

ADKI: UN SISTEMA WEB DE ADQUISICIÓN DE DATOS BAJO LINUX

*Pedro Jurado Maqueda, Jorge Juan Chico¹, Paulino Ruiz de Clavijo Vázquez¹
Manuel J. Bellido Díaz¹, Alejandro Millán Calderón¹
David Guerrero Martos¹, Enrique Ostua Arangüena*

Departamento de Tecnología Electrónica. Universidad de Sevilla.
E.T.S. Ingeniería Informática. Av. Reina Mercedes s/n. 41012 Sevilla. España.
Tlf: +34 - 954 55 61 61. Fax: +34 - 954 55 27 64

¹También con el Instituto de Microelectrónica de Sevilla
Centro Nacional de Microelectrónica -CNM-CSIC- España

melenas@kdehispano.org, {jjchico,paulino,bellido,amillan,guerre,ostua}@dte.us.es

Resumen

Esta contribución presenta una aplicación genérica de adquisición de datos y control para sistemas simples que incorpora el registro histórico de la actividad del sistema y la capacidad de consulta y gestión de la aplicación de forma remota vía web. Las características del sistema lo hacen idóneo para su aplicación en el control de granjas o la gestión de pequeños edificios: iluminación, calefacción, ventilación, etc. La aplicación ha sido desarrollada íntegramente mediante software libre en el marco de un proyecto fin de carrera de la titulación de Ingeniería Técnica en Informática de Sistemas de la Universidad de Sevilla, mostrando la idoneidad de este tipo de plataforma tanto en el ámbito industrial como académico.

1. INTRODUCCIÓN

Desde hace tiempo, el ordenador es un elemento clave en muchos sistemas de control. La gran ventaja del uso del ordenador se derivan de sus características como sistema de propósito general que puede ser programado para realizar la tarea de control requerida pero que además, puede llevar a cabo multitud de tareas complementarias como el registro de la actividad y la generación de informes e históricos que permiten, entre otras cosas, el estudio de la operación del sistema, optimización, corrección de errores, etc. La facilidad de uso de los sistemas informáticos actuales ofrece también atractivas opciones como la posibilidad de adaptación del sistema a nuevas necesidades de forma rápida.

Por otra parte, el extraordinario desarrollo de las redes de ordenadores y en especial el de Internet permite incorporar a los sistemas de control basados en ordenador capacidades adicionales y de gran utilidad como la monitorización y gestión remota del sistema, basada en la conexión a través de la red al ordenador encargado del control.

De este modo, las posibilidades del ordenador hacen posible imaginar y llevar a la práctica sistemas de control con características inimaginables hace pocos años. Los relativamente bajos costes de los sistemas informáticos actuales hacen que estas capacidades estén, además, al alcance de pequeñas y medianas empresas e incluso de usuarios particulares.

Uno de los aspectos fundamentales que motiva este trabajo es mostrar que en la actualidad es posible la implementación de un sistema de control basado en ordenador con características avanzadas como las descritas anteriormente: registro de actividad, monitorización y configuración remota, etc. con costes bajos y asequibles. La clave en cuanto a la reducción de costes se basa, por una parte, en los precios de los sistemas informáticos actuales y en el uso de software libre [1], que puede ser adquirido por un coste marginal. Este tipo de software, el software libre, ha sufrido un espectacular desarrollo en los últimos años, proporcionando hoy en día un amplio conjunto de herramientas y aplicaciones de alta calidad: sistemas operativos, servidores de Internet, bases de datos y, en general, todos los componentes necesarios para la realización de un sistema de control basado en ordenador.

El sistema descrito en esta contribución es relativamente sencillo pero suficiente para un buen número de aplicaciones de control que no precisen de una respuesta en tiempo real. El sistema, por una parte, recoge datos a través de un interfaz de adquisición de datos y permite accionar diversos actuadores en función de los datos de entrada. El significado de las señales de entrada así como de las reglas de actuación son completamente configurables, lo que permite la adaptación del sistema a múltiples aplicaciones prácticas como el control de granjas, sistemas de riego, iluminación, calefacción, gestión de pequeños edificios, etc. Uno de los aspectos más interesantes del sistema es que permite la consulta de datos y acceso a las reglas de control mediante una interfaz web, lo que permite el acceso desde cualquier lugar con acceso a Internet que disponga de un navegador web estándar.

Es importante destacar que el trabajo presentado se desarrolló como tema de Proyecto Fin de Carrera para la obtención del título de Ingeniero Técnico en Informática de Sistemas de la Universidad de Sevilla durante el curso 2001/02 obteniendo la máxima calificación. Este hecho muestra las posibilidades del software libre y de las plataformas abiertas para ser usadas en el ámbito docente para el desarrollo de proyectos aplicados. La madurez alcanzada por estas plataformas permite, por una parte, la formación de los alumnos en proyectos tecnológicos y, por otra parte, la obtención de resultados extrapolables al sector productivo, como los que se presentan en esta contribución.

El resto de esta contribución se organiza como sigue: en el siguiente apartado se presenta una descripción general del sistema, su arquitectura e implementación. En el tercer apartado se describe el uso y configuración del sistema. En el cuarto apartado se esbozan algunas de las aplicaciones del sistema desarrollado. Finalmente se resumen las conclusiones más relevantes.

2. DESCRIPCIÓN DEL SISTEMA

2.1. Requisitos funcionales

Los requisitos funcionales del sistema pueden resumirse en los siguientes puntos:

- El sistema debe adquirir datos desde las entradas analógicas de una tarjeta de adquisición de datos. Estos datos pueden ser almacenados y podrán tomarse decisiones de control en función del valor de los mismos.

- El intervalo con el que se monitorizan y se almacenan los datos de entrada debe ser configurable. Los datos adquiridos por el sistema deben poder transformarse mediante operaciones sencillas antes de ser almacenados con objeto de convertirlos a valores con un significado físico útil.
- Un sistema de activación/desactivación actuará sobre las salidas digitales de la tarjeta de adquisición de datos en función de unas reglas programables y dependientes de los valores de las señales de entrada.
- El sistema debe ser independiente del hardware de adquisición de datos concreto.
- El sistema será accesible remotamente mediante una interfaz web pudiendo usarse para ello cualquier navegador web convencional.
- El acceso web debe permitir la lectura cómoda del histórico de datos almacenados, así como la configuración general del sistema en sí y de las reglas de control.
- El acceso al sistema dispondrá de un mecanismo de control mediante nombre de usuario y contraseña.
- Con el fin de cumplir los objetivos generales mencionados en la Introducción, el sistema debe ser implementado en su totalidad empleando software libre.

2.2. Arquitectura del sistema

El sistema desarrollado ha sido denominado ADKI. La arquitectura general del sistema ADKI se muestra en la figura 1.

El sistema se puede dividir en dos partes bien diferenciadas:

Una sería un programa binario que ejecutado en forma de demonio se encarga de leer las configuraciones que describen las entradas y salidas utilizadas así como su relación entre ellos para que dependiendo de los valores de entrada, se active o desactive las salidas. Además se encarga de introducir las medidas tomadas por las entradas en la base de datos (tras ser tratadas por una fórmula matemática para ajustarla a una medida real) así como el intervalo de tiempo entre cada medida.

Por la otra parte estaría la aplicación web que permitiría consultar las medidas tomadas por la aplicación, diferenciando además mediante autenticación por contraseña el tipo de consultas que puede hacer el usuario, llegando incluso el caso de que si tiene suficientes permisos, pueda cambiar remotamente los archivos de configuración para un mejor ajuste.

2.3. Implementación

La figura 2 muestra en detalle los componentes que forman la implementación del sistema. Estos componentes se describen a continuación:

- Una tarjeta de adquisición de datos compatible con Comedi [2] y con al menos 16 entradas analógicas.
- Drivers Comedi: Es un paquete software desarrollado en la Universidad de California en Berkeley que nos sirve para controlar, leer y escribir datos con una tarjeta de adquisición bajo Linux gracias a que integra una cantidad enorme de controladores para un gran número de tarjetas de adquisición de datos para Linux, y que en

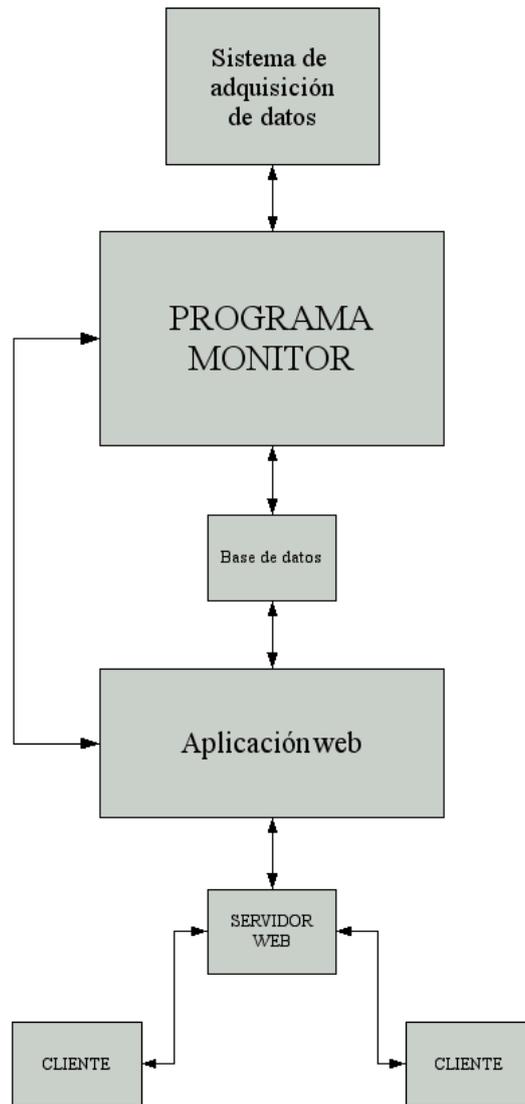


Figura 1: Diagrama de bloques general de ADKI.

la actualidad reúne drivers para más de 250 tarjetas de fabricantes como National Instruments, Advantech, Adlink y otros. Además proporcionan un esqueleto para que los usuarios puedan escribir sus propios controladores con compatibilidad completa con el propio Comedi. Y por el otro lado tenemos

- Librerías Comedi: las librerías Comedi, que se encargan de proporcionarnos un paquete bastante extenso de funciones en C para el control, lectura y escritura de datos de las tarjetas de adquisición, abstrayéndonos así del modelo en particular. Así, al utilizarlo, nos aseguramos de que la aplicación que utiliza esas funciones, debería funcionar sin problemas en una tarjeta de otro modelo e incluso distinto fabricantes, siempre que esté soportado por Comedi, y utilicemos sus controladores para instalarla. Como todo lo anterior es software libre.
- Programa ADKI: El corazón de todo el sistema y se encarga de todo el manejo de la propia tarjeta y de los datos, según lo indicado en los archivos de configuración `adki.conf` y `adki_out.conf`. Por motivos de eficiencia, está programado en C y emplea

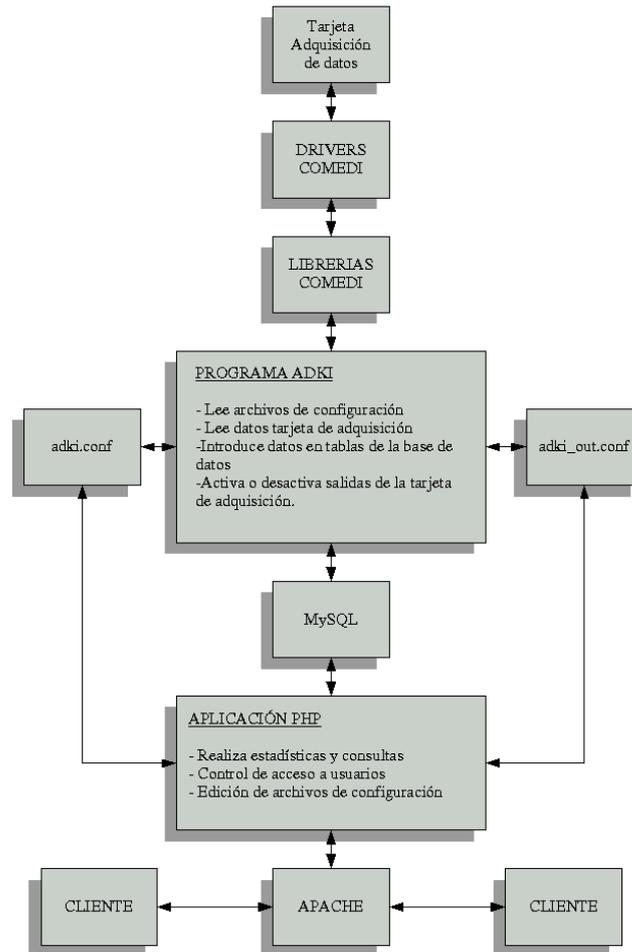


Figura 2: Diagrama de bloques de la implementación realizada de ADKI.

el sistema de temporizadores propio de UNIX [3] para la realizar las tareas de monitorización y toma de datos.

- MySQL [4]: Es un sistema gestor de base de datos totalmente compatible con el lenguaje SQL y también como todo lo anterior, software libre. Entre sus ventajas están la inclusión de unas librerías propias para el lenguaje C, y la gran rapidez de inserción y búsqueda de datos en las bases de datos con un costo de recursos hardware mínima. Por el contrario tiene el inconveniente de que el bloqueo y desbloqueo de tablas para el acceso concurrente en escritura no está muy bien implementado, aunque en nuestro caso, esto no tiene mucha relevancia, al haber sólo una conexión activa que altere la base de datos, y que será el programa ADKI. Este inconveniente no afecta a la posibilidad de realizar diversas consultas simultáneas desde varios clientes WEB.
- Aplicación PHP: Se encarga de generar las interfaces mediante PHP que es un lenguaje de alto nivel para programación de páginas web dinámicas (de creación en el mismo momento de la consulta). Su sintaxis es muy parecida al C, y su inclusión

en el código HTML nativo es muy fácil de hacer [5, 6]. La elección de este lenguaje se debe a las siguientes razones:

- Gran interacción con otros lenguajes, incluso con funciones propias para su manejo, al contrario de, por ejemplo Perl.
 - Todas las acciones se realizan en el servidor, con lo cual el cliente no se ve sobrecargado de trabajo ni compromete su propia seguridad al ejecutar código extraño que no se ve directamente.
 - Totalmente libre y no dependiente de la plataforma (al contrario que ASP).
- Apache [7]: Servidor web rápido, seguro y libre, líder en su campo y que se integra perfectamente con PHP y MySQL.

2.4. Herramientas y entorno de desarrollo

El proyecto se ha desarrollado sobre plataforma IBM-PC con sistema operativo GNU/Linux [8, 9]. Aunque no se han hecho pruebas, el sistema debería funcionar sin cambios sobre cualquier otra arquitectura soportada por GNU/Linux siempre que el sistema de adquisición de datos esté soportado por Comedi en esa arquitectura.

Para el desarrollo del proyecto se han empleado dos de las distribuciones GNU/Linux más populares del momento como son Debian [10] y Red Hat [11], ambos con núcleos del sistema operativo personalizados de la serie 2.4.x.

Las herramientas utilizadas para el desarrollo han sido las siguientes, siendo todas software libre:

- KDevelop [12]: Sin duda el IDE libre para Linux por excelencia, integrado en el sistema de escritorio de KDE, nos permite realizar proyectos en lenguajes como C, C++ y Objective-C , además de que en la nueva versión a publicar, se añadirán otros como Java e incluso PHP, aunque sólo si está compilado como cgi-bin. Además de tener resaltado de sintaxis, depuración, trazas y prácticamente todo lo que se le pueda pedir a un interfaz de programación.
- Para la edición del código PHP se ha empleado el programa Vim (*vi improved*) un clónico mejorado del Vi con resaltado de sintaxis que facilita la localización de errores simples como el cierre de comillas o el emparejamiento de corchetes. También se trata de software libre.
- En la parte de MySQL se ha empleado el propio monitor de sistema de la aplicación. Aunque la aplicación se encarga de crear la base de datos y todas las tablas, además de introducir los datos automáticamente sin intervención alguna de usuario o administrador, el monitor se utilizó para comprobar y corregir la sintaxis de los comandos introducidos por el programa así como la correcta creación de las tablas y de la introducción de datos.
- Para el análisis léxico y sintáctico de los archivos de configuración se han empleado las herramientas *flex* y *bison* respectivamente, ambas pertenecientes al proyecto GNU. Estos dos programas son los herederos de los originales *lex* y *yacc*, presentes en la mayoría de los sistemas UNIX, mejorándolos en diversos aspectos.

| #canal | nombre | unidad | fórmula | actualización |
|--------|---------|---------|----------|---------------|
| 0 | Ruido | 'db' | 3+log(x) | 25 |
| 3 | Temper | 'C°' | 1/(x+1) | 46 |
| 7 | Humedad | '%' | sin(x+3) | 90 |
| 1 | Luz | 'Lumen' | sqrt(x) | 10 |

Figura 3: Ejemplo de archivo adki.conf.

```

if Temper > 40 active 3;
if Temper > 30 and Humedad > 80 active 3;
if Temper < 40 deactivate 3;
if Temper < 30 and Humedad < 80 deactivate 3;
if Luz > 3000 active 2;
if Luz < 3000 deactivate 2;

```

Figura 4: Ejemplo de archivo adki_out.conf.

- Como ya hemos comentado, el servidor web que se ha utilizado es el popular Apache, por su alta integración con MySQL y PHP, que viene “de serie” con el propio paquete binario del servidor, sin tener que recompilarlo, con el consiguiente ahorro de tiempo y esfuerzo.

3. USO DEL SISTEMA

3.1. Monitor de adquisición de datos

El programa de configuración se maneja mediante dos archivos de texto simple, uno de configuración general *adki.conf* y otro que controla la activación/desactivación de salidas digitales llamado *adki_out.conf*.

Viendo el ejemplo del archivo *adki.conf* (figura 3) observamos que hay cinco campos:

- *#canal* indica a qué canal físico nos estamos refiriendo.
- *nombre* es un identificador del canal.
- *unidad*, nos dará la unidad en la que se mide ese canal.
- *fórmula* define la fórmula a aplicar al dato en voltios para obtener el dato final.
- *actualización* es el tiempo que transcurrirá entre dos inserciones de datos en la tabla correspondiente al canal.

Por otro lado, el archivo *adki_out.conf* tiene el esquema que se muestra en la figura 4.

Como podemos ver, se trata de unas comparaciones de las medidas que se recogen en las entradas analógicas, que si se cumplen ejecutan la acción que se indica al final de la línea, pudiendo ser la activación (*active*) o desactivación (*deactive*) de cualquier salida digital, indicando ese número el canal físico.

| .: Medidas actuales :. | | |
|------------------------|-----------------|--------|
| Logout (salir) | | |
| CANAL | MEDIDA | |
| Humedad | -19660.50000000 | % |
| Ruido | 5.99984741 | dB |
| Luz | 2.99984741 | Lumen2 |
| Viento | 5.00015259 | m/s |

| .: Escoger acción :. | |
|-----------------------------|-----------------------|
| Su nombre de usuario: pedro | Su nivel de acceso: 3 |

Figura 5: Interfaz web. Acceso de usuario de nivel 3.

| .: Medidas actuales :. | | |
|------------------------|-----------------|--------|
| Logout (salir) | | |
| CANAL | MEDIDA | |
| Humedad | -19660.50000000 | % |
| Ruido | 5.99984741 | dB |
| Luz | 2.99984741 | Lumen2 |
| Viento | 5.00015259 | m/s |

| .: Escoger acción :. | |
|--|-----------------------|
| Su nombre de usuario: Jorge | Su nivel de acceso: 2 |
| <input type="button" value="Histórico"/> | |

Figura 6: Interfaz web. Acceso de usuario de nivel 2.

3.2. Aplicación web

El interfaz web sólo puede ser accedido tras dar un nombre de usuario y contraseña. Según el nivel de usuario se nos permite hacer las siguientes acciones:

- Nivel 3: Como se puede comprobar en la figura 5, tan sólo se permite comprobar la última medida realizada, sin más opciones.
- Nivel 2: En la figura 6, se puede ver que, además de lo incluido en el nivel 3, también se permite acceder al histórico mediante el botón que está justo debajo de las últimas medidas realizadas. Esto nos llevaría a otro menú en el que podríamos elegir el intervalo de tiempo a examinar.
- Nivel 1: Se puede apreciar en la figura 7 que, además de lo incluido en el nivel 2, también se permite gestionar los usuarios añadiendo los nuevos que sean necesarios y borrar los que no sean ya útiles; así como cambiar los permisos a los usuarios ya

| .: Medidas actuales .: | | |
|------------------------|-----------------|--------|
| LogOut (salir) | | |
| CANAL | MEDIDA | |
| Humedad | -19660.50000000 | % |
| Ruido | 5.99984741 | dB |
| Luz | 2.99984741 | Lumen2 |
| Viento | 5.00015259 | m/s |

| .: Escoger acción .: | |
|---------------------------|-----------------------|
| Su nombre de usuario: Fe | Su nivel de acceso: 1 |
| Gestionar usuarios | |
| Histórico | |

Figura 7: Interfaz web. Acceso de usuario de nivel 1.

existentes (excepto cambios al nivel 0). Para ello pulsaremos el botón “Gestionar usuarios”.

- Nivel 0: Por último, en la figura 8, se muestra el menú para los usuarios de nivel 0, en el que se permite todo lo dicho en el nivel 1 así como cambiar el usuario al nivel 0, y también editar los archivos de configuración de la aplicación remotamente. Para cada archivo existe una interfaz de edición que se corresponde con cada uno de los dos nuevos botones que aparecen en este nivel.

4. APLICACIONES

Las aplicaciones prácticas del sistema desarrollado son muchas, tanto en el ámbito industrial como científico. La construcción de aplicaciones concretas pasaría por la instalación del sistema desarrollado junto con los dispositivos concretos de transducción y actuación adecuados para la aplicación deseada. La calibración del sistema se podría realizar fácilmente a partir de las medidas tomadas en tiempo real y mediante la adaptación de los ficheros de configuración (fórmulas y reglas de actuación) a la lógica propia de la aplicación.

En el marco del Proyecto Fin de Carrera en que se ha desarrollado el sistema, se han realizado pruebas de funcionamiento basadas en la toma de datos de elementos transductores simples: potenciómetros, células fotoeléctricas, etc, y se han implementado algoritmos de actuación mediante el lenguaje de reglas propio de la aplicación. Obteniendo resultados favorables en todos los casos, tanto en la lectura en tiempo real de la información de entrada como en el almacenamiento de históricos y cumplimiento de las reglas de actuación.

Aunque consideramos que los detalles de implementaciones concretas del sistema quedan fuera del ámbito de este trabajo, a continuación describimos algunas de sus posibles aplicaciones:

| .: Medidas actuales .: | | |
|------------------------|------------------|--------|
| LogOut (salir) | | |
| CANAL | MEDIDA | |
| Humedad | -19660.500000000 | % |
| Ruido | 5.99984741 | dB |
| Luz | 2.99984741 | Lumen2 |
| Viento | 5.00015259 | m/s |

| .: Escoger acción .: | |
|---|-----------------------|
| Su nombre de usuario: Admin | Su nivel de acceso: 0 |
| <input type="button" value="Editar adki.conf"/> | |
| <input type="button" value="Editar adki_out.conf"/> | |
| <input type="button" value="Gestionar usuarios"/> | |
| <input type="button" value="Histórico"/> | |

Figura 8: Interfaz web. Acceso de usuario de nivel 0.

- Casa inteligente: como se explicó en la introducción de este proyecto, una aplicación del sistema desarrollado es la de conectar distintos sensores como de luz, temperatura, humedad y otros, para el encendido automático de luces, calefacciones, aire acondicionado, etc. La aplicación, obviamente, debe ser configurada para realizar tales acciones, en la línea de calibración y definición de reglas comentada anteriormente.
- Estación meteorológica remota: bastaría configurar un PC correctamente con una tarjeta de adquisición de datos y los sensores adecuados en un habitáculo correctamente aislado de las inclemencias del exterior y alimentarlo con energía solar. Los datos serían tomados diariamente o según el intervalo que indicara el administrador. Así, el traspaso de esos datos se facilitaría enormemente al utilizar cualquier medio de almacenamiento compatible con los PC. Para estaciones situadas en puntos de difícil acceso, la conexión podría establecerse de forma inalámbrica usando tecnologías convencionales (pruebas recientes han llegado a enlazar dos puntos a 50km de distancia con dos simple tarjetas de red inalámbricas compatibles con el protocolo 802.11b) y recibir los datos en vivo a muy bajo coste.
- Control de seguridad de un edificio: para el control de un edificio, se podrían conectar sensores de detección de movimiento y, adaptando la interfaz web, mostrar planos del edificio donde se identificasen posibles intrusos en las diferentes estancias del edificio. El sistema podría completarse con sensores de temperatura y humos para la detección de incendios, control de calefacción, etc.
- Granja avícola, invernadero: en una serie de aplicaciones agropecuarias es necesario mantener unas condiciones ambientales controladas (temperatura, luz y humedad), como es el caso del cultivo en invernadero o la cría de aves. En estos casos no es

necesaria una alta precisión en el control de las condiciones, pero es muy útil disponer de un histórico de la evolución de las condiciones ambientales para poder detectar a posteriori posibles fallos en la producción o diseñar estrategias de optimización de la misma. El sistema desarrollado en este proyecto se adapta perfectamente a estas aplicaciones y, de hecho, fue el tipo de aplicación que motivó inicialmente el desarrollo del sistema.

5. CONCLUSIONES

Las principales conclusiones derivadas de este trabajo son las siguientes:

- El progreso experimentado en los últimos años por las tecnologías relacionadas con los ordenadores permite construir sistemas de control y monitorización basados en ordenador con bajo coste tanto en su desarrollo como implementación.
- Se ha presentado un sistema de control de propósito general basado en ordenador que incluye la monitorización y gestión completas a través de la web.
- El sistema permite la definición de la relación entre las variables de entrada y las señales de actuación, permitiendo su adaptación a un amplio número de aplicaciones concretas.
- Además de las tareas de control, el sistema almacena en una base de datos el histórico de valores de entrada registrados, permitiendo tareas de seguimiento, análisis de errores, estrategias de optimización, etc.
- El sistema ha sido implementado íntegramente usando software libre, mostrando así las capacidades y ventajas de esta plataforma a reunir todos los componentes necesarios para la elaboración del sistema.
- El software libre es un aliado idóneo para estas aplicaciones, tanto por su bajo coste por su alta compatibilidad, flexibilidad y eficiencia. En particular, la combinación del sistema operativo GNU/Linux junto con el servidor Apache, el lenguaje PHP y sistemas de base de datos como MySQL constituyen una plataforma avanzada y de alta productividad para el desarrollo de todo tipo de aplicaciones Web, como es el caso del tema de esta contribución.

6. REFERENCIAS

- [1] Fundación para el software libre. <http://www.fsf.org>.
- [2] Web del Proyecto Comedi. <http://stml.lbl.gov/comedi>.
- [3] Kay A. Robbins. "UNIX Programación Práctica". Prentice Hall, 1997. ISBN 968-3556-L.
- [4] Web y documentación del la base de datos MySQL. <http://www.mysql.com>.
- [5] Web y documentación del lenguaje PHP. <http://www.php.net>.
- [6] Pedro Pablo Fábrega. "PHP 4". Prentice Hall, 2000. ISBN 84-205-3112-X.
- [7] Web del proyecto Apache. <http://www.apache.org>.
- [8] Web del proyecto GNU. <http://www.gnu.org>.

- [9] Información general sobre Linux. <http://www.linux.org>, <http://www.linux.net>.
- [10] Web del proyecto Debian. <http://www.debian.org>.
- [11] Web de la distribución Red Hat Linux. <http://www.redhat.com>.
- [12] Web del proyecto KDevelop. <http://www.kdevelop.org>.