

“Desarrollo e implementación de una red de información meteorológica automática en los partidos de San Nicolás y Ramallo”

Ricardo Martín Fernández, Alberto Mayol, Bruno Competelli, Nicolás Murisengo.

Facultad Regional San Nicolás, Universidad Tecnológica Nacional, Argentina.

rmfernandez@frsn.utn.edu.ar, albertomayol@gmail.com,
bcompetelli@hotmail.com, nico_murisengo@hotmail.com.

Resumen. Debido a la poca homogeneidad de las lluvias de los últimos años, es necesario contar con más puntos de muestreo para obtener conclusiones más acertadas.

El proyecto nace con la necesidad de obtener varios puntos de medición de pluviometría en el partido de San Nicolás y alrededores. Actualmente solo se obtienen datos desde una estación para toda la zona.

El proyecto consta de una estación meteorológica de bajo costo a base de Arduino (plataforma de código abierto). La misma recolecta datos de sus sensores, los almacena en su memoria interna y luego los envía al servidor vía conexión GPRS por internet. Tiene la capacidad de recuperar datos no recibidos por el servidor de cualquier nodo y a petición en cualquier momento.

El prototipo consta de sensores de Presión Atmosférica, Temperatura, Humedad y Pluviometría.

Palabras clave: trazabilidad – pluviómetro – Nodos remotos.

1 Introducción

A través del Instituto de Tecnología Agropecuaria INTA delegación San Nicolás, surge la inquietud de desarrollar un sistema a medida para la recolección remota de datos y variables atmosféricas, en particular lluvia caída en distintos puntos del partido de San Nicolás.

El diseño debía tener como premisa la trazabilidad de los datos, la información meteorológica debe ser completa sin pérdida de información, ya que provocaría un posterior análisis incompleto o erróneo.

El desarrollo del software contempla el almacenaje de los datos de forma segura hasta que el servidor confirma la correcta recepción de los mismos.

A su vez, el hardware, debía ser de bajo coste para poder replicar nuevos nodos de información con facilidad y economía.

1.1 Hardware

El desarrollo del hardware fue realizado en base a la placa de desarrollo Arduino Mega 2560. La misma fue pensada como un Shield de la misma placa. Esto es, un módulo que se adosa al Arduino y que trabajan en conjunto.

El hardware consta de:

- Arduino Mega 2560
- Módulo de lectura de memoria SD-CARD
- Sensor de temperatura y humedad de gabinete **DHT22**
- Sensor de temperatura y humedad externa **AM2305** (compatible con **DHT22**)
- Sensor de Presión Atmosférica (Barómetro) **BMP085**
- Módulo GSM **SIM800L**
- Pluviómetro tipo Balancín Genérico.

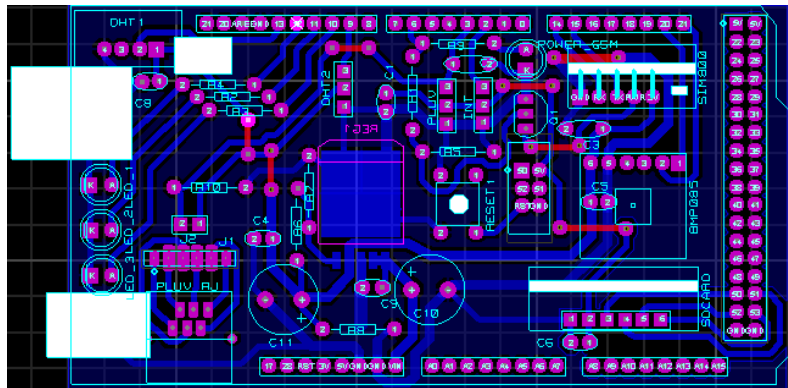


Fig. 1. Arduino Mega placa base microcontrolador.

El controlador (Arduino) recolecta la información suministrada por los sensores a una base de tiempo prefijada, la almacena ordenada en la memoria SD y la envía vía GPRS al servidor (IP fija o DNS Dinámico). Una vez enviados dichos datos, el servidor responde OK si no necesita ningún dato faltante, o responde con la petición de otro dato que la estación tenga almacenada en memoria. Repite este comportamiento hasta que no requiera más datos faltantes, con lo que finaliza la comunicación.

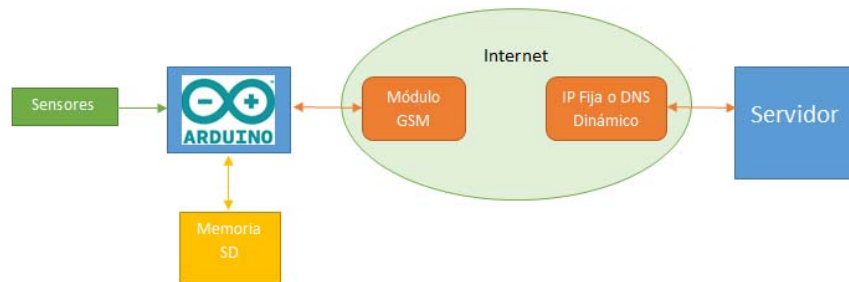


Fig. 2. Diagrama en bloques del sistema propuesto.

2 Software

La comunicación entre el controlador y el servidor es realizada por medio el módulo de comunicación **SIM800L**. Dicho modulo se comunica con el controlador vía puerto USART. La conexión entre el **SIM800L** y el servidor se realiza vía puerto TCP, a la IP fija o DNS Dinámico del servidor. La estación meteorológica actúa de cliente TCP, y el servidor como servidor TCP controlando la comunicación y almacenando todos los datos suministrados por cada nodo/estación. Esto se definió de esta forma debido a que los servicios de telefonía asignan IP dinámica a cada teléfono/nodo, y no hay forma de que el servidor pueda conectarse a las estaciones, debido a que no tienen siempre la misma IP. Por lo tanto, se debe contar con una IP fija en el servidor o servicio de DNS Dinámico, para que todas las estaciones se conecten a ésta.

Cada estación tiene un número de Nodo/Estación. Dicho número es enviado en la trama de datos, junto con la fecha, la hora y el número de **Índice**. Cada vez que la estación actualiza el valor de sus sensores, dichos valores son almacenados en la memoria SD con un índice. Dicho índice es único e irrepetible. Este índice se va incrementando en cada lectura de sensores. En caso que la estación no pueda conectarse con el servidor, o algún dato no haya llegado, el servidor va a poder detectar la falta del mismo debido a la cadencia de cada dato, y de esta forma poder solicitar el reenvío del mismo.

2.1 Visualización de los datos

El programa servidor también tiene una interfaz gráfica donde se pueden ver por un lado los datos más actuales que se van recibiendo de las estaciones, una tabla donde se van observando los datos históricos y una sección de gráficas.

Apenas se abre el programa servidor, se muestra los datos que se van recibiendo, mostrando los más actuales (izquierda de la pantalla), y se ve además una tabla con datos históricos, los cuales se pueden seleccionar por estación y por fecha. A continuación, se muestra una captura de pantalla de lo indicado:

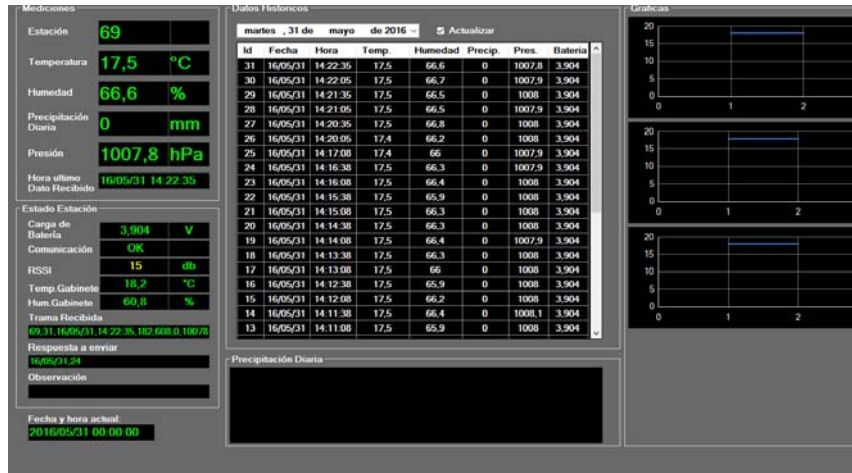


Fig. 3. Datos en el servidor en tiempo real.

También, se puede acceder desde el menú a una parte de visualización de gráficas, donde se selecciona que estación se quiere ver los datos y se muestra una gráfica mensual de las precipitaciones ocurridas cada día, además se puede hacer una representa visual de la ubicación geográfica de las estaciones. A continuación, se muestra una captura de pantalla:

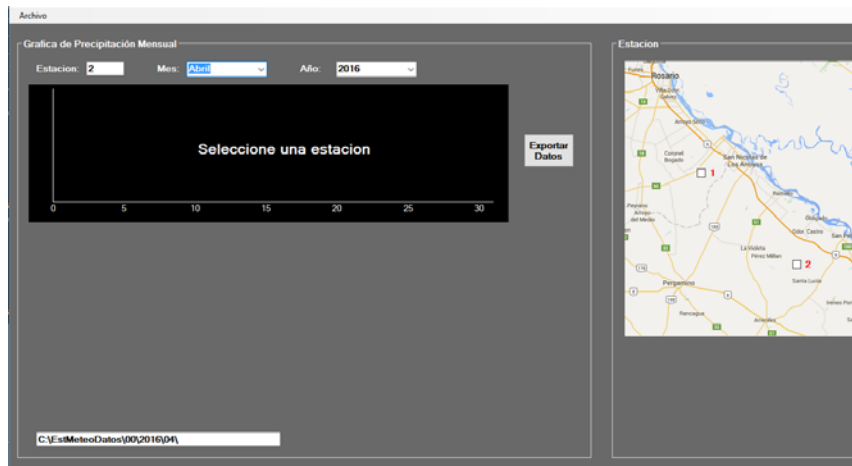


Fig. 4. Pantalla selección de nodo.

2.2 Comunicación

La comunicación se considera de vital importancia, ya que se desarrolló un mecanismo de recepción y recuperación de los datos (trazabilidad de datos), partiendo de la premisa de que ningún dato que se encuentre guardado en la memoria de las estaciones meteorológicas se pierda, sino que pueda ser recuperado.

El programa servidor, en la parte de comunicación funciona como un servidor TCP. Los clientes que envíen datos al servidor simplemente tienen que abrir un Socket a un puerto específico y a la IP de dominio público que tenga la computadora designada donde se va a instalar el programa servidor.

Una vez que se inicia el programa servidor, abre el puerto designado y queda en modo de escucha, esperando recibir algún dato de alguna estación, mientras, el usuario puede usar el resto de las funcionalidades del programa como, por ejemplo, ver datos históricos. Una vez que alguna estación envía algún dato, se empiezan a recibir éstos como tramas TCP, se decodifican, se verifica que la trama haya llegado completa y sin problemas, se almacenan los datos y por último se activa el mecanismo de recuperación de datos.

2.3 Almacenamiento de datos

El almacenamiento de los datos se hace con el uso de archivos de texto plano, sin usar algún motor de base de datos, si bien las bases de datos brindan una serie de ventajas, resulta más difícil portar de una computadora a otra el servidor, requiriendo incluso de un experto en IT para la instalación y configuración de la misma. Al usarse archivos de texto plano para guardar la información, el programa servidor se puede utilizar en cualquier computador sin la necesidad de tener que instalar y configurar aplicaciones terceras.

La forma en que se guardan los datos en archivos de texto plano tiene el mismo formato en la que están almacenados en la memoria SD de la estación meteorológica, con las distintas mediciones separadas por coma con un orden específico.

Luego, se creó una estructura de directorios para ordenar los datos de las distintas estaciones, y almacenados por fechas según lo siguiente:

1. Directorio raíz
 - (a) ID estación
 - (i) Año
 - (1) Mes
 - (a) Archivo

Con esta estructura, se guarda un archivo diario de datos por cada estación meteorológica disponible.

2.4 Mecanismo de recuperación de datos.

La comunicación entre las estaciones meteorológicas y el programa servidor es realizado mediante las redes de telefonía celular, específicamente, a través de la red GPRS, gracias a que este tipo de comunicación cubre un área bastante amplia de cobertura, lo cual permite ubicar las estaciones meteorológicas en lugares aislado, siempre y cuando, claro está, las operadoras de telefonía celular ofrezcan cobertura.

Durante la realización del proyecto se pudo constatar que este tipo de comunicación GPRS no es muy confiable en cuanto la cantidad de datos que fallan al ser enviados. Debido a ello, se tuvo que implementar un mecanismo para que, ningún dato almacenado en las estaciones meteorológicas se pierda y pueda ser recuperado por el servidor.

El mecanismo de recuperación se basa en lo siguiente:

El que lleva el control de los datos, si están completos o no, para todas las estaciones es el programa servidor. De esta manera se libera a las estaciones a tener control e implementar algoritmos complicados pensados a que están desarrollados con Arduino, que, si bien brinda un poder de procesamiento interesante, siempre va a ser inferior a un programa que corra en una computadora.

El Arduino (estación meteorológica) es quien se va a encargar de llevar la secuencia del índice de cada dato almacenado. Es quien recupera el valor en caso de pérdida de alimentación del mismo y quien resetea el valor del índice al comienzo del día.

Cuando una estación envía un dato existen dos opciones, que el dato se envíe correctamente o que falle. En caso de que falle, al servidor nunca le llega un dato, y no pasa nada. Ahora, si el dato enviado fue exitoso, el servidor lo registra, y revisa sus archivos en busca de si existe un dato anterior faltante, acá pueden hacer dos opciones, que no falte ningún dato, en ese caso, el servidor devuelve un OK, o caso contrario el servidor devuelve el índice faltante a la estación y ésta reenvía el dato solicitado al servidor. Dicho servidor continúa el ciclo revisando si están faltando datos anteriores y solicitándolo a la estación, hasta que finalmente todos los datos sean recuperados y el servidor devuelva un OK.

En la siguiente imagen puede verse el sistema recuperando los datos perdidos. Al servidor ha llegado un dato con Nro. de ID (índice) 25, pero detecta que le faltan los registros de ID desde el 13 hasta éste. Por lo que comienza el ciclo de recuperación de datos.

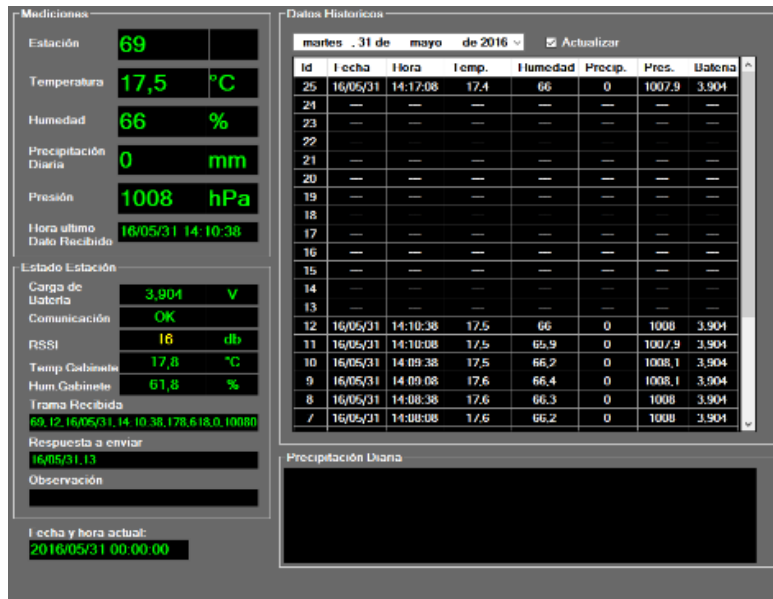


Fig. 5. Recuperación de datos.

Con este procedimiento se asegura que ningún dato se pierda a pesar de haber fallos en los envíos de datos por comunicación.

2.5 Estructura de datos

Los datos de los sensores son almacenados en la memoria SD en archivos que se ordenan por Número de nodo y fecha. Esto es, los datos almacenados por la estación número 31 el día 27 de mayo de 2016 se van a encontrar en el archivo 31160527.CSV (Nro. de Estación)(Año)(Mes)(Dia).CSV

En cada archivo, los datos se almacenan por renglón y separados cada valor por un carácter prefijado, en nuestro caso el carácter “;” pero podría ser cualquiera. Cada renglón tiene un número de índice único para poder identificar cada dato, y así enviarlo al servidor en caso que éste lo solicite.

La estructura de cada registro es la siguiente:

(Nro. Estación), (Nro. Índice), (Fecha), (Hora), (Temp Estación), (Humedad Estación), (Pluviometría), (Presión),
(Tensión Batería), (RSSI GSM), (Temp. Exterior), (Humedad Exterior), (Vel Viento), (Dir Viento),
(Tiempo Resp. TX), (Tiempo Resp. RX)

Nro. Estación: Número de Estación/Nodo

Nro. índice: Numero único de índice para identificar el registro

Fecha: Fecha del registro (AA/MM/DD)

Hora: Hora del registro (HH/MM/SS)

Temp Estación: Temperatura medida dentro de la Estación Meteorológica (gabinete)

Humedad Estación: Humedad leída dentro de la estación (gabinete)

Pluviometría: Pluviometría acumulada diaria

Presión: Presión atmosférica

Tensión Batería: Tensión de la batería de la estación (2da etapa del proyecto)

RSSI GSM: Intensidad de la señal GSM en RSSI

Temp Exterior: Temperatura Ambiente

Humedad Exterior: Humedad Ambiente

Vel Viento: Velocidad del viento (2da etapa)

Dir Viento: Dirección del viento (2da etapa)

Tiempo Resp. TX: Tiempo en que el servidor confirma la recepción del dato (valor solo para depuración del sistema)

Tiempo Resp. RX: Tiempo de respuesta del servidor (valor solo para depuración del sistema)

Por ejemplo, éste es el registro de índice 0 de la estación 69, el día 27 de mayo de 2016 a las 4hs, 57 minutos y 40 segundos.

69,0,16/05/27,04:57: 40,187,547,0,10212,3898,16,175,621,0,0,5201,530

Nro. Estación: 69
Nro. índice: 0
Fecha: 27 de mayo de 2016
Hora: 4hs, 57 minutos y 40 segundos
Temp Estación: 18.7C
Humedad Estación: 54.7%
Pluviometría: 0mm
Presión: 1021.2
Tensión Batería: 3,898V
RSSI GSM: 16
Temp Exterior: 17.5C
Humedad Exterior: 62.1%
Vel Viento: 0
Dir Viento: 0
Tiempo Resp. TX: 5201 ms
Tiempo Resp. RX: 530 ms

3 Diagramas de flujo

Se describe a continuación los diagramas de flujo principales del proyecto, en los mismos se detalla las secuencias de inicialización de/los nodos remotos, la sincronización para la comprobación de la correcta recepción de todos los datos transmitidos sin pérdida de información y por último el proceso de petición de dato faltante en el servidor para garantizar la trazabilidad de las variables meteorológicas.

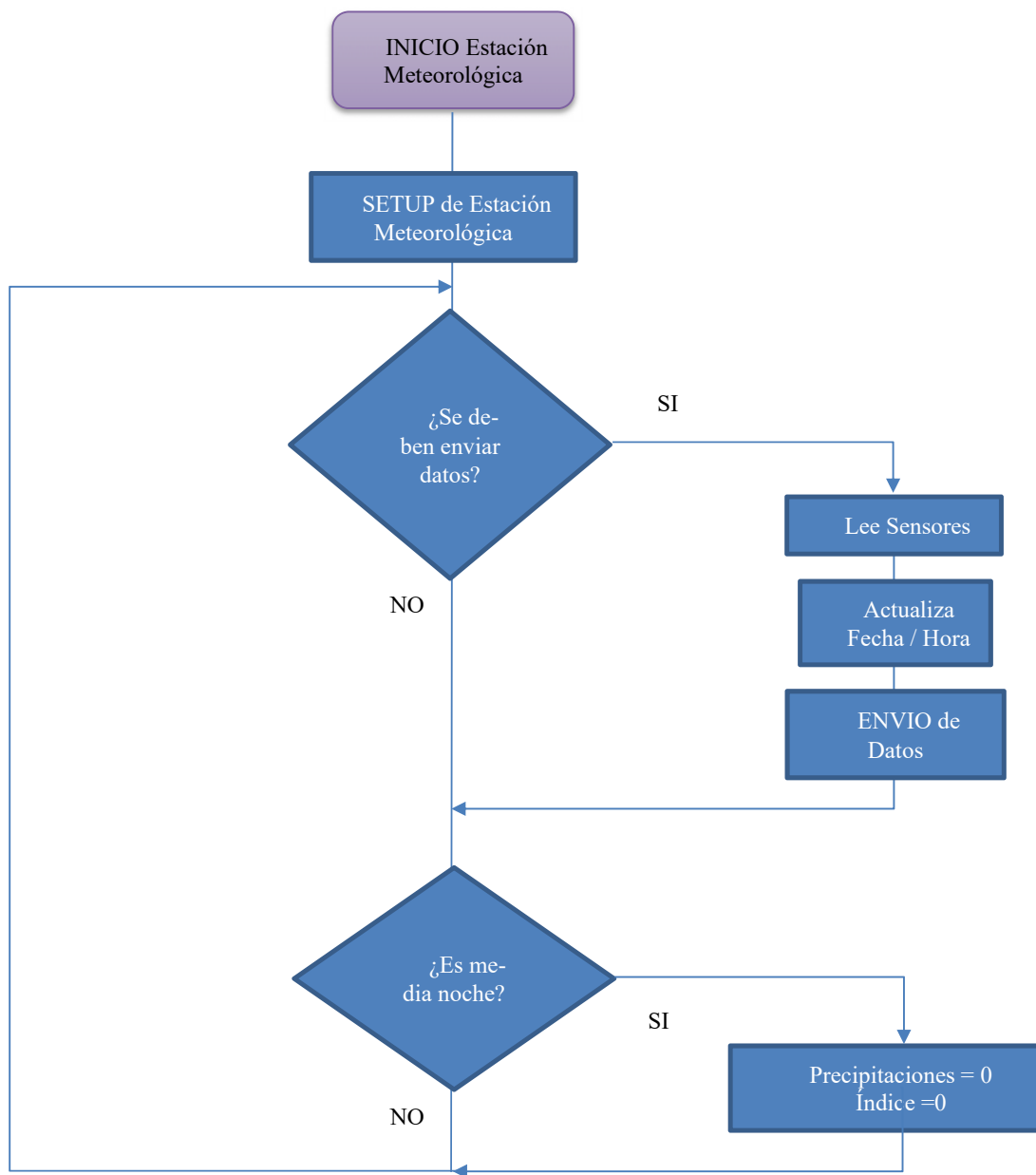


Fig. 6. Inicialización de la estación, actualización de valores de los sensores.

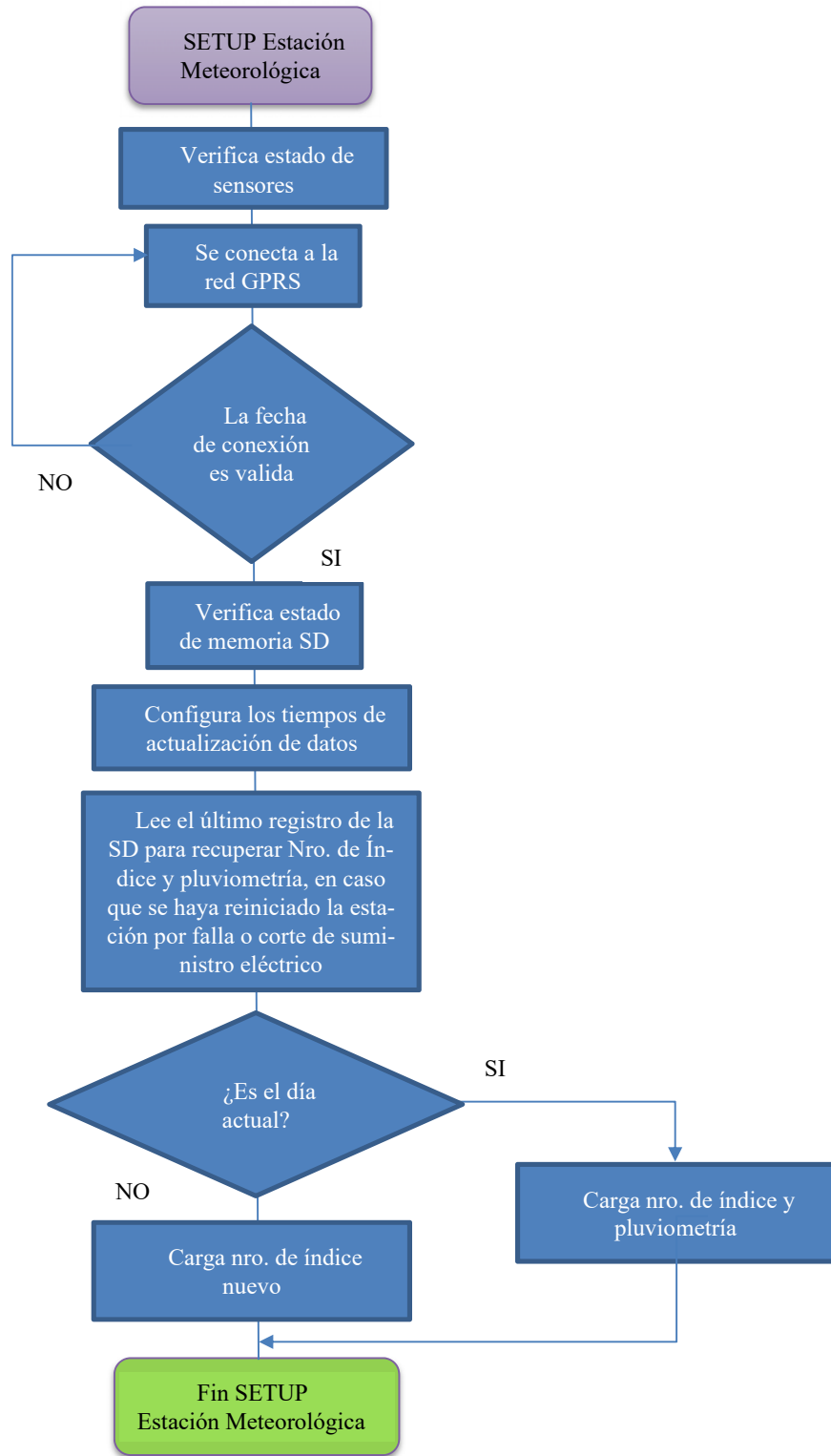


Fig. 7. Comprobación internada de almacenamiento local de datos y actualización de punteros de programa.

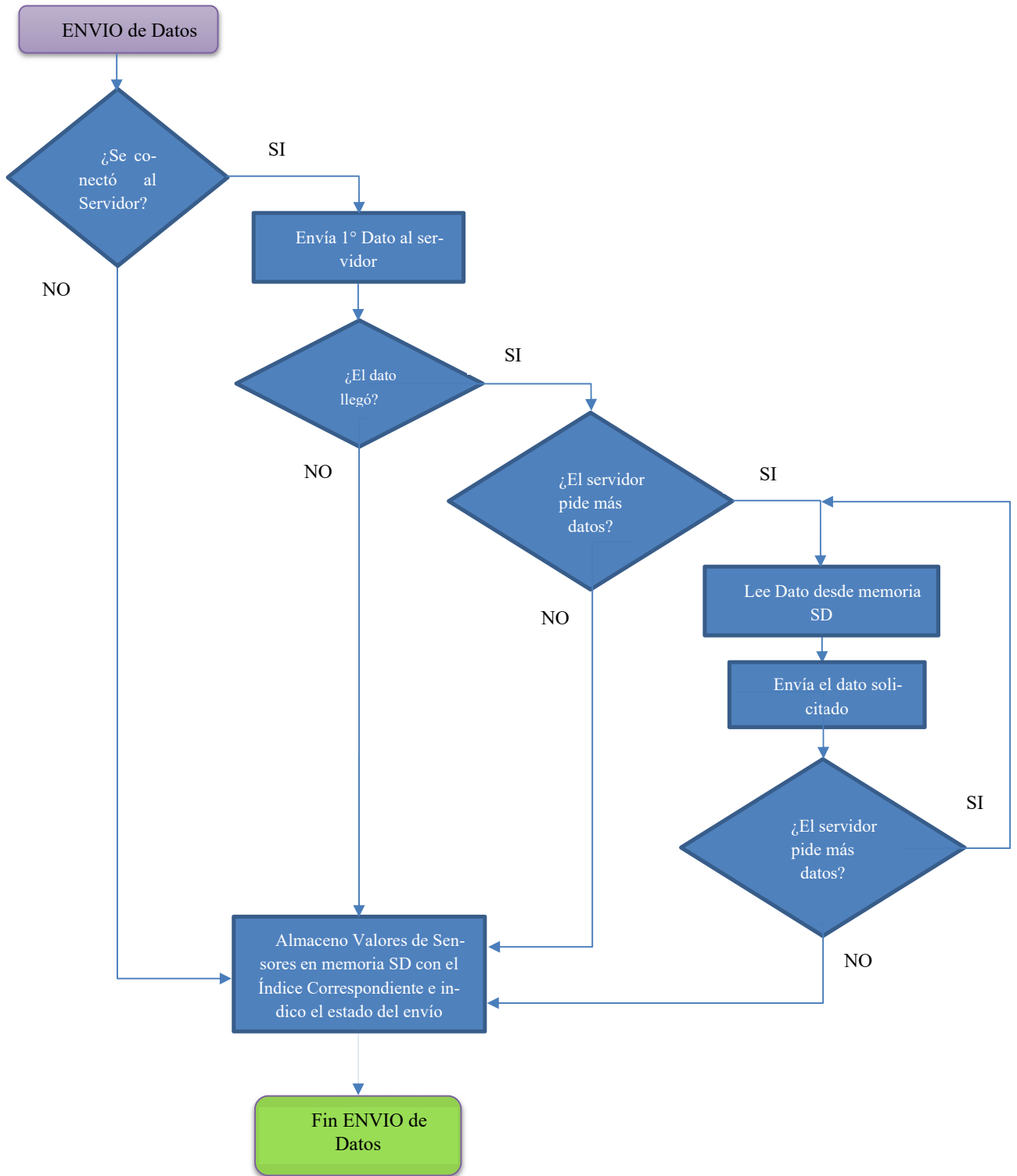


Fig. 8. Secuencia de envío de datos y comprobación de recepción correcta de las tramas en el servidor.

4 Futuras etapas del proyecto

En cuanto a las futuras etapas o mejoras que tendrá el proyecto, del lado del servidor se pretende lo siguiente:

Separar las partes de recepción de datos y visualización en dos programas diferentes, esto permite dar independencia entre ambas partes y trabajar así las aplicaciones de forma modular, siendo más adaptable a cambios y mejoras.

La parte de visualización de datos desarrollarla en aplicaciones basadas en web, de tal forma que el usuario pueda acceder a través de un explorador web sin la necesidad de tener que instalar o portar un software.

Como otra futura mejora es que la visualización de los datos se desarrolle en aplicaciones para ser usadas en celulares inteligentes y/o tabletas, acompañando a las nuevas tendencias tecnológicas.



Fig. 9. Prototipo desarrollado.

5 Referencias

1. Aragón Hernández, J. P. (16 de febrero de 2016). Arduinos y control Web: El control automático a través de la web. EAE.
2. Reyes Cortes, F. e. (1º de marzo de 2015). Arduino: aplicaciones en robótica, mecatrónica e ingenierías. Marcombo.
3. Ruiz, A. (2 de septiembre de 2010). Guía práctica de sensores. Creaciones Copyright.
4. Vara Mesa, J. M. (30 de mayo de 2012). Desarrollo web en entorno servidor. RA-MA S.A. Editorial y Publicaciones.