

Sistema de Control de un Invernadero

1. Parte I: Funcionalidad básica (2 puntos, incluyendo DTE)

Se trata de diseñar una aplicación C-S para implementar el control de un invernadero que posee varias placas de sensores distribuidos a lo largo del mismo. Una placa podrá disponer de varios sensores y cada sensor controlará una variable. En la descripción que sigue a continuación se ejemplifican algunas variables susceptibles de ser controladas de forma remota por cada placa. Además, sobre cada variable se podrán realizar varias acciones. Para la implementación del proyecto cada grupo habrá de contemplar varias variables de control, las cuales pueden monitorizarse y controlarse para cada una de las placas de sensores. Por cada placa se almacenarán los siguientes datos:

- Identificativo de la placa (ej: "Placa1")
 - Por cada variable:
 - Variable (ej: "Temperatura")
 - Función principal (ej: "Regulación climatización")
 - Estado_Variable (ON, OFF)
 - Última acción (ej: "Encender a.c / Encender calefacción")
- Estado_Placa (ON, OFF)
- Foto (ej: imagen de la placa)

Otros ejemplos de variables bajo control pueden ser: humedad, luminosidad, humo, velocidad del viento, etc.

Todos los datos estarán almacenados en el Servidor y las acciones a efectuar sobre ellos proceden de una aplicación Cliente remota. El servidor ha de permitir la conexión de varios clientes simultáneamente.

Se debe implementar tanto el Servidor, que debe recibir los comandos del cliente, debe procesarlos y devolverle los resultados, como el Cliente que debe enviar los comandos al Servidor según lo que pida el usuario, desde su interfaz gráfica.

La aplicación debe funcionar tanto ejecutando el Cliente y el Servidor en la misma máquina, como en máquinas distintas, por lo que al arrancar la ventana de registro de un cliente (donde se pide *username* y *password*), se ha de solicitar la **dirección IP del servidor** de la aplicación al que se desea conectar.

A continuación se describen los comandos que se deben implementar: el nombre del comando aparece en mayúsculas (y se debe respetar) y los parámetros aparecen en minúsculas (y podrán ser modificados según se considere oportuno).

En las tablas se proponen ejemplos de cada comando, en la primera columna se especifica el comando que envía el cliente al servidor y en la segunda se especifican las posibles respuestas ante cada comando. Se debe implementar como mínimo todos los posibles casos expuestos.

- **USER nombre.** Este comando es el primero que debe enviar el cliente al servidor, ya que sirve para validar si el usuario “nombre” está registrado en el servidor. Todo usuario que quiera acceder a la información debe introducir su nombre. El servidor debe validar si el nombre introducido existe, así como si no se ha introducido nombre alguno.

Por ejemplo:

| | |
|---------------|-------------------------------------|
| USER Markel | 200 OK Bienvenido Markel. |
| USER | 400 ERR Falta el nombre de usuario. |
| USER Fernando | 401 ERR Usuario desconocido. |

- **PASS clave.** Este comando debe enviarlo el cliente al servidor después de enviar el comando anterior, ya que sirve para validar si el usuario tiene privilegios para utilizar esta aplicación.

La clave se valida para un nombre de usuario específico (enviado en el comando anterior), luego el servidor habrá de guardar el usuario, hasta validarlo con su clave).

Por ejemplo:

| | |
|------------|---------------------------------|
| PASS 94475 | 201 OK Bienvenido al sistema. |
| PASS mqm | 401 ERR La clave es incorrecta. |
| PASS | 402 ERR Falta la clave. |

- **LISTADO.** Este comando devuelve una línea **ELEM** por cada objeto almacenado en la BBDD, con todos sus datos separados por espacios (si un campo puede contener espacios, será necesario definir otro carácter separador) y **FINLISTA** cuando no hay más.

Por ejemplo:

| | |
|---------|---|
| LISTADO | ELEM 01 Placa1; Temperatura; Regulación climatización; ON; subir a.c. ELEM 02 Placa1; Humedad; Sistema de riego; ON; activar sistema de riego. ELEM 03 Placa2; Temperatura; Regulación climatización; OFF;bajar a.c. --Línea en blanco-- 202 FINLISTA |
|---------|---|

- **BUSCAR opción patrón.** Este comando devuelve una línea **ELEM** por cada objeto almacenado cuyo campo “opción” cumpla el patrón y **FINLISTA** cuando no hay más.

El patrón utiliza dos caracteres especiales: *, para indicar cualquier combinación de caracteres y ?, para indicar cualquier carácter.

El * puede estar al principio y al final de la palabra y el ? en cualquier posición de la palabra. No debe ser sensible a mayúsculas y minúsculas.

Por ejemplo:

- an*: indica cualquier palabra que empiece por “an”.
- an?: indica cualquier palabra que empiece por “an” y esté formada por tres letras.

| | |
|-------------------------|--|
| BUSCAR variable Tem* | ELEM 01 Placa1; Temperatura; Regulación climatización; ON; subir a.c. ELEM 03 Placa2; Temperatura; Regulación climatización; OFF;bajar a.c. --Línea en blanco-- 202 FINLISTA |
| BUSCAR estado O? | ELEM 01 Placa1; Temperatura; Regulación climatización; ON; subir a.c. ELEM 02 Placa1; Humedad; Sistema de riego; ON; activar sistema de riego. --Línea en blanco-- 202 FINLISTA |

- **ON id_placa id_variable.** Este comando posibilita obtener el control de la variable cuando su estado previo era OFF. Para que tenga efecto es imprescindible comprobar que el estado de la variable es OFF.

Por ejemplo:

| | |
|-----------------------|---------------------------------------|
| ON placa1 temperatura | 203 OK Control de temperatura activo. |
| ON placa1 temp | 403 ERROR id_variable no existe. |
| ON placa1 temperatura | 404 ERROR id_variable en estado ON. |

- **OFF id_placa id_variable.** Este comando libera el control de la variable cuando se encuentra en el estado ON. Para que tenga efecto es imprescindible comprobar que el estado previo es ON.

Por ejemplo:

| | |
|------------------------|--|
| OFF placa1 temperatura | 204 OK Control de temperatura desactivado. |
| OFF placa1 temp | 405 ERROR id_variable no existe. |
| OFF placa1 temperatura | 406 ERROR id_variable en estado OFF. |

- **ACCION id_placa id_variable, acción.** Este comando permite realizar una determinada acción sobre una variable. Por ejemplo, en el caso de la regulación de la temperatura, será accionar el aire acondicionado o bien la calefacción. Por tanto, sobre cada variable se podrán realizar varias acciones.

Se ha de tener en cuenta que para implementar esta acción los parámetros a recibir serán diferentes en función del tipo de variable que estemos controlando.

Para poder ejecutar la acción es necesario comprobar que:

- 1º- el estado de la placa (Estado_Placa) esté ON.
- 2º- el estado de la variables (Estado_Variable) esté ON.

Por ejemplo:

| | |
|--|--------------------------------------|
| ACCION Placa1 temperatura encender a.c | 205 OK Esperando confirmación. |
| ACCION Placa1 temper encender calf | 407 ERROR id_variable no existe. |
| ACCION Placa1 temperatura encender a.c | 408 ERROR id_variable en estado OFF. |

- **CONFIRMAR_ACCION parámetros.** Este comando confirma la acción a realizar sobre la variable y envía como parámetros (a través de un cuadro de diálogo) los datos necesarios para ejecutar la acción específica anteriormente de esa variable. Es posible que no sean necesarios parámetros. Se ha de actualizar en la BBDD la última acción efectuada sobre la variable. De esta forma quedará reflejada la última acción efectuada.

Por ejemplo:

| | |
|--------------------|--|
| CONFIRMAR_ACCION 5 | 206 OK Acción sobre el sensor confirmada <ul style="list-style-type: none"> • <i>Explicación ejemplo: encender a.c. duran 5 minutos si antes se había indicado ACCION id_placa temperatura encender a.c</i> |
| CONFIRMAR_ACCION | 409 ERR Faltan datos. |

- **RECHAZAR_ACCION.** Este comando cancela la acción solicitada, a través del cuadro de dialogo (se hayan introducido los datos con los parámetro del comando o no).

Por ejemplo:

| | |
|-----------------|-------------------------|
| RECHAZAR_ACCION | 207 OK Acción cancelada |
|-----------------|-------------------------|

- **OBTENER_FOTO id_placa.** Este comando permite pedirle al servidor que le envíe la foto referente a la placa fotovoltaica seleccionada. También se podrá enviar una foto de los elementos destinados a monitorizar variables, como por ejemplo un termómetro, un anemómetro, etc. El archivo podrá guardarse en el cliente en una carpeta temporal, o bien mostrarse en una ventana, sin que se guarde una copia.

Como respuesta, el servidor le indica el inicio del envío y además le indica el nombre del fichero y su tamaño.

Por ejemplo:

| | |
|---------------------|-----------------------------------|
| OBTENER_FOTO placa1 | 206 OK 30564 bytes transmitiendo. |
| OBTENER_FOTO placa2 | 403 ERR Identificador no existe. |

- **SALIR.** Este comando permite al usuario desconectarse del servidor. El usuario debe poder desconectarse de la aplicación desde su ventana principal (una vez se ha validado).

Por ejemplo:

| | |
|-------|---------------|
| SALIR | 208 OK Adiós. |
|-------|---------------|

2. Parte II: Administrador del Servidor (1 punto)

El servidor debe poseer una interfaz gráfica en la que se pueden llevar a cabo las siguientes acciones por el administrador de manera local:

- Controlar dinámicamente el número de conexiones activas pudiendo establecer un número máximo, por encima del cual una nueva petición del cliente (conexión nueva) sería rechazada.
- Listar los usuarios conectados en cada momento pudiendo ser desconectados por el administrador.
- Gestionar altas, bajas y modificaciones de cualquier usuario (conectado o no conectado) asignando sus permisos.
- Editar las acciones a ejecutar sobre una variable determinada (dependerá de cada enunciado de proyecto). Modificar la última acción, o último valor registrado, resetear el registro sobre esa variable en la BBDD, etc.

3. Ejemplo Posibles Diálogos Proyecto

