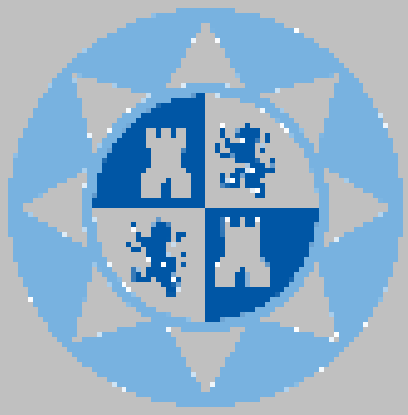
Universidad Politécnica de Cartagena

UPCT



Monitorización en tiempo real y registro de parámetros de control en granjas porcinas

Ingeniería Técnica Industrial
Esp.: Electrónica Industrial

Dpto.: Ingeniería de Sistemas y Automática

Alumno: Juan Antonio Pérez Martínez
Director del Proyecto: Roque Torres Sánchez

MEMORIA

ÍNDICE DE LA MEMORIA

1. INTRODUCCIÓN

1.1. RESUMEN

Este proyecto desarrolla una solución integral para la monitorización remota y registro de los valores de los parámetros físicos más significativos en la explotación de una granja ganadera porcina de cebo. Todos estos datos, tanto los últimos recibidos (actualizaciones cada minuto) como los archivados de cualquier período anterior, pueden ser consultados desde cualquier dispositivo-navegador conectado a internet.

Hemos creído conveniente la inclusión de un sistema para la introducción de comentarios por parte del usuario local (en granja) legibles por cualquier usuario universal (Internet), así como un sistema de alertas configurables por el usuario local, siendo opción que estas alertas sean simplemente visibles en web y/o enviadas vía e-mail a el/los usuario/s que se crea oportuno.

1.2. ANTECEDENTES

La mayor parte de las empresas ganaderas y agrícolas son familiares y por tanto, la mano de obra también es mayoritariamente la aportada por personas de este núcleo, lo cual no quiere decir que en determinadas circunstancias o de continuo exista mano de obra asalariada. Trabajar con seres vivos presenta también unas condiciones especiales; existe una serie de fases o secuencias que no se pueden interrumpir, por ejemplo, necesidad de asistencia a partos, ordeños ineludibles, recolecciones, etc.; por contra en la industria, salvo en la siderurgia y en la industria química, se puede interrumpir el proceso, por ejemplo puedo interrumpir una cadena de montaje de coches y no haber colocado el motor en los últimos 20, sin embargo, un trabajo equivalente, en ganadería, no sería una solución factible. Todo ello hace muy difícil poder llegar a una regulación de la jornada laboral de trabajo, sobre todo, en pequeñas explotaciones.

Podemos decir que una explotación como la que nos ocupa (granja porcina de cebo intensiva) no nos va a requerir la aleatoriedad de los partos, por ejemplo, pero sí que tenemos que garantizar ciertos servicios mínimos en todo momento, como son el agua de bebida, la comida y unos parámetros ambientales más o menos acotados.

El autor del proyecto ha estado explotando desde hace unos años y hasta la actualidad una granja porcina de cebo situada en la localidad de Tallante, pedanía perteneciente al municipio de Cartagena. Por motivos obvios se escogieron algunas de las instalaciones de la misma para el montaje y seguimiento del sistema objeto del presente proyecto.

Es fácil de imaginar como en este entorno de negocio y otros muchos similares y no tan similares, se puede tener la necesidad o, simplemente, la conveniencia de conocer el valor de ciertos parámetros en un momento dado sin necesidad de desplazarse al lugar, así como de tenerlos registrados a lo largo del tiempo para su consulta y análisis.

En las conversaciones mantenidas con los profesores Ana Toledo Mareo y Roque Torres Sánchez del Departamento de Tecnología Electrónica, y el alumno de Ingeniería Técnica Industrial especialidad en Electrónica Industrial de la Universidad Politécnica de Cartagena, con motivo de la realización del proyecto final de carrera de dicho alumno se aceptó la idea y se empezaron a concretar los medios a usar, los cuales serán descritos posteriormente en los apartados dedicados a tal efecto.

El nombre clave utilizado para el proyecto es PigFarmProject, por lo que en las ocasiones en las que aparezca esta palabra, se referirá al proyecto en sí.

Este documento es el resultado de la elaboración del proyecto y se presenta a efectos de reconocimiento académico para la obtención del título de Ingeniero Técnico Industrial especialidad en Electrónica Industrial.

1.3. ALCANCE

El presente trabajo engloba a las siguientes subtareas que hemos intentado ordenar según una secuencia lo más lógica posible:

- Diseño e implantación física del sistema captador de datos de los parámetros más interesantes de la granja.
- Programación de los microcontroladores necesarios en el sistema captador mencionado en el punto anterior y la transmisión de la información pre-procesada al ordenador conectado a internet y situado en la misma granja.
- Programación de las aplicaciones java ejecutables en el ordenador de la granja que son las encargadas de recepción de estos valores y el posterior envío de la información procesada (ordenada e identificada para su localización automática) a una base de datos remota.
- Diseño de las tablas necesarias en dicha base de datos.
- El diseño y la programación de la interfaz (páginas web) que hará posible la consulta de los valores de los parámetros (últimos en tiempo real e históricos) desde cualquier terminal con navegador y conectado a internet.
- El diseño y la programación de un sistema de inclusión de comentarios de interés.
- Diseño y programación de un sistema de alertas que pueden ser visuales en web y/o enviadas a dirección de correo electrónico.

1.4. ORGANIZACIÓN DEL PROYECTO

Para la consecución del presente trabajo hemos considerado dividir la tarea en una serie de bloques y sub-bloques con una clara dependencia entre ellos pero con diferencias de ubicación, naturaleza y/o objetivos.

El siguiente diagrama de bloques (figura 1) representa un sistema teórico indefinido donde podemos ver los diferentes módulos y la comunicación entre ellos, que hemos creído necesarios para el sistema del presente proyecto.

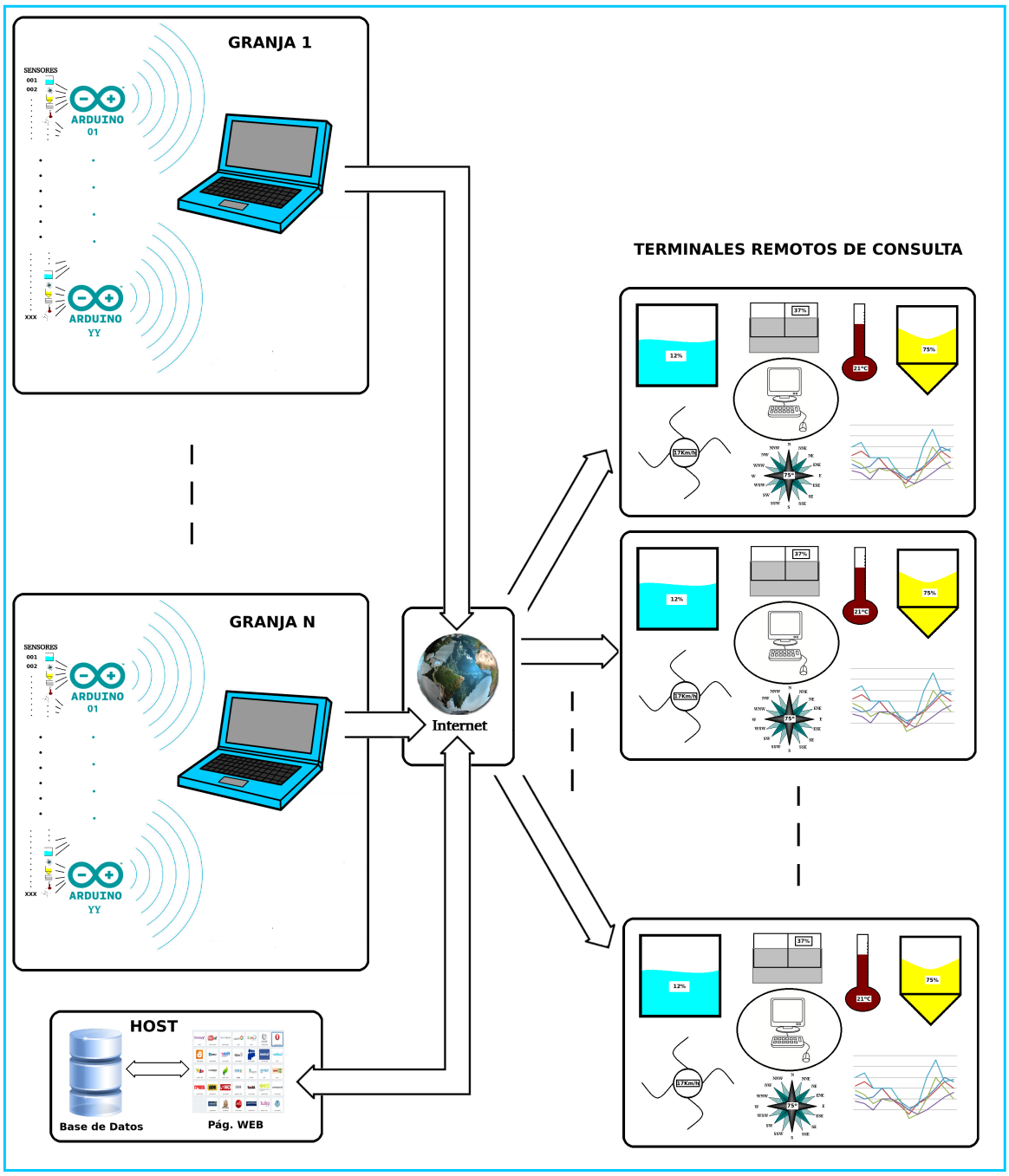


Fig. 1

En la siguiente figura (figura 2) vemos la reducción que hemos hecho de elementos de origen para que el sistema pudiera ser llevado a cabo física y económicamente por el autor. Con lo que se reduce de N granjas a 1 granja y el número de sensores necesarios a 11 (más otros 7 que conseguiremos por otros medios analizados más adelante) y no 30 como estimamos que una granja como la objeto del estudio necesitaría. El número de sensores es 11 para abarcar al menos un sensor de cada tipo de los considerados necesarios o, al menos, aconsejables por la experiencia del propio autor del proyecto.

La instalación denominada “Nave 1” ha sido diseñada y montada en granja al completo, “Nave 2” solo en parte, y “Servicios Generales” aporta solamente los datos de los parámetros que podemos recopilar automáticamente, mediante una aplicación incluida en el presente proyecto, de la estación meteorológica de Cartagena (escogida por cercanía a la granja) gracias a la web oficial de la Agencia Estatal de Meteorología (AEMET).

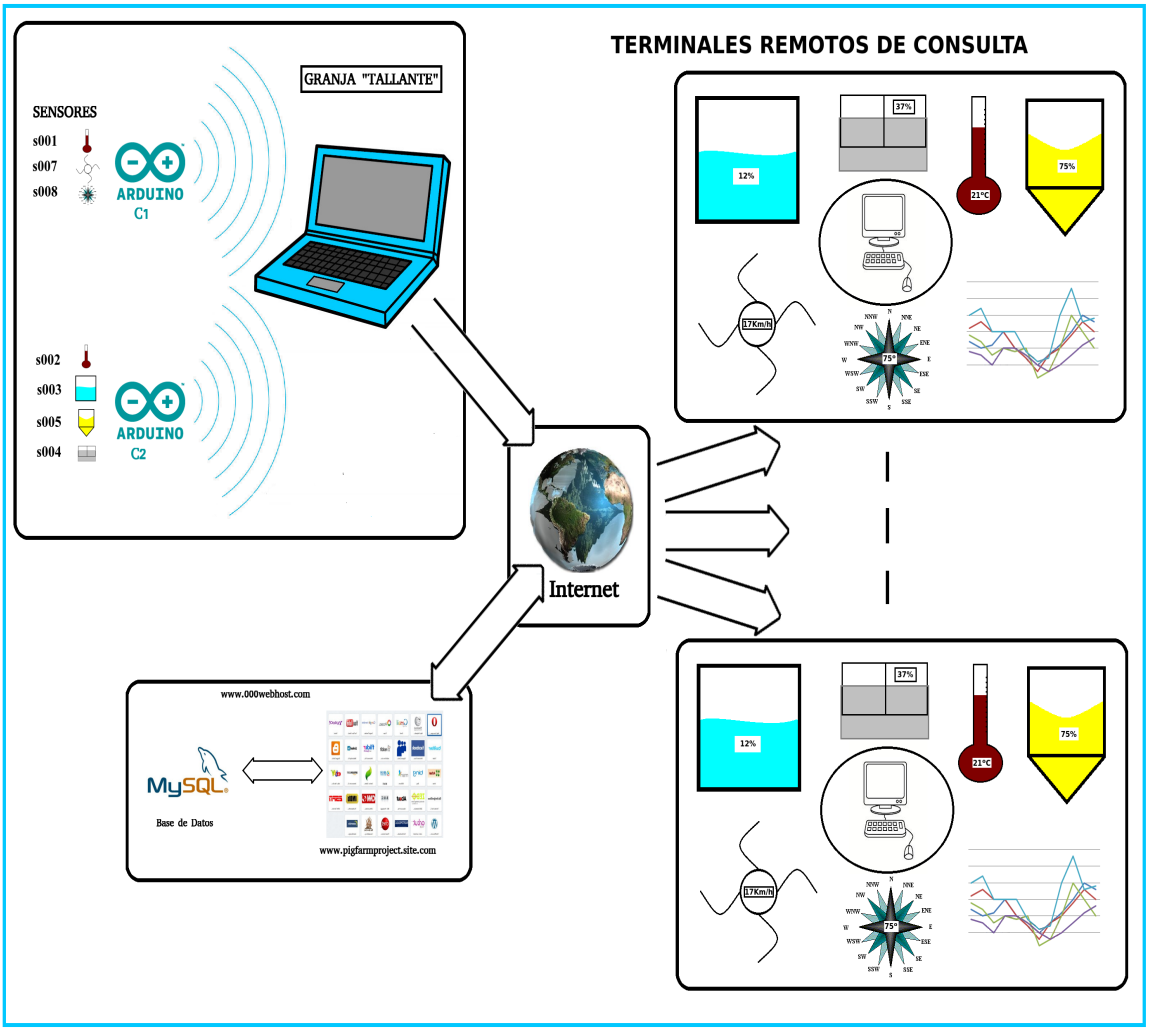


Fig. 2

2. NECESIDAD DE INSTRUMENTACIÓN DE UNA GRANJA PORCINA

2.1. INTRODUCCIÓN AL BIENESTAR ANIMAL

Se puede definir el bienestar animal como todo lo relativo al confort animal, y que está más allá de la mera falta de enfermedad, abarcando el completo estado de bienestar físico. Es la realidad que considera al animal en un estado de armonía en su ambiente y la forma por la cual reacciona frente a los problemas del medio, tomando en cuenta su confort, su alojamiento, trato, cuidado, nutrición, prevención de enfermedades, cuidado responsable, manejo, etc.

2.2. CONDICIONES AMBIENTALES

Los cerdos, lógicamente, reaccionan negativamente a las altas y bajas temperaturas. Esta reacción puede suponer cambios en su comportamiento (reducción de la productividad y la sanidad de los animales). Estos problemas se pueden evitar proporcionando al cerdo una ventilación y una climatización continua y estable.

El grado de confort dependerá de la interacción entre: temperatura del aire, humedad de la piel, corrientes de aire, humedad ambiental, tipo de suelo, raza, así como cantidad y composición de las dietas.

El efecto de las temperaturas en la fase que nos ocupa de cebo porcino, los animales son muy sensibles al frío en la entrada y más susceptibles al calor en la fase final. En condiciones de altas temperaturas se reduce la ingestión y, por tanto, también los crecimientos.

Por encima de 30°C el apetito de los cerdos se reduce de forma significativa; primera consecuencia, a menor consumo de pienso, menos energía disponible para el crecimiento. Esta marcada reducción en la ingesta de energía da lugar a una severa reducción de los crecimientos que es más notable en los cerdos alimentados a voluntad. Por encima de esta temperatura de 30°C los cerdos sólo comen el 20% del total de la capacidad de la ingesta, haciéndolo durante la noche.

"En épocas calurosas el consumo de pienso disminuye perjudicando muy gravemente el estado de las carnes".

Por cada 250 Kcal/día de reducción en la ingesta, el crecimiento disminuye alrededor de 25 gramos/día.

Así pues si, por ejemplo, la temperatura exterior es de 36°C y la ingesta sólo es el 20% de lo usual (±2Kg), un cerdo a esta temperatura sólo comerá 400-500 gramos de pienso al día. Si un pienso tiene 3.200 Kcal/Kg Y ha existido una reducción/día de 1.5Kg., la reducción de energía/día será de 1.5 x 3.200 = 4.800 Kcal. 250 Kcal/día = 19.2 x 25 gramos/día = 480

gramos/día de reducción en el crecimiento. Por lo tanto, si suponemos un crecimiento standard es de 650 gramos/día, en épocas muy calurosas podemos encontramos con crecimientos tan ridículos como 170-200 gramos/cerdo/día. Por esta razón en verano el comentario general entre los ganaderos de cebo es que sus cerdos están "parados".

El equipamiento necesario de la instalación porcina conlleva especial atención en el problema de renovación de aire para que la vida de los animales en su interior sea lo más estable posible para facilitar su engorde.

Este tipo de animales es muy susceptible a los cambios que se puedan producir en el ambiente y a la concentración de NOx , CO2 , por lo que la climatización de la nave tiene que cumplir cierta estabilidad que, para nuestra zona climática en concreto, podremos conseguir con el equipamiento que se lista a continuación:

- Instalación eléctrica.
- Instalación de agua.
- Sistema de iluminación.
- Sistemas de ventilación/extracción de aire.
- Sistema de distribución de piensos.

2.3. CONDICIONES DE ALIMENTACIÓN

Otro de los factores fundamentales que incluye el bienestar animal es, por supuesto, la nutrición de los animales. La existencia de pienso a demanda del animal (conviene recordar que estamos hablando de una granja porcina de cebo) debe de ser continua. Con el suministro del agua de bebida ocurre lo mismo, aunque conviene destacar que el caso del agua es aún más importante; ya que en pocas horas, la falta de la misma, podría poner en peligro la salud e incluso la vida de los animales en según que condiciones.

3. SOLUCIONES ADOPTADAS

3.1. SOLUCIONES DE OTRAS GRANJAS Y OTROS SISTEMAS

La tendencia de este tipo de instalaciones es, lógicamente, el aumento de la automatización de tantas tareas como sea posible y rentable llevar a cabo. Desde el, ya consolidado desde hace varias décadas, transporte automático de pienso hasta sistemas más modernos, ciertamente extendidos en la actualidad, como el control de la temperatura interior con elevadores automáticos de ventanas, ventiladores/extractores, etc. nos confirma esta idea. Igualmente podemos imaginar otros sistemas que no son tan exclusivos del sector; por ejemplo, los sistemas automáticos para suministro y almacenaje del agua de bebida.

Nuestra granja ya tiene instaladas algunas de estas actuaciones automáticas, las consideradas convenientes para nuestro trabajo y posibilidades; como son el transporte automático de pienso en todas las naves, regulación automática de ventanas según criterio ajustable de temperatura interior de las naves (que nos parece interesante explicar con más detalle a continuación) en 3 de las 4 naves existentes, además de un sistema auxiliar y portable de una nave a otra según necesidad, de control automático de temperatura, en este caso, variando en este caso la velocidad de ventilador/es portátiles.

Nos centraremos más en el control de la temperatura en la granja por medio de la ventilación estática, es decir, cerrando y abriendo las ventanas. Este sistema, en modo manual, sin dejar de ser efectivo, pasaba a no ser tan efectivo como debería debido a épocas con frecuentes cambios de temperatura a los que en ganadero no era posible que respondiera en todo momento por pendiente que estuviera.

Esto cambió con la aparición de los elevadores automáticos de ventanas, que incorporan un microprocesador que recibiendo continuamente las lecturas de una sonda de temperatura colocada en el interior de la nave y controlando un motorreductor se encarga de mantener en todo momento las ventanas de la granja en una posición que es la exactamente adecuada para conseguir la temperatura deseada.

Estos equipos controladores de ventanas ha ido evolucionando y hoy en día pueden ser programados para tener en cuenta otros factores adicionales como un cierre mínimo y máximo de ventana, conectarse con equipos de ventiladores para trabajar conjuntamente o con equipos de alarma que envían al usuario un mensaje de texto a su teléfono móvil en caso de que la temperatura rebase unos niveles establecidos.

Modernizar la granja y estar al día no es una opción, es una necesidad que obliga a todas las explotaciones a estar actualizadas ya que terminará retirando del mercado a las explotaciones que no lo hagan.

El presente trabajo, sin embargo, trata únicamente los datos tomados para generar diversas informaciones derivadas de los mismos, pero no la de actuación sobre elemento físico alguno de la granja. No es objeto del presente trabajo entrar a mejorar en algún aspecto

alguna/s de éstas u otras soluciones que la industria especializada ya nos ofrece.

Usar los datos obtenidos para proceder a actuaciones automáticas o manuales remotas sería fácilmente implementable dentro de nuestro sistema aprovechando algunos de los mismos recursos que hemos usado, aunque obligaría a adquirir otra serie de equipamiento industrial (tornos automáticos de ventanas, reguladores de potencia de ventiladores/extractores de aire, sistema de electro-válvulas, alimentación de los equipos electro-mecánicos, etc) de precio y coste de instalación mucho más elevados. Por este motivo y por la mencionada existencia en el mercado múltiples soluciones cada día más usadas en el sector, nuestro trabajo se limitará a la toma, almacenamiento y tratamiento de los valores de los parámetros físicos captados para convertirlos en información útil y accesible para el ganadero y/o personal técnico responsable de la granja.

3.1.1. EQUIPOS Y DISPOSITIVOS.

Respecto a equipamiento lógico (computadoras, controladores, etc.) y electro-mecánico (nos referimos sobre todo a sensorización) disponemos de la gran cantidad de recursos que nos proporciona la industria más o menos especializada del sector.

En la figura 3 podemos ver las computadoras de un sistema de control de temperatura que actúa variando la altura de hasta 8 filas de ventanas y el sistema arranque con paro automático de 4 líneas de pienso.

Fig. 3

En la figura 4 vemos un motor elevador instalado para una fila de ventanas exteriores . Podemos apreciar fácilmente en la imagen la sencilla lógica de funcionamiento e imaginar

la tarea que realiza mecánicamente el dispositivo.

Fig. 4

Para la parte de sensorización necesaria, tanto de estos dispositivos actuadores que estamos viendo en esta sección como de los de tratamiento de información que ocupan el presente proyecto, disponemos en el mercado de una amplísima variedad de alternativas para cada tipo de parámetro. Por lo tanto, múltiples son las soluciones que, de una forma más o menos sofisticada, podemos adoptar como medidores y detectores en un proyecto de este tipo. Sensores de temperatura, de velocidad, de posición, detectores de presencia de sustancias nocivas en el ambiente (sobre todo amoniaco), células de pesada, etc.

Por ejemplo, para conocer la cantidad de pienso en silos o el agua en depósitos con gran exactitud, son de uso común las células de pesada. Sin embargo su precio y, sobre todo, coste de instalación en equipos ya existentes unido a la no necesidad de una lectura tan precisa, aconsejaron estudiar desde el inicio otras alternativas.

3.2. REQUISITOS QUE LLEVAN A NUESTRA SOLUCIÓN

3.3. SITUACIÓN GEOMÉTRICA DE LAS INSTALACIONES

En la foto aérea de la figura 5 podemos observar donde están ubicados los equipos en la granja.

En el interior de una oficina ubicada en la edificación de un antiguo molino de pienso se encuentra nuestro ordenador-granja conectado a internet (esquina inferior derecha de la figura). Los dos módulos de campo, a los que hemos llamado, Arduino/XBee/sensores C1

y C2, instalados físicamente para este proyecto, están ubicados en las entradas de las naves 1 y 2 respectivamente. Podemos ver en la figura las distancias a salvar de forma inalámbrica entre los tres equipos mencionados.

Diremos, aprovechando esta perspectiva, que la única posibilidad de comunicación del módulo C2 con el ordenador-granja tiene que pasar por el módulo C1 usado como nodo repetidor de red.

Hacemos observar la necesidad de que C1 y C2 tengan visión libre directa entre ellos, no así entre C1 y ordenador-granja que admite, por estar a una distancia más reducida, varios obstáculos entre ellos (muro de bloques y puerta metálica).

Fig. 5

Ampliar el sistema a las dos naves restantes y un módulo más para controlar los depósitos de agua de los Servicios Generales no entrañaría más dificultades de comunicación, al contrario, facilitaría las comunicaciones y la independencia de funcionamiento de cada uno de ellos con respecto a los demás por razones obvias de cantidad de nudos y geometría de los mismos. Podemos ver en la siguiente foto aérea (figura 6) el equipamiento y distribución física de un sistema completo.

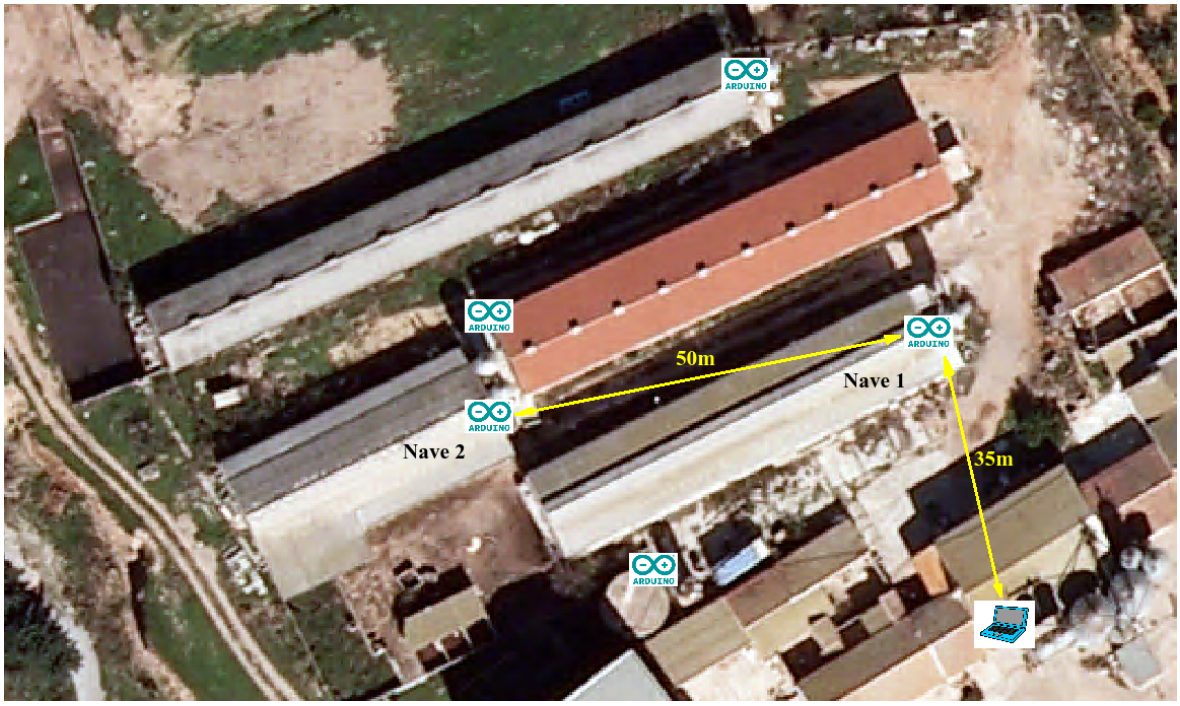


Fig. 6

3.4. HARDWARE

Los motivos de elección son los mismos prácticamente en todos los casos: versatilidad, precio y disponibilidad, tanto de los dispositivos físicos como de información en general sobre los mismos (software asociado, herramientas de desarrollo, hojas de características, etc.).

3.4.1 DESCRIPCIÓN

3.4.1.1. SENSORES

Los sensores elegidos para nuestra versión de muestra del sistema son:

- Sensor de Temperatura LM35.
- Medidor de Distancia por Ultra-Sonidos.
- Resistencias Variables Multivuelta (como medidores de pequeñas distancias).
- Sensor anemométrico (tipo tacómetro).

3.4.1.2. MICROCONTROLADORES

Para las bases de control de sensores y recopilación de la información captada por los mismos, hemos optado por la plataforma electrónica para la creación de prototipos denominada *Arduino*.

Arduino es una plataforma de hardware libre, basada en una placa con un microcontrolador y un entorno de desarrollo, diseñada para facilitar el uso de la electrónica en proyectos multidisciplinares.

3.4.1.3. MÓDULO DE RECOLECCIÓN Y ENVÍO DE DATOS

Como plataforma inteligente encargada de la tarea de recibir la información enviada por los microcontroladores (ver figura 2, bloque llamado Granja "Tallante"), su tratamiento y envío a la base de datos remota, optamos por un computador mini-portátil conectado a internet, con un único dispositivo extra acoplado de comunicación inalámbrica por radiodifusión digital (ver apartado siguiente "comunicaciones en granja"). A partir de ahora llamaremos a este conjunto Ordenador-Granja.

Cualquier ordenador configurado para el ámbito doméstico puede servir perfectamente para realizar las tareas que se le van a encomendar. Los requerimientos son, uno, conexión a internet y , dos, un puerto USB 2.0 libre.

3.4.1.4. COMUNICACIONES EN GRANJA

Para facilitar y hacer inalámbrica la comunicación entre todos los módulos microcontroladores y el Ordenador-Granja elegimos los módulos de comunicación XBee XB24-Z7CIT Series 2, basados en tecnología ZigBee.

ZigBee es el nombre de la especificación de un conjunto de protocolos de alto nivel de comunicación inalámbrica para su utilización con radiodifusión digital de bajo consumo, basada en el estándar IEEE 802.15.4 de redes inalámbricas de área personal (*wireless personal area network*, WPAN). Su objetivo son las aplicaciones que requieren comunicaciones seguras con baja tasa de envío de datos y maximización de la vida útil de sus baterías.

3.4.1.5. ORDENADOR-GRANJA

Como plataforma inteligente capaz de recolectar, tratar y enviar la información necesaria a la base de datos usaremos un ordenador mini-portátil, que deberá de estar acompañado por uno de los dispositivos de comunicación mencionados en el apartado "COMUNICACIONES GRANJA" y, además, con conexión a internet.

3.4.2. EQUIPOS EN GRANJA

Nos referimos al bloque que muestra la figura 7, el cual es una parte del bloque general de la figura 2 que vimos en su momento.

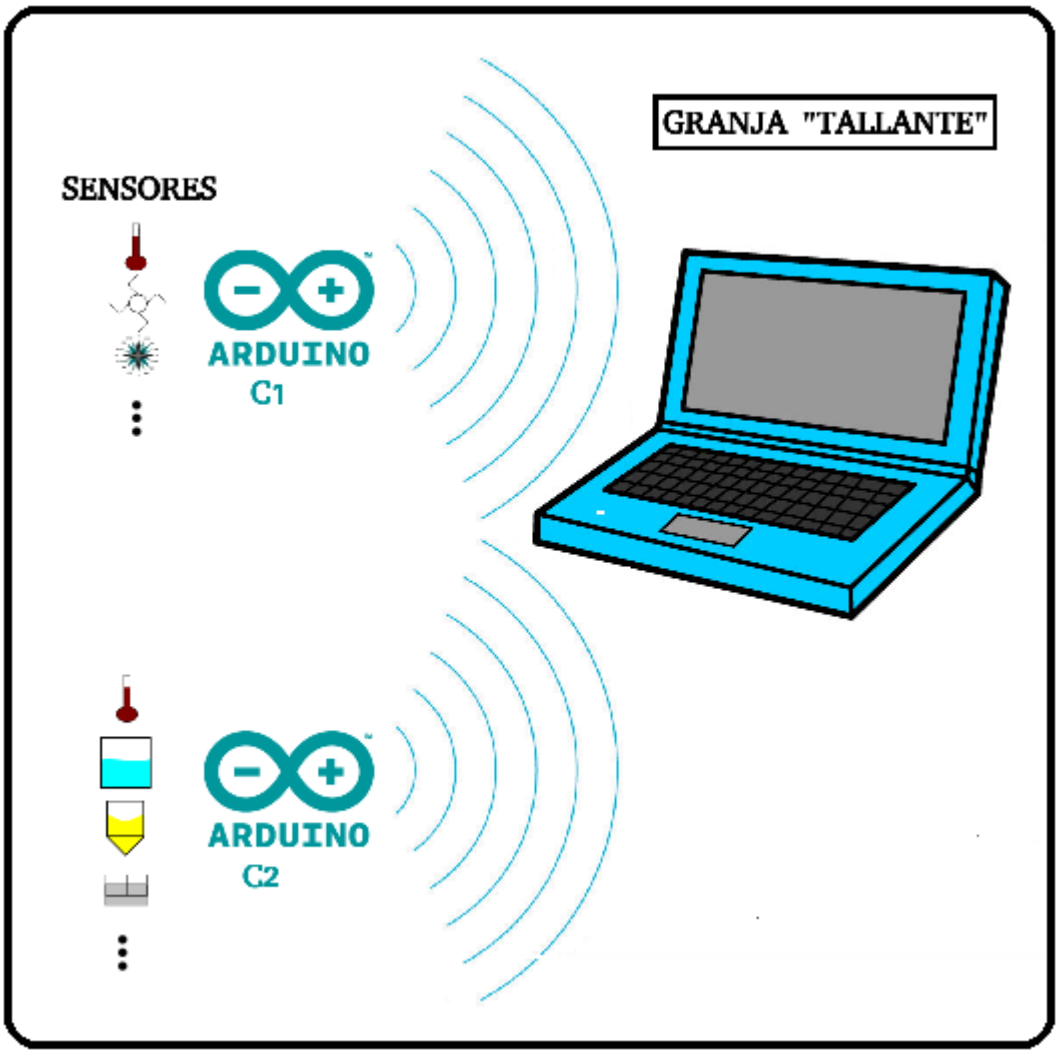


Fig. 7

3.4.2.1. MÓDULOS ARDUINO/XBee/SENSORES

Cada modulo de campo con microcontrolador, gracias a los mencionados dispositivos de comunicación Xbee, se convierten en una red donde cada uno de ellos, además de emitir los datos de sus propios sensores, servirá como repetidor de las señales de otros equipos análogos más alejados del Ordenador-Granja. En la figura 8 podemos observar de forma muy esquematizada el bloque con el paquete de sensores, el microcontrolador Arduino y el módulo de comunicaciones Xbee (representado como ondas simuladas de emisión/recepción).

Cada uno de estos módulos está programado para emitir cada cierta cantidad de segundos el valor de un sensor perfectamente etiquetado para su interpretación en el Ordenador-Granja. Se pretende con ésto que cada módulo haya emitido en 60 segundos el valor

captado por todos y cada uno de los sensores asociados a él.

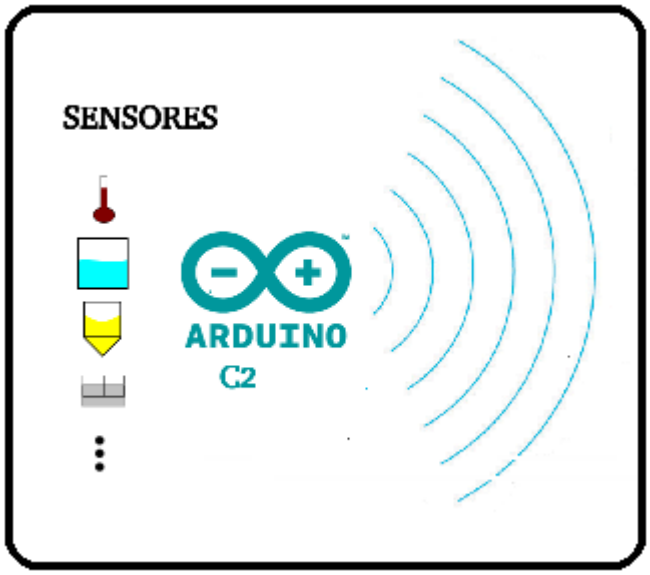


Fig. 8

A continuación mostramos la información que concierne a uno de los dos módulos tipo instalados.

3.4.2.1.1. COMPONENTES

- 1 Arduino UNO con ATMega328
- 1 Arduino Shield Wireless Xbee SD
- 1 Xbee 2mW con Antena Chip Serie 2
- 1 fuente alimentación 220V-USB
- 1 placa de prototipos
- 1 Cable USB TipoA/TipoB (USB ordenador a impresora)
- 1 LM35 + 1 resistencia 75ohm +1 condensador electrolítico 1uF
- 4 resistencias variables multivuelta como medidores de distancias (altura de ventanas y nivel de depósito de agua)
- 1 Sensor de Distancia por Ultrasonidos Ping (nivel de pienso en silo)
- Resistencia y Condensador Electrolítico de baja potencia
- 25 m cableado (cable de señal de 3 pares)
- 6 m de cable apantallado doble.

3.4.2.1.2. ESQUEMA

En la figura 9 vemos el esquema completo del módulo Arduino/Sensores encargado de los

parámetros de la Nave 1. Nos sirve también como esquema típico del módulo de cada nave. Realmente la diferencia solo estribará en el nombre de nuestro módulo de campo y, en este caso "Arduino C1", y en el de los sensores. Tal como hemos concebido el sistema los sensores pertenecientes a la Nave 1 irán desde "s010" hasta "s019", los de la Nave 2, que controlaría el módulo "Arduino C2", irían desde el "s020" al "s029", y así sucesivamente. Los nombres del "s000" al "s009" los reservamos al grupo de parámetros que denominamos de "Servicios Generales" y que serían controlados (de haber sido instalada esta parte) con el módulo al que llamaríamos "Arduino C0".

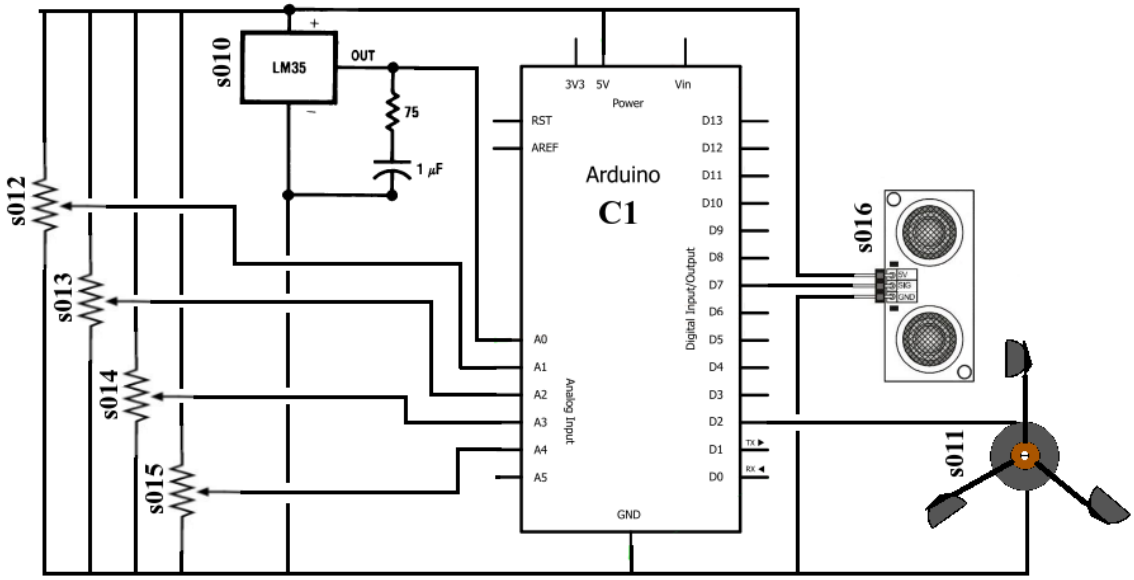


Fig. 9

3.4.2.2. "ARDUINO C0", "ARDUINO C2" Y SUCESIVOS.

El módulo "Arduino C0" queda únicamente con la misión de leer el nivel de los dos depósitos de agua de Servicios Generales, de mayores dimensiones pero de idéntica complicación que sus homólogos más pequeños (los particulares de cada nave). El resto de entradas (ventanas, silos pienso, temperatura, etc) al Arduino "C0" deberían ser obviadas en hardware y anuladas en software (ver apartado "microcontroladores arduino" y el "código 1" incluido en él).

En cuanto al resto de naves, éstas deberían de tener exactamente la misma cantidad y tipo de dispositivos instalados, por lo que detallar por separado cada una de ellas sería inútilmente redundante. Lo único que cambiaría serían los nombres que hemos puesto a cada módulo Arduino y la de la denominación de los propios sensores de cada módulo (definidos en el propio software del microcontrolador) según criterios que adelantamos en el punto anterior.

3.4.3. ORDENADOR-GRANJA. COMPONENTES

- 1 Ordenador Mini-Portátil con conexión a internet.
- 1 Xbee Explorer USB.
- 1 Xbee 2mW con Antena Chip Serie 2.
- 1 Cable USB a mini-USB.

3.5. PRESUPUESTO DE EQUIPOS, COMPONENTES Y MATERIALES

A pesar de no haber montado el sistema completo estimado como necesario para la granja objeto del estudio, debido a su carácter perfectamente modular, podemos hacer la estimación global de materiales necesarios para el montaje del sistema ideado en su totalidad.

Se valora en este presupuesto solo los componentes y materiales físicos a adquirir, no se incluye ni se valora el trabajo de desarrollo y adaptación de software alguno ni los trabajos de montaje e instalación en campo del sistema.

3.5.1. PRESUPUESTOS PARCIALES

3.5.1.1. HARDWARE ORDENADOR-GRANJA

Descripción	Unidades	Precio Unitario	Precio Total
Ordenador Mini-Portátil c/SO. y prog. inst. y config.	1	300	300
Xbee Explorer USB	1	19,90	19,90
Xbee 2mW con Antena Chip Serie 2	1	23,50	23,50
Cable USB a mini-USB	1	2,95	2,95
SUBTOTAL 1.			**346,35**

Tabla 1

3.5.1.2. HARDWARE MÓDULO ARDUINO/XBEE POR INSTALACIÓN TÍPICA

Descripción	Unidades	Precio Unitario	Precio Total
Arduino UNO con ATMega328	1	21,90	21,90
Arduino Shield Wireless Xbee SD	1	20,90	20,90
Xbee 2mW con Antena Chip Serie 2	1	23,50	23,35
SUBTOTAL 2.			**66,15**

Tabla 2

3.5.1.3. ALIMENTACIÓN/CABLES/CONEXIONES/ETC POR INSTALACIÓN TÍPICA

Descripción	Unidades	Precio Unitario	Precio Total
fuente alimentación 220V/USB	1	9,40	9,40
Cable de señal de 3 pares	25 m	0,50	12,50
Cable doble apantallado	6 m	0,73	4,38
Cable USB TipoA/TipoB	1	7,50	7,50
proto-board	1	5,50	5,50
Caja IP-55	1	4,00	4,00
Conector 3 pines	14	1,80	25,2
Conector 6 pines	2	2,35	4,70
Resistencia	1	0,07	0,07
Condensador	1	0,10	0,10
SUBTOTAL 3.			**73,35**

Tabla 3

3.5.1.4. SENSORES POR INSTALACIÓN TÍPICA

Descripción	Unidades	Precio Unitario	Precio Total
Resistencia Variable Multi-Vuelta	4	7,00	28,00
Sensor Temperatura LM35	1	2,14	2,14
Sensor Anemométrico	1	16,50	16,50
Sensor Distancia por Ultra-Sonidos PING	1	19,50	19,50
SUBTOTAL 4.			**66,14**

Tabla 4

3.5.1.5. SOPORTACIÓN/POLEAS POR INSTALACIÓN TÍPICA

Descripción	Unidades	Precio Unitario	Precio Total
Soporte Sensor Depósito	2	10	20
Soporte Sensor Ventanas	2	10	20
Soporte/Caja Sensor Distancia	1	10	10
Polea Tipo Ventana	4	0,80	3,20
SUBTOTAL 5.			**53,20**

Tabla 5

3.5.1.6. OTROS MATERIALES DE MONTAJE POR INSTALACIÓN TÍPICA

En este apartado hacemos una estimación bastante libre del material fungible de montaje electrónico, eléctrico y mecánico; tales como, estaño, pines, pequeños cables con terminal, alargaderas, cinta aislante, tubo termo-retráctil, bridas de cremallera, silicona, cola adhesiva, broca, tornillería/fijación , cuerda, etc.

Estimamos un gasto aproximado de 25 euros en cada instalación en granja (nave).

SUBTOTAL 6.	**25**

Tabla 6

3.5.2. PRESUPUESTO TOTAL

Descripción	Unidades	Precio Unitario	Precio Total
SUBTOTAL 1.	1	286,35	346,35
SUBTOTAL 2.	5	66,15	330.75
SUBTOTAL 3.	5	73,35	366,75
SUBTOTAL 4.	5	66,14	330,70
SUBTOTAL 5.	5	53,20	266,00
SUBTOTAL 6.	5	25,00	125,00
TOTAL			1.434,80
IVA (21 %)			**301,31**
TOTAL (IVA INCL.)			**1.736,11**

Tabla 7

4. SOFTWARE

4.1. MICROCONTROLADORES ARDUINO

Ya hablamos sobre la semejanza entre los programas de los diferentes microcontroladores Arduino a instalar en cada módulo de campo; por esta misma razón veremos aquí únicamente el código correspondiente del que hemos denominado módulo "Arduino C1" a ubicar físicamente en la "Nave 1" de nuestras instalaciones.

Añadimos a cada línea del código 1 una etiqueta con el nombre del parámetro físico a que corresponde su función para una fácil localización y comprensión del funcionamiento del programa en sí.

Sin etiqueta alguna nos quedan los nombres y las llaves que delimitan las partes principales en que se debe elaborar un programa para la plataforma *Arduino;* la linea que configura la velocidad de transmisión del puerto serie ("`Serial.begin(9600);`") y por otra parte los retardos (ejemplo "`delay(7000);`"), destinados a separar el envío de datos de cada sensor del anterior y del siguiente. No debemos olvidar que el tiempo de espera particular en enviar entre el valor de un sensor y el siguiente no es lo más importante. Sí que lo es ajustar estos retardos globalmente de manera que un ciclo de envío de los valores de todos los sensores de cada módulo tarde lo más exactamente posible 60 segundos en completarse. Diremos, a modo de aclaración, que para módulos Arduino que controlen más sensores, estos retardos deberían ser, en general, más reducidos y al contrario si el número de sensores a controlar es menor.

```
/*[VelocidadViento]*/ #define uint  unsigned int
/*[VelocidadViento]*/ #define ulong unsigned long

/*[VelocidadViento]*/ #define PIN_ANEMOMETER  2
/*[VelocidadViento]*/ #define MSECS_CALC_WIND_SPEED 5000

/*[VelocidadViento]*/ volatile int numRevsAnemometer = 0;
/*[VelocidadViento]*/ ulong nextCalcSpeed;
/*[VelocidadViento]*/ ulong time;


/*[VelocidadViento]*/ #define NUMDIRS 8
/*[VelocidadViento]*/ ulong   adc[NUMDIRS] = {26, 45, 77, 118, 161, 196,
220, 256};

/*[Temperatura]*/ float tempC;
/*[Temperatura]*/ float temp1;
/*[Temperatura]*/ int c;
/*[Temperatura]*/ int tempPin = 0;

/*[Nivel DepAgua 1]*/ int analogPin1 = 1;
/*[Nivel DepAgua 2]*/ int analogPin2 = 2;
```

```
/*[Apertura Ventanas SUR 3]*/ int analogPin3 = 3;
/*[Apertura Ventanas NOR 3]*/ int analogPin4 = 4;

/*[Nivel DepAgua 1]*/ float EntradaAnalogicaA1 = 0.0;
/*[Nivel DepAgua 1]*/ float ResistenciaA1 = 0.0;
/*[Nivel DepAgua 1]*/ float X1A1 = 1023.0;
/*[Nivel DepAgua 1]*/ float Y1A1 = 0.0;
/*[Nivel DepAgua 1]*/ float X2A1 = 0.0;
/*[Nivel DepAgua 1]*/ float Y2A1 = 100.0;

/*[Nivel DepAgua 2]*/ float EntradaAnalogicaA2 = 0.0;
/*[Nivel DepAgua 2]*/ float ResistenciaA2 = 0.0;
/*[Nivel DepAgua 2]*/ float X1A2 = 1023.0;
/*[Nivel DepAgua 2]*/ float Y1A2 = 0.0;
/*[Nivel DepAgua 2]*/ float X2A2 = 0.0;
/*[Nivel DepAgua 2]*/ float Y2A2 = 100.0;

/*[Apertura Ventanas SUR]*/ float EntradaAnalogicaA3 = 0.0;
/*[Apertura Ventanas SUR]*/ float ResistenciaA3 = 0.0;
/*[Apertura Ventanas SUR]*/ float X1A3 = 1023.0;
/*[Apertura Ventanas SUR]*/ float Y1A3 = 0.0;
/*[Apertura Ventanas SUR]*/ float X2A3 = 337.0;
/*[Apertura Ventanas SUR]*/ float Y2A3 = 75.0;

/*[Apertura Ventanas NOR]*/ float EntradaAnalogicaA4 = 0.0;
/*[Apertura Ventanas NOR]*/ float ResistenciaA4 = 0.0;
/*[Apertura Ventanas NOR]*/ float X1A4 = 1023.0;
/*[Apertura Ventanas NOR]*/ float Y1A4 = 0.0;
/*[Apertura Ventanas NOR]*/ float X2A4 = 337.0;
/*[Apertura Ventanas NOR]*/ float Y2A4 = 75.0;

/*[Nivel Silo Pienso]*/ const int pingPin1 = 7;
/*[Nivel Silo Pienso]*/ long duration1, cm1, dist1;

void setup() {

   Serial.begin(9600);
/*[VelocidadViento]*/    pinMode(PIN_ANEMOMETER, INPUT);
/*[VelocidadViento]*/    digitalWrite(PIN_ANEMOMETER, HIGH);
/*[VelocidadViento]*/    attachInterrupt(0, countAnemometer, FALLING);
/*[VelocidadViento]*/    nextCalcSpeed = millis() +
MSECS_CALC_WIND_SPEED;
/*[Temperatura]*/    pinMode(tempPin, INPUT);
/*[Nivel DepAgua 1]*/    pinMode(analogPin1, INPUT);
/*[Nivel DepAgua 2]*/    pinMode(analogPin2, INPUT);
/*[Apertura Ventanas SUR]*/    pinMode(analogPin3, INPUT);
/*[Apertura Ventanas NOR]*/  pinMode(analogPin4, INPUT);
}

void loop() {

/*[Temperatura]*/ temp1=0.0;
/*[Temperatura]*/ for (c=0;c<10;c++)
/*[Temperatura]*/ {
/*[Temperatura]*/ tempC = analogRead(tempPin);
```

```
/*[Temperatura]*/ temp1 += (5.0 * tempC * 100.0)/1024.0;
/*[Temperatura]*/ delay(500);
/*[Temperatura]*/ }
/*[Temperatura]*/ Serial.print("/s010=");
/*[Temperatura]*/ Serial.print(temp1/10.0);
/*[Temperatura]*/ Serial.print(' ');

delay(7000);

/*[VelocidadViento]*/   time = millis();

/*[VelocidadViento]*/   if (time >= nextCalcSpeed) {
/*[VelocidadViento]*/       calcWindSpeed();
/*[VelocidadViento]*/       nextCalcSpeed = time +
MSECS_CALC_WIND_SPEED;
   }


/*[Nivel DepAgua 1]*/   EntradaAnalogicaA1 = analogRead(analogPin1);
/*[Nivel DepAgua 1]*/   ResistenciaA1 = ((EntradaAnalogicaA1-
X1A1)*(Y2A1-Y1A1)/(X2A1-X1A1))+Y1A1;
/*[Nivel DepAgua 1]*/   Serial.print("/s012=");
/*[Nivel DepAgua 1]*/   Serial.print(ResistenciaA1);
/*[Nivel DepAgua 1]*/   Serial.print(" ");
  delay(8500);


/*[Nivel DepAgua 2]*/   EntradaAnalogicaA2 = analogRead(analogPin2);
/*[Nivel DepAgua 2]*/   ResistenciaA2 = ((EntradaAnalogicaA2-
X1A2)*(Y2A2-Y1A2)/(X2A2-X1A2))+Y1A2;
/*[Nivel DepAgua 2]*/   Serial.print("/s013=");
/*[Nivel DepAgua 2]*/   Serial.print(ResistenciaA2);
/*[Nivel DepAgua 2]*/   Serial.print(" ");
  delay(8500);


/*[Apertura Ventanas SUR]*/   EntradaAnalogicaA3 =
analogRead(analogPin3);
/*[Apertura Ventanas SUR]*/   ResistenciaA3 = ((EntradaAnalogicaA3-
X1A3)*(Y2A3-Y1A3)/(X2A3-X1A3))+Y1A3;
/*[Apertura Ventanas SUR]*/   Serial.print("/s014=");
/*[Apertura Ventanas SUR]*/   Serial.print(ResistenciaA3);
/*[Apertura Ventanas SUR]*/   Serial.print(" ");
  delay(8500);


/*[Apertura Ventanas NOR]*/   EntradaAnalogicaA4 =
analogRead(analogPin4);
/*[Apertura Ventanas NOR]*/   ResistenciaA4 = ((EntradaAnalogicaA4-
X1A4)*(Y2A4-Y1A4)/(X2A4-X1A4))+Y1A4;
/*[Apertura Ventanas NOR]*/   Serial.print("/s015=");
/*[Apertura Ventanas NOR]*/   Serial.print(ResistenciaA4);
/*[Apertura Ventanas NOR]*/   Serial.print(" ");
  delay(8500);
```

```
/*[Nivel Silo Pienso]*/   pinMode(pingPin1, OUTPUT);
/*[Nivel Silo Pienso]*/   digitalWrite(pingPin1, LOW);
/*[Nivel Silo Pienso]*/   delayMicroseconds(2);
/*[Nivel Silo Pienso]*/   digitalWrite(pingPin1, HIGH);
/*[Nivel Silo Pienso]*/  delayMicroseconds(5);
/*[Nivel Silo Pienso]*/   digitalWrite(pingPin1, LOW);

/*[Nivel Silo Pienso]*/   pinMode(pingPin1, INPUT);
/*[Nivel Silo Pienso]*/   duration1 = pulseIn(pingPin1, HIGH);

/*[Nivel Silo Pienso]*/   cm1 = microsecondsToCentimeters(duration1);

/*[Nivel Silo Pienso]*/   if (cm1 >= 400.0)
/*[Nivel Silo Pienso]*/ {
/*[Nivel Silo Pienso]*/   cm1 = 400.0;
/*[Nivel Silo Pienso]*/ }
/*[Nivel Silo Pienso]*/   dist1 = 100.0-(cm1/4.0);

/*[Nivel Silo Pienso]*/   Serial.print("/s016=");
/*[Nivel Silo Pienso]*/   Serial.print(dist1);
/*[Nivel Silo Pienso]*/   Serial.print(" ");
  delay(5000);
}
/*[Nivel Silo Pienso]*/ long microsecondsToCentimeters(long
microseconds)
/*[Nivel Silo Pienso]*/ {
/*[Nivel Silo Pienso]*/   return microseconds / 29.0 / 2.0;
/*[Nivel Silo Pienso]*/ }

/*[VelocidadViento]*/ void countAnemometer() {
/*[VelocidadViento]*/    numRevsAnemometer++;
/*[VelocidadViento]*/ }

/*[VelocidadViento]*/ void calcWindSpeed() {
/*[VelocidadViento]*/    int x, iSpeed;
/*[VelocidadViento]*/    long speed = 14920;
/*[VelocidadViento]*/    speed *= numRevsAnemometer;
/*[VelocidadViento]*/    speed /= MSECS_CALC_WIND_SPEED;
/*[VelocidadViento]*/    iSpeed = speed;

/*[VelocidadViento]*/    Serial.print("/s011=");
/*[VelocidadViento]*/    x = iSpeed / 6,2137;
/*[VelocidadViento]*/    Serial.print(x);
/*[VelocidadViento]*/    Serial.print(' ');
   delay(8500);

/*[VelocidadViento]*/    numRevsAnemometer = 0;
/*[VelocidadViento]*/ }
```

Código 1

4.2. CONFIGURACIÓN MÓDULOS DE COMUNACIÓN Xbee s2

Con la ayuda de una tabla (ver tabla 8) y del software XCTU configuramos los 3 módulos

de comunicaciones que tenemos: uno en cada módulo arduino (configurados como router) y otro (configurado como coordinador) destinado al puerto USB del Ordenador-Granja a través de un soporte-adaptador **Xbee Explorer USB**.

	Coordinador	Remoto 1	Remoto 2
	CO	**R1**	**R2**
Pan ID	2332	2332	2332
SH	13A200	13A200	13A200
SL	4091A2A0	4098D7E2	4091A15B
DH	0	0	0
DL	FFFF	0	0
NI	CO	R1	R2
	USB/PC	**ARDUINO C1**	**ARDUINO C2**

Tabla 8

En la tabla 9 podemos ver lo simple que resultaría la configuración de los nuevos Xbee si decidiésemos ampliar el sistema.

	Coordinador	Remoto 0	Remoto 1	Remoto 2	Remoto 3	…	Remoto n
	CO	**R0**	**R1**	**R2**	**R3**	**…**	**Rn**
Pan ID	2332	2332	2332	2332	2332	…	2332
SH	13A200	Dato Fab.	13A200	13A200	Dato Fab.	…	Dato Fab.
SL	4091A2A0	Dato Fab.	4098D7E2	4091A15B	Dato Fab.	…	Dato Fab.
DH	0	0	0	0	0	…	0
DL	FFFF	0	0	0	0	…	0
NI	CO	R0	R1	R2	R3	…	Rn
	USB/PC	**ARDUINO C0**	**ARDUINO C1**	**ARDUINO C2**	**ARDUINO C3**	**…**	**ARDUINO Cn**

Tabla 9

4.3. ALOJAMIENTO REMOTO DE DATOS

4.3.1. INTRODUCCIÓN

Las primeras versiones de los programas en java a ejecutar en Ordenador-Granja fueron desarrollados y probados en comunicación a una base de datos MySQL hospedada en el propio equipo (“localhost” en voz inglesa).

Los problemas surgen cuando descubrimos que no hay hostings gratuitos que permitan leer y, mucho menos, escribir en la base de datos desde cualquier código que no sea ejecutado en el propio hosting.

Sopesamos la opción de contratar un hosting de pago para la ejecución del presente proyecto, pero el problema de acceso a la base de datos también existía, aunque, como medida de gracia, se barajaba la posibilidad de permitir un acceso configurado

manualmente para una única dirección IP estática y única. Entonces empezamos a barajar la posibilidad de convertir la dirección IP de nuestro conexión a internet de la granja en estática.

Esta solución presenta, por una parte, el inconveniente de tener que cambiar código y gestionar nuevos permisos, por ejemplo, por un simple cambio de compañía suministradora de acceso a internet. Por otra parte, tenemos muchas dudas de si una conexión vía modem USB de telefonía móvil tendría la opción de poder fijar su dirección IP. La opción "internet móvil de banda ancha" podría ser en muchos casos la mejor (o única) posibilidad de tener acceso a internet desde una granja que, por su naturaleza, suelen encontrarse a cierta distancia de las zonas residenciales e industriales. Por lo que esta opción quedó desestimada.

Después de probar una serie de alojamientos gratuitos con la opción base de datos y de inspeccionar algunos de pago, nos decantamos por el servicio gratuito ofrecido por "www.000webhost.com". La base de datos disponible e incluida en la propia cuenta es MySQL, con lo que, por este motivo y por haber sido casualmente el tipo usado en nuestras primeras pruebas en "localhost", la opción de usar otro tipo de base de datos no fue sopesada.

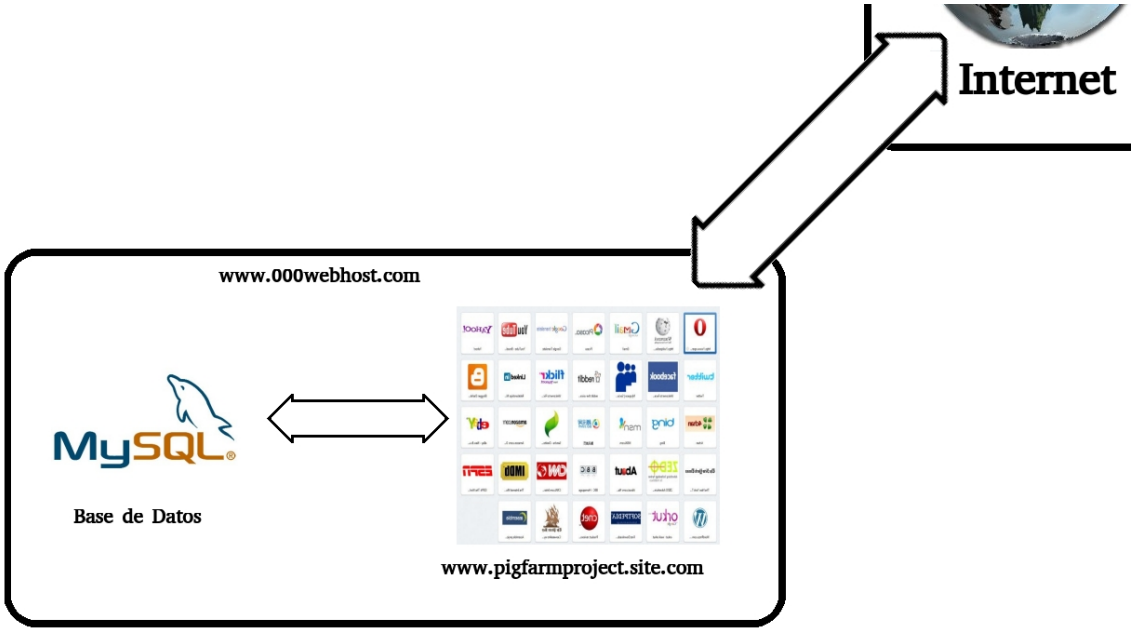


Fig. 10

Ya podemos explicar con mayor éxito porqué en el diagrama de bloques de la figura 10 las

comunicaciones con internet y el alojamiento remoto se ha representado conectadas exclusivamente a las páginas web. Serán éstas (su código), por lo tanto, la única posibilidad de que podamos acceder desde el exterior del alojamiento remoto a leer y escribir en dicha base de datos.

4.3.2 BASE DE DATOS

Un servicio de alojamiento remoto, como el que hemos usado, nos proporciona incluido en el mismo paquete, la posibilidad de alojar nuestra base de datos.

Hemos considerado necesario tener dos tipos de tablas almacenadas en nuestra base de datos; un tipo de tabla destinado a registrar los valores de todos los parámetros correspondientes a un mes natural, y un segundo tipo compuesto por una única tabla, que estará destinado a contener información general de cada sensor o parámetro.

Ambos tipos de tabla se detallan en los dos siguientes sub-apartados.

4.3.2.1. TABLA DE INFORMACIÓN GENERAL DE CADA PARÁMETRO

Tendremos por una parte una tabla (única por granja) a la que hemos llamado “TllTexto”, que incluirá, entre otra información, la básica para la identificación de cada campo con su sensor correspondiente. Contendrá también los comentarios que el usuario quiera que sean visibles en la ubicación que corresponda en la web, etc.

En la captura de pantalla de la figura 11 podemos ver la estructura de nuestra tabla. Bajo la primera columna, a la que hemos llamado “que” se encuentran los títulos de cada registro, el cual define el contenido del resto de los campos del mismo. Cada campo contendrá el tipo de información definida por el título del registro correspondiente al sensor del título de la columna. Pasamos, a continuación, a enumerar y describir, si procede, cada uno de estos registros:

- Registro “Comentarios” (Ver el apartado “Comments.jar”).
- “NombresCortos”. Contiene una identificación inequívoca aunque muy abreviada de cada sensor. Se usará, bre todo, en la parte de visualización de gráficos históricos de líneas y como texto en los e-mails de aviso del sistema de alertas.
- “NombresLargos”. Se usará para identificar cada sensor en el resto de aplicaciones y páginas web donde no haya tal limitación de espacio.

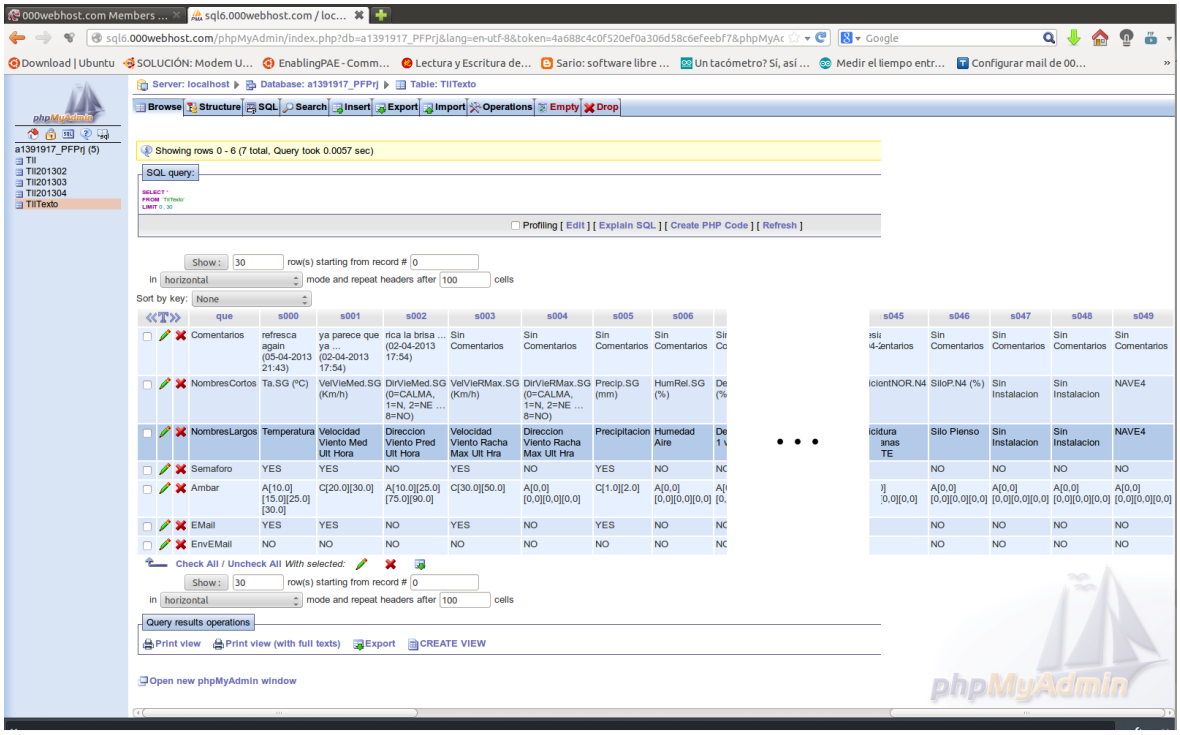


Fig. 11

Los cuatro registros restantes merecen una consideración aparte ya que guardan toda la información relativa al sistema de alertas/alarmas personalizable (ver también apartado relacionado "Warnings"). Y son:

– Registro "Ambar". Cada campo nos indica, con su primer carácter, el tipo de configuración queremos para la alerta del parámetro, pueden ser de tipo A, B o C; el resto del campo puede contener dos o cuatro valores numéricos entre corchetes. Ésto dependerá del tipo de configuración elegido. Hemos incorporado tres configuraciones diferentes:

· Tipo A: La zona considerada adecuada es una zona delimitada entre un valor mínimo y otro máximo; los valores más extremos podrían ser motivo de algún tipo de alerta. Ésta es una configuración conveniente, por ejemplo, para parámetros como la temperatura.

· Tipo B: Las zonas de riesgo son los valores bajos de la gama. Por ejemplo, conveniente para depósitos de agua y silos de pienso.

· Tipo C: Las zonas de riesgo son los valores altos de la gama. Es el caso, por ejemplo, de la velocidad del viento en el interior de una nave habitada por animales de corta edad.

- Registro "Semaforo". "YES" nos indica que queremos ver en nuestra web, junto a la imagen del parámetro correspondiente, un semáforo con uno de los tres colores iluminado (verde, amarillo o rojo) en función del valor de dicho parámetro y nuestra configuración incluida en el registro "Ambar" anteriormente descrito. "NO" nos indica que no queremos este servicio, en cuyo caso veremos en web un pequeño semáforo tachado.

- Registro "EMail". "YES" nos indica que queremos que sea enviado un aviso a nuestra dirección de correo electrónico si el valor del parámetro correspondiente está en zona roja de semáforo. "NO" indica que no queremos este servicio. En este caso será visible en web un pequeño sobre (tipo correspondencia ordinaria) tachado.

- Registro "EnvEMail". Un campo "YES" nos indica que un e-mail de alerta por valor extremo (zona roja semáforo) de este parámetro ya ha sido enviado y que el sistema de alertas correspondientes quedará en stand-by hasta que se proceda a la reinicialización del aviso por el método correspondiente (en este caso veríamos en web un sobre de correspondencia de color rojo). Esta medida extra (tener que reinicializar la alerta tras cada aviso por correo electrónico) fue adoptada para evitar la posibilidad de que cada minuto pudiera ser enviado un correo electrónico por un mismo motivo, lo cual causaría una molestia y, quizá, otros problemas indirectos(*) innecesarios. Normalmente una zona roja para alertas por e-mail se configura para situaciones donde es necesario acudir y actuar in situ, por lo que el procedimiento a seguir será casi siempre recibir la alerta, acudir al lugar, solucionar el/los problema/s que generaron dicha alerta y reinicializar la alarma para que nos vuelva a avisar si se produce una nueva situación digna de alerta. Análogamente a casos anteriores "NO" significará que ningún correo de alerta ha sido enviado desde la última reinicialización de la alerta. Ver apartado Email.jar.

(*) Algunos servidores de correo pueden interpretar como spam u otra amenaza la recepción sospechosamente reiterada de correos electrónicos enviados desde ciertos orígenes.

Es necesario crear esta tabla manualmente previamente a la puesta en marcha de nuestro sistema. Ésto se realiza de forma fácil y rápida importando una especie de plantilla de la tabla inicial que podemos tener guarda en un archivo de texto. Este archivo de texto no es

más que una lista de comandos SQL que al ejecutarse nos crea nuestra tabla con la estructura necesaria y la información básica de partida para nuestras necesidades. Ver código 2.

```
-- phpMyAdmin SQL Dump
-- version 2.11.4
-- http://www.phpmyadmin.net
--
-- Host: localhost
-- Generation Time: May 04, 2013 at 11:21 AM
-- Server version: 5.1.57
-- PHP Version: 5.2.17

SET SQL_MODE="NO_AUTO_VALUE_ON_ZERO";

--
-- Database: `a1391917_PFPrj`
--

-- --------------------------------------------------------

--
-- Table structure for table `TllTexto`
--

CREATE TABLE `TllTexto` (
 `que` varchar(20) COLLATE utf8_spanish2_ci NOT NULL DEFAULT '',
 `s000` varchar(120) COLLATE utf8_spanish2_ci DEFAULT NULL,
 `s001` varchar(120) COLLATE utf8_spanish2_ci DEFAULT NULL,
 `s002` varchar(120) COLLATE utf8_spanish2_ci DEFAULT NULL,
 `s003` varchar(120) COLLATE utf8_spanish2_ci DEFAULT NULL,
 `s004` varchar(120) COLLATE utf8_spanish2_ci DEFAULT NULL,
 `s005` varchar(120) COLLATE utf8_spanish2_ci DEFAULT NULL,
 `s006` varchar(120) COLLATE utf8_spanish2_ci DEFAULT NULL,
 `s007` varchar(120) COLLATE utf8_spanish2_ci DEFAULT NULL,
 `s008` varchar(120) COLLATE utf8_spanish2_ci DEFAULT NULL,
 `s009` varchar(120) COLLATE utf8_spanish2_ci DEFAULT NULL,
 `s010` varchar(120) COLLATE utf8_spanish2_ci DEFAULT NULL,
 `s011` varchar(120) COLLATE utf8_spanish2_ci DEFAULT NULL,
 `s012` varchar(120) COLLATE utf8_spanish2_ci DEFAULT NULL,
 `s013` varchar(120) COLLATE utf8_spanish2_ci DEFAULT NULL,
 `s014` varchar(120) COLLATE utf8_spanish2_ci DEFAULT NULL,
 `s015` varchar(120) COLLATE utf8_spanish2_ci DEFAULT NULL,
 `s016` varchar(120) COLLATE utf8_spanish2_ci DEFAULT NULL,
 `s017` varchar(120) COLLATE utf8_spanish2_ci DEFAULT NULL,
 `s018` varchar(120) COLLATE utf8_spanish2_ci DEFAULT NULL,
 `s019` varchar(120) COLLATE utf8_spanish2_ci DEFAULT NULL,
 `s020` varchar(120) COLLATE utf8_spanish2_ci DEFAULT NULL,
 `s021` varchar(120) COLLATE utf8_spanish2_ci DEFAULT NULL,
 `s022` varchar(120) COLLATE utf8_spanish2_ci DEFAULT NULL,
 `s023` varchar(120) COLLATE utf8_spanish2_ci DEFAULT NULL,
```

```
 `s024` varchar(120) COLLATE utf8_spanish2_ci DEFAULT NULL,
 `s025` varchar(120) COLLATE utf8_spanish2_ci DEFAULT NULL,
 `s026` varchar(120) COLLATE utf8_spanish2_ci DEFAULT NULL,
 `s027` varchar(120) COLLATE utf8_spanish2_ci DEFAULT NULL,
 `s028` varchar(120) COLLATE utf8_spanish2_ci DEFAULT NULL,
 `s029` varchar(120) COLLATE utf8_spanish2_ci DEFAULT NULL,
 `s030` varchar(120) COLLATE utf8_spanish2_ci DEFAULT NULL,
 `s031` varchar(120) COLLATE utf8_spanish2_ci DEFAULT NULL,
 `s032` varchar(120) COLLATE utf8_spanish2_ci DEFAULT NULL,
 `s033` varchar(120) COLLATE utf8_spanish2_ci DEFAULT NULL,
 `s034` varchar(120) COLLATE utf8_spanish2_ci DEFAULT NULL,
 `s035` varchar(120) COLLATE utf8_spanish2_ci DEFAULT NULL,
 `s036` varchar(120) COLLATE utf8_spanish2_ci DEFAULT NULL,
 `s037` varchar(120) COLLATE utf8_spanish2_ci DEFAULT NULL,
 `s038` varchar(120) COLLATE utf8_spanish2_ci DEFAULT NULL,
 `s039` varchar(120) COLLATE utf8_spanish2_ci DEFAULT NULL,
 `s040` varchar(120) COLLATE utf8_spanish2_ci DEFAULT NULL,
 `s041` varchar(120) COLLATE utf8_spanish2_ci DEFAULT NULL,
 `s042` varchar(120) COLLATE utf8_spanish2_ci DEFAULT NULL,
 `s043` varchar(120) COLLATE utf8_spanish2_ci DEFAULT NULL,
 `s044` varchar(120) COLLATE utf8_spanish2_ci DEFAULT NULL,
 `s045` varchar(120) COLLATE utf8_spanish2_ci DEFAULT NULL,
 `s046` varchar(120) COLLATE utf8_spanish2_ci DEFAULT NULL,
 `s047` varchar(120) COLLATE utf8_spanish2_ci DEFAULT NULL,
 `s048` varchar(120) COLLATE utf8_spanish2_ci DEFAULT NULL,
 `s049` varchar(120) COLLATE utf8_spanish2_ci DEFAULT NULL,
 PRIMARY KEY (`que`)
) ENGINE=MyISAM DEFAULT CHARSET=utf8 COLLATE=utf8_spanish2_ci;

--
-- Dumping data for table `TllTexto`
--

INSERT INTO `TllTexto` VALUES('Comentarios', 'Sin Comentarios', 'Sin
Comentarios', 'Sin Comentarios', 'Sin Comentarios', 'Sin Comentarios', 'Sin
Comentarios', 'Sin Comentarios', 'Sin Comentarios', 'Sin Comentarios', 'Sin
Comentarios', 'Sin Comentarios', 'Sin Comentarios', 'Sin Comentarios', 'Sin
Comentarios', 'Sin Comentarios', 'Sin Comentarios', 'Sin Comentarios', 'Sin
Comentarios', 'Sin Comentarios', 'Sin Comentarios', 'Sin Comentarios', 'Sin
Comentarios', 'Sin Comentarios', 'Sin Comentarios', 'Sin Comentarios', 'Sin
Comentarios', 'Sin Comentarios', 'Sin Comentarios', 'Sin Comentarios', 'Sin
Comentarios', 'Sin Comentarios', 'Sin Comentarios', 'Sin Comentarios', 'Sin
Comentarios', 'Sin Comentarios', 'Sin Comentarios', 'Sin Comentarios', 'Sin
Comentarios', 'Sin Comentarios', 'Sin Comentarios', 'Sin Comentarios', 'Sin
Comentarios', 'Sin Comentarios', 'Sin Comentarios', 'Sin Comentarios', 'Sin
Comentarios', 'Sin Comentarios', 'Sin Comentarios', 'Sin Comentarios', 'Sin
Comentarios');
INSERT INTO `TllTexto` VALUES('NombresCortos', 'Ta.SG (°C)', 'VelVieMed.SG
(Km/h)', 'DirVieMed.SG (0=CALMA, 1=N, 2=NE … 8=NO)', 'VelVieRMax.SG
(Km/h)', 'DirVieRMax.SG (0=CALMA, 1=N, 2=NE … 8=NO)', 'Precip.SG (mm)',
'HumRel.SG (%)', 'Dep1v70m.SG (%)', 'Dep2v20m.SG (Courier New%)',
```

```
'ServiciosGenerales', 'Ta.N1 (°C)', 'VelVie.N1 (Km/h)', 'Dep1v2m.N1 (%)',
'Dep2v2m.N1 (%)', 'ApVentSUR.N1 (%)', 'ApVentNOR.N1 (%)', 'SiloP.N1 (%)',
'Sin Instalacion', 'Sin Instalacion', 'NAVE1', 'Ta.N2 (°C)', 'VelVie.N2
(Km/h)', 'Dep1v2m.N2 (%)', 'Dep2v2m.N2 (%)', 'ApVentSUR.N2 (%)',
'ApVentNOR.N2 (%)', 'SiloP.N2 (%)', 'Sin Instalacion', 'Sin Instalacion',
'NAVE2', 'Ta.N3 (°C)', 'VelVie.N3 (Km/h)', 'Dep1v2m.N3 (%)', 'Dep2v2m.N3
(%)', 'ApVentSUR.N3 (%)', 'ApVentNOR.N3 (%)', 'SiloP.N3 (%)', 'Sin
Instalacion', 'Sin Instalacion', 'NAVE3', 'Ta.N4 (°C)', 'VelVie.N4 (Km/h)',
'Dep1v2m.N4 (%)', 'Dep2v2m.N4 (%)', 'ApVentSUR.N4 (%)', 'ApVentNOR.N4
(%)', 'SiloP.N4 (%)', 'Sin Instalacion', 'Sin Instalacion', 'NAVE4');
INSERT INTO `TllTexto` VALUES('NombresLargos', 'Temperatura', 'Velocidad
Viento Med Ult Hora', 'Direccion Viento Pred Ult Hora', 'Velocidad Viento
Racha Max Ult Hra', 'Direccion Viento Racha Max Ult Hra',
'Precipitacion', 'Humedad Aire', 'Deposito Agua 1 v70m3', 'Deposito Agua 2
v20m3', 'ServiciosGenerales', 'Temperatura', 'Velocidad Viento', 'Deposito
Agua 1 v2m3', 'Deposito Agua 2 v2m3', 'Apertura Ventanas SUR', 'Apertura
Ventanas NORTE', 'Silo Pienso', 'Sin Instalacion', 'Sin Instalacion',
'NAVE1', 'Temperatura', 'Velocidad Viento', 'Deposito Agua 1 v2m3',
'Deposito Agua 2 v2m3', 'Apertura Ventanas SUR', 'Apertura Ventanas
NORTE', 'Silo Pienso', 'Sin Instalacion', 'Sin Instalacion', 'NAVE2',
'Temperatura', 'Velocidad Viento', 'Deposito Agua 1 v2m3', 'Deposito Agua
2 v2m3', 'Apertura Ventanas SUR', 'Apertura Ventanas NORTE', 'Silo
Pienso', 'Sin Instalacion', 'Sin Instalacion', 'NAVE3', 'Temperatura',
'Velocidad Viento', 'Deposito Agua 1 v2m3', 'Deposito Agua 2 v2m3',
'Apertura Ventanas SUR', 'Apertura Ventanas NORTE', 'Silo Pienso', 'Sin
Instalacion', 'Sin Instalacion', 'NAVE4');
INSERT INTO `TllTexto` VALUES('Semaforo', 'NO', 'NO', 'NO', 'NO', 'NO', 'NO',
'NO', 'NO', 'NO', 'NO', 'NO', 'NO', 'NO', 'NO', 'NO', 'NO', 'NO', 'NO', 'NO',
'NO', 'NO', 'NO', 'NO', 'NO', 'NO', 'NO', 'NO', 'NO', 'NO', 'NO', 'NO', 'NO',
'NO', 'NO', 'NO', 'NO', 'NO', 'NO', 'NO', 'NO', 'NO', 'NO', 'NO', 'NO', 'NO',
'NO', 'NO', 'NO', 'NO', 'NO');
INSERT INTO `TllTexto` VALUES('Ambar', 'A[0,0][0,0][0,0][0,0]', 'A[0,0]
[0,0][0,0][0,0]', 'A[0,0][0,0][0,0][0,0]', 'A[0,0][0,0][0,0][0,0]',
'A[0,0][0,0][0,0][0,0]', 'A[0,0][0,0][0,0][0,0]', 'A[0,0][0,0][0,0]
[0,0]', 'A[0,0][0,0][0,0][0,0]', 'A[0,0][0,0][0,0][0,0]', 'A[0,0][0,0]
[0,0][0,0]', 'A[0,0][0,0][0,0][0,0]', 'A[0,0][0,0][0,0][0,0]', 'A[0,0]
[0,0][0,0][0,0]', 'A[0,0][0,0][0,0][0,0]', 'A[0,0][0,0][0,0][0,0]',
'A[0,0][0,0][0,0][0,0]', 'A[0,0][0,0][0,0][0,0]', 'A[0,0][0,0][0,0]
[0,0]', 'A[0,0][0,0][0,0][0,0]', 'A[0,0][0,0][0,0][0,0]', 'A[0,0][0,0]
[0,0][0,0]', 'A[0,0][0,0][0,0][0,0]', 'A[0,0][0,0][0,0][0,0]', 'A[0,0]
[0,0][0,0][0,0]', 'A[0,0][0,0][0,0][0,0]', 'A[0,0][0,0][0,0][0,0]',
'A[0,0][0,0][0,0][0,0]', 'A[0,0][0,0][0,0][0,0]', 'A[0,0][0,0][0,0]
[0,0]', 'A[0,0][0,0][0,0][0,0]', 'A[0,0][0,0][0,0][0,0]', 'A[0,0][0,0]
[0,0][0,0]', 'A[0,0][0,0][0,0][0,0]', 'A[0,0][0,0][0,0][0,0]', 'A[0,0]
[0,0][0,0][0,0]', 'A[0,0][0,0][0,0][0,0]', 'A[0,0][0,0]Courier New[0,0]
[0,0]', 'A[0,0][0,0][0,0][0,0]', 'A[0,0][0,0][0,0][0,0]', 'A[0,0][0,0]
[0,0][0,0]', 'A[0,0][0,0][0,0][0,0]', 'A[0,0][0,0][0,0][0,0]', 'A[0,0]
[0,0][0,0][0,0]', 'A[0,0][0,0][0,0][0,0]', 'A[0,0][0,0][0,0][0,0]',
'A[0,0][0,0][0,0][0,0]', 'A[0,0][0,0][0,0][0,0]', 'A[0,0][0,0][0,0]
[0,0]', 'A[0,0][0,0][0,0][0,0]', 'A[0,0][0,0][0,0][0,0]');
INSERT INTO `TllTexto` VALUES('EMail', 'NO', 'NO', 'NO', 'NO', 'NO', 'NO',
```

```
'NO', 'NO', 'NO', 'NO', 'NO', 'NO', 'NO', 'NO', 'NO', 'NO', 'NO', 'NO',
'NO', 'NO', 'NO', 'NO', 'NO', 'NO', 'NO', 'NO', 'NO', 'NO', 'NO', 'NO',
'NO', 'NO', 'NO', 'NO', 'NO', 'NO', 'NO', 'NO', 'NO', 'NO', 'NO', 'NO',
'NO', 'NO', 'NO', 'NO', 'NO');
INSERT INTO `TllTexto` VALUES('EnvEMail', 'NO', 'NO', 'NO', 'NO', 'NO', 'NO',
'NO', 'NO', 'NO', 'NO', 'NO', 'NO', 'NO', 'NO', 'NO', 'NO', 'NO', 'NO',
'NO', 'NO', 'NO', 'NO', 'NO', 'NO', 'NO', 'NO', 'NO', 'NO', 'NO', 'NO',
'NO', 'NO', 'NO', 'NO', 'NO', 'NO', 'NO', 'NO', 'NO', 'NO', 'NO', 'NO',
'NO', 'NO', 'NO', 'NO', 'NO');
```

4.3.2.2. TABLAS DONDE SE REGISTRA LA INFORMACIÓN DE LOS PARÁMETROS CAPTADOS

En este apartado vamos a describir las tablas donde irán siendo registrados los valores de los parámetros captados por los sensores así como los que obtenemos aprovechando otros medios como es el caso de los valores ambientales del exterior, los cuales corresponden a la estación meteorológica ubicada en el municipio de Cartagena y son recopilados de la propia página web de la Agencia Estatal de Meteorología (AEMET) como veremos en el apartado "DescargarFicherosDatosAemet.jar" que describe la aplicación java que se encarga de realizar dicha tarea.

Tendremos una tabla para cada granja y por cada mes natural del calendario donde se hayan registrado datos. El nombre de cada tabla estará compuesto por los caracteres "Tll" seguidos del año (4 dígitos) y el mes (siempre 2 dígitos, ej.: "03" equivale al mes de marzo) correspondiente.

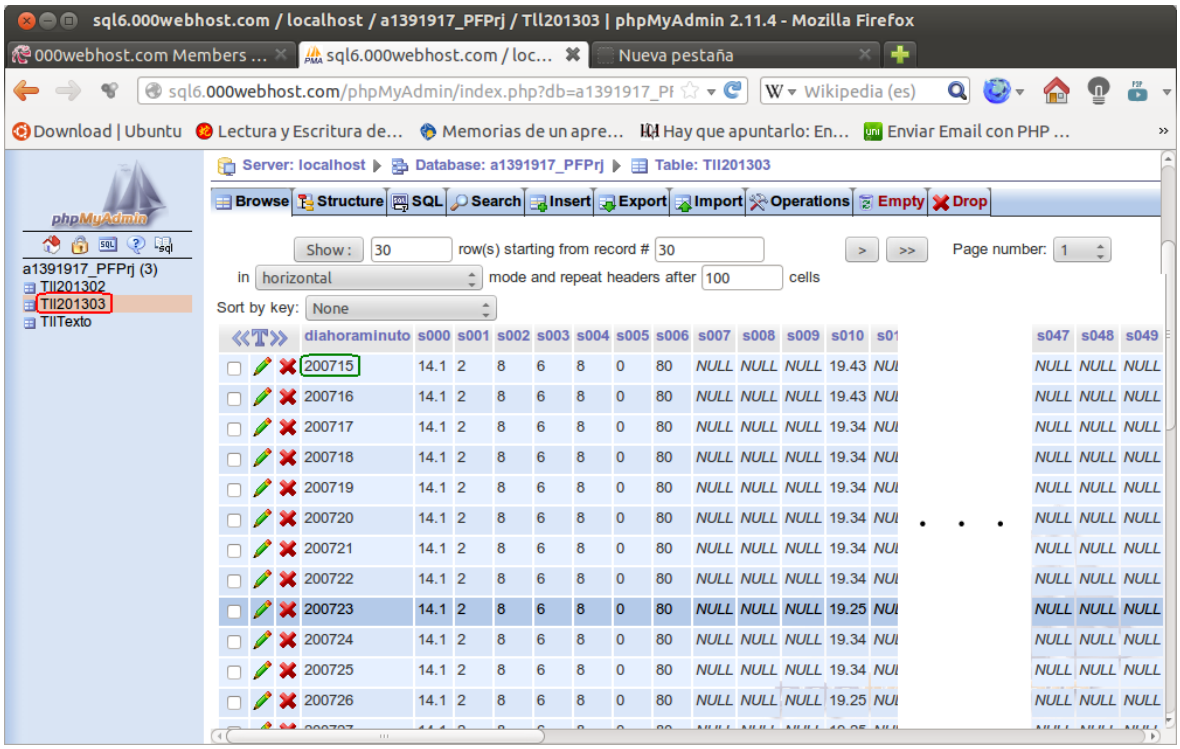


Fig. 12

Enmarcado en rojo en la figura 12 (en la izquierda) está el nombre de la tabla de la granja de Tallante ("Tll" en clave abreviada, sería fácil la inclusión de otras granjas de común propiedad y/o explotación en nuestra solución "PigFarmProject") y al año y mes correspondientes, por este orden. En este caso concreto hemos enmarcado la leyenda "Tll201303" que nos indica que la tabla contiene todos los registros de datos que corresponden al mes de marzo de 2.013 de la granja de Tallante.

Enmarcado en verde vemos un ejemplo que nos sirve para explicar la manera en que vamos a ir nombrando cada registro. Tal y como podemos adivinar por el título de la columna ("diahoraminuto") los dos primeros caracteres del campo corresponden al día del mes (siempre 2 dígitos), los dos siguientes a la hora del día en formato 24 horas y los dos de la derecha al minuto del tiempo real de toma de los datos de ese registro.

De esta manera podremos acceder a cualquier dato o rango temporal de datos de una forma lógica y fácilmente automatizable en cualquier lenguaje de programación elegido. En nuestro caso, y por las limitaciones ya descritas, será siempre en lenguaje PHP y ejecutándose desde el mismo alojamiento que la base de datos. Usaremos para ello las funciones de dicho lenguaje con posibilidad de relacionarse con la base de datos usando líneas de comandos propias del lenguaje SQL que requiere esta última.

Cada una de las tablas destinadas a contener los valores correspondientes a cada mes natural, será creada automáticamente cuando llegue la necesidad de guardar el primer registro de un nuevo mes.

4.4. INTERFACES DE USUARIO EN ORDENADOR-GRANJA

4.4.1. PigFarmProject.jar

El primer programa que será ejecutado al arrancar el sistema (configurado por nosotros a través de "Aplicaciones al Inicio" de ubuntu) será el que hemos llamado *PigFarmProject.jar* y se ejecutarán los siguientes pasos por este orden:

- Abrirá el navegador firefox con una página de inicio.
- Esperará unos segundos para asegurarse de que el paso anterior se ha ejecutado antes de pasar al punto siguiente.
- Ejecutará el programa DescargarFicherosDatosAemet.jar (ver apartado "DescargarFicherosDatosAemet.jar").
- Ejecutará el programa SerialTalk0.jar. (ver apartado "serialtalk0")
- Abrirá una consola (captura mostrada en la figura 13) desde donde podremos abrir otras opciones de nuestro sistema, como son la de "Introducción de Comentarios Visibles desde Web" (ver apartado "Comments.jar"), la de "Introducción de parámetros para la configuración de alertas" (ver apartado "warnings") y otra que nos proporciona una manera rápida de "reinicializar las alertas enviadas por e-mail" y así reactivarlas rápidamente si ése fuera nuestro deseo (ver apartado "EMail").

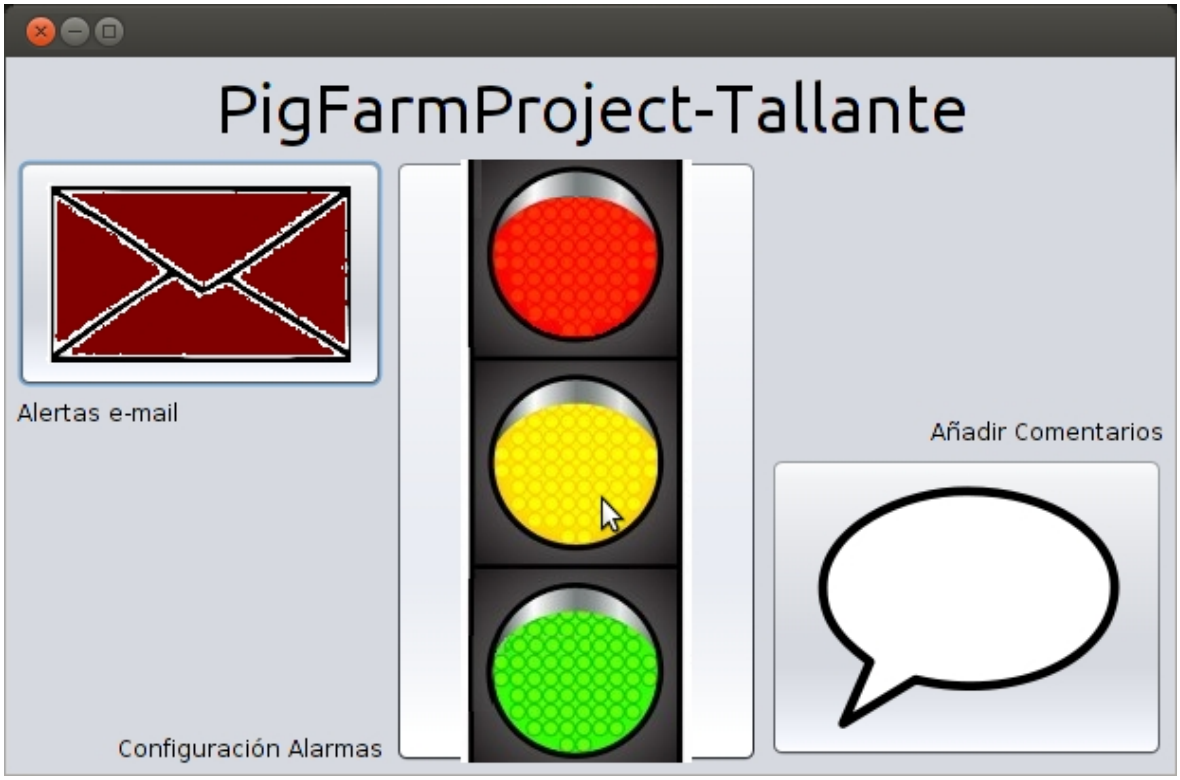


Fig. 13

4.4.2. SerialTalk0.jar

Es el programa que podríamos considerar como principal, ya que es el encargado de recopilar los datos de los sensores y enviarlos cada minuto a la base de datos remota con ayuda de una página web descrita más adelante.

Si el programa fuera ejecutado desde un terminal veríamos como cada algunos segundos nos van apareciendo algo como lo que vemos entre comillas en cada línea, a la derecha y entre paréntesis hemos añadido una breve explicación. Vamos a ver como funcionaría un ciclo cualquiera de toma de datos:

- Última actualización.
- "/s010=22.5" (el valor de la temperatura interior de Nave 1 es de 22,5ºC).
- "/s011=0" (la velocidad del viento en el interior de Nave 1 es de 0 Km/h).
- "/s020=21.8" (el valor de la temperatura interior de Nave 2 es de 22,5ºC).
- "/s012=56" (el depósito de agua nº 1 de Nave 1 está al 56% de su capacidad máxima).
- "/s022=15" (el depósito de agua nº 1 de Nave 2 está al 15% de su capacidad máxima).
- "/s013=21" ...
- …
- "/s010=23.5" <---- primer sensor repetido desde la última actualización (primer

punto de la presente lista). En este punto el programa procede al envío de los valores de todos los sensores desde la última actualización hasta el anterior al actual y se reinician los valores de todos los sensores excepto del actual (este valor será considerado el primero del siguiente ciclo). Como decimos, un nuevo ciclo ya ha comenzado y será idéntico a todos los anteriores y a todos los siguientes, hasta el "infinito".

Este mecanismo fue desarrollado con el fin de tener un sistema que pudiera seguir en funcionamiento aunque alguno/s módulo/s Arduino/XBee/Sensores deje/n de hacerlo o presente/n problemas de comunicación.

Entrecomillamos "infinito" porque vimos la necesidad de terminar y re-arrancar el programa cada cierto período para evitar problemas que observamos que se generaban con el tiempo de ejecución. Debido a que nos apoyamos en procedimientos y librerías que no conocemos en profundidad, la solución de terminar/re-arrancar con otro mini-programa java auxiliar nos pareció la más adecuada por sencillez y eficacia, a pesar de ser un procedimiento no demasiado elegante ni ortodoxo.

La tarea de grabar/leer los registros en la base de datos se la encargamos a una página web, varias en realidad, que son las que realizarán todas las operaciones con la base de datos. Ésta es una necesidad (o alternativa) que surge de los problemas que fueron tratados con mayor detalle en el apartado "Alojamiento Remoto".

La forma en que este programa envía los datos a la base de datos es abriendo una dirección url en un navegador pasando toda la información necesaria en la misma barra de direcciones (método GET de php).

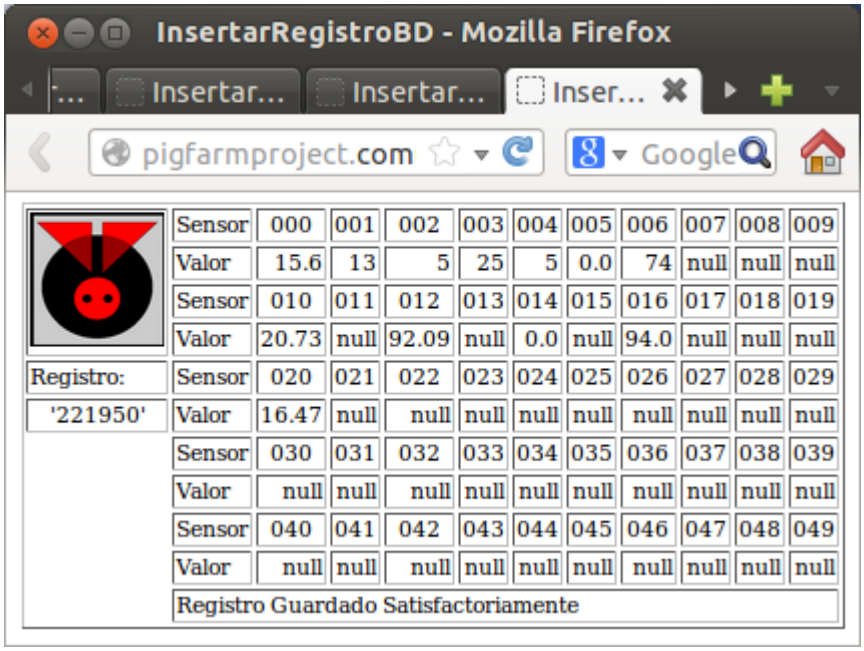
Incluimos, a modo de ejemplo, el código (codigo 3) que corresponde a la función del programa java considerado como "principal" (SerialTalk0.jar) encargada de abrir la dirección URL correspondiente, ya que las necesidades de comunicación del resto de programas java se realizará de un modo casi idéntico a éste.

```
public static void abrirURL(String[] a)throws Exception
{
    String direccion="http://pigfarmproject.comuv.com/DataInsert.php";
    String gets="?";
    gets+="a="+a[0]+"&";
    gets+="m="+a[1]+"&";
    gets+="dhm="+a[2]+"&";
    gets+="s000="+a[3]+"&";
    gets+="s001="+a[4]+"&";

      ...

    gets+="s049="+a[52];

    Desktop.getDesktop().browse(new URI(direccion+gets));
```

```
}
```

Cód. 2

Esta función, y más en concreto su línea "`Desktop.getDesktop().browse(new URI(direccion+gets));`", abrirá una nueva pestaña(**) en nuestro navegador por defecto con una página que nos dará el aspecto que se ve en la figura 14 en ausencia de problemas.



Sensor	000	001	002	003	004	005	006	007	008	009
Valor	15.6	13	5	25	5	0.0	74	null	null	null
Sensor	010	011	012	013	014	015	016	017	018	019
Valor	20.73	null	92.09	null	0.0	null	94.0	null	null	null
Sensor	020	021	022	023	024	025	026	027	028	029
Valor	16.47	null	null	null	null	null	null	null	null	null
Sensor	030	031	032	033	034	035	036	037	038	039
Valor	null	null	null	null	null	null	null	null	null	null
Sensor	040	041	042	043	044	045	046	047	048	049
Valor	null	null	null	null	null	null	null	null	null	null

Fig. 14

Este programa (página web) alojado en el hosting será tratado en la sección correspondiente. Por ahora, y para este punto basta decir que solo será llamado desde el programa java de ejecución en Ordenador-Granja ya que no tiene conexión alguna ni desde la portada ni desde cualquier otra página relacionada con la interfaz de usuario remoto por motivos de funcionamiento y, sobre todo, de seguridad.

Dentro de este programa, posteriormente al proceso de registro de los últimos datos recopilados, se ejecuta el procedimiento encargado de comprobar si alguno/s de los últimos datos recibidos debe/n ser motivo de envío de una alerta vía e-mail; en caso de ser así, el programa ejecutará la rutina encargada de enviar dicho correo electrónico. En el programa, al que llamamos emailsender.php, tendremos configurados la/s dirección/es de correo electrónico a enviar, además el procedimientos que elaborará el contenido textual del mensaje: encabezado, texto donde se indican los parámetros que fueron motivo de alerta (número/s de señal/es, descripción/es y valor/es), etc.

Ejempo e-mail de alerta:

Asunto: PigFarmProject - Sistema de Alarmas

> Este mensaje fue enviado por Juan A. Pérez, de la empresa PigFarmProject
> Su e-mail es: el_ignaky@pigfarmproject.comuv.com
> Mensaje:
>
> - Alarma en: s012 - Dep1v2m.N1 (%) - Con valor: 8.89
>
> - Alarma en: s020 - Ta.N2 (?C) - Con valor: 17
>
> Enviado el 08-04-2013 22:57

(**) Este programa abrirá una nueva pestaña en nuestro navegador instalado por defecto **cada minuto**, por lo que recomendamos el uso de un navegador con limitador automático de pestañas. Nosotros usamos el navegador Firefox 20.0 de Mozilla con el complemento (extensión) Window and Tab Limiter 4.28.

4.4.3. Comments.jar

Antes de proceder a explicar el funcionamiento de dicho programa, creo obligado comentar el motivo por el cual creímos en la necesidad o, mejor dicho, en la conveniencia de incluir un sistema de comentarios que solo puedan ser escritos desde el Ordenador-Granja y que puedan ser leídos desde el terminal-navegador conectado a internet; y lo vamos a ver de una forma muy sencilla con unos ejemplos.

Para situarnos diremos que, por ejemplo, el nivel de agua en un depósito de una nave destinado al consumo de los animales, por lo general, siempre debería ser el máximo (100%) ya que, en su funcionamiento normal, la cantidad de agua que consumen los cerdos se repone automáticamente mediante un simple sistema de bolla; o sea, cualquier nivel lejano más o menos alejado del 100% podría ser indicador de un problema en el suministro particular o general de agua. Prácticamente solo cuando usamos el depósito de agua para suministrar un tratamiento (para tal caso se dosifica la cantidad de sustancia correspondiente en la medida de agua adecuada y se cierra la entrada de agua al depósito para no modificar la concentración de la misma) donde se deja consumir por los animales el tiempo necesario hasta su consumo total. Todo dependerá de la duración del producto una vez disuelto, de la cantidad y tamaño de los animales a tratar, de la capacidad de los depósitos, etc. En este caso, avisados por el comentario (Ejemplo: "(23-04-13 15:30) Agua medicada con Aspirina"), iremos observando con el paso del tiempo como el nivel de agua de ese depósito va menguando pero dentro de la normalidad más absoluta. Solo nos quedará estar pendientes para actuar cuando el nivel llegue al nivel mínimo que consideremos oportuno o esperar algún tipo de aviso si así lo tenemos configurado (como podremos observar en el apartado que trata sobre los diferentes sistemas de avisos previstos en el presente trabajo).

Otro ejemplo podría ser con el pienso o el ambiente de una instalación cuando ya no hay

animales y está en proceso de limpieza y desinfección. En estos casos los valores nulos de nivel de agua en depósitos, cantidad de pienso en el silo o valores ambientales no aptos, no nos deberán de preocupar si vemos en la página web un aviso, por ejemplo, como "Nave Vacía".

La consola del programa que estamos describiendo se muestra en la figura 15 y su manejo es tan sencillo como se intuye.

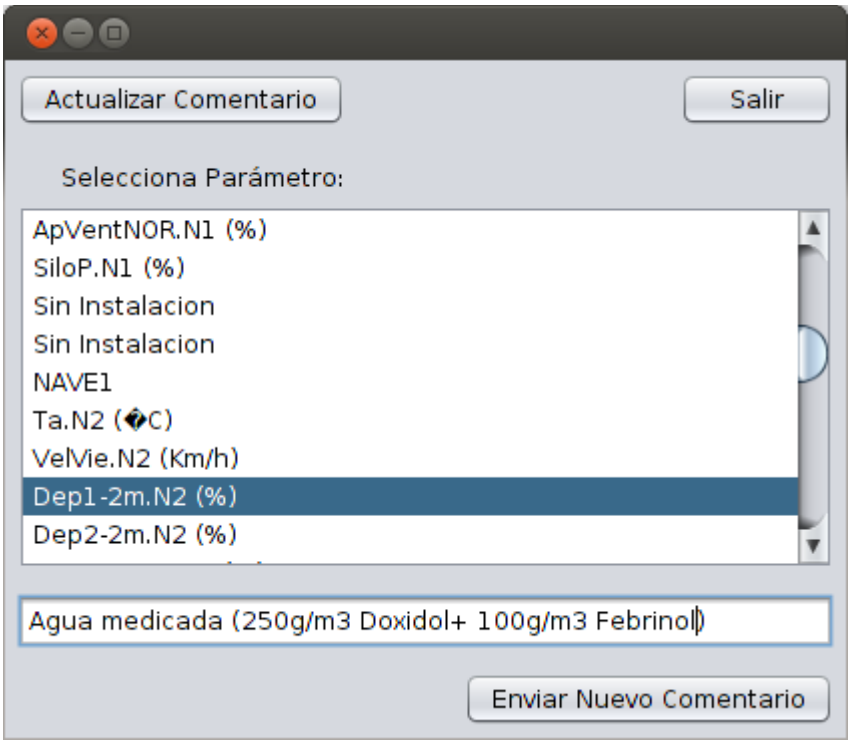


Fig. 15

Este programa envía los datos a la base de datos de una forma muy similar a la usada en el programa SerialTalk0.jar objeto del apartado anterior; o sea, a través de una página web que tampoco será accesible desde ningún punto de la interfaz de usuario en el Terminal-Navegador.

En la figura 16 podemos ver el resultado en el navegador de un registro de comentario terminado con éxito.

El comentario será guardado en el lugar adecuado y correspondiente al sensor de la tabla en la base de datos con la fecha y hora de introducción del mismo por parte del usuario en granja, que puede ser una manera lógica de detectar si algún comentario pudiera resultar, por ejemplo, obsoleto.

También hemos previsto ir actualizando un archivo de texto con un histórico de comentarios que podría servir al ganadero en un momento dado para un listado rápido de de tratamientos y otras incidencias en la granja.

Fig. 16

Hemos cambiado un poco la estética de presentación con respecto al mensaje que apareció en el apartado anterior con el único fin de que el usuario pueda identificar fácilmente que, después de enviar la actualización del módulo java, efectivamente el navegador consiguió el objetivo; que no es otro que insertar nuestro nuevo comentario en el campo del registro correspondiente de la tabla "TllTexto" de la base de datos.

4.4.4. DescargarFicherosDatosAemet.jar

Este programa descargará la última información meteorológica de la estación ubicada en el municipio de Cartagena de la Agencia Estatal de Meteorología (AEMET), leyendo directamente del código fuente de la página: "http://www.aemet.es/.../ultimosdatos?k=mur&l=7012C&w=0&datos=det&x=h24&f=Todas", la cual nos ofrece en las casillas que hemos enmarcado en color rojo en la figura 17, la información buscada a incluir en nuestra base de datos en sus posiciones correspondientes (en nuestro caso s000, s001, … s006).

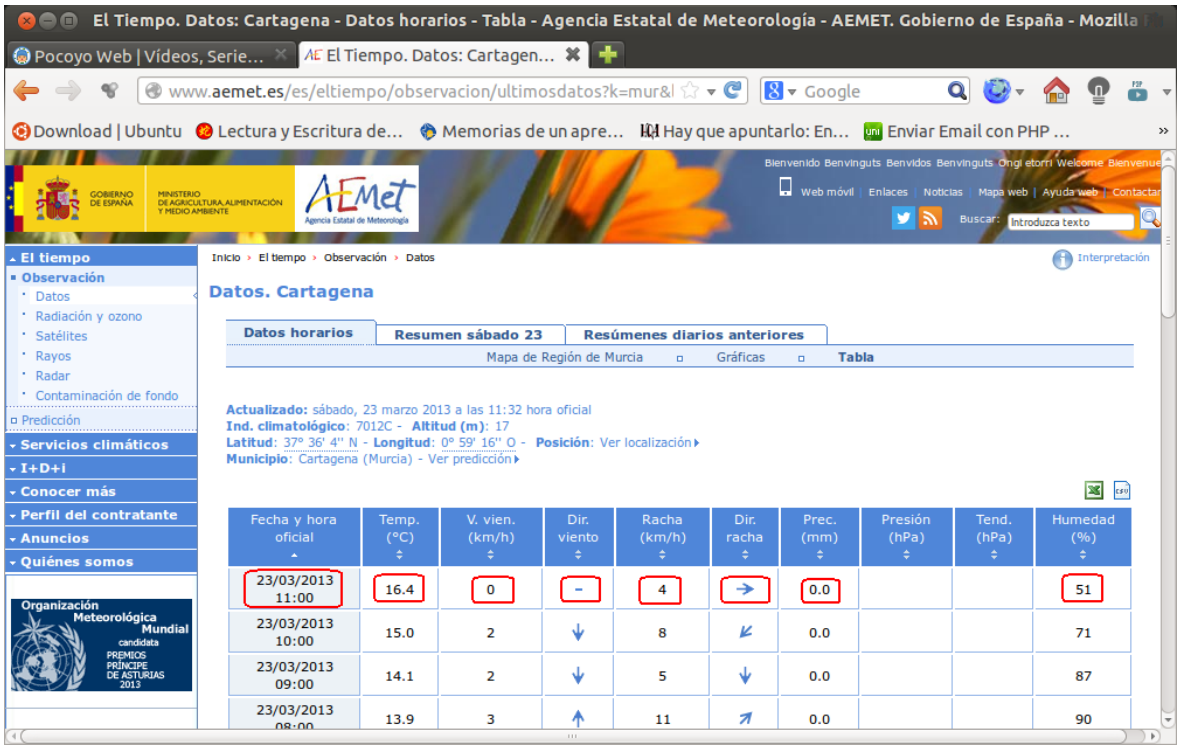


Fig. 17

Leyendo en el código fuente (html) de la página localizamos las casillas buscadas y guardamos la información que nos interesa en un archivo de texto local que va a disponer siempre de los últimos datos publicados en la web oficial de AEMET. Hemos programado un chequeo en busca de nuevas actualizaciones de datos cada 20 minutos (tiempo fácilmente modificable).

Al terminar de ejecutar cada uno de estos chequeos a la página de AEMET comenzará a ejecutarse el programa que pasamos a explicar a continuación será ejecutado.

4.4.5. LeerDatosAemet.jar

Este programa leerá el archivo de texto comentado en el punto anterior y se encargará de hacer las gestiones necesarias para actualizar los datos de los registros del período ya guardados. Así intentamos subsanar el problema del retraso de la página web de Aemet.

Como todos los que se tienen que comunicar con la base de datos remoto habrá que apoyarse en una página web alojada en el mismo servidor que dicha base de datos. En la figura 18 vemos un ejemplo de la información que nos presentará en el navegador.

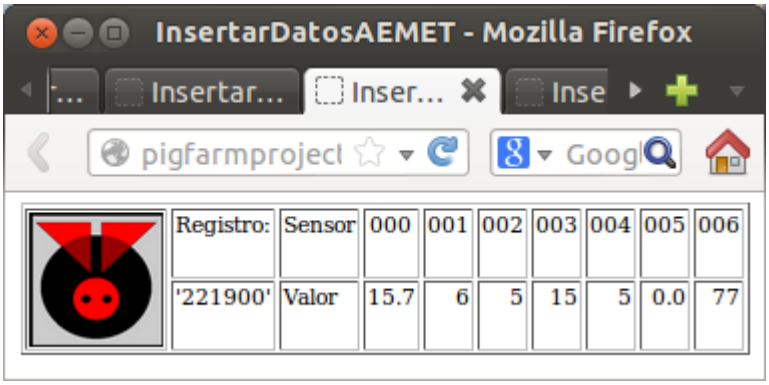


Fig. 18

4.4.6. Warnings.jar

Es la consola donde podemos configurar individualmente nuestro sistema de alertas visuales en web y alarmas vía e-mail.

- Pasamos a explicar el manejo de dicha consola (ver captura de pantalla de la figura 19). Primeramente debemos indicar el parámetro de la lista en el cual queremos modificar el sistema de alertas/alarmas, seguidamente seleccionaremos el/los tipo/s de alerta/s (semaforo en web y alerta e-mail) que queremos activar, pudiendo elegir uno de ellos, ambos o ninguno; posteriormente seleccionaremos un tipo de alerta (A, B o C) y, si procede, los valores límite. La filosofía de este sistema de avisos ya fue tratada en profundidad en el apartado "Tabla de información general de cada parámetro".

Este programa java se ayudará, como en todos los casos anteriormente descritos, de una página web para modificar la información necesaria en la base de datos.

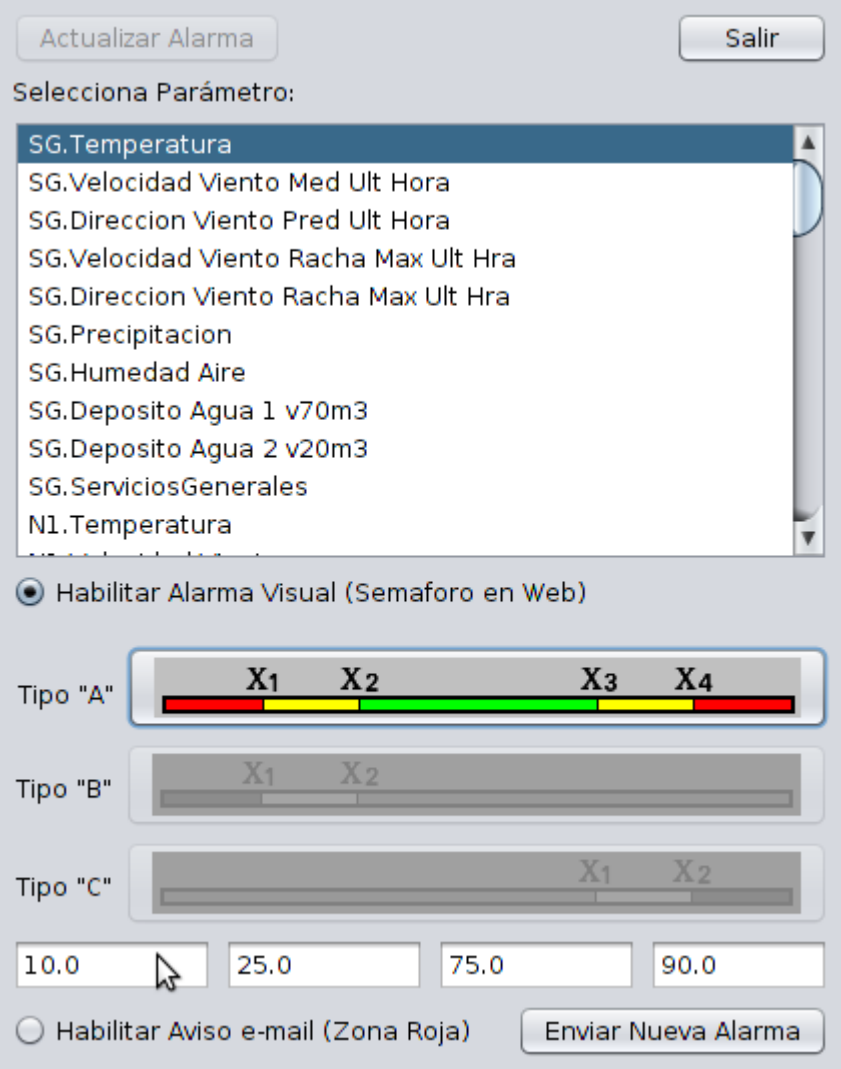


Fig. 19

4.4.7. EMail.jar

- Es una consola destinada únicamente a poder hacer una visualización rápida de las alertas enviadas vía e-mail y poder reiniciar las elegidas fácil y rápidamente (ver figura 20). Para mejor comprensión de la necesidad o, al menos, conveniencia de disponer de esta herramienta pueden ver también el apartado "Tabla de información general de cada parámetro".

Fig. 20

4.5. PROGRAMAS PARA LA ESCRITURA DE REGISTROS EN LA BASE DE DATOS

Aunque ya hemos hablado de ellos en la descripción de los programas java asociados que las requieren, a modo de introducción, pasamos a enumerar las características más importantes comunes a todos ellos:

- Son páginas lanzadas únicamente por su programa java asociado correspondiente y que deberá estar instalado y en ejecución en Ordenador-Granja.

- Por seguridad e integridad de nuestra base de datos, no existe ninguna posibilidad de acceder a ninguna de ellas desde enlace alguno de nuestras páginas web.

- Todas reciben la información por el método GET de php; ésto es, pasando toda la información necesaria a continuación de la dirección url y del símbolo "?" en la propia barra de direcciones (para conocer más acerca de éste método podemos visitar la página: "http://www.webestilo.com/php/php09b.phtml", por ejemplo).

La breve descripción de cada uno de ellos que ya fue adelantada en los apartados correspondientes de los programas java que requieren de su apertura, no va a evitar que en los apartados siguientes sean tratados en mayor profundidad.

Para mayor información sobre las tablas y registros que crean, leen y/o actualizan los programas en php que vamos a ver a continuación se pueden volver a consultar los apartados de este trabajo donde ya se abordaron; éstos son, "Tabla de información general de cada parámetro" y "Tablas donde se registra la información de los parámetros captados".

4.5.1. AemetInsert.php

Esta página, que ya apareció en el presente trabajo cuando describimos "su" programa java asociado (LeerDatosAemet.jar), tiene como misión principal la de actualizar los registros guardados correspondientes al último período ofrecido en la página web de la estación meteorológica de Cartagena de la Agencia Estatal de Meteorología (AEMET). La realización de esta tarea resulta necesaria ya que dichos datos pueden ser publicados en la web con un retraso considerable, normalmente de 30 a 45 minutos.

A pesar de este retraso, disponer de esta información guardada correctamente nos puede ser útil para, por ejemplo, estudiar con posterioridad el comportamiento térmico de las instalaciones en según qué condiciones.

Este programa se encarga de mantener actualizado un fichero de texto con los últimos datos del AEMET (la utilidad de este fichero la vamos a entender mejor tras examinar el siguiente apartado.

La captura de esta página web la pudimos ver en la figura 18.

4.5.2. DataInsert.php

Este programa (página web) es lanzado por su correspondiente programa java (SerialTalk0.jar) incluyendo en la barra de direcciones todos los valores de los sensores que tenemos en funcionamiento. Captura de pantalla en figura 14.

El propio "DataInsert.php" tomará los datos recibidos a través de la barra de direcciones (método GET), añadirá los datos de la última lectura de la web de AEMET (incluidos en el archivo de texto mencionado en el apartado anterior) e insertará todos ellos en los campos destinados a tal efecto en un nuevo registro de la tabla correspondiente al mes en curso de la base de datos. Ver apartado de base de datos tabla de registro de información captada.

4.5.3. InsertComments.php

Esta página sera lanzada por el programa java "Comments.jar" programa que es equivalente al anterior pero, en este caso, se encarga de actualizar la tabla llamada "TllTexto" con nuevos datos y comentarios, concretamente en los campos del registro de título "Comentarios". El funcionamiento y la filosofía es la misma que el caso de "DataInsert.php" y sucesivos de esta misma sección por lo que podemos dejar aquí su descripción.

La captura de pantalla la pudimos ver en la figura 16.

4.5.4. AlarmInsert.php

Esta página sera lanzada por el programa java "Warnings.jar", este programa en php se encarga de actualizar la tabla llamada "TllTexto" con nuevos datos; más concretamente los registros llamados "Ambar", "Semaforo" y "Email". La función y tipo de información que guardan estos registros fue descrita en el apartado "Tabla de información general de cada parámetro".

4.5.5. EnvEMailInsert.php

Esta página sera lanzada por el programa java "EMail.jar", este programa se encarga de actualizar la tabla llamada "TllTexto" con nuevos datos, concretamente el registro "EnvEMail". La función de este registro fue tratada con detalle en el apartado "Tabla de información general de cada parámetro".

4.6. INTERFACES PARA CONSULTA DE USUARIO EN INTERNET

4.6.1. default.php

Accederemos a default.php (portada) (figura 21) a través de la dirección url "pigfarmproject.comuv.com". En ella podemos ver una vista aérea de la granja y algunas de las instalaciones y servicios a los que podemos acceder gracias al mapeo de la imagen como se puede ver en el código que mostramos (código 4). Podremos acceder a la página donde seleccionar los gráficos históricos a visualizar, o ver en una página todos los depósitos de agua, o ver en otra todos los silos de pienso, o ver en otra todos los parámetros ambientales, o, por el contrario, ver los parámetros que corresponden a cada una de, lo que nosotros hemos llamado, las cinco instalaciones principales de la presente granja por separado; que son:

– Servicios Generales: Temperatura, velocidad y dirección de viento, humedad ambiental, precipitaciones (todos en exterior) y niveles de los depósitos de agua generales.

– Nave 1: Temperatura y velocidad de viento en el interior de la nave, apertura de ventanas, nivel de los depósitos de agua propios de la nave y nivel de pienso en el silo.

– Naves 2, 3 y 4: Idéntica cada una de ellas a Nave 1 en cantidad y tipo de parámetros.

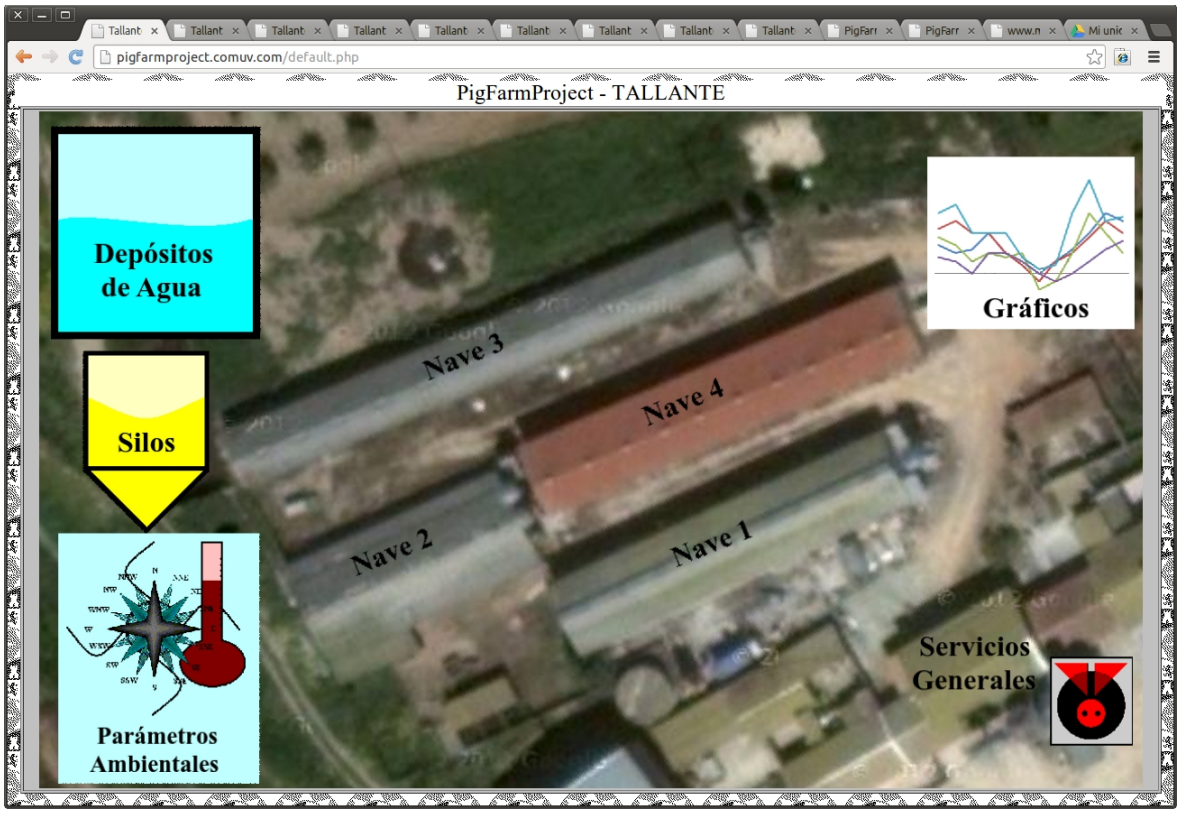


Fig. 21

```
<html>
	<head>
		<meta content="text/html; charset=ISO-8859-1" http-
equiv="content-type">
		<title>Tallante-PigFarmProject</title>
		<style type="text/css">body, p, li { background-position:
center top; background-image: url(Imagenes/upct0.png);}</style>
	</head>
	<body>
		<table style="text-align: left; margin-right: auto; width:
1266px; margin-left: auto; height: 719px; background-color: rgb(192,
192, 192);" border="1" cellpadding="2" cellspacing="2">
			<tbody>
				<tr>
					<td style="vertical-align: middle; text-
align: center; width: 800px;">
						<map name="Tallante">
							<area title="Graficos"
shape="poly" coords="977,50,977,240,1205,240,1205,50"
href="PigFarmProjectTallanteParamGrafico.php">
							<area title="Depositos"
shape="poly" coords="15,15,15,250,240,250,240,15"
href="PigFarmProjectTallanteDepositosAgua.php">
							<area title="Silos"
shape="poly" coords="50,260,50,395,120,460,188,395,188 ,260"
```

```
href="PigFarmProjectTallanteSilosPienso.php">
                                        <area title="Parametros
Ambientales" shape="poly" coords="21,465,21,740,240,740,240,465"
href="PigFarmProjectTallanteParametrosAmbientales.php">
                                        <area title="Nave 3"
shape="poly" coords="200,360,240,420,780,180,760,120"
href="PigFarmProjectTallanteNave3.php">
                                        <area title="Nave 2"
shape="poly" coords="260,500,320,600,540,500,500,400"
href="PigFarmProjectTallanteNave2.php">
                                        <area title="Nave 4"
shape="poly" coords="520,360,560,440,980,260,940,180"
href="PigFarmProjectTallanteNave4.php">
                                        <area title="Nave 1"
shape="poly" coords="560,520,600,600,960,440,920,360"
href="PigFarmProjectTallanteNave1.php">
                                        <area title="Servicios
Generales" href="PigFarmProjectTallanteServiciosGenerales.php"
shape="DEFAULT">
                                    </map>
                                    <big><span style="font-weight:
bold;"></span></big>
                                    <img src="Imagenes/TallanteUPS.png"
style="width: 1224px; height: 750px;" usemap="#Tallante" alt="Tallante">
                                </td>
                            </tr>
                        </tbody>
                    </table>
                </body>
</html>
```

Cód. 4

4.6.2. PigFarmProjectTallanteParamGrafico.php

Hemos incluido un enlace con imagen visible e ilustrativa (figura 22) a esta página desde cada una de las demás, ya que consideramos que es una de las de mayor interés junto con la del siguiente apartado, llamada "PigFarmProjectTallanteGraficos.php" y que es la consecuencia directa de lo que hagamos/seleccionemos en la presente antes de solicitarla.

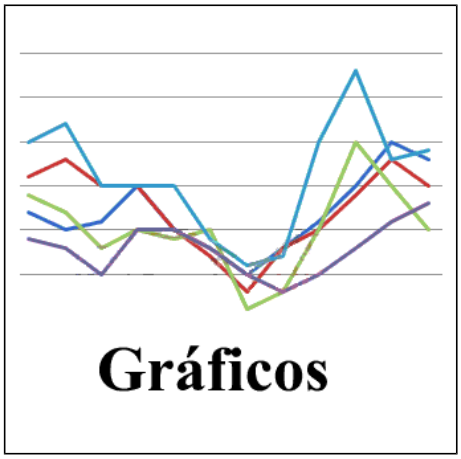


Fig. 22

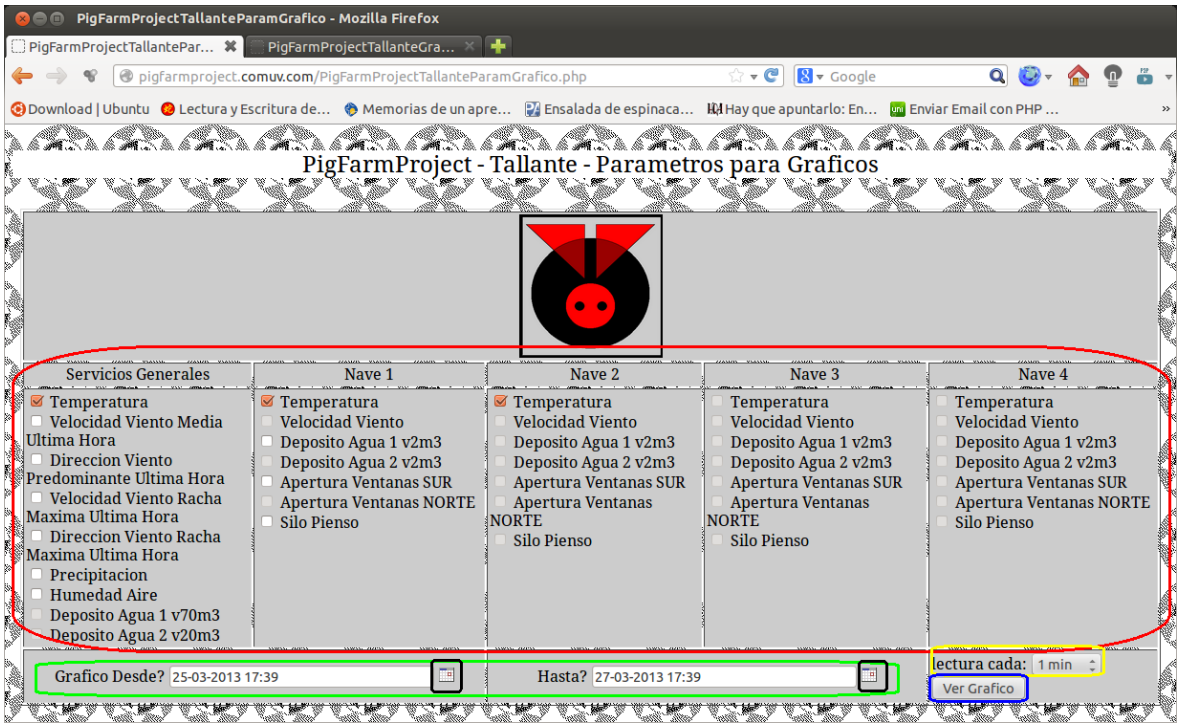


Fig. 23

En la figura 23 vemos una captura de pantalla con indicaciones para su mejor comprensión. La hemos dividido en 5 partes principales que describimos a continuación:

– Enmarcado en rojo. Aquí podemos seleccionar los parámetros que deseemos visualizar simultáneamente en la gráfica de tiempo. (En esta versión están habilitados solo los que hemos dispuesto como muestra práctica del presente proyecto). Hemos dejado activados (clickeados) por defecto algunos parámetros (de los realmente operativos) para facilitar la posibilidad de visualizar rápidamente una muestra del funcionamiento de esta parte de nuestro sistema.

– Enmarcado en verde vemos la parte donde se ha de introducir los datos de fecha y hora requeridos; o sea, principio y fin de nuestro gráfico temporal. Hemos dejado por defecto el momento actual como tiempo final del gráfico y 48 horas atrás como principio. El motivo es el mismo que el descrito al final del párrafo anterior.

– Enmarcados en negro están unos botones que nos van a desplegar una herramienta muy común en infinidad de páginas web llamada en voz inglesa "Date and Time Picker". La ardua tarea de introducir con el teclado fecha y hora exactamente en el formato requerido, podrá ser sustituida en su totalidad por simples clicks de ratón gracias a dicha herramienta. Podemos ver su aspecto en la figura 24.

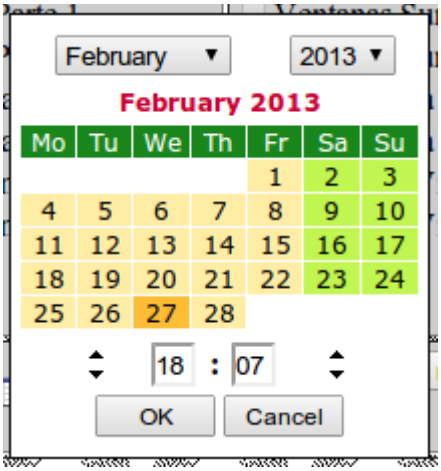


Fig. 24

– En amarillo vemos el lugar donde seleccionar, mediante menú auto-desplegable, el intervalo de tiempos deseado entre punto y punto a visualizar. Por defecto estará seleccionado en "5 minutos". Hemos dispuesto para este intervalo de tiempos de 1, 5, 10 y 30 minutos, así como de 1 y 2 horas.

– En azul tenemos el botón (tipo "Submit" en voz inglesa) usado para requerir la apertura de otra página (PigFarmProjectGraficos.php, ver apartado siguiente) enviando por el método GET de php todos los parámetros que hayamos dejado seleccionados.

De nuestro rigor a la hora de seleccionar toda la información en esta página dependerá la claridad y/o utilidad de los gráficos que se mostrarían a continuación de pulsar el botón "Ver Gráfico". Aunque ésto no debería preocuparnos en exceso más allá de los comienzos de uso de nuestro sistema, ya que, con un poco de experiencia será fácil obtener los gráficos de nuestro interés, de fácil interpretación y evitando retardos excesivos e innecesarios en la carga de los mismos.

4.6.3. PigFarmProjectGraficos.php

Ésta es la página que muestra el gráfico hayamos solicitado siguiendo las instrucciones del apartado anterior.

Para la creación del gráfico hemos usado una herramienta on-line de Google para la

creación de todo tipo de gráficas llamado “Google API Chart”.

Podemos satisfacer nuestra curiosidad acerca del mismo visitando:

“https://developers.google.com/chart/interactive/docs/gallery/linechart”.

El proceso básico con los datos ya prefijados del ejemplo, que podemos ver en la página web de la herramienta, se llega a complicar bastante al tener que automatizarlo para abarcar todas las posibilidades de visualización dispuestas en el presente trabajo.

En la figura 25 se puede ver un ejemplo de visualización gráfica.

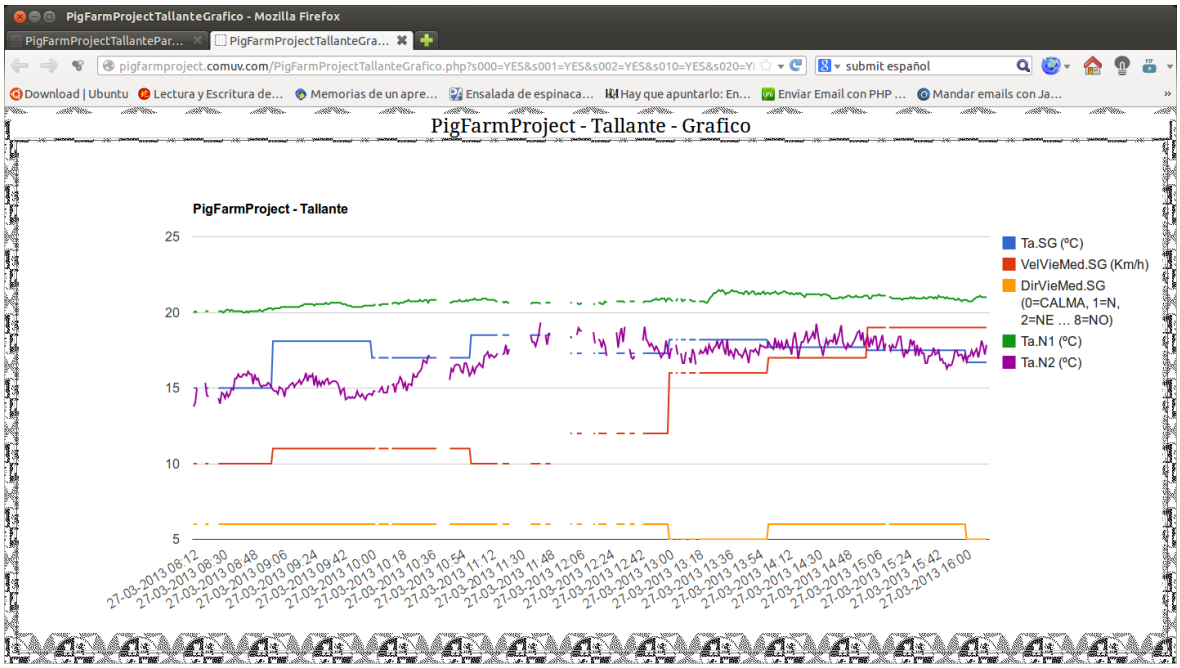


Fig. 25

Y aquí el código (codigo 5) resumido de la misma:

```
<html>
      <head>
            <meta content="text/html; charset=ISO-8859-1" http-
equiv="content-type">
            <title>PigFarmProjectTallanteGrafico</title>
            <?php include "conexion.php";


                  if ($_GET['s000']) {$s000=true;} else $s000=false;
                  if ($_GET['s001']) {$s001=true;} else $s001=false;
                  //.............................................
                  if ($_GET['s049']) {$s049=true;} else $s049=false;
```

```
                $fechahorainicial= new DateTime($_GET['fhi']);
                $fechahorafinal= new DateTime($_GET['fhf']);
                $intervalo= $_GET['intervalo'];

                $sql="SELECT * FROM `TllTexto` WHERE `que` =
'NombresCortos'"; $result=mysql_query($sql, $conexion) or
die(mysql_error());
                $NombresCortos=@mysql_fetch_array($result);

                $titulocolumnas="Fecha y Hora";


                if ($s000) {$titulocolumnas.="-".
$NombresCortos['s000'];}
                if ($s001) {$titulocolumnas.="-".
$NombresCortos['s001'];}
                //.............................................
                if ($s049) {$titulocolumnas.="-".
$NombresCortos['s049'];}



                $cadena = "";
                $fechahora=new DateTime();
                for ($fechahora = $fechahorainicial;
$fechahora<=$fechahorafinal; $fechahora->modify($intervalo))
                {
                        $table="`Tll".$fechahora->format('Ym')."`";
                        $diahoraminuto=$fechahora->format('dHi');
                        $sql="select * from $table WHERE diahoraminuto =
$diahoraminuto";
                        $result=mysql_query($sql, $conexion) or
die(mysql_error());
                        if ($row=@mysql_fetch_array($result))
                        {
                                $cadena.= $fechahora->format('d-m-Y H:i');


                                if ($s000) {if($row['s000']!="")
{$cadena.="+".$row['s000']."";} else {$cadena.="+"."null"."";}}
                                if ($s001) {if($row['s001']!="")
{$cadena.="+".$row['s001']."";} else {$cadena.="+"."null"."";}}
                                //.....................................
                                if ($s049) {if($row['s049']!="")
{$cadena.="+".$row['s049']."";} else {$cadena.="+"."null"."";}}


                                $cadena.="*";
                        }
                        else
                        {
                                $cadena.= $fechahora->format('d-m-Y H:i');
```

```
                              if ($s000) {$cadena.="+"."null".";}
                              if ($s001) {$cadena.="+"."null".";}
                              //.....................................
                              if ($s049) {$cadena.="+"."null".";}

                              $cadena.="*";
                        }
                  }
                  include "desconectar.php";
            ?>
            <script type="text/javascript"
src="https://www.google.com/jsapi"></script>
            <script type="text/javascript">
                  var titulocolumnas= "<?php echo $titulocolumnas; ?>";
                  var cadena = "<?php echo $cadena; ?>";
                  google.load("visualization", "1", {packages:
["corechart"]});
                  google.setOnLoadCallback(drawChart);

                  function drawChart()
                  {
                        var data = new google.visualization.DataTable();
                        var titulo = titulocolumnas.split("-");
                        data.addColumn('string', titulo[0]);
                        for (var i=1;i<titulo.length;i++)
                        {data.addColumn('number', titulo[i]);}
                        var registro = cadena.split("*");
                        for (var j=0;j<registro.length;j++)
                        {
                              var valor = registro[j].split("+");
                              for (var k=1;k<titulo.length;k++)
                              {valor[k] = parseFloat(valor[k]);}
                              data.addRow(valor);
                        }
                        var options = {title: 'PigFarmProject -
Tallante'}
                        var chart = new
google.visualization.LineChart(document.getElementById('chart_div'));
                        chart.draw(data, options);
                  }
            </script>
            <style type="text/css">
                  body, p, li {background-position: center top;
background-image: url(Imagenes/upct0.png);}

            </style>
      </head>
      <body>
            <div style="text-align: center; background-color: rgb(255,
255, 255);"><font size="+2">PigFarmProject - Tallante -
Grafico</font></div>
            <table style="text-align: left; width: 100px; margin-left:
auto; margin-right: auto;" border="1" cellpadding="2" cellspacing="2">
                  <tbody>
                        <tr>
```

```
                                <td colspan="2" rowspan="1"
style="vertical-align: middle; text-align: center;">
                                    <div id="chart_div" style="width:
1400px; height: 600px;"></div><br>
                                </td>
                            </tr>
                        </tbody>
                    </table>
            </body>
</html>
```

Cód. 5

4.6.4. PigFarmProjectTallanteServiciosGenerales.php

Si hacemos “click” en cualquier área no delimitada específicamente en el mapeo de la portada (default) de nuestro proyecto para otro fin especificado llegaremos a la página objeto de este apartado. A modo de ejemplo, en la captura de pantalla de la figura 26 podemos ver la estructura elegida. Sobria, quizá en exceso, debido a que se priorizó la claridad y sencillez sobre un diseño más atractivo.

Debajo de los dos logos de enlace a la portada y a la página (en la figura 26 han quedado ocultos por la parte superior) donde poder seleccionar los gráficos históricos a visualizar, tenemos la fila de la tabla que nos indica el nombre de la instalación de la granja que estamos viendo, a la derecha y en la misma linea podemos ver el comentario que concernirá a toda la instalación denominada, en este caso,“Servicios Generales”.

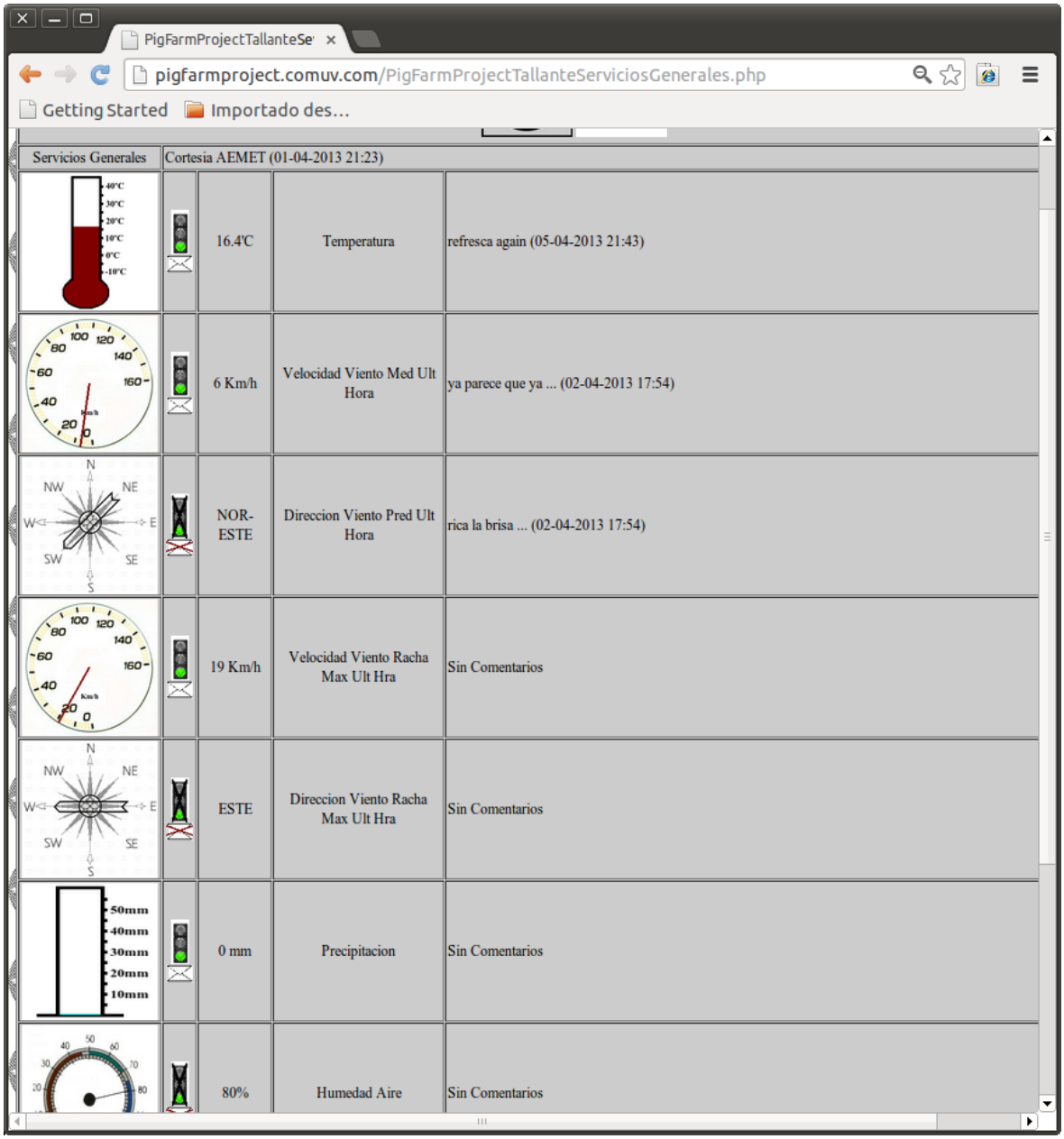


Fig. 26

Cada una de las siguientes filas se refiere únicamente a un parámetro físico de los observables en el sistema. Cada fila se divide a su vez en varias celdas que, de izquierda a derecha, pasamos a describir:

– La primera nos muestra una imagen dinámica. Ésto es, la imagen, además de ser un icono sencillo pero que identifica claramente el tipo de parámetro al que corresponde, nos representa en forma gráfica los valores observados (altura del nivel del mercurio en un termómetro graduado, aguja indicadora en un velocímetro, veleta marcando la dirección del viento, nivel de los depósitos de agua, etc). Podemos ver unas muestras en la figura 27 donde cada figura muestra gráficamente los valores que correspondan. En este caso vemos

como las figuras representan un valor del 75% para los casos de apertura de ventanas, el nivel de agua en el depósito y el nivel de pienso en silo; el termómetro marca una temperatura de 10ºC, y el velocímetro marca 25 Km/h de velocidad de viento.

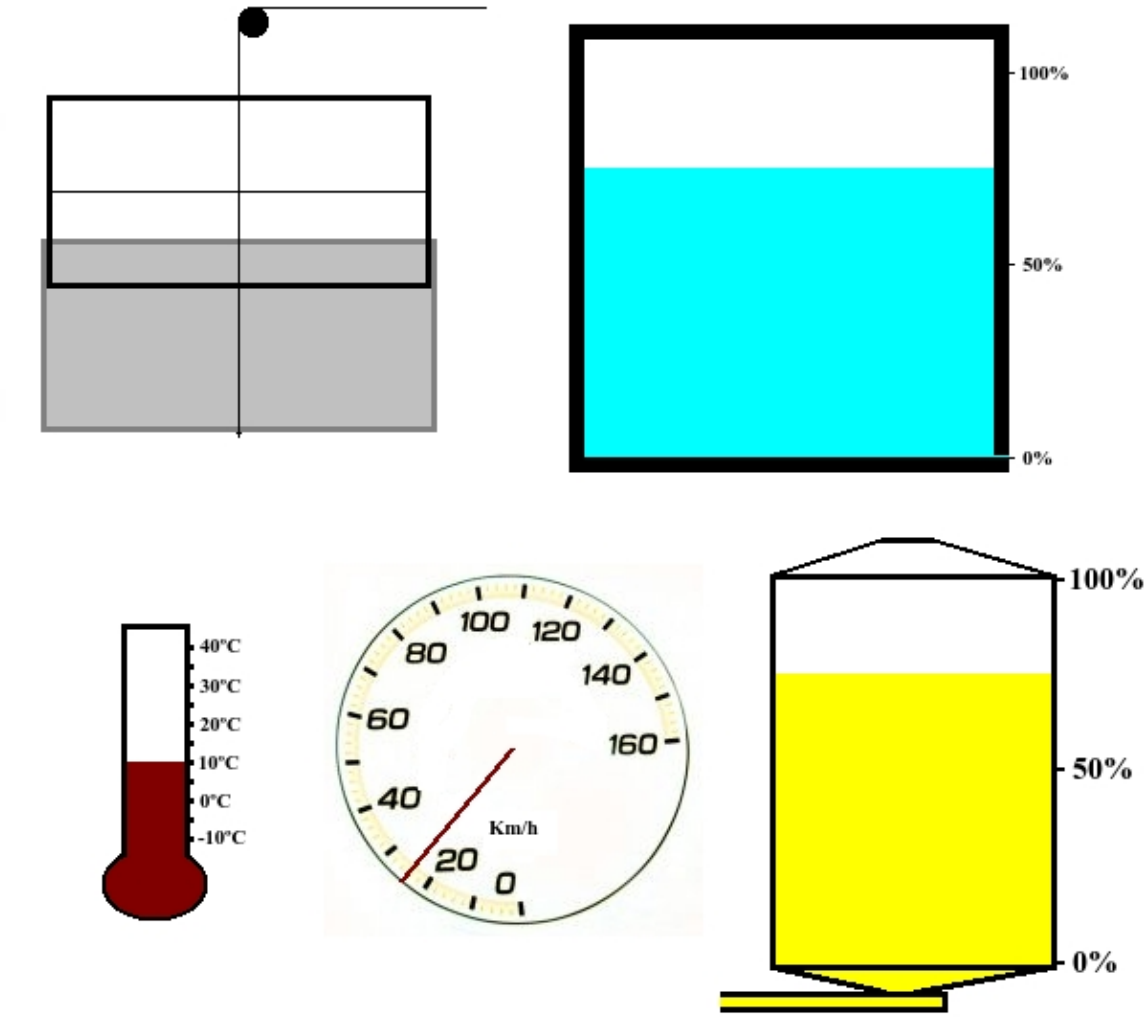


Fig. 27

Usamos algunos de los comandos disponibles en la librería GD de php (http://php.net/manual/es/ref.image.php). Dicha librería nos permite dibujar solo unas pocos tipos de figuras simples (puntos, líneas, rectángulos, etc) pero resultó suficiente para nuestras necesidades. El procedimiento consiste en: primero, mostrar la imagen base del parámetro (termómetro vacío, esfera del velocímetro sin aguja, depósito de agua vacío, etc.) y, segundo y último, dibujar sobre ella el indicador correspondiente compuesto por figuras simples; en nuestro caso, rectángulos y líneas). Para los indicadores de nivel (por ejemplo los depósitos de agua), primeramente mostramos la imagen base de la instalación y después dibujamos sobre ella un rectángulo ubicado en la posición adecuada, relleno del color correspondiente y de una altura que será función del valor del parámetro que

se muestra. Globalmente, lo que veremos será la representación esquemática de un depósito con un nivel de agua determinado. Caso especial es el indicador de dirección del viento, donde la rutina php lo que hace realmente es hacer visible una figura u otra. Lo cual es posible ya que este tipo de parámetro puede presentar, en nuestro sistema, un número finito y muy reducido de valores (enteros del 0 al 8, ambos incluidos). Ver figura 26 (imágenes 3 y 5 contadas de arriba a abajo – vemos una flecha indicando la dirección sobre una rosa de los vientos).

– La segunda nos muestra la información de las alertas (visuales en web y de envío de correos electrónicos a la dirección de el/los responsable/s prefijados) según hayamos dispuesto en la configuración. El la figura 28 podemos ver cada una de las imagenes con que pueden ser mostradas. Cada una de ellas facilita la siguiente información (de izquierda a derecha):

· Semáforo rojo: Sistema de alertas visuales en web habilitado para este parámetro y valor en rango calificado como "zona de alerta" por el usuario.

· Semáforo amarillo: Sistema de alertas visuales en web habilitado para este parámetro y valor en rango calificado de riesgo medio.

· Semáforo verde: Sistema de alertas visuales en web habilitado para este parámetro y valor en rango calificado como adecuado.

· Semáforo verde tachado: Sistema de alertas visuales en web NO habilitado para este parámetro.

· Sobre tipo correspondencia blanco: Sistema de alertas al correo electrónico habilitado para este parámetro y, además, ninguna alerta e-mail ha sido enviada después de la última reinicialización particular del mismo.

· Sobre rojo: Sistema de alertas e-mail activado y, además, ha sido enviada una de estas alertas al correo electrónico el cual aún no ha sido reinicializado (se encontrará no operativo mientras tanto).

· Sobre blanco tachado: Sistema de alertas e-mail desactivado para este parámetro y no hay ningún aviso pendiente de reinicializar.

· Sobre rojo tachado: Sistema de alertas e-mail desactivado para este parámetro pero, en este caso, tenemos aún una alerta e-mail de un periodo anterior a la desactivación sin reinicializar por parte del usuario.

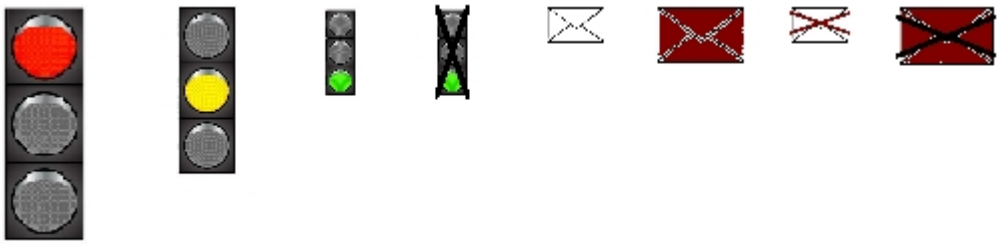


Fig. 28

– La tercera nos muestra el último valor captado del parámetro en su unidad; ºC, %, Km/h, etc.

– La cuarta parte es el lugar destinado a mostrar los comentarios específicos del parámetro observado.

4.6.5. PigFarmProjectTallanteNave1.php

Si hacemos "click" en el área de la página web portada del proyecto que delimita la instalación denominada "Nave 1" llegaremos a esta página. Es interesante describirla pese a su fácil comprensión ya que es una instalación típica, lo que nos indica que, aproximadamente, un 90% de las instalaciones de este tipo tendrán esta misma configuración.

La primera fila de la tabla después de dos logos enlazados a otras páginas del proyecto es la que corresponde a la información general de la instalación; ésto es, nombre de la misma y un comentario, si lo hubiera, concerniente a toda la instalación. Por ejemplo: "Nave en periodo de limpieza y desinfección" para indicarnos que valores fuera de rangos lógicos no deberían ser preocupantes dado que no hay ganado dentro de la instalación.

Cada una de las siguientes filas se puede describir exactamente como las del apartado anterior salvando, por ejemplo, que en las páginas de las naves típicas no creemos necesario tener información de la dirección del viento en el interior de la misma, la humedad del aire, precipitación, etc. y sí que nos interesa saber la apertura de las ventanas y la cantidad de pienso en el silo.

En la figura 29 vemos una captura de pantalla para mejor comprensión.

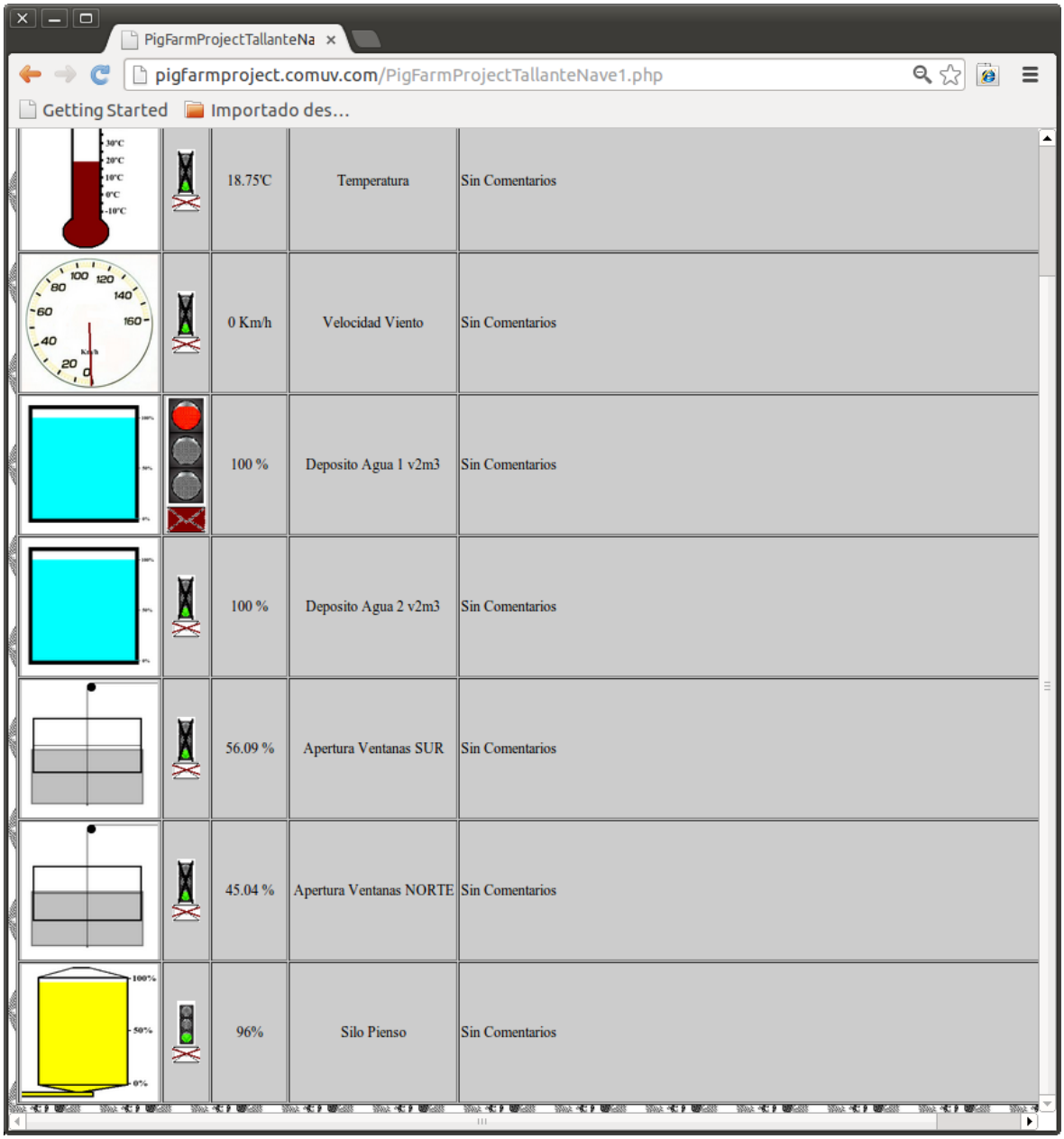


Fig. 29

Obviaremos la presentación y descripción específica del resto de naves ya que, como hemos mencionado anteriormente, es una instalación típica que puede describir la mayor parte de las instalaciones de este tipo.

4.3.6. PigFarmProjectTallanteParametrosAmbientales.php,
PigFarmProjectTallanteSilosPienso.php Y
PigFarmProjectTallanteDepositosAgua.php

Como podemos ver en la "portada", además del acceso a cada una de las instalaciones de la granja por separado, hay tres imágenes ordenadas verticalmente a la izquierda de la

misma. Nos pareció muy conveniente, por ejemplo, poder conocer de un solo vistazo el estado de todos los silos para prever las necesidades del siguiente pedido, especialmente interesante cuando se aproximan fines de semana, fiestas, paradas por avería o mantenimiento de la fábrica de piensos, etc. Un razonamiento parecido justifica la inclusión de una página que nos muestra el estado de todos los niveles de agua de bebida existentes en la granja y una página más que incluirá la totalidad de los parámetros ambientales.

Como ejemplo vemos en la figura 30 la captura de la página que presenta la información de los silos de las cuatro naves.

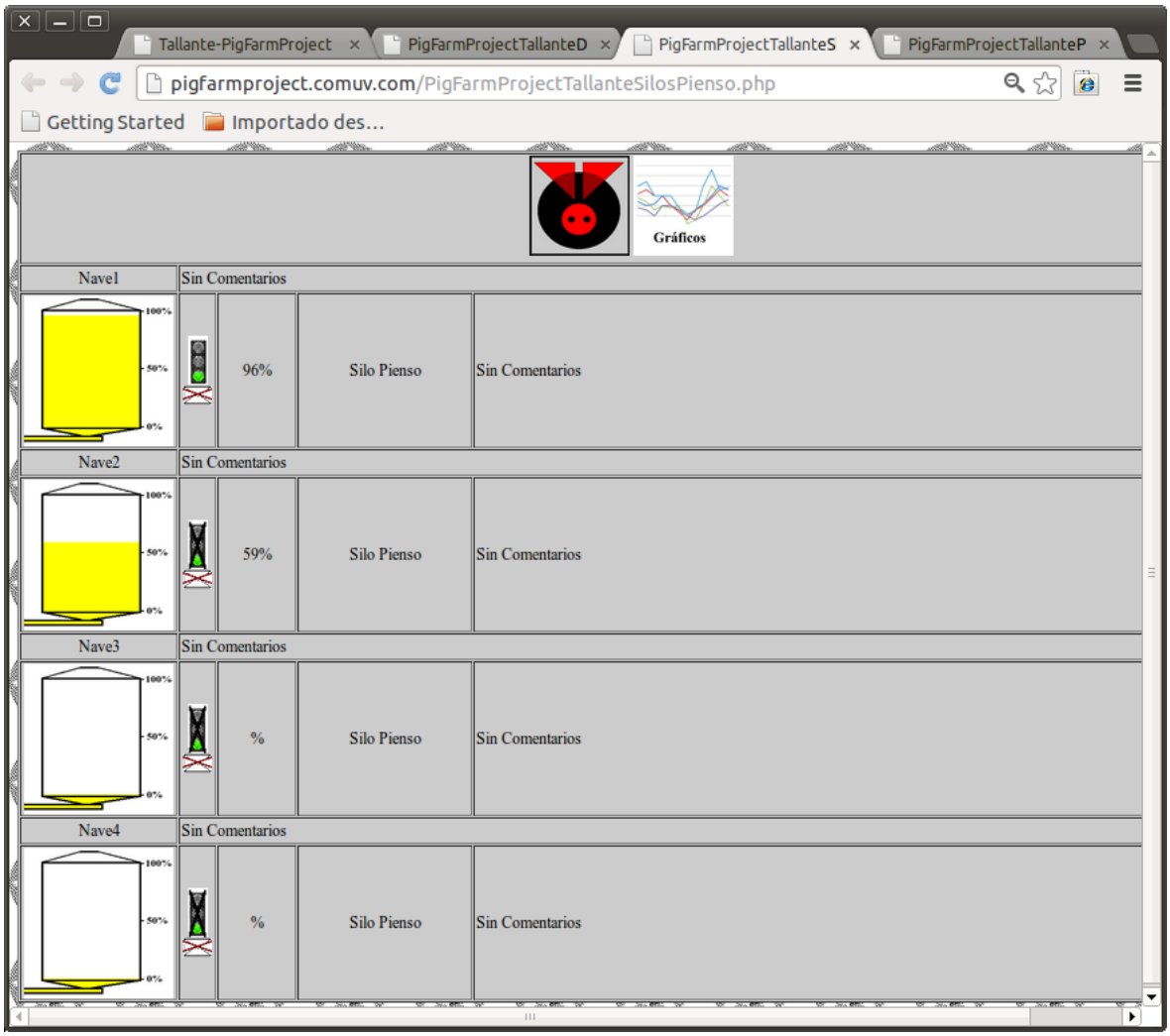


Fig. 30

5. NORMAS Y REFERENCIAS

5.1. DISPOSICIONES LEGALES Y APLICADAS

- Ley de prevención de riesgos laborables.
- Reglamento electrotécnico para baja tensión.
- Norma ISA de instrumentación.
- Reglamento de instalaciones molestas, insalubres, nocivas y peligrosas.

5.2. BIBLIOGRAFÍA

- Real Decreto 842/2002 . Reglamento Electrotécnico para Baja tensión.
- Normas ISA para Instrumentación.
- Seguridad laboral en explotaciones ganaderas . Alfredo Quesada Valero - Enrique Castaño Molina - Antonio Vicente Abellán - María José Abellán Millá - Emilio José Sánchez Infer . Departamento de Prevención de Riesgos Laborales de la Federación de Cooperativas Agrarias de Murcia.
- Reglamento de instalaciones molestas, insalubres, nocivas y peligrosas. Aprenda Java como si estuviera en primero. Campus Tecnológico de la Universidad de Navarra. Javier García de Jalón • José Ignacio Rodríguez • Iñigo Mingo • Aitor Imaz • Alfonso Brazález • Alberto Larzabal • Jesús Calleja • Jon García.
- Guía de Usuario de Arduino. Rafael Henríquez Herrador. 13 de noviembre de 2.009.
- Mini Manual de Uso de FileZilla. Enero 2.008. Servicio de Informática y Comunicaciones. Universidad de Huelva.
- Tutorial para la Configuración de Xbee. Álvaro Neira Ayuso. Concurso Universitario de Software Libre.
- XBee OEM RF Modules- ZigBee - v1.x1x [2007.06.01]. 2007 Digi International, Inc.
- Manual de JAVASCRIPT . Jose Antonio Rodríguez.
- Programación en PHP5. Nivel Básico . Carlos Vázquez Mariño . Ferrol, Septiembre de 2008.
- LM35 - Precision Centigrade Temperature Sensors (DataSheet).
- PING)))TM Ultrasonic Distance Sensor (DataSheet).

5.3. PROGRAMAS DE CÁLCULO, DISEÑO Y DESARROLLO DE SOFTWARE

Este proyecto ha sido desarrollado y documentado íntegramente usando software gratuíto y de libre distribución. Y son:

- KompoZer 0.7.10 (www.kompozer.net/download.php)
- Mozilla Firefox 21.0 (Incluido distribución Ubuntu 12.04 LTS)

- Window and Tab Limiter (Complemento para Firefox)
- KolourPaint 4.8.4 (www.kolourpaint.org)
- NetBeans IDE 7.2.1 (netbeans.org/downloads/)
- Gedit (Incluido distribución Ubuntu 12.04 LTS)
- Arduino 1.0.3 (arduino.cc/en/Main/Software)
- OpenOffice (Incluido distribución Ubuntu 12.04 LTS)
- FileZilla 3.5.3 (filezilla-project.org/)
- Wine 1.0.1 (www.winehq.org/download/)
- XCTU 5.2.7.5 (www.digi.com/support/productdetail?pid=3352)

La distribución de linux "Ubuntu 12.04 LTS" es el sistema operativo usado para la realización íntegra del proyecto (diseño, calculo, desarrollo y funcionamiento) . Dicha distribución y otras se encuentran disponibles para su libre descarga en http://www.ubuntu.com/download.

5.4. OTRAS REFERENCIAS

- arduino.cc
- developers.google.com/chart/interactive/docs/gallery/linechart
- Podríamos incluir un larguísimo etcétera de blogs, tutoriales, video-tutoriales y foros de internet, gracias a los cuales ha sido posible la realización de este multidisciplinar proyecto desde cero.

6. PRUEBAS Y AJUSTES DE LOS EQUIPOS INSTALADOS

· Comprobaciones previas al montaje de cada módulo con microcontrolador:

- Pruebas de software de microcontrolador en modo de simulación.
- Comprobación de sensores con software de simulación en microcontrolador.
- Comprobación independiente de cables, pines y conectores en banco de pruebas.
- Comprobación en mínima distancia (eliminando alargaderas de señal) de funcionamiento de todo el módulo en modo de simulación.
- Montaje y prueba de cada sistema a corta distancia.

· Comprobación de comunicaciones entre los diferentes módulos y el ordenador-granja ya instalados en sus ubicaciones definitivas.

· Comprobaciones y ajustes tras la instalación del equipamiento:

- Se instalan las mini-poleas necesarias (apertura de ventanas y niveles de agua) en los ejes de los correspondientes potenciómetros e instalaremos nuestro sistema de cuerda con un extremo sujeto a un elemento indicador y el otro a un objeto para contrapeso, de forma tal que, esta cuerda, en su rango posible de recorrido (0 y 100% en estos casos) no intente forzar al potenciómetro a sobrepasar el tope (máximo y mínimo) de dicho potenciómetro. Este caso provocará el desajuste del sensor lo que llevará a lecturas erróneas.
- Ahora tendremos que adaptar el valor de las lecturas recopiladas por el microcontrolador de parte de nuestro sistema de sensores a información entendible por el usuario. Debido al carácter lineal, tanto de nuestros potenciómetros multivuelta como de la naturaleza de la medida (ventanas de forma rectángular y depósitos de sección constante, todos cilíndricos en posición vertical en nuestro caso particular), podremos solucionar la adaptación de los valores entregados por los sensores a valores de medida real en las unidades deseadas integrando una simple función (recta bidimensional) en el código del microcontrolador donde podremos ajustar los parámetros deseados. Podemos ver el código 6 que no es más que una parte del código completo del programa del módulo Arduino C1 (código 1).

```
...

/*[Nivel DepAgua 1]*/ float EntradaAnalogicaA1 = 0.0;
/*[Nivel DepAgua 1]*/ float ResistenciaA1 = 0.0;
/*[Nivel DepAgua 1]*/ float X1A1 = 1023.0;
/*[Nivel DepAgua 1]*/ float Y1A1 = 0.0;
/*[Nivel DepAgua 1]*/ float X2A1 = 0.0;
/*[Nivel DepAgua 1]*/ float Y2A1 = 100.0;

...
```

```
/*[Nivel DepAgua 1]*/   EntradaAnalogicaA1 = analogRead(analogPin1);
/*[Nivel DepAgua 1]*/   ResistenciaA1 = ((EntradaAnalogicaA1-
X1A1)*(Y2A1-Y1A1)/(X2A1-X1A1))+Y1A1;

...
```

Código 6

A modo de ejemplo hemos seleccionado la parte de código correspondiente al al sensor del nivel de agua en el Depósito nº 1 de cualquiera de las naves. La variable que recoge el valor de la entrada analógica A1 de nuestro microcontrolador (EntradaAnalogicaA1 en nuestro caso) puede adoptar desde el valor 0 (correspondiente a 0V-GND) al valor 1023 (tensión de alimentación-5Vcc), los cuales corresponden, en el código mostrado, a 0 y 100 (variables con inicial "Y"), respectivamente. El ajuste final corresponde, simplemente, a la sustitución de dos pares de valores {(X1A1,Y1A1), (X2A1,Y2A1)} por los dos pares de valores finales correctos. Así la variable a la cual hemos llamado "ResistenciaAnalógicaA1", tomará el valor final adecuado para su registro y muestras.

– Actuaremos de forma análoga, prácticamente, con todas las señales que recibiremos del resto de tipos de sensores, LM35 de temperatura, anemómetro/tacómetro, medidor de distancias por ultrasonidos, así como, cualquier otro tipo de sensor que hubieramos instalado.

7. CONCLUSIONES

Hemos diseñado, desarrollado y puesto en funcionamiento un sistema que permite, entre otros objetivos, la monitorización remota y universal de una serie de parámetros medibles.

Un resumen de todos los servicios ofrecidos en el sistema PigFarmProject podría ser:

- Monitorización remota y en tiempo real de los parámetros sensorizados.
- Registro de datos y sistema de representación gráfica histórica de los mismos.
- Sistema de alertas configurables por el usuario.
- Inclusión de comentarios por el usuario, modificables localmente y visibles de forma remota.

El objeto de este proyecto se centró en las características y necesidades de una granja ganadera en particular, pero queremos volver a hacer constar que el sistema sería fácilmente adaptable a otras necesidades, ya sean similares a la descrita o de una naturaleza muy distinta, tanto de ámbito doméstico o industrial. Así mismo, tendremos que tener muy en cuenta si nuestro proceso admite la fiabilidad y disponibilidad no comprometida de los servicios ajenos contratados; en nuestro caso hablamos de la conexión a internet de banda ancha en ubicación geográfica con media/baja cobertura, el alojamiento gratuito para los procedimientos remotos y base de datos, etc.

Así mismo, en el propio ámbito de una granja ganadera se podrían haber instrumentalizado muchos más parámetros, los cuales pueden ser de muy interesante incorporación. Es el caso de, por ejemplo, detectores de presencia o presión de agua en algunas líneas, consumos generales de agua y electricidad, etc. En conclusión, sería difícil poner un límite ajeno a razones puramente económicas.

Para futuros desarrollos y versiones de PigFarmProject, sería interesante abordar un sistema de protección contra cortes en el suministro eléctrico (uso de un SAI o UPS como lo llaman los americanos) y de re-arranque automático tras un corte más prolongado por encima de las posibilidades del sistema anterior (muchos ordenadores tienen esta opción en la configuración de la propia Bios del equipo).

Posiblemente, algunos de los elementos instalados no sean las mejores soluciones pensando en un medio o largo plazo de funcionamiento (no debemos olvidar, entre otras cosas, la atmósfera húmeda y ciertamente corrosiva de los interiores de las naves, el ambiente polvoriento común en este tipo de instalaciones, etc.); éstos problemas habría que solventarlos, en caso de que se presenten, según vayan apareciendo con el tiempo de uso. En este caso, y si así lo creyéramos oportuno, podremos optar por una nueva solución más adecuada (generalmente más costosa) y simplemente adaptar la parte de código correspondiente en el programa del microcontrolador.

8. CONTENIDO DEL CD

El CD, el cual hemos etiquetado como "PigFarmProject v1.0", va a contener una copia de cada uno de los siguientes documentos y/o carpetas:

- Memoria:
  · PigFarmProject.pdf
- Presentación:
  · PigFarmProject.pps
- Programas para microcontroladores:
  · Arduino_C1
  · Arduino_C2
- Proyectos NetBeans (en carpeta principal "Java"):
  · SerialTalk0
  · DescargarFicherosDatosAEMET
  · LeerFicherosDatosAEMET
  · Comments
  · Warnings
  · EMail
- Software en alojamiento remoto (en carpeta principal "www"):
  · AemetInsert.php
  · agua.php
  · AlarmInsert.php
  · class.phpmailer.php
  · class.smtp.php
  · comentariosHistorial.xml
  · CommentInsert.php
  · conexion.php (modificado para ocultar datos privados)
  · DataInsert.php
  · datetimepicker_css.js
  · DatosAemet.xml
  · default.php
  · desconectar.php
  · emailsender.php ((modificado para ocultar datos privados))
  · EnvEMailInsert.php
  · envEMail.xml
  · green.php
  · humedad.php
  · Imagenes (subcarpeta con imagenes que aparecen en web)
  · images2 (subcarpeta con imagenes que aparecen en web – exclusivas de la utilidad "Date & Time Picker")
  · lluvia.php
  · nolights.php
  · nombreslargos.xml
  · PigFarmProjectTallanteDepositosAgua.php

· PigFarmProjectTallanteGrafico.php
· PigFarmProjectTallanteNave1.php
· PigFarmProjectTallanteNave2.php
· PigFarmProjectTallanteNave3.php
· PigFarmProjectTallanteNave4.php
· PigFarmProjectTallanteParametrosAmbientales.php
· PigFarmProjectTallanteParamGrafico.php
· PigFarmProjectTallanteServiciosGenerales.php
· PigFarmProjectTallanteSilosPienso.php
· red.php
· SaveEnvEMail.php
· semaforo.php
· silo.php
· temperatura.php
· veleta.php
· velocidadviento.php
· ventanas.php
· yellow.php

– Hojas de datos en pdf de componentes usados (en carpeta principal “Anexos”):
  · Propuesta de Proyecto Fin de Carrera Específico
  · LM35 - Precision Centigrade Temperature Sensors
  · PING)))TM Ultrasonic Distance Sensor (DataSheet)

– Reglamentos oficiales aplicados y/o consultados (en carpeta Legislación):
  · Real Decreto 842/2002 . Reglamento Electrotécnico para Baja tensión.
  · Reglamento de instalaciones molestas, insalubres, nocivas y peligrosas.
  · Ley de Prevención de Riesgos Laborales y Desarrollo Normativo. FREMAP.

– Otras referencias consultadas.
  · Seguridad laboral en explotaciones ganaderas . Alfredo Quesada Valero - Enrique Castaño Molina - Antonio Vicente Abellán - María José Abellán Millá - Emilio José Sánchez Infer . Departamento de Prevención de Riesgos Laborales de la Federación de Cooperativas Agrarias de Murcia.