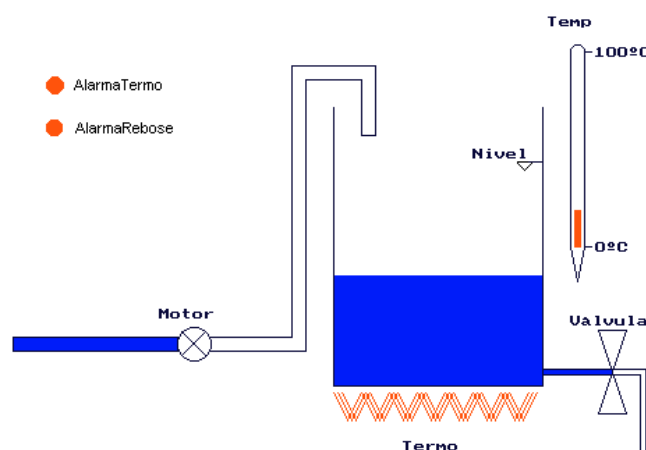
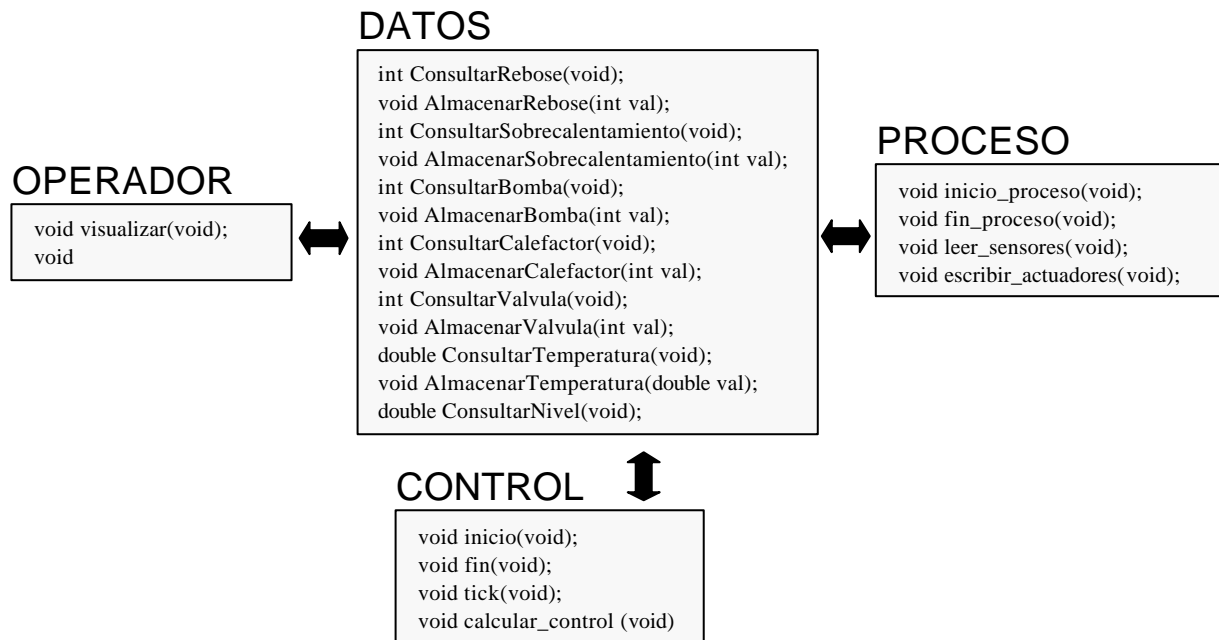


Figura 3. Diagrama del proceso

Partiendo de los conocimientos previos de los alumnos y con el objetivo de que aprendan a diseñar y desarrollar un proyecto con una interfaz adecuada con el usuario y con el proceso, se

emplean metodologías no formales de diseño *top-down*, donde se hace especial hincapié en las reglas heurísticas de “*descomposición modular*” y de “*ocultación de la información*”.

Con las anteriores reglas se les propone lo que podría ser una descomposición modular básica mínima en forma de módulos C. La Figura 4 representa esta descomposición.

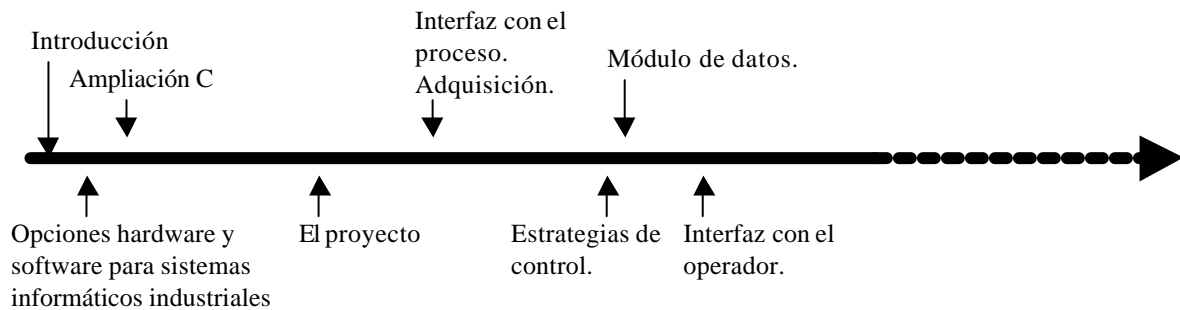


**Figura 4. Estructura modular propuesta**

Cada módulo tiene un propósito muy específico, que es:

- ? El *módulo de datos* contiene una imagen del estado del proceso, por ejemplo, la temperatura del líquido del depósito, la temperatura deseada del líquido, etc., y unas funciones que permiten depositar en el módulo y consultar esta imagen. El propósito de este módulo es evitar la interdependencia entre el resto de módulos, facilitando mucho el desarrollo por separado de cada uno de ellos y permitiendo, como efecto beneficioso para la docencia, que pueda explicarse cada aspecto de la aplicación por separado.
- ? El *módulo de proceso* se encarga de implementar la interfaz con el proceso. Debe ser capaz de emplear un sistema de adquisición de datos determinado y unos sensores y actuadores para conseguir depositar los valores recogidos por los sensores en el módulo de datos y modificar el proceso mediante actuadores, reflejando el estado deseado del proceso representado en el módulo de datos.
- ? El *módulo de operador* se encarga de implementar la interfaz con el operador, representado gráficamente el estado del proceso y permitiendo la introducción de consignas de funcionamiento.
- ? Finalmente, el *módulo de control* se encarga de ejecutar las distintas tareas (funciones C) que forman la aplicación en un determinado orden y a intervalos predefinidos de tiempo. Estas tareas son: lectura de los sensores, cálculo de la acción de control, modificación de actuadores y actualización de la interfaz de usuario. También se encarga de implementar una estrategia de control (un regulador PID, una tabla de verdad, etc.).

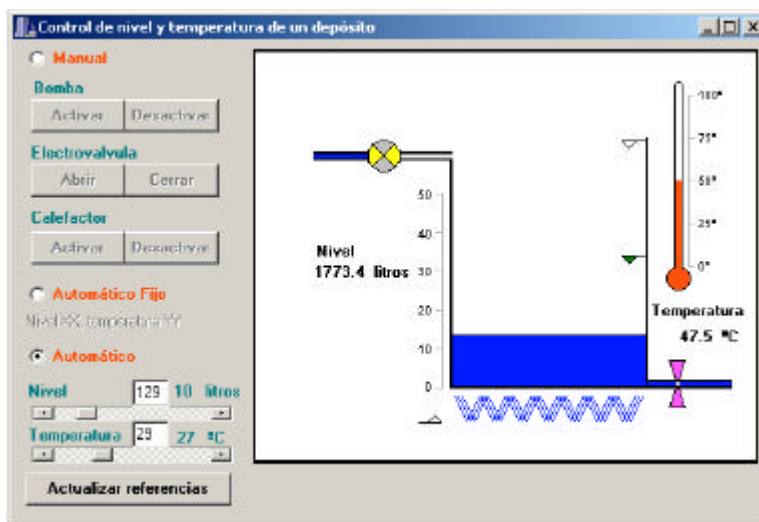
El uso del proyecto real simplificado como hilo conductor de la docencia y la propuesta de descomposición modular acoplan perfectamente para planificar temporalmente la docencia. La Figura 5 representa esto de manera gráfica.



**Figura 5. Distribución temporal de los contenidos de la asignatura**

Para la docencia se elige como sistemas microprocesador el computador tipo PC, para el sistema operativo se elige Windows, como entorno de desarrollo se elige Borland C++ Builder y como sistema de adquisición de datos se emplea una tarjeta de adquisición de datos modelo PCI-9112 de ADLink. Dicha tarjeta posee entradas y salidas digitales, entradas y salidas analógica y contadores.

La aplicación de la propuesta lleva a conseguir un programa con el aspecto mostrado en la Figura 6(a) y que puede ser probado en una maqueta como la mostrada en la Figura 6 (b).



a)



b)

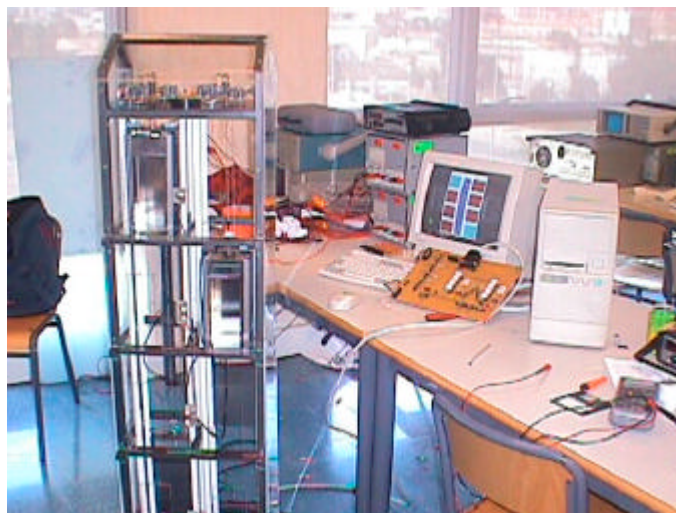
**Figura 6. Aspecto de la aplicación (a) y maqueta de pruebas (b)**

### 3. LA EVALUACIÓN MEDIANTE EL "MINIPROYECTO"

Uno de los pilares fundamentales de la metodología empleada en la asignatura recae en la evaluación, que pretende facilitar que se induzcan las aptitudes descritas en el apartado de introducción.

Es difícil lograr estos objetivos empleando sólo pruebas escritas clásicas, por lo que se ha optado por emplear una combinación de distintas pruebas: un test eliminatorio (20%), una prueba de problemas (50%), y un *miniproyecto* (30%). Cabe destacar el *miniproyecto*, que es un proyecto sencillo de sistema informático industrial que el alumno ha de resolver en grupo y presentar públicamente. De esta forma, además del avance en el aprendizaje de temas relacionados con la especialidad, los alumnos adquieren otras habilidades como el trabajo en grupo, la exposición pública de ideas, la redacción técnica de documentos, etc.

El sistema informático industrial a desarrollar deberá emplear como herramientas un ordenador personal, el entorno de programación Borland C++ Builder para Windows y la tarjeta de adquisición de datos ADLink modelo PCI-9112, de manera que puedan basarse fácilmente en el proyecto ejemplo empleado en la docencia de la asignatura. La Figura 7 muestra un ejemplo del resultado; una maqueta de ascensor diseñada por los alumnos y controlada por el software desarrollado por ellos.



**Figura 7. Ejemplo de trabajo realizado por alumnos**

El desarrollo del miniproyecto se sincroniza con la docencia, concretándose y refinándose a través de sucesivas entregas, que permiten al profesor supervisar el trabajo de grupo y realimentarlo corrigiendo los posibles defectos.

Estas entregas corresponden a:

- El documento de requerimientos del proyecto.
- El documento de especificaciones.
- La propuesta de descomposición en módulos.
- Cada uno de los módulos desarrollados individualmente.
- La integración de los módulos en la aplicación final.
- La guía de usuario de la aplicación desarrollada.

Esta propuesta adolece de un problema importante, que es la necesidad de laboratorios capacitados para el desarrollo de la aplicación disponibles para el alumnado. Ello supone un enorme costo (material y de personal) que no es actualmente asumible por una asignatura troncal bastante masificada como la nuestra.

#### **4. SUPERACIÓN DE LAS LIMITACIONES MEDIANTE UN SIMULADOR**

La experiencia de estos años nos ha mostrado que la disponibilidad, en condiciones adecuadas, de la tarjeta de adquisición de datos es el mayor handicap para obtener los mejores frutos de este enfoque. Por otro lado, las aulas informáticas y los domicilios de los alumnos disponen de parte de la infraestructura necesaria, que no es más que el ordenador personal y el entorno de programación. Únicamente faltan la tarjeta de adquisición de datos y los dispositivos conectados a ellas.

Nos hemos planteado la pregunta de si es posible desarrollar la aplicación sin disponer de la tarjeta de adquisición de datos. La solución consiste en lograr un simulador que permita desarrollar la aplicación informática sin la necesidad de dicha tarjeta. Dicho simulador permitirá lograr los siguientes objetivos:

- ? Los alumnos pueden hacer parte del desarrollo de su aplicación en su casa o en aulas de informática, consiguiéndose así orientar mucho mejor el trabajo no lectivo del alumno, pues puede practicar los aspectos fundamentales de la asignatura sin necesidad de acudir al laboratorio.
- ? El alumno se centra en el problema del diseño y desarrollo de la aplicación, sin invertir un esfuerzo excesivo en el montaje físico de los circuitos electrónicos conectados a la tarjeta.
- ? La menor necesidad de tarjetas de adquisición de datos permite aplicar esta aproximación a asignaturas masificadas como la nuestra.
- ? Disminuyen los fallos y averías debidos a problemas de conexionado con la tarjeta o de inadecuado diseño de los circuitos de acondicionamiento de sensores y actuadores.
- ? Se reservan las tarjetas de adquisición de datos para las prácticas de laboratorio y para las etapas finales del proyecto.

Como premisas de su diseño del simulador, se ha pretendido que sea transparente al programador, independiente del modelo particular de sistema de adquisición de datos, y adecuado a distintos entornos de programación bajo Windows.

El resultado ha sido un simulador que imita las señales eléctricas físicas que se conectan a los terminales de una tarjeta de adquisición de datos. Con ello se puede desarrollar y probar un programa que use una determinada tarjeta de adquisición de datos sin tener que recurrir a instalar la tarjeta y sin necesidad de montar dispositivos en las entradas/salidas de la tarjeta. Una vez terminado el programa, éste se puede probar sobre un computador que posea la tarjeta de adquisición de datos y conectar a dicha tarjeta los dispositivos que se desee controlar.

Para ilustrar el planteamiento del simulador, la Figura 8 muestra un ejemplo típico de diagrama de bloques de los componentes que intervienen en una aplicación informática industrial que emplee un sistema de adquisición de datos. Normalmente, la aplicación a desarrollar interactúa con los *drivers* de la tarjeta a través de una biblioteca suministrada por el fabricante. Dicha biblioteca es la que accede al hardware del sistema de adquisición de datos.



Figura 8. Relación entre los distintos bloques que componen la aplicación

La Figura 9 muestra como se introduce el simulador en la aplicación. Todas las llamadas a las funciones de biblioteca de la tarjeta son capturadas, pudiéndose entonces redirigir al simulador o continuar usando la tarjeta.

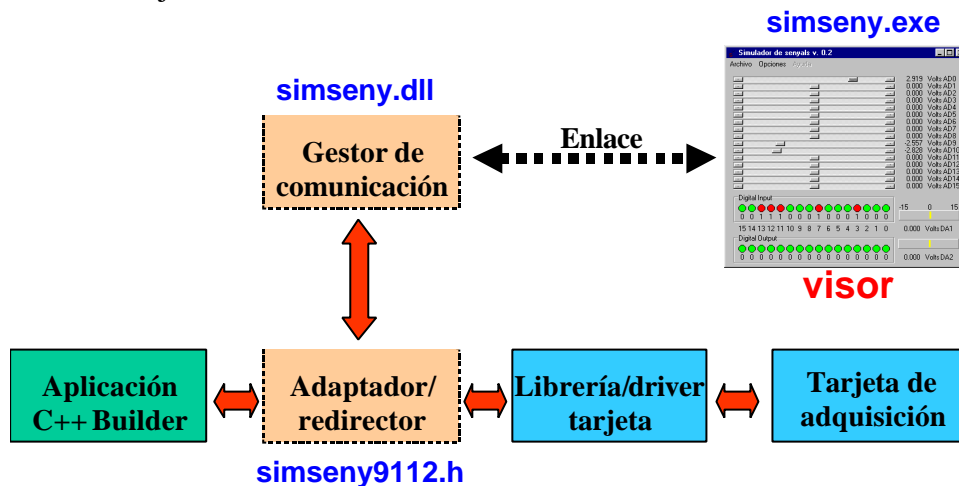


Figura 9. Relación del simulador con la aplicación.

Los elementos que componen el simulador son:

- ? Un **fichero de cabecera** específico para el modelo de tarjeta de adquisición de datos utilizada, que sobrecarga las funciones que hacen uso de la tarjeta y permite redirigir éstas al simulador, convirtiendo previamente la información de control de la tarjeta a un formato independiente.
- ? Una **librería de enlace dinámico** (DLL), que se carga sólo cuando se requiere el uso del simulador y que permite la interacción de la aplicación de control con la interfaz de usuario del simulador. El uso de una DLL para el enlace con el simulador hace que esta herramienta sea aplicable a entornos de programación como Microsoft Visual C++, Borland C++ Builder, Microsoft Visual Basic, Borland Delphi, National Instruments LabView/LabWindows, HP VEE, etc.
- ? Un **visor de señales**, que permite mostrar las salidas físicas de la tarjeta e introducir valores físicos en las entradas.

La Figura 10 muestra el aspecto del visor del simulador. Este imita las entradas analógicas y digitales y las salidas analógicas y digitales de una tarjeta de adquisición de datos al nivel de señales físicas que se conectan a los terminales de la tarjeta. Esta interfaz ha sido cuidadosamente diseñada para facilitar su uso por los alumnos de forma intuitiva.



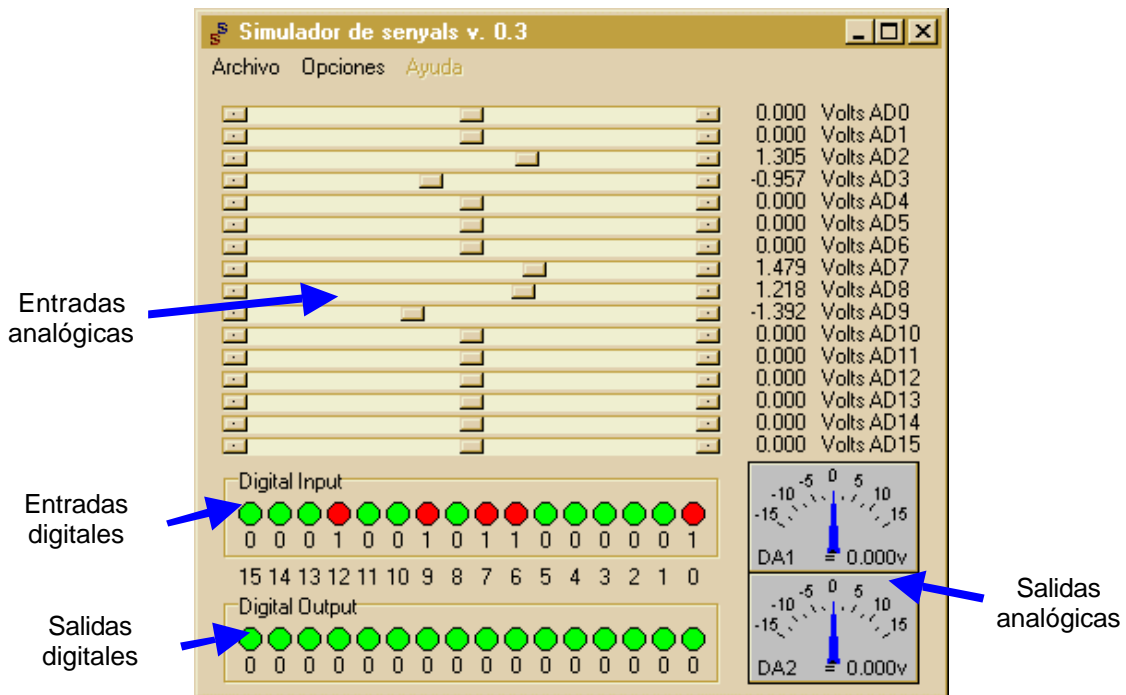


Figura 10. Visor de entradas y salidas.

Para facilitar aún más el aprendizaje del alumno, el simulador incorpora maquetas virtuales que imitan el comportamiento físico de un determinado proceso. Por ejemplo, la Figura 11 muestra la ventana de la maqueta virtual de un motor conectado a un encoder. Esta maqueta es la que el grupo de alumnos debe controlar en el miniproyecto planteado en el presente curso académico.

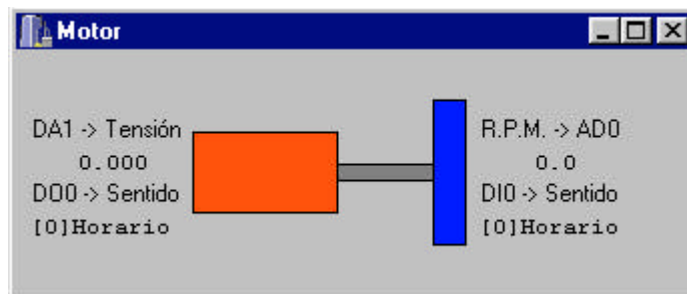


Figura 11. Maqueta virtual de un control de velocidad de motor.

Con el simulador se proporciona un fichero de cabecera (`simseny9112.h`) particular para su uso en aplicaciones para la tarjeta de adquisición de datos ADLink PCI-9112. Los programas que se desarrollen para dicha tarjeta y empleen dicho fichero podrán emplear tanto la tarjeta como el simulador.

Para que el programa desarrollado para la tarjeta PCI-9112 pueda ser usado con el simulador basta con añadir el fichero de cabecera `simseny9112.h` inmediatamente después del fichero de cabecera `9112.h` suministrado por el fabricante. Por ejemplo:

```
#include "9112.h"
#include "simseny9112.h"
```

Con este cambio, la aplicación desarrollada por el alumno preguntará al principio si se desea emplear el simulador o la tarjeta.

## 5. CONCLUSIONES

La metodología presentada aquí es una suma de métodos clásicos como la lección magistral, las técnicas orientadas al problema, el trabajo en grupo, distintas técnicas de evaluación, etc. que permiten controlar todos los aspectos del aprendizaje del alumno.

Esta aproximación está en consonancia con las expectativas de la propuesta de futura reforma de la Universidad española y de su adaptación al marco Europeo; siendo inmediata su adaptación al crédito ECTS.

La metodología es el fruto de más de 8 años "experimentando y refinando" con nuestros alumnos y, al final, tiene sus ventajas e inconvenientes. Como ventaja, la metodología aquí propuesta consigue que el protagonista de la docencia sea el alumno, pasando a ser el profesor un orientador de este aprendizaje, consiguiendo fomentar en los alumnos los objetivos expuestos en la introducción.

La principal desventaja es el exceso de trabajo que supone, tanto para los alumnos como para los profesores, en relación con la enseñanza tradicional basada en la lección magistral. Nosotros lo aplicamos en enormes aulas-laboratorio con más de 60 alumnos, que suponen un agotador esfuerzo por parte del profesor. Esta limitación desaparecería con grupos de alumnos más pequeños.

## 5. BIBLIOGRAFÍA

- [1] J. Albaladejo, A. Perles, C. Dominguez, H. Hassan, J.M.Martínez. Aprendizaje activo. Alternativas a la lección magistral. Actas 1er Congreso Internacional: "Docencia Universitaria e Innovación". 2.000
- [2] A. Perles, C. Dominguez, J.M.Martínez, H. Hassan. Enseñanza de la Informática Industrial mediante Proyectos Reales Simplificados. Actas V Jornadas Sobre La Enseñanza Universitaria En Informática (JENUI'99). 1.999
- [3] A. Perles, J.M.Martínez, C. Dominguez, J. Albaladejo. SimSeny: Un simulador didáctico para tarjetas de adquisición de datos. Actas IV Congreso de Tecnologías Aplicadas a la Enseñanza de la Electrónica (TAEE2000). 2.000
- [4] Angel Perles; Juan M. Martínez; Houcine Hassan; José Albaladejo; Carlos Domínguez. Descripción del Simulador de Tarjetas de Adquisición de Datos SimSeny. Actas del congreso JENUI 2001. 2.001