

# Indice de Anexos

## ANEXO I: HARDWARE

- **Display HD44780U: Datasheet**
- **Raspberry Pi: Planos electrónicos del fabricante**
- **WTW multi 350i: Información sobre la publicación de datos por puerto serie.**

## ANEXO II: SOFTWARE

- **RTU**  
Código: conf.sh  
Código: datos\_tr.sh  
Código: haltb.py  
Código: minicom.sh  
Código: watchdog.sh  
Código: status.py  
Código: rc.local
- **Servidor Principal**  
Código: juan.html  
Código: web\_tr.sh

<b>2. Envío de datos desde la RTU al Servidor Principal</b>	
2.1. Elección del protocolo de comunicación	7
2.1.1. Estándares de comunicación	7
2.1.2. Conclusiones	12
2.2. El Protocolo SCP	14
2.2.1. Uso del protocolo SCP	14
<b>3. Adquisición de datos del instrumento de medida desde la RTU</b>	17
3.1. El analizador multiparamétrico WTW Multi 350i	17
3.1.1. Generalidades	17
3.1.2. Características de la publicación de datos del WTW 350i	19
<b>4. Uso del estándar RS-232</b>	21
4.1. Concepto del estándar RS-232.	21
4.2. Características y modos de operación del RS-232.	21
4.3. Conectores y cables en RS-232	24
4.4. Señales utilizadas en la operación del estándar RS-232 para la comunicación con el analizador	25
4.4.1. TxD – Transmitted Data	26
4.4.2. RxD – Received Data	26
4.4.3. RTS/ CTS – Request To Send/ Clear To Send	26

4.5. Características de la interfaz de comunicación del dispositivo de medida	26
4.6. Conexión entre el analizador multiparamétrico y la unidad remota	27
4.6.1. Adaptador USB a RS-232	28
4.6.2. Cable DE-9F a terminales de conexión	28
5. La unidad terminal remota (RTU)	30
5.1. Concepto	30
5.2. Elección del dispositivo RTU	30
5.2.1. Requisitos del ordenador de la RTU	30
5.2.2. Alternativas consideradas	32
5.3. Concepto y diseño de los módulos de operación de la RTU	34
5.4. Descripción y funcionamiento de los módulos de la RTU	35
5.4.1. Módulo de carga de la configuración “conf.sh”	36
5.4.2. Módulo de comunicación serie “minicom.sh”	36
5.4.3. Módulo de copia de datos en el Servidor Principal “datos_tr.sh”	36
5.4.4. Módulo Watchdog “watchdog.sh”	37
5.4.5. Módulo de control del pulsador de apagado “haltb.py”	37
5.4.6. Módulo de gestión de la pantalla “status.py”	38
5.5. Arranque automático del sistema de la RTU	38
5.6. Gestión de usuarios del sistema de la RTU	39
5.7. Construcción de la RTU	41
5.7.1. Objetivo	41

5.7.2. Elementos de la RTU	41
5.8 Funcionamiento del terminal	49
5.8.1. Pantalla de la RTU	49
5.8.2. Administración local de la RTU	52
 6. El Servidor Principal	 54
6.1. Concepto	54
6.2. Requisitos del Servidor Principal	55
6.2.1. Sistema operativo Linux	55
6.2.2. Hardware	55
6.2.3. Software	55
6.2.4. Características de red	56
6.3. Módulos del Servidor Principal	56
6.4. Funcionamiento del Servidor Principal	57
 7. Mejoras Futuras	 59
 8. Bibliografía	 60

# 1. Introducción

## 1.1. Introducción general

Cualquier proceso de investigación o experimentación se basa de forma inherente en la recopilación de datos que sirvan como fundamento a la hora de enunciar resultados. Es por ello que los investigadores deben centrar gran parte de sus esfuerzos en asegurarse la obtención de la mayor cantidad de información, y que ésta sea lo más fidedigna posible.

Por tanto, una de las cuestiones críticas que deben abordarse con extrema atención de forma previa al inicio de una investigación, es decidir qué instrumentación deberá usarse en la ejecución del proyecto.

En muchos de los casos, los investigadores cuentan con los instrumentos de laboratorio necesarios, pero no les resulta posible recopilar la información provista por estos dispositivos de forma óptima, debido a las limitaciones impuestas por las herramientas informáticas dispuestas por los fabricantes para la adquisición y gestión de los datos.

## 1.2. Objetivos

### 1.2.1. Propuesta de proyecto

Entre las múltiples líneas de investigación que el grupo TAR tiene en marcha se encuentra un proyecto particularmente ambicioso titulado: *“Tratamiento de aguas residuales de industrias agroalimentarias mediante codigestión anaerobia en el tratamiento de fango de las estaciones depuradoras de aguas residuales”*. En este proyecto se investigan las técnicas de procesamiento de residuos que podrán aplicarse en un futuro próximo, y que harán posible que las E.D.A.R. que implanten estos métodos puedan optimizar su rendimiento tanto económico como ecológico.

Este proyecto liderado por D. Julián Lebrato, está siendo desarrollado en las instalaciones del grupo TAR en los Campos Experimentales Blanco White, a las afuera de Sevilla. Actualmente, esta investigación cuenta con un digestor piloto en el cual deben monitorizarse de forma minuciosa y continuada ciertos parámetros clave en el desarrollo de este proyecto. Esta tarea es llevada a cabo por los investigadores de forma presencial, obteniendo manualmente los datos de medidas de instrumentos como el analizador múltiparamétrico WTW Multi 350i.



*Figura 1. Co-digestor piloto*

La propuesta de proyecto planteada por D. Julián Lebrato es desarrollar un sistema que permita que los datos proporcionados por los aparatos de medida puedan ser monitorizados en remoto, y que estos queden almacenados de forma automática para su posterior procesamiento y análisis.

### **1.3.2. Objetivos específicos**

Con este proyecto se persigue el desarrollo de una plataforma integral que permita mediante una unidad terminal remota (RTU), capturar en tiempo real los datos publicados por un instrumento de medida de laboratorio, para luego transmitir estos a través de internet hacia un servidor principal. De esta forma, la información podrá encontrarse a disposición de los investigadores en cualquier momento.

A continuación se exponen de forma más detallada los objetivos en base a los cuales se ha dirigido el desarrollo de este proyecto:

- **Minimizar el uso de instrumental específico para cada proyecto**

En la mayoría de los casos, los dataloggers (registradores de datos) y pasarelas de datos utilizados en laboratorio requieren del uso de sondas con tecnología privada que no son compatibles con otros modelos de analizadores, o están limitados a un cierto número de entradas analógicas o digitales. Ante esta situación, resulta indispensable en la mayoría de los casos la adquisición de dispositivos específicos para cada aplicación.

Poder utilizar el instrumental de laboratorio del que ya se dispone es uno de los objetivos principales de esta prueba de concepto. Muchos de los instrumentos de medida estudiados durante el desarrollo de este proyecto, como son los modelos PCD 650 de la casa Eutech, o el multi 350i de WTW, cuentan con interfaces de comunicación serie basados en el estándar RS-232. Gracias a esta característica común, se ha buscado aislar la capa de instrumentación de la gestión de los datos ya procesados, permitiendo al técnico de laboratorio el uso de dispositivos con los que ya se encontraba familiarizado, agilizando así las fases de calibración, montaje, configuración del instrumental, etc.

- **Empleo exclusivo de software de licencia libre y estándares abiertos.**

En la actualidad la mayoría de fabricantes de dataloggers e instrumental de medida, ponen a disposición de sus clientes herramientas de comunicación con sus dispositivos, como por ejemplo *HOBOnode Viewer Utility* de la casa Onset o *Multi Lab Pilot* del fabricante WTW. Este tipo de aplicaciones requieren en todos los casos del uso de PCs completos (monitor, teclado, ratón, etc.), además de obligar al usuario a utilizar versiones concretas de sistemas operativos compatibles. Por otra parte, en la mayoría de los casos, este tipo de software necesita que el técnico de laboratorio gestione manualmente los datos, y sólo permite exportar la información recogida en una variedad limitada de formatos. En contrapunto, algunos de los principales objetivos perseguidos en el desarrollo del dispositivo presentado en este proyecto, son crear una pasarela de comunicación independiente, basada en software libre y que no implique el uso de material informático extra dedicado. De esta manera, será posible reducir de forma drástica los costes durante las investigaciones realizadas tanto en laboratorio como en campo.

- **Disponibilidad de los datos de forma remota.**

Actualmente algunos fabricantes de sistemas de adquisición y gestión de medidas incluyen para algunos de sus modelos más avanzados, la posibilidad de contar con plataformas online a través de las cuales es posible consultar los datos registrados como es el caso de *HOBO Link* de Onset. Aparte de las limitaciones que pueden imponer los proveedores del servicio en cuanto al formato y modo de acceso a los datos, debido a la necesidad de gestionar y almacenar dicha información, estos servicios pueden estar sujetos a contratación.

Por otra parte, si se desean utilizar las capacidades online de este tipo de dispositivos, puede ser necesaria la configuración de ciertos parámetros

avanzados de red como IPs estáticas o reglas en los firewall para el direccionamiento de puertos de comunicación. Es por esta razón que otro de los objetivos principales de este proyecto, es poner a disposición de los investigadores la posibilidad de acceder de forma remota a los datos tanto históricos, como en tiempo real, provistos por sus instrumentos de medida, procurando minimizar tanto los requisitos, como la configuración necesaria de la red local que permitirá el acceso a esta información en cualquier parte del mundo.

Por ello, asimismo se ha contemplado en el alcance de este proyecto la creación de un Servidor Principal basado en Linux, el cual permite tanto monitorizar en tiempo real los datos suministrados por los instrumentos de medida a través de una web, como alojar y acceder a los datos recopilados en forma de históricos.

## **2. Envío de datos desde la RTU al Servidor Principal**

### **2.1. Elección del protocolo de comunicación**

En la actualidad diversos protocolos de comunicación en tiempo real son utilizados en el ámbito de aplicación de los SCADAs dependiendo de las necesidades y de las características de los entornos en los que se desean implementar.

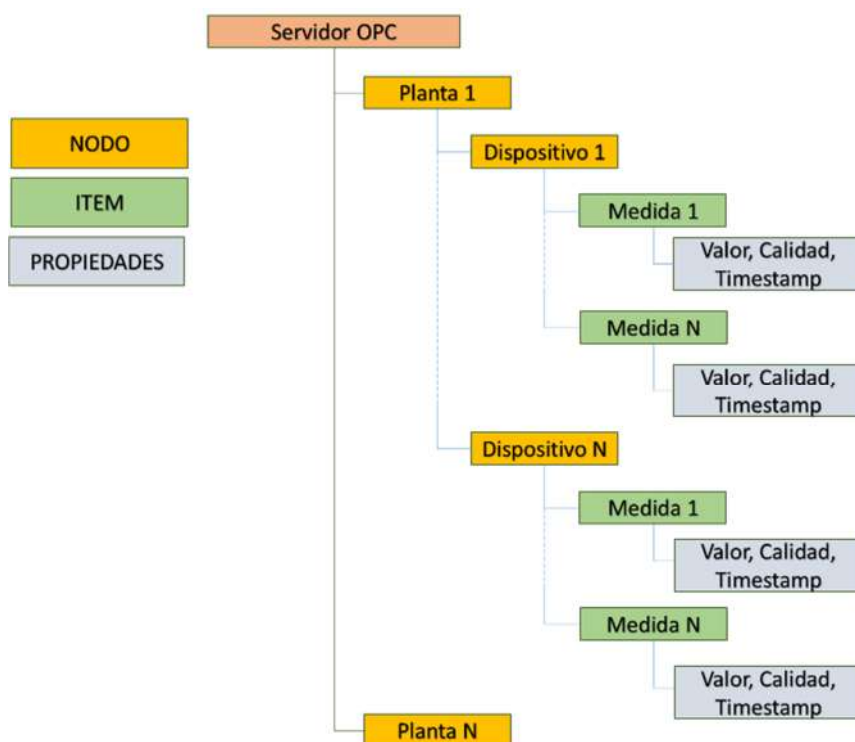
#### **2.1.1. Estándares de comunicación**

Debido al objetivo principal del proyecto consistente en permitir la transmisión de datos a través de una conexión a internet común, únicamente fueron contemplados los protocolos creados para trabajar sobre redes TCP/IP. A continuación se describen brevemente algunas de las alternativas contempladas:

- **OPC**



Las siglas OPC hacen referencia a OLE (Object Linking and Embedding) for Process Control, aunque a partir del 2011 el consorcio OPC Foundation cambió el significado de sus siglas por Open Platform Communication. Originalmente creado en el año 1996 y basado en tecnologías propietarias de Microsoft como OLE y DCOM (Distributed Component Object Model), consiste en una serie de especificaciones estándar que en la actualidad se han posicionado como uno de los protocolos de interoperabilidad más extendidos en la industria. Este consta de una arquitectura cliente-servidor basada en el protocolo RCP (Remote Procedure Call) mediante el cual es posible ejecutar código en una máquina remota. Este compendio de estándares de comunicación permite que diferentes equipos compartan datos tanto en tiempo real como históricos, estructurando la información en forma de árbol en los cuales los puntos de medida son llamados “items” (elementos) que a su vez se encuentran agrupados en nodos. Cada ítem cuenta además con ciertas propiedades asociadas como son los permisos de acceso, “timestamp”, o el tipo de dato.



*Figura 2. Ejemplo de estructura de datos en un servidor OPC.*

Debido al uso de la tecnología DCOM de Microsoft, la configuración de los equipos que desean ser intercomunicados a través de este protocolo no resulta de ningún modo trivial. Es por esta razón sumada a otros inconvenientes propios

de este estándar, como son ciertos problemas de seguridad, o la necesidad de utilizar Microsoft Windows, lo que ha movido a la OPC Foundation al lanzamiento en 2006 de un nuevo estándar conocido como OPC UA (Unified Architecture) que busca solventar los puntos débiles del protocolo original.

- **IEC 104 (IEC 60870-5-104)**

Consiste en una extensión del protocolo IEC 101 (IEC 60870-5-101) el cual fue creado para funcionar sobre únicamente conexión serie (por ejemplo RS-232 o RS-485). Se trata de una norma especialmente utilizada en la actualidad para la supervisión y control remoto en tiempo real en el ámbito de la producción de la energía eléctrica.

La arquitectura de este protocolo es del tipo Maestro-Esclavo. Una vez establecido el enlace entre la unidad de control principal (Master) y la unidad remota (Slave), éstas se comunican a través de datagramas en los que se empaquetan de forma estructurada junto con el valor de la medida, las características de la información transmitida como son el remitente, calidad de la señal, tipo de dato, causa de la transmisión, etc.

```
IEC 60870-5-104-Asdu: ASDU=5 M_ME_TF_1 Spont IOA[16]=13190,... 'measured value'
  TypeId: M_ME_TF_1 (36)
  0... .... = SQ: False
  .001 0000 = NumIx: 16
  ..00 0011 = CauseTx: Spont (3)
  .0.. .... = Negative: False
  0... .... = Test: False
  OA: 0
  Addr: 5
  IOA: 13190
    IOA: 13190
    Value: 2.77815
  QDS: 0x00
    .... ...0 = OV: No overflow
    ...0 .... = BL: Not blocked
    ..0. .... = SB: Not Substituted
    .0.. .... = NT: Topical
    0... .... = IV: valid
  CP56Time: Feb 18, 2015 10:41:23.499000000 Hora estándar romance
```

*Figura 3. Ejemplo de paquete de datos sobre protocolo IEC104*

- **MODBUS TCP**

Modbus consiste en un protocolo de comunicación serie publicado por la compañía Modicon en 1979 para ser utilizado en PLCs. Modbus es un protocolo abierto, por lo que cualquier fabricante o desarrollador puede hacer uso de este estándar sin necesidad de pagar royalties. Esta es una de las razones por la cual desde hace años, es uno de los medios más comunes de comunicación entre dispositivos electrónicos.

La arquitectura de esta norma es del tipo Maestro-Esclavo. Un único equipo actuando como Maestro puede comunicar con hasta 247 dispositivos esclavos a los que mediante una operación de petición, consulta o escribe los datos almacenados en los registros de cada esclavo. Estos registros ocupan diferentes direcciones de acceso en el esclavo, dependiendo del tipo de dato, como pueden ser medidas analógicas o estados booleanos, que desea contenerse en cada uno de ellos.

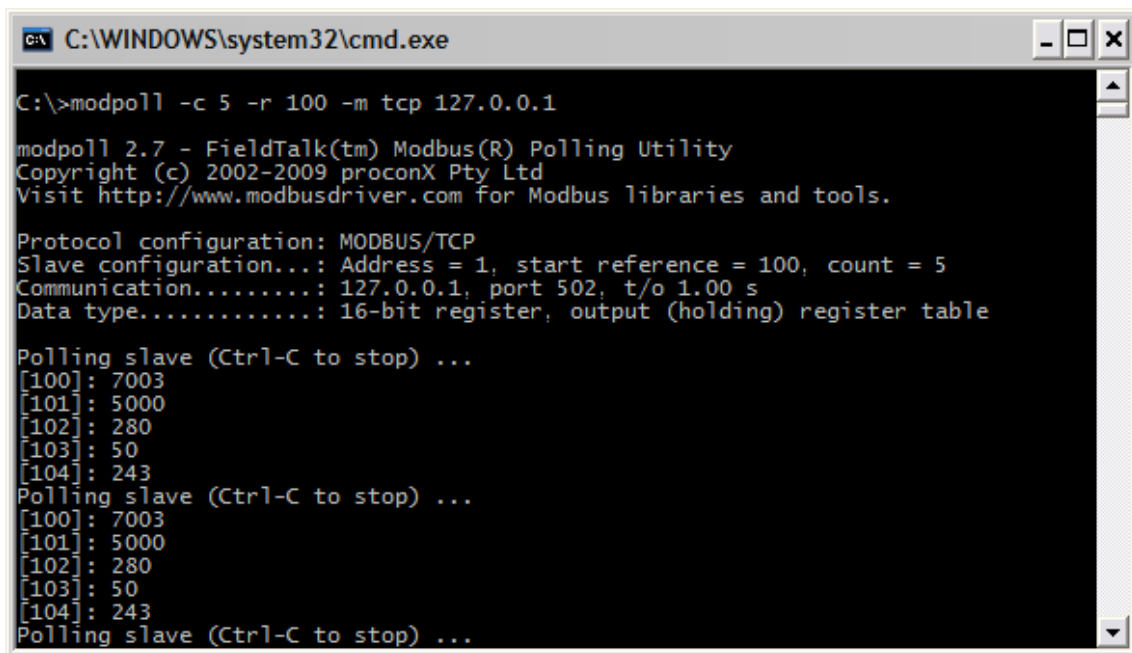
				Function Codes			
				code	Sub code	(hex)	Section
Data Access	Bit access	Physical Discrete Inputs	Read Discrete Inputs	02		02	6.2
		Internal Bits Or Physical coils	Read Coils	01		01	6.1
			Write Single Coil	05		05	6.5
			Write Multiple Coils	15		0F	6.11
	16 bits access	Physical Input Registers	Read Input Register	04		04	6.4
			Read Holding Registers	03		03	6.3
		Internal Registers Or Physical Output Registers	Write Single Register	06		06	6.6
			Write Multiple Registers	16		10	6.12
			Read/Write Multiple Registers	23		17	6.17
			Mask Write Register	22		16	6.16
			Read FIFO queue	24		18	6.18
	File record access		Read File record	20		14	6.14
		Write File record	21		15	6.15	
Diagnostics			Read Exception status	07		07	6.7
			Diagnostic	08	00-18,20	08	6.8
			Get Com event counter	11		0B	6.9
			Get Com Event Log	12		0C	6.10
			Report Slave ID	17		11	6.13
			Read device Identification	43	14	2B	6.21
Other			Encapsulated Interface Transport	43	13,14	2B	6.19

Figura 4. Tipos de datos Modbus y sus Function Codes asociados.

En el caso de que se desee usar este medio de comunicación a través de una red TCP/IP en lugar de una conexión serie como podría ser RS-232 o RS-485, existe una versión específica llamada Modbus TCP, en la cual los mensajes Modbus son transmitidos utilizando una envoltura TCP/IP. Esto es, cuando la información es transmitida utilizando estos protocolos, mediante TCP (Transmission Control Protocol) a los datos originales Modbus se le adjunta

cierta información añadida necesaria para su transmisión, luego el protocolo IP coloca todos los datos en un paquete (datagrama) el cual es finalmente transmitido.

Asimismo, debe tenerse en consideración que al hacerse uso del protocolo TCP antes de transferir ningún tipo de dato, debe existir una conexión establecida entre el Maestro y el Esclavo Modbus TCP; el Esclavo permanecerá a la espera de una conexión entrante del Maestro, una vez establecida dicha conexión, el Esclavo responderá a la peticiones realizadas sobre este hasta que el Maestro termine su conexión.



```
C:\WINDOWS\system32\cmd.exe
C:\>modpoll -c 5 -r 100 -m tcp 127.0.0.1

modpoll 2.7 - FieldTalk(tm) Modbus(R) Polling Utility
Copyright (c) 2002-2009 proconX Pty Ltd
Visit http://www.modbusdriver.com for Modbus libraries and tools.

Protocol configuration: MODBUS/TCP
Slave configuration...: Address = 1, start reference = 100, count = 5
Communication.....: 127.0.0.1, port 502, t/o 1.00 s
Data type.....: 16-bit register, output (holding) register table

Polling slave (Ctrl-C to stop) ...
[100]: 7003
[101]: 5000
[102]: 280
[103]: 50
[104]: 243
Polling slave (Ctrl-C to stop) ...
[100]: 7003
[101]: 5000
[102]: 280
[103]: 50
[104]: 243
Polling slave (Ctrl-C to stop) ...
```

*Figura 5. Ejemplo de operación de petición de datos (poll) a un esclavo Modbus TCP mediante la aplicación Modpoll.*

### 2.1.2. Conclusiones

Estos protocolos de comunicación poseen como característica común una eficacia probada como estándares industriales en cuanto a estabilidad, capacidad e integridad en la transmisión de información. De igual forma, y debido al extensivo uso de estos protocolos, existe una amplia gama de herramientas de análisis de comunicación como pueden ser *Matrikon OPC Explorer*, *Wireshark*, *ModPoll*, etc. Estas aplicaciones permiten a los administradores y mantenedores de los sistemas detectar errores, ayudando tanto a prevenir como a corregir posibles incidencias en su funcionamiento.

Sin embargo, se ha descartado el uso de estos protocolos de comunicación anteriormente descritos y de otros similares, debido a que en todos los casos estos estándares de comunicación basan la transmisión de la información en objetos individuales mapeados, llamados “tags” o “items”, los cuales contienen el dato o los datos que desean ser transferidos.

A modo de ejemplo, si se desean transmitir mediante protocolo IEC 104 varias medidas de temperatura obtenidas por diferentes sondas desde un terminal remoto hasta una unidad central, debe declararse para cada punto de medida un *tag* único como podría ser ‘SONDA\_1\_TEMP\_A\_DEGC’, ‘SONDA\_1\_TEMP\_B\_DEGC’, etc. Estos elementos deben siempre ser definidos de forma idéntica en las bases de datos de los drivers de comunicación Maestro y Esclavo, y siguiendo criterios de configuración de parámetros que dependerán de las características intrínsecas de las señales que se manipulen, tales como son el origen de la medida, el tipo de dato, o el rango de valores válidos.

Unidad Remota 1, CA (Common Address) = 1			
Nombre de la medida	Valor	Tipo de dato	IOA (Information Object Address)
SONDA_1_TEMP_A_DEGC	20,3	Coma Flotante	100
SONDA_1_TEMP_B_DEGC	22,5	Coma Flotante	200
SONDA_1_TEMP_C_DEGC	21,4	Coma Flotante	300

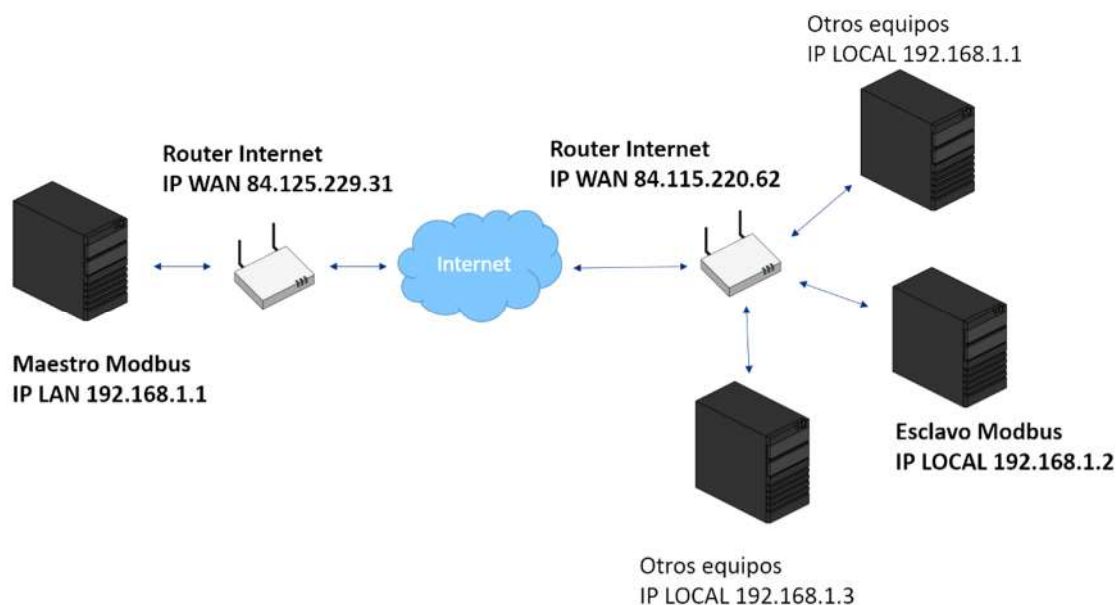
*Figura 6. Ejemplo simplificado de datos utilizados por el protocolo IEC-104*

Una integración incorrecta o poco estudiada tanto de las señales implicadas en la transmisión, como de los parámetros de configuración del canal de conexión que debe establecerse entre el Maestro y el Esclavo, impactará de forma negativa en el rendimiento global de la comunicación. En sistemas implementados de forma deficiente, tendrán lugar retardos, pérdida de paquetes de información, o incluso problemas de integridad punto a punto en los datos transferidos.

Por lo tanto, la integración de un sistema de transferencia de medidas basado en “tags”, conlleva en todos los casos un estudio pormenorizado de las señales a transmitir, así como un proceso de ingeniería de datos en el que deben definirse los parámetros tanto cualitativos como cuantitativos de todas las señales con las que trabajará el sistema. Este método de transmisión de datos aunque potente y fiable, una vez puesto en marcha no admite ninguna variación en cuanto al número o tipo de señales, no sin antes realizar una nueva revisión y posterior actualización del sistema.

Por otra parte, resulta necesario si se desea utilizar cualquiera de estos protocolos que tanto la unidad remota como la red de conexión a la que esta se encuentra conectada cumplan ciertos requisitos, como puede ser la necesidad de utilizar una IP estática o un enrutado de puertos para permitir la comunicación desde un punto exterior a la red local.

A modo de ejemplo, si se opta por la alternativa Modbus TCP sobre internet, será imprescindible definir en el Maestro que accederá a los datos una dirección IP (84.115.220.62) invariable para el Esclavo Modbus del que pretenden obtenerse los datos, así como una regla de enrutamiento en la red local que permita a una petición Modbus TCP típicamente realizada sobre el puerto 502 dirigirse a la IP local del equipo (192.168.1.2) en el que se ejecuta el Esclavo Modbus.



*Figura 7. Ejemplo de red Modbus TCP.*

Partiendo de la premisa original de crear un entorno flexible que pueda admitir la transmisión de las medidas obtenidas de cualquier tipo de

instrumento, y publicar esta información a un servidor principal utilizando un acceso a internet común con la menor necesidad de configuración, se ha optado por el uso de SCP (Secure Copy) siendo éste un método de transferencia de datos que permite abstraer la transmisión de la información de las características particulares de los datos con los que se trabaja.

## 2.2. El Protocolo SCP

El servicio SCP consiste en un medio para la copia de archivos basado en el protocolo SSH (Secure Shell). Este protocolo extensamente utilizado en la informática para transferir archivos de forma segura entre equipos, es asimismo utilizado actualmente en la industria de los SCADAs (por ejemplo *Symphony Plus* y *Network Manager* de ABB) para compartir ficheros de datos históricos almacenados en las unidades remotas, tras sufrir una periodo en el que no ha sido posible la publicación por parte de la unidad remota de datos en tiempo real debido a una incidencia en la comunicación.

Algunas de las ventajas añadidas del uso de este método de transferencia de archivos frente a otros como pudiera ser FTP (File Transfer Protocol), se basan en el uso del servicio SSH el cual aporta un nivel óptimo de seguridad en las transacciones, además de encontrarse disponible de forma nativa en la mayoría de las distribuciones Linux.

### 2.2.1. Uso del protocolo SCP

El uso del protocolo SCP ha sido implementado de forma que todas las transacciones son iniciadas desde el terminal remoto. De esta forma, es posible permitir que el dispositivo local pueda funcionar incluso cuando se le asigna una IP dinámica ya que sólo necesita conocer la IP estática del servidor principal para realizar la transferencia de ficheros. El único requisito que debe cumplir el punto de acceso a internet de la RTU es tener el puerto TCP estándar SSH (TCP 22) abierto. Este puerto TCP 22 suele estar siempre abierto en la configuración por defecto de cualquier red.



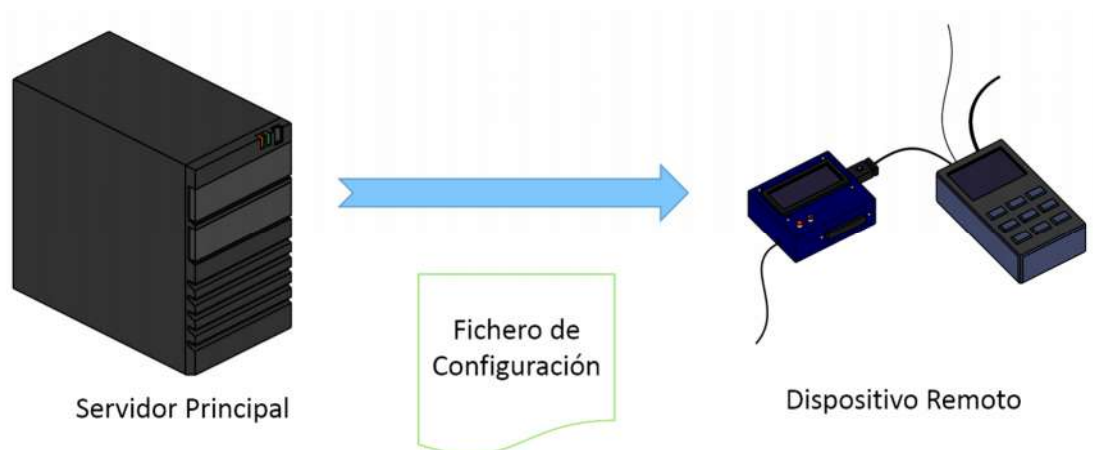
- **Autenticación de la Unidad Remota en el sistema principal.**

Con el fin de permitir que la unidad remota pueda realizar las operaciones de transferencia de ficheros de forma automática, se ha hecho uso de un sistema de claves pre-compartidas (PSK) el cual permite a la unidad remota acceder al sistema principal de forma segura.

- **Operaciones realizadas mediante SCP**

- **Copia de archivos desde el Servidor Principal.**

Mediante una operación de copia de archivos desde el Servidor Principal e iniciada en todos los casos desde la RTU, puede importarse a la unidad remota un archivo de configuración que servirá para establecer los parámetros deseados de comunicación con el instrumento de medida, o la información de la conexión Wi-Fi a la que se conectará la RTU.



*Figura 8. Copia de archivos desde el servidor principal.*



La estructura del comando Linux ejecutado desde la unidad remota para obtener el fichero de configuración es el siguiente:

```
scp usuario@direccion_ip_servidor_remoto:/ruta_remota /ruta_local
```

*Figura 9. Estructura del comando SCP para la copia de archivos desde un equipo remoto.*

### – Copia de archivos en el Servidor Principal

A través de esta acción, se copian en el Servidor Principal de forma periódica los ficheros de datos generados a partir de la comunicación con el instrumento de medida.



*Figura 9. Copia de archivos hacia el servidor principal.*

Estos datos crudos podrán ser consultados en todo momento, tanto en tiempo real como en forma de históricos, por los usuarios de la plataforma para supervisar de forma remota los valores servidos por el instrumento de medida.

Del mismo modo, se envían al Servidor Principal otros archivos auxiliares, que permitirán al administrador del sistema conocer el estado de funcionamiento de la unidad remota. Estos ficheros contienen información acerca de la conexión a internet establecida o el uso de recursos del sistema.

La estructura del comando Linux ejecutado desde la unidad remota para obtener el fichero de configuración es el siguiente:

```
scp /ruta_local usuario@dirección_ip_servidor_remoto:/ruta_remota
```

*Figura 10. Estructura del comando SCP para la copia de archivos en un equipo remoto.*

## **3. Adquisición de datos del instrumento de medida desde la RTU**

### **3.1. El analizador multiparamétrico WTW Multi 350i**

El instrumento de medida seleccionado para la prueba de concepto del sistema ha sido el analizador multiparamétrico Multi 350i de la casa WTW.

Debido a sus características y especificaciones, que serán expuestas a continuación, este dispositivo resulta ideal para demostrar las capacidades del sistema de supervisión remota de instrumentos de medida desarrollado en este proyecto.

#### **3.1.1. Generalidades**

El fabricante alemán WTW es parte del grupo Xylem el cual es uno de los mayores proveedores globales de tecnologías para el análisis y tratamiento del agua en contextos de servicios públicos, residenciales y comerciales y mercados agrícolas e industriales.

Por su parte, la casa WTW cuenta con más de 60 años de experiencia en el ámbito de la instrumentación destinada al análisis del agua. Es por ello que se presentan como el fabricante con la línea de productos más completa y reconocida a nivel mundial.

## Sistema de supervisión remota para instrumentos de medida

Comercializado desde el año 2004, el aparato multiparamétrico Multi 350i cuenta con las siguientes características según su catálogo comercial:

- **Multifuncional con la máxima precisión**
- **Flexible**
- **De uso universal**

*Precisión compacta sin problemas.*

*pH, mV, ISE, oxígeno, conductividad: el Multi 350i mide todos estos parámetros y, si se desea, también pH, oxígeno, conductividad y temperatura al mismo tiempo: en el laboratorio con el ConOx combinado con sensor de conductividad y oxígeno, en el campo con la sonda multiparamétrica MPP 350.*

*Además, se pueden conectar todos los sensores actuales de pH, ISE combinado, oxígeno y conductividad.*

*Alta resolución, alta precisión, sencillo control mediante menús. En condiciones de escasa visibilidad la pantalla retroiluminada brinda una clara lectura de los valores. Un registrador de datos, 1.800 espacios de almacenamiento y un reloj en tiempo real con fecha permiten la captación de datos conforme a las buenas prácticas de laboratorio. La batería recargable incorporada de NiMH brinda hasta 1000 horas de mediciones continuas y viene con adaptador para corriente eléctrica*



*Figura 10. Extracto del catálogo de WTW*

### 3.1.2. Características de la publicación de datos del WTW 350i

El dispositivo está preparado para que vía interfaz RS-232 sea posible transmitir datos a un PC o una impresora. En ambos casos es necesario el uso de un cable de conexión provisto por separado por el fabricante.

```
3.11.03 15:48:08
Multi 350i ser. no. 12345678
ID number: 1
ISE 0.316 mg/l 22 °C , AR
Oxi 6.32 mg/l 24.8 °C , AR
Sal = 0.7
Cond 1413 uS/cm 24.8 °C
C = 0.975 1/cm, Tref25, Lin, TC = 2.000 %/K

03.11.03 09:56:20
Multi 350i ser. no. 12345678
ID number: 1
pH 6.12 24.8 °C , AR
Oxi 7.46 mg/l 24.8 °C
Sal = 0.0
Res 69.0 kOhm*cm 24.8 °C , AR
C = 0.010 1/cm, Tref25, Lin, TC = 2.000 %/K

03.11.03 09:27:24
Multi 350i ser. no. 12345678
ID number: 1
pH 7.13 24.8 °C Cond-TP, AR
Oxi 5.95 mg/l 24.8 °C , AR
Sal = 0.7
Res 0.708 kOhm*cm 24.8 °C , AR
C = 0.975 1/cm, Tref25, Lin, TC = 2.000 %/K
```

*Figura 11. Ejemplo de datos publicados por el analizador.*

La siguiente tabla muestra qué datos son transmitidos según el modo de operación seleccionado:

Datos	Control	Operación / descripción
Valor actual medido de todos los sensores	Manual	Pulsando brevemente la tecla <PRT> Simultaneo con cada ciclo de almacenamiento de datos
	Automático a intervalos	Pulsando prolongadamente la tecla <PRT>. El intervalo de transmisión puede ser seleccionado en este menú. Simultaneo con cada ciclo de almacenamiento de datos
Valores medidos almacenados	Manual	Pulsando <PRT> se transmitirá la colección de datos recuperada del almacenamiento.
Registros de calibración	Manual	Pulsando <PRT> se transmitirán los registros de calibración recuperados del almacenamiento.
	Automático	Registros de cada respectivo sensor al terminar el proceso de calibrado.

Figura 12. Modos de publicación de datos

Dado que el objetivo del proyecto es obtener datos periódicamente en tiempo real, y de forma autónoma para ser almacenados y supervisados en remoto, la publicación de datos por parte del instrumento de medida se realizará en modo ‘Valor actual medido de todos los sensores’ en control automático.

## 4. Uso del estándar RS-232

### 4.1. Concepto del estándar RS-232.

Publicado por la Electronics Industries Association (IEA) en los años 60, el RS-232 es la interfaz serie I/O más extendida. Este estándar define formalmente entre otras las características eléctricas de la capa física de comunicación, la temporización y significado de las señales, así como el pinyado de los conectores a utilizar. Pero por el contrario, no establece otras limitaciones como podrían ser protocolos a utilizar o una cierta codificación de los datos.

Sobre este estándar se pueden implementar protocolos de comunicación serie como pueden ser Modbus RTU o IEC 101 (IEC 60870-5-101).

Aunque el uso del interfaz RS-232 se considera obsoleto en el ámbito de la informática de consumo, sigue siendo uno de los medios más comunes de comunicación en máquinas industriales, en el equipamiento de red profesional, o en instrumentos de laboratorio por citar algunos ejemplos.

### 4.2. Características y modos de operación del RS-232.

Este estándar se basa en el envío de series temporales de bits, y aunque en este proyecto se hace exclusivamente uso comunicación de tipo asíncrona, cabe destacar que también es posible la comunicación de forma síncrona mediante RS-232.

En esta norma, un “1” lógico también llamado “mark”, es representado por una tensión entre -3 y 25 voltios entre la línea y la masa, mientras que un “0” o “space”, corresponde a un valor de tensión entre +3 y +25 voltios. El rango de valores de tensión situado entre -3 y +3 voltios se considera indefinido.

Los datos útiles transmitidos mediante este estándar se agrupan en bloques (normalmente de 1 byte), y éstos están a su vez son contenidos en paquetes de bits más extensos también llamados “chunks” o “frames” de bits. Estos paquetes contienen además de los bits de datos que se desean transmitir, otros bits auxiliares los cuales intervienen en la correcta transmisión asíncrona de los datos. Estos son:

- **Bit de arranque (Start Bit)**

Este bit indica el inicio de un nuevo paquete de información. El bit de arranque siempre viene indicado por una línea en estado inactivo (idle) que pasa del valor lógico “1” a “0”.

- **Bit de paridad (Parity Bit)**

Es un bit utilizado para comprobar a bajo nivel que no se han producido errores en la transmisión de la información. Este bit se calcula a partir de los datos que se desean enviar. El uso de este bit no es muy común ya que implica un decrecimiento en la tasa de comunicación, y la necesidad de implementar un sistema de manejo de errores tanto en el emisor como en el receptor. Es por eso que sólo suele emplearse en ambientes muy propensos al ruido eléctrico.

- **Bits de parada (Stop Bit)**

Indica el fin de un paquete llevando el estado lógico de la línea a “1”. En la configuración de la comunicación puede también definirse si se usarán 1 o 2 bits de parada.

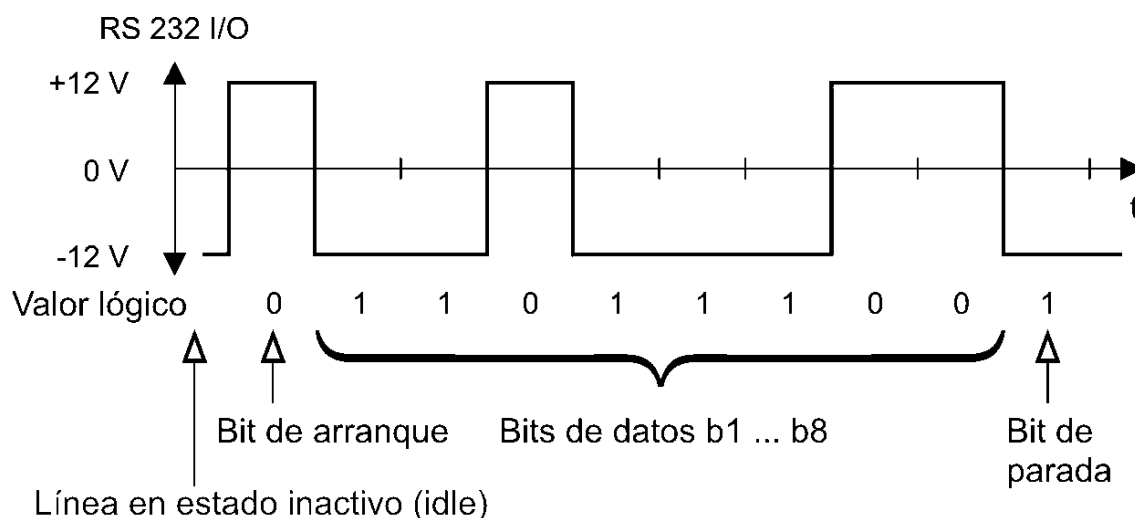


Figura 13. Transferencia de una serie temporal de bits sobre RS-232

En el gráfico expuesto sobre estas líneas, se representa la transferencia de una cadena de 8 bits de datos, precedidos por un bit de arranque, sin bit de paridad, y finalizado por un bit de parada. Esta configuración del ejemplo, es muy común en comunicaciones serie y es conocida como 8N1.

El tiempo (t) que debe mantenerse la línea en cada estado lógico para que la comunicación se ejecute satisfactoriamente, viene definido por el “baud rate”. A mayor “baud rate”, mayor será la velocidad de transmisión. Existen ciertos valores estándares como son 1200, 2400, 4800, 19200, 38400, 57600, y 115200 siendo este el límite común de velocidad en RS-232.

Por otra parte, debe tenerse en cuenta un último aspecto en este estándar, el “endianness”. Esto es, definir si el bit más significativo de la cadena es el primero o el último. Generalmente, si no se hace referencia a este parámetro, puede asumirse que la comunicación se realiza de modo que el bit menos significativo es el primero.

En adición a los circuitos de datos por los que circula la información que desea ser transferida, esta norma contempla el uso de otros circuitos auxiliares de control los cuales mediante un procedimiento de “handshaking” o “apretón de manos”, hacen posible mediante hardware controlar el flujo de datos en ambos sentidos, y operar así en modo simplex, half-duplex o full-duplex. Esto es, permitir que la comunicación se realice entre dispositivos en un solo sentido, en ambos sentidos por turnos, o en ambos sentidos de forma simultánea.

Cabe notar que el estándar original sólo soportaba comunicación half-duplex, y fue a partir de finales de los años 80 cuando se desarrolló una nueva revisión que incluía la capacidad de operar en modo full-duplex.

Por otra parte, existen dos tipos diferentes de realizar el proceso de handshaking; mediante software o hardware.

- **Handshake software**

El sistema más común de handshake por software, es utilizar los caracteres XON y XOFF (caracteres ASCII 17 y 18 respectivamente). Mediante el envío de estos caracteres por la línea de datos, el receptor puede con XON notificar al emisor que está listo para recibir los datos, mientras que con XOFF indica al emisor que debe pausar el envío de datos. Este método sólo puede ser utilizado cuando se opera en modo full-duplex.



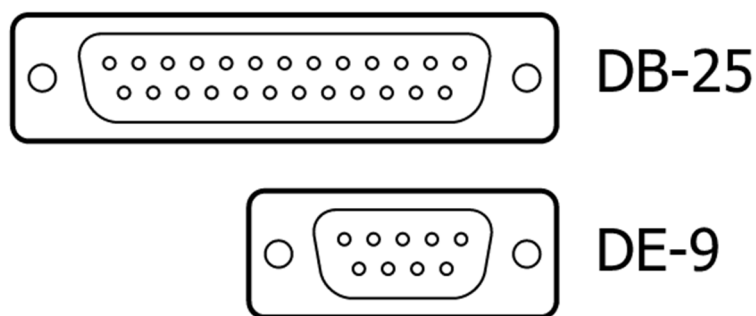
- **Handshake hardware**

En el caso del handshake hardware, se utilizan líneas de conexión extra dedicadas exclusivamente al control del flujo de la comunicación. Mediante el valor lógico atribuido a estas líneas, tanto el emisor como el receptor pueden indicar a la otra parte la disponibilidad para publicar o recibir datos.

Todos estos parámetros descritos anteriormente, deben estar configurados de la misma exacta manera en los dispositivos que se desean intercomunicar para que los datos se transmitan de forma satisfactoria.

### 4.3. Conectores y cables en RS-232

Aunque el estándar por su parte recomienda el uso del conector de 25 pines DB-25 (D-Subminiature 25-pin), al no ser esta una característica obligatoria, es común el uso del conector más compacto de nueve pines DE-9 (D-sub 9-pin).



*Figura 14. Conectores comunes RS-232*

Con respecto al cable de conexión a utilizar, el estándar indica que no se deben superar los 50 pies de longitud (unos 15 metros). Esta limitación está basada en evitar el uso de línea que superen los 2500 pF de capacitancia total, por lo tanto, siempre que no se supere esta cifra pueden utilizarse cables de mayor longitud. Por ejemplo, si se hace uso de cable UTP CAT-5 con una capacitancia típica de 17 pF/pie y despreciando la capacitancia propia de las unidades a comunicar, entonces la longitud máxima permitida para el cable será de 147 pies (casi 45 metros).

La capacitancia de la línea de conexión influirá de manera decisiva en la velocidad máxima de transmisión soportada. Según experimentos realizados por Texas Instruments, puede considerarse la siguiente tabla como relación entre la velocidad máxima en baudios por segundo y la longitud del cable de conexión:

Velocidad de transmisión (Baudios por segundo)	Longitud máxima del cable (pies)
19200	50
9600	500
4800	1000
2400	3000

*Figura 15. Tabla experimental que relaciona la distancia en pies del cable con la velocidad máxima en baudios por segundo*

#### 4.4. Señales utilizadas en la operación del estándar RS-232 para la comunicación con el analizador

Las líneas de conexión comúnmente manejadas para la comunicación sobre RS-232 son las siguientes:

Pin N° (DB-25)	Pin N° (DE-9)	Nombre y uso de la señal
2	<b>3</b>	<b>TxD – Transmitted Data</b>
3	2	<b>RxD – Received Data</b>
4	7	<b>RTS – Request To Send</b>
5	<b>8</b>	<b>CTS – Clear To Send</b>
6	6	DSR – Data Set Ready
7	5	<b>GND – Signal Ground</b>
8	1	DCD – Received Line Signal Detector (Carrier Detect)
20	4	DTR – Data Terminal Ready
22	9	RI – Ring Indicator

*Figura 16. Líneas hardware más comunes empleadas por el estándar RS-232*

En negrita han sido destacadas las señales (junto con sus correspondientes pines DE-9) que serán utilizadas en la comunicación con el dispositivo de medida.

Desde el punto de vista del dispositivo remoto que adquiere los datos provenientes del instrumento de medida, estas señales son:

#### 4.4.1. TxD – Transmitted Data

Línea mediante la cual la unidad remota envía datos en forma de serie de bits hacia el instrumento de medida.

#### 4.4.2. RxD – Received Data

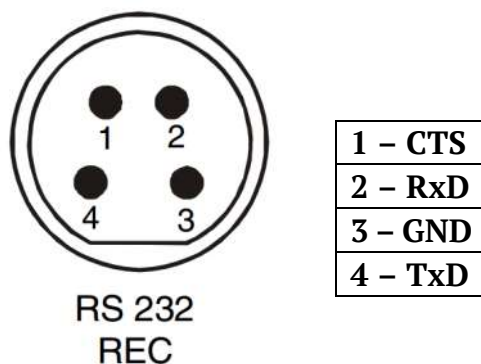
Línea mediante la cual la unidad remota recibe los datos publicados por el instrumento de medida en forma de serie de bits.

#### 4.4.3. RTS/ CTS – Request To Send/ Clear To Send

Cuando la unidad remota está lista para recibir datos, activa la línea RTS; este valor será recibido por el instrumento de medida en su línea pareja CTS, notificando a éste que puede comenzar a publicar datos a través de su línea TxD hacia la línea RxD de la unidad remota.

### 4.5. Características de la interfaz de comunicación del dispositivo de medida

El analizador multiparámetro WTW 350i dispone en su puerto serie de un conector no estandarizado de cuatro pines (similar a un conector Mini-DIN de 4 pines) con la siguiente configuración:



*Figura 17. Puerto RS-232 del analizador multiparamétrico WTW multi 350i*

<b>Cuota de transmisión (en bauds/s)</b>	Seleccionable entre: 1200, 2400, 4800, 9600, 19200
<b>Handshake</b>	RTS/CTS
<b>Paridad</b>	Sin
<b>Bits de datos</b>	8
<b>Bits de parada</b>	2

*Figura 18. Configuración RS-232 del analizador multiparamétrico WTW 350i*

La cuota de transmisión (baudrate) utilizada ha sido de 4800 bauds/s, siendo ésta la configuración por defecto del instrumento. Por tanto, el driver de comunicación de la unidad remota debe ser configurado de la siguiente forma:

<b>Cuota de transmisión (en bauds/s)</b>	4800
<b>Handshake</b>	RTS/CTS
<b>Paridad</b>	Sin
<b>Bits de datos</b>	8
<b>Bits de parada</b>	2

*Figura 19. Configuración RS-232 del módulo de comunicación serie de la unidad remota*

### 4.6. Conexión entre el analizador multiparamétrico y la unidad remota

Para realizar la conexión física entre la unidad remota (Raspberry Pi) y el instrumento de medida (WTW multi 350i) ha sido necesario el uso de dos elementos: un adaptador USB a RS-232, y un cable de conexión de DE-9F (hembra) a cuatro terminales que puedan ser enchufados en el conector RS-232 del instrumento de medida.

4.6.1. Adaptador USB a RS-232

Este elemento permite el uso del estándar RS-232 al ser conectado a uno de los puertos USB de la unidad remota. El lado USB es tipo USB Type-A, y el lado RS-232 dispone de un conector tipo DE-9M (macho).



Figura 20. Adaptador USB a RS-232

El modelo utilizado en este proyecto está basado en un chip Prolific PL-2303.

4.6.2. Cable DE-9F a terminales de conexión

Dado que no se dispone del cable original fabricado por WTW, ha sido necesaria la construcción de éste.

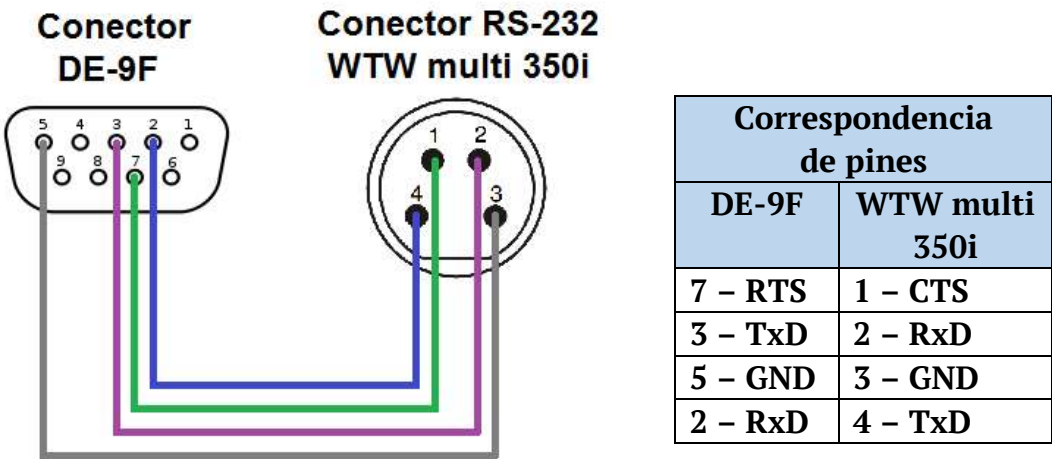


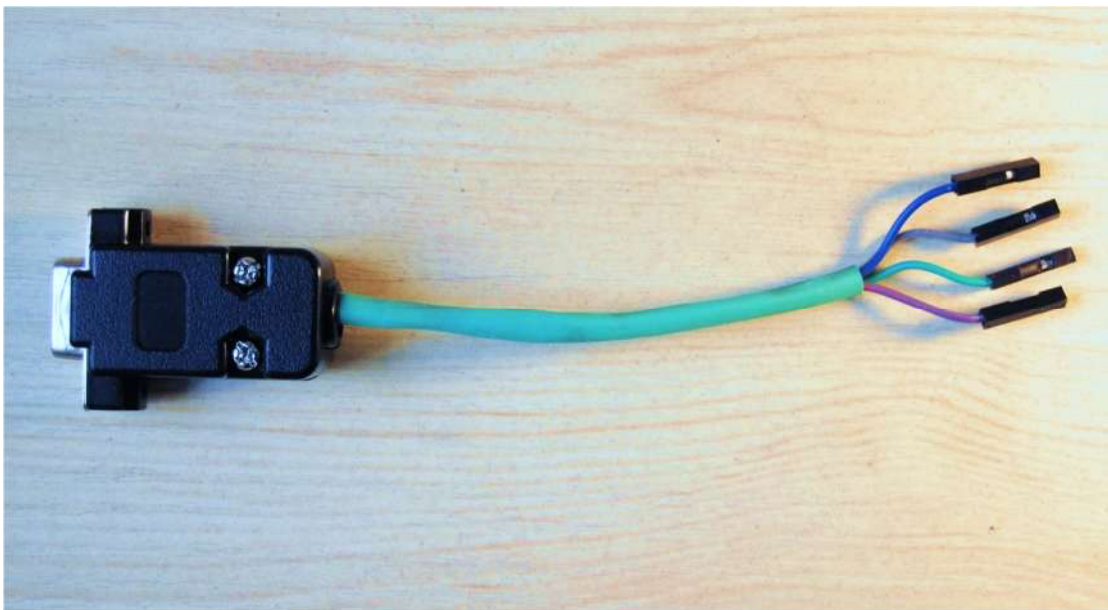
Figura 21. Diagrama de conexión

## Sistema de supervisión remota para instrumentos de medida

El cable consiste en un conector DE-9F al que le ha sido soldado en cada pin necesario un cable tipo “female jumper wire”.



Figura 22. Detalles de los conectores DE-9F y female jumper wire



*Figura 23. Cable de conexión DE-9F a terminales individuales*

## **5. La unidad terminal remota (RTU)**

### **5.1. Concepto**

La unidad terminal remota o RTU por sus siglas en inglés, consiste en un dispositivo electrónico basado en un microprocesador (comúnmente un ordenador), cuyo cometido es actuar como interfaz con los elementos físicos de la instalación local donde se encuentra integrada. Estos elementos pueden ser instrumentos de medida o control, permitiendo a su vez desde un sistema Maestro recibir datos mediante telemetría, o enviar comandos de control remoto.

En la actualidad, prácticamente cualquier actividad en la que intervengan dispositivos de medida o actuadores, puede ser monitorizada y operada de forma remota mediante un sistema SCADA. Es por esta razón que las aplicaciones y funcionalidades de las RTUs se encuentran en plena expansión.

### **5.2. Elección del dispositivo RTU**

Partiendo de la necesidad del sistema de utilizar diversas herramientas y estándares informáticos actuales, con el objetivo de sacar partido de la potencia que puede proporcionar la conectividad a internet para la transmisión de datos, y a su vez disponer de un equipo de tamaño y precio reducidos, para maximizar así el potencial acceso al sistema propuesto en este proyecto, se ha optado por el uso de un ordenador de placa única o “Single Board Computer” (SBC).

Este tipo de ordenadores disponen de todos los elementos básicos de un equipo personal como son el procesador, memoria RAM, tarjeta de video, etc., ya incluidos en una única placa base. Estos dispositivos están disponibles en el mercado desde hace años para su uso en aplicaciones industriales como ordenadores embebidos, pero ha sido en los últimos años cuando se han hecho populares para el gran público debido al abaratamiento de las tecnologías que permiten alcanzar el grado de integración necesario para construir uno de estos equipos.

#### **5.2.1. Requisitos del ordenador de la RTU**

La oferta actual de equipos de placa única es extensa, y cada vez más fabricantes incluyen en sus catálogos nuevos modelos para competir en la demanda de estos dispositivos día a día más extendidos, ya que debido a su

relación precio, potencia y flexibilidad, resultan ideales para aplicaciones enfocadas al concepto del “Internet de las cosas”.

Las características básicas con las que debe contar el mini ordenador a utilizar en el desarrollo del proyecto son las siguientes:

- **Funciones de Red**

La placa debe integrar al menos un Puerto Ethernet RJ-45 que permita conectar por cable la unidad a un router o switch de red para proveer al dispositivo de acceso a internet. Asimismo, dispondrá de módulo Wi-Fi integrado, o en su defecto deberá soportar la inclusión de un adaptador Wi-Fi USB para permitir el acceso a internet sin cables.

- **S.O. Linux**

Siendo Linux el sistema operativo de código libre por antonomasia, no se han contrastado otras alternativas (como BSD) en vista de la inconmensurable cantidad de aplicaciones y documentación disponible para este SO.

Por otra parte, como se ha descrito en puntos anteriores, el sistema hará uso en su ejecución de aplicaciones incluidas de forma nativa en la mayoría de distribuciones Linux. Es por ello que el mini ordenador seleccionado debe ser suficientemente potente para trabajar con este sistema operativo.

- **Entradas y salidas**

El dispositivo debe contar con al menos 8 pines I/O digitales: 6 para controlar la pantalla de información del dispositivo basada en el controlador Hitachi HD44780, un pin para el pulsador de apagado, y un pin más para permitir incluir un LED indicador.

- **Tamaño**

Aunque durante el proceso de elección de la placa las dimensiones de ésta no debían cumplir ningún requerimiento estricto, se ha valorado que el tamaño sea reducido con el fin de obtener un terminal remoto que sea fácil de almacenar y transportar.



- **Precio**

Dado que el objetivo del proyecto es crear una plataforma destinada a equipos de investigación en laboratorios universitarios o independientes, se ha buscado en todo momento contener el presupuesto del proyecto.

### 5.2.2. Alternativas consideradas

Partiendo de los requerimientos indicados en el punto anterior, y tras considerar diversas opciones de mini ordenadores para acometer el proyecto, la elección quedó finalmente acotada entre las dos placas más populares en la actualidad. Estas son: la Beagle Bone Black y la célebre Raspberry Pi.

Desarrollada por Texas Instruments, la Beagle Bone Black cuenta con un potente procesador ARM Cortex-A8 a 1 GHz y con 512 MB de RAM DDR3. Es una de las placas más potentes del mercado en cuanto a capacidad procesamiento y versatilidad de conexiones ya que dispone de buses I2C, CAN, SPI, múltiples entradas analógicas, etc. Por otro lado, y al igual que sucede con la Raspberry Pi, al no integrar conectividad Wi-Fi resulta imprescindible conectar un adaptador Wi-Fi USB que ocuparía el único puerto USB del que dispone. Ante esta limitación, resulta inevitable el uso de un concentrador USB para poder hacer un uso simultáneo del conversor RS232-USB necesario para comunicar con los instrumentos de medida. Ya que esta placa dispone de hasta seis puertos serie con lógica TTL, esta necesidad bien puede solventarse mediante el uso de un adaptador TTL-RS232. El precio de venta al público de este dispositivo se sitúa en torno a los 52€.

Repasando ahora las características técnicas del mini ordenador Raspberry Pi, éste se basa en un procesador ARM 1176JZF-S a 700 MHz. En su versión B, dispone de hasta 512 MB de memoria SDRAM, 2 puertos USB, y entre sus periféricos de bajo nivel cuenta con 8 GPIO, un UART y buses I2C y SPI.

La placa Raspberry Pi model B, más una tarjeta de memoria SD de 8GB necesaria para su uso con la distribución Linux Raspbian preinstalada, puede adquirirse por un precio inferior a 30€.

Aunque de rendimiento y conectividad objetivamente inferiores a la Beagle Bone Black, dispone de capacidad suficiente para ejecutar todos los procesos que serán necesarios para la captación, procesamiento y publicación de datos así como la gestión de la pantalla de información rápida.

Continuando con el enfoque aplicado al proyecto en el que se aspira a que el sistema pueda ser adaptado para virtualmente comunicar con cualquier instrumento de medida utilizando un protocolo serie, debe tomarse en consideración la extensa comunidad de usuarios con la que cuenta la plataforma Raspberry Pi. Debido a la proverbial popularidad de este mini ordenador, para casi cualquier estándar de comunicación, o nueva rutina que desee implementarse en el sistema, puede encontrarse documentación así como foros muy activos en los que es posible resolver, o hallar orientación en caso de bloqueos en el desarrollo. Además, a causa de su notoriedad, la alternativa Raspberry Pi dispone de un granado entorno de fabricantes que continuamente incluyen en sus catálogos nuevos productos destinados a extender y mejorar las posibles aplicaciones de esta familia de ordenadores de placa única.

Por otra parte, cabe notar que en el primer trimestre del año 2015 Raspberry Pi Foundation lanzó la segunda generación de su exitoso modelo B, la Raspberry Pi 2 Model B. Entre las mejoras más destacables de esta actualización, se incluyen un nuevo procesador ARM Cortex-A7 Quad-Core a 900 Mhz, 1 GB de RAM, 4 puertos USB, así como hasta 40 pines GPIO. Aun así, el fabricante ha indicado que esta nueva revisión es totalmente compatible con la anterior, y que además será distribuida al mismo precio que su versión predecesora.

Es por todo ello que la alternativa Raspberry Pi se presenta como la opción mejor posicionada en cuanto a continuidad a medio o largo plazo y compatibilidad entre versiones de la placa. Así que todo ello, sumado a que el precio de venta de la placa Beagle Bone Black es superior al de la Raspberry Pi, hacen que definitivamente se ha optado por la elección de la Raspberry Pi como placa a utilizar en el desarrollo del proyecto.



*Figura 24. Mini ordenador Raspberry Pi model B.*

### **5.3. Concepto y diseño de los módulos de operación de la RTU**

El funcionamiento de la unidad remota es gestionado completamente a través de lenguaje de programación interpretado. Continuando con el fundamento principal del proyecto el cual persigue que tanto el uso, aprendizaje y modificación de todos los elementos del proyecto sean lo más sencillo posible, se ha optado en todo momento por el uso de código script. De esta forma, se agiliza enormemente el proceso de codificación, depuración y ejecución de programas en el sistema al eludir la necesidad de compilar, apoyando así el posterior desarrollo de módulos que aporten nuevas funcionalidades al sistema.

- **Linux Shell Scripts**

Un Shell de Unix consiste en un intérprete de comandos el cual proporciona al usuario una interfaz mediante la cual realizar operaciones sobre un sistema operativo Unix, en el caso de este proyecto, Linux. Estos comandos pueden asimismo estructurarse en un fichero de texto plano que puede ser

posteriormente ejecutado de forma automática al arrancar el equipo o mediante una tarea programada por ejemplo. Comúnmente los archivos script en Linux suele usar la extensión “.sh”.

- **Python**

Python es un lenguaje interpretado de alto nivel el cual ha experimentado un auge creciente en los últimos años debido a su potencia, flexibilidad y claridad de su sintaxis. Distribuido bajo licencia GNU (General Public License), cuenta con una extensa y activa comunidad de desarrolladores compuesta tanto por programadores profesionales como aficionados, por lo que es sencillo encontrar ejemplos así como soporte para casi cualquier aplicación que se desee desarrollar. Es por ello que este lenguaje de programación entronca directamente con la filosofía hacia la que ha sido orientada la plataforma desarrollada en este proyecto. Los archivos script de Python son identificados mediante la extensión “.py”.

```
sudo apt-get install python-rpi.gpio
```

*Figura 25. Comando de instalación de la biblioteca RPi GPIO*

El intérprete Python viene instalado de serie en el sistema operativo Raspbian. Concretamente para el alcance de este proyecto, sólo es necesario el uso de una biblioteca extra que igualmente viene preinstalada en las versiones más recientes de este SO. Esta biblioteca es *RPi GPIO*, y es la encargada de facilitar el acceso al interfaz GPIO de la Raspberry.

Si esta no se encontrase por defecto en el sistema, puede instalarse fácilmente mediante el comando:

## 5.4. Descripción y funcionamiento de los módulos de la RTU

Los módulos en los que se basa el funcionamiento de la unidad remota son los siguientes:

### 5.4.1. Módulo de carga de la configuración “conf.sh”

Obtiene a través de SCP los archivos de configuración del Servidor Principal y los sitúa en las rutas necesarias para que se apliquen los nuevos parámetros de la red Wi-Fi a la que se desea conectar, las características de la conexión serie del instrumento de medida, o los requisitos de la publicación de datos al Servidor Principal.

El comando básico que utiliza este módulo es la copia de ficheros desde el Servidor Principal.

### 5.4.2. Módulo de comunicación serie “minicom.sh”

Este Bash Script contiene los parámetros obtenidos del Servidor Principal necesarios para configurar la aplicación Minicom encargada de establecer la comunicación serie con el instrumento de medida WTW multi 350i.

La aplicación de código libre Minicom es un programa basado en texto para la emulación de un terminal serie en Linux.

```
# Comunicación serie sobre el puerto USB0 a 4800 baud/s
# Capturar los datos en el fichero /home/pi/WTW_COMM/raw_wtw
sudo minicom -b 4800 -D /dev/ttyUSB0 -o -C /home/pi/WTW_COMM/raw_wtw
```

*Figura 26. Contenido básico del script minicom.sh*

### 5.4.3. Módulo de copia de datos en el Servidor Principal “datos\_tr.sh”

Es el encargado de copiar en el Servidor Principal mediante el servicio SCP los datos obtenidos por Minicom desde el instrumento de medida. Este Script comprueba en cada ciclo el tamaño del fichero de datos capturados por Minicom y en función de su dimensión en bytes realiza una de las siguientes acciones:

- **Copia de datos en tiempo real en el Servidor Principal**

A medida que los datos son captados por Minicom, el módulo datos\_tr.sh procede a copiar en el Servidor Principal los registros recogidos desde el instrumento de medida. De esta forma se permite la monitorización remota en tiempo real del dispositivo.

- **Almacenamiento de datos históricos en el Servidor Principal**

Una vez que el archivo de datos recogidos en tiempo real por Minicom supera un cierto tamaño en bytes, el módulo `datos_tr.sh` genera internamente un fichero copia a partir de éste, el cual queda identificado mediante su fecha y hora de creación. Acto seguido, este módulo reinicia el fichero de datos en el que Minicom registra la información capturada. Luego almacena en el Servidor Principal mediante copia SCP el fichero cerrado con su fecha y hora, para su futuro uso como registro histórico de datos obtenidos del instrumento de medida.

Mediante este procedimiento se cumplen dos objetivos: mantener en todo momento un registro histórico de medidas en el Servidor Principal, y optimizar la transferencia de información mediante SCP al poder fijar un tamaño máximo para los ficheros transmitidos.

Cabe destacar que el valor en bytes que disparará el almacenamiento y reinicio del fichero de datos, puede ser configurado según se desee en el módulo `datos_tr.sh`.

#### **5.4.4. Módulo Watchdog “`watchdog.sh`”**

De forma periódica este Script Linux genera un archivo de texto plano con información relativa al estado del sistema RTU, para acto seguido transferirlo al Servidor Principal. Gracias a este módulo, es posible conocer de forma remota el estado de funcionamiento del terminal y su rendimiento. Algunos de los datos publicados son un listado de redes Wi-Fi cercanas, la IP WAN y LAN del dispositivo, tiempo del sistema en marcha, carga promedio del procesador, etc. Tanto la velocidad de refresco como la información enviada, son totalmente configurables en este Script.

#### **5.4.5. Módulo de control del pulsador de apagado “`haltb.py`”**

Se trata de un Script Python el cual mediante una operación de poll a la interfaz GPIO, comprueba el estado del pulsador de apagado del sistema. Si dicho botón se mantiene pulsado durante un cierto número de segundos, lanzará un comando de sistema que permite apagar el dispositivo de forma segura.

#### 5.4.6. Módulo de gestión de la pantalla “status.py”

Consiste en un programa escrito en código Python el cual se encarga de la gestión del display de 4x20 caracteres compatible con el Standard Hitachi HD44780. El control de esta pantalla se realiza mediante el uso de una biblioteca específica para este estándar. Este recurso se puede obtener de forma libre en la web del fabricante Adafruit.

De forma periódica, este Script comprueba el estado de la red y lee los registros temporales obtenidos de la unidad de medida para mostrar esta información por pantalla.

### 5.5. Arranque automático del sistema de la RTU

Utilizando el servicio de arranque automático de la distribución Linux Raspbian para Raspberry Pi, el dispositivo es capaz de arrancar de forma secuencial los módulos de operación anteriormente descritos. Para ello, en el fichero *rc.local* albergado en la ruta del sistema */etc* deben estar contenidos los comandos que desean ser ejecutados al inicio del sistema operativo.

```
printf "display..."

python /home/pi/Adafruit-Raspberry-Pi-Python-
Code/Adafruit_CharLCD/status.py &

printf "boton halt..."

python /home/pi/Adafruit-Raspberry-Pi-Python-
Code/Adafruit_CharLCD/haltb.py &

sleep 5

printf "conf remota..."

sh /home/juan/conf.sh &

sleep 5

printf "watchdog..."

sh /home/juan/watchdog.sh &

printf "datos tr..."

sh /home/juan/datos_tr.sh &

printf "com serie..."

sh /home/pi/WTW_COMM/minicom.sh

exit 0
```

*Figura 27. Extracto del archivo /etc/rc.local de la RTU*

De igual forma, pueden incluirse tiempos de espera entre comandos destinados a asegurar que los Scripts y aplicaciones arranquen de debidamente de forma secuencial.

Como nota al pie, cabe mencionarse que los Scripts Python se inician anteponiendo *python* a la ruta del Script que se desea ejecutar. Para el caso de los Bash Scripts de Linux, se debe usar *sh*.

## 5.6. Gestión de usuarios del sistema de la RTU

Obsérvese en el extracto del Script de arranque del sistema expuesto en el apartado anterior que los diferentes módulos de operación han sido albergados en dos rutas de usuario diferentes: */home/pi* y */home/juan*.

La primera, correspondiente al usuario “pi”, contiene los programas que gestionan las actividades locales como son: “status.py”, “haltb.py” y “minicom.sh”. En el caso de que se construyesen e integrasen más RTUs en la plataforma de supervisión presentada en este proyecto, este usuario estará presente por defecto en todas las unidades remotas basadas en la placa Raspberry Pi.

Por el contrario, el segundo usuario del sistema “juan”, ha sido creado para identificar de manera única e inequívoca a la RTU presentada como prototipo de la prueba de concepto. Los Scripts contenidos en la ruta */home* de este usuario, son los destinados al intercambio de información con el Servidor Principal. Estos son: *conf.sh*, *watchdog.sh* y *datos\_tr.sh*.

En el código de los Scripts alojados en la ruta */home/juan*, se hace uso de forma explícita del nombre de usuario “juan” identificativo de la RTU y de la dirección IP del Servidor Principal. A continuación se expone un ejemplo aclaratorio:

```
sudo scp /home/juan/watchdog.log juan@tar:/home/juan/watchdog.log
```

*Figura 28. Extracto de código del módulo watchdog.sh*

Con el comando SCP sobre estas líneas, se ejecuta la copia del fichero de información sobre la remota en el Servidor Principal. Este comando consta de dos parámetros clave: la dirección IP del Servidor Principal y el nombre del usuario que realiza la acción.



- **Dirección IP del Servidor Principal**

En la prueba de concepto presentada en este proyecto, el Servidor Principal se encuentra situado en el departamento de Tratamiento de Aguas Residuales de la Escuela Politécnica Superior de la Universidad de Sevilla, y es por ello que se ha creado una entrada en el fichero */etc/hosts* del sistema operativo Linux de la RTU, que permite utilizar el nemónico “tar” en lugar de la IP WAN estática asignada a este equipo.

- **Nombre de usuario**

Como se apuntaba anteriormente, en la RTU existirán dos usuarios diferentes: un usuario genérico “pi”, y un usuario único que identifique la RTU en cuestión. En el caso del usuario único, este debe existir con el mismo nombre en el sistema que actúe como Servidor Principal. De esta forma se podrá mediante un proceso de clave SSH pre-compartida (PSK), autenticar cada unidad remota de forma segura e inequívoca en el Servidor Principal.

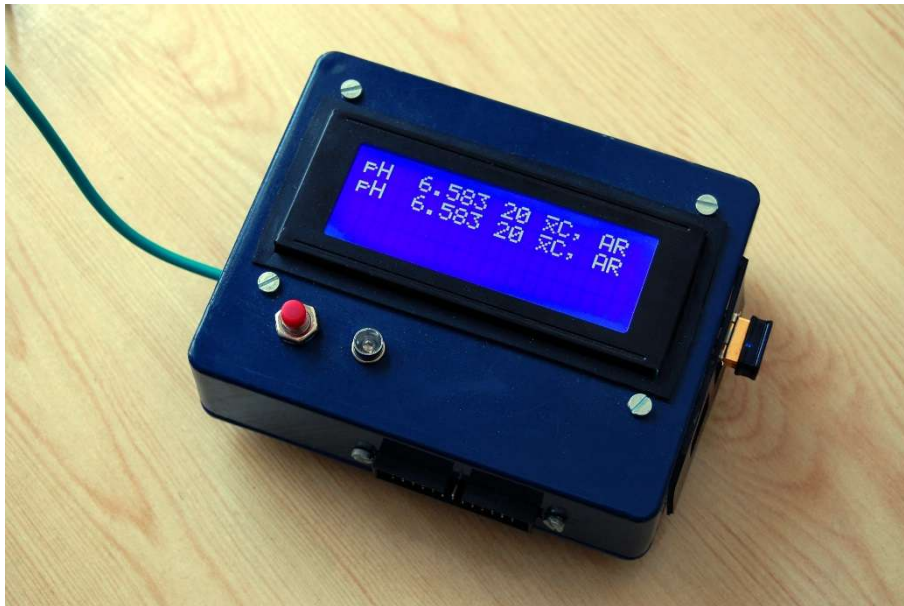
El objetivo de esta medida es que cada RTU disponga de un directorio único y exclusivo en el Servidor Principal. En el ejemplo de la figura 28, se observa que el módulo “watchdog.sh” copia el fichero “watchdog.log” del directorio local */home/juan*, al directorio del Servidor Principal */home/juan*; ninguna otra RTU podrá acceder a este directorio.

De esta manera se asegura la escalabilidad del sistema sin comprometer la organización y estructura de la información, o la seguridad del acceso al Servidor Principal.

## 5.7. Construcción de la RTU

### 5.7.1. Objetivo

Continuando con la filosofía aplicada en el proyecto global en la que se ha perseguido en todo momento brindar al usuario una plataforma sencilla, económica y robusta, se ha hecho especial hincapié en la aplicación de estos términos a la hora de diseñar la unidad remota física.



*Figura 29. Unidad RTU*

### 5.7.2. Elementos de la RTU

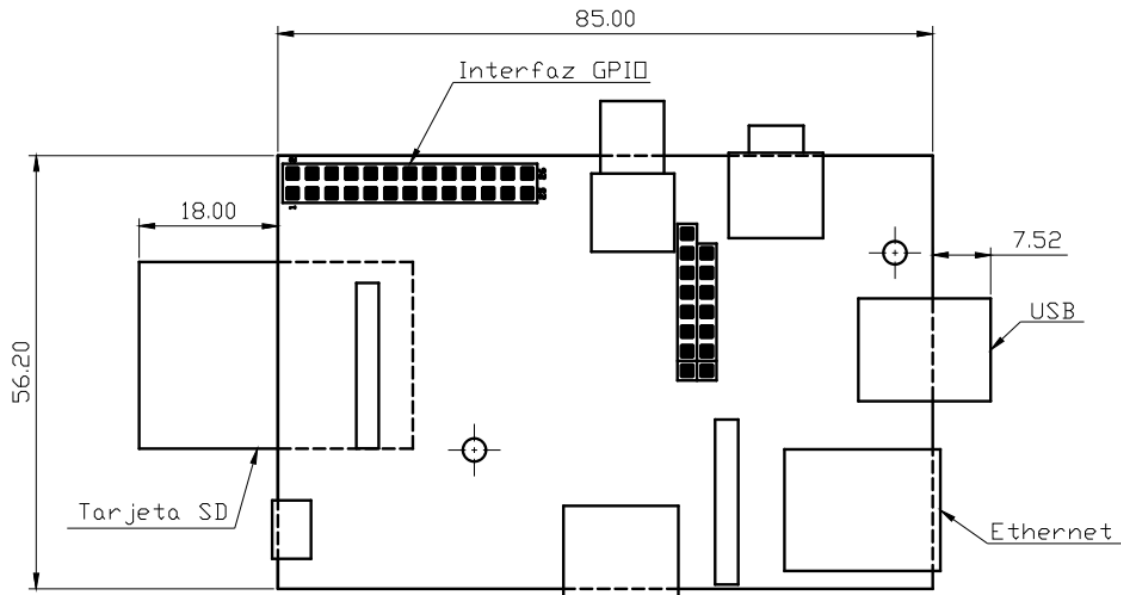
La unidad remota consta de los siguientes elementos básicos:

- **Raspberry Pi model B**

Como ya se trató en el punto “5.2 Elección del dispositivo RTU”, la unidad remota está basada en el ordenador de placa única Raspberry Pi Model B. Esta placa se basa en un procesador ARM 1176JZF-S a 700 MHz, con 512 MB de SDRAM y dispone a su vez de una tarjeta de 8GB como almacenamiento inter<sup>o</sup>no.

## Sistema de supervisión remota para instrumentos de medida

La alimentación del mini ordenador se realiza mediante un conector micro-USB. La tensión necesaria es de 5 V DC con un consumo de corriente medio de 700 mA.



*Figura 30. Placa Raspberry Pi*

Los puertos de entrada y salida de los que se ha hecho uso en este proyecto son:

- **Puerto Ethernet 10/100 con clavija RJ-45**

A través de este puerto es posible conectar el equipo a internet para transmitir los datos obtenidos del instrumento de medida, o a una red local con el fin de administrar la configuración de la RTU de forma manual.

- **Dos puerto USB 2.0**

Destinados a la conexión de los adaptadores Wi-Fi y USB-RS232

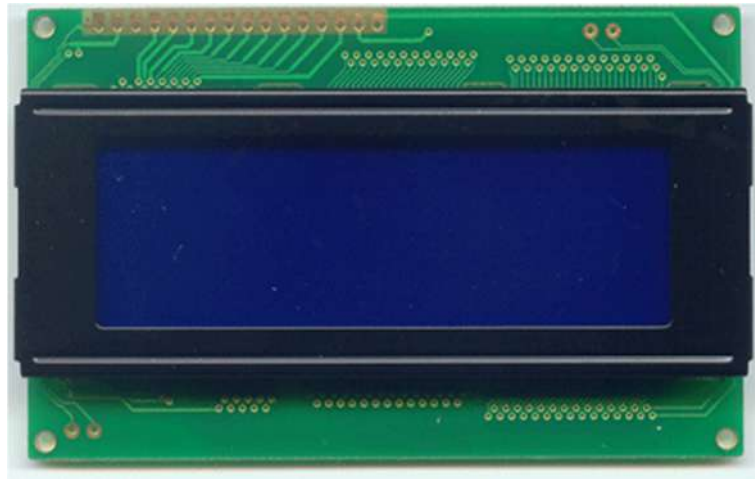
- **8 pines GPIO**

Se hará uso de los 8 terminales GPIO (general purpose input/output) disponibles como medio de conexión con la pantalla, el pulsador de apagado y el led de estado.

- **Pantalla**

El controlador compatible con el estándar industrial Hitachi HD44780 integrado en la placa de la pantalla, es el encargado de gestionar la matriz de puntos que compone los hasta 20x4 caracteres que pueden ser representados.

Como se ha indicado anteriormente, esta pantalla será controlada a través de 6 pines de la interfaz GPIO de la Raspberry Pi. Ver documento detallado de conexiones en el anexo.



*Fig 31. Pantalla LCD de matriz de puntos retroiluminada*

- **Adaptador Wi-Fi USB**

Mediante este dispositivo será posible conectar la RTU a una red inalámbrica Wi-Fi. El modelo seleccionado es el EW-7811Un del fabricante Edimax. Cabe notar que el sistema operativo Raspbian incluye los drivers necesarios para este dispositivo.



*Figura 32. Adaptador Wi-Fi USB Edimax EW-7811Un*

- **Cargador micro USB**

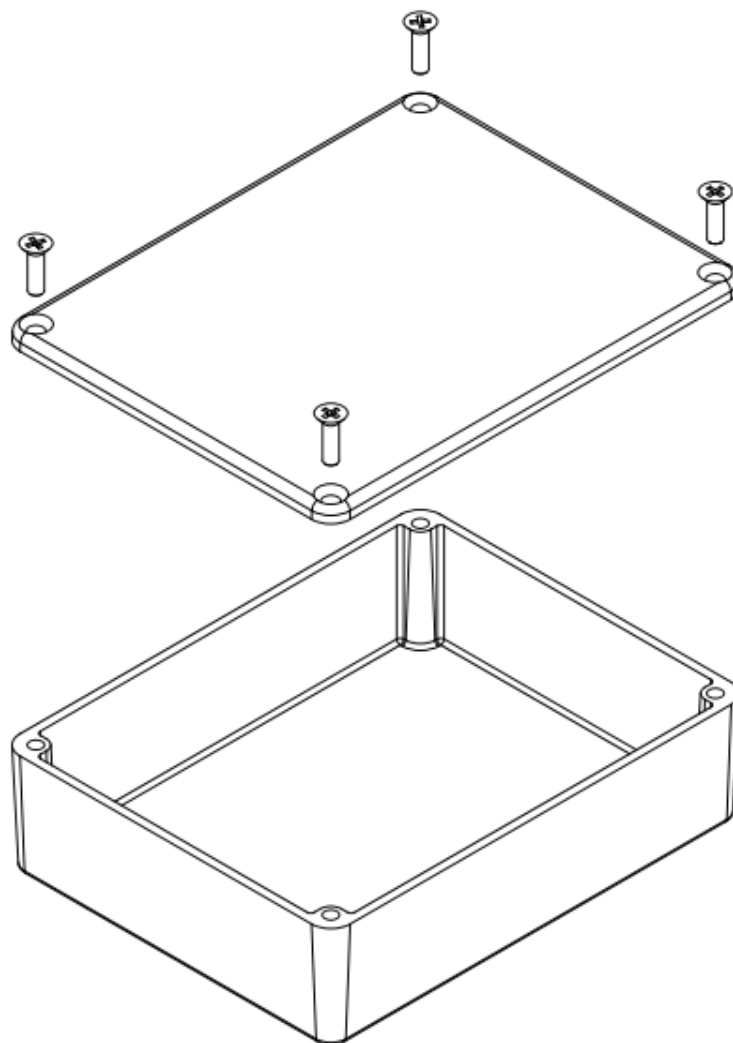
Un cargador micro USB proporciona la alimentación a la unidad RTU. Se recomienda el uso de un dispositivo de buena calidad y con una capacidad suficiente para entregar una corriente máxima de salida de al menos 1 A. La unidad remota puede asimismo ser alimentada mediante un PC o una batería portátil, siempre que estos cumplan los requisitos mínimos en cuanto a potencia necesaria.



*Figura 33. Cargador Micro USB*

- **Carcasa**

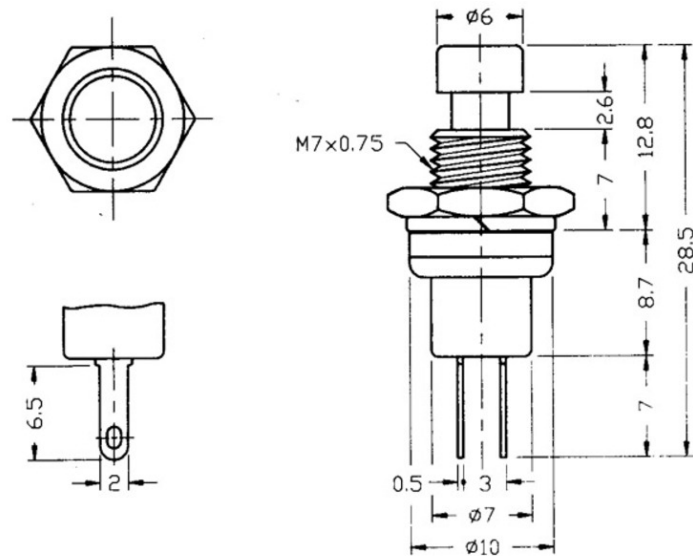
La carcasa elegida para contener el equipo ha sido el modelo 1590BB del fabricante Hammond. Está fabricada en aluminio, finalizada con un pintado electrostático y cuenta con cuatro tornillos para el cierre de la tapa. Esta robusta caja está especialmente destinada a aplicaciones electrónicas, ha sido mecanizada para albergar todos los componentes necesarios de la RTU. Las dimensiones exteriores aproximadas de esta carcasa son 120 x 94 x 34 mm. Más especificaciones de la carcasa pueden encontrarse en el anexo.



*Fig 34. Carcasa Hammond 1590BB*

- **Pulsador momentáneo**

Pulsador unipolar modelo PB-11.



*Figura 35. Pulsador momentáneo PB-11*

- **LED y porta-LED**

Un led rojo de alta intensidad de 3 mm actuará como indicador auxiliar del estado de la RTU. Dicho led está alojado en una montura para led de 3 mm.



*Figura 36. Led y montura de éste.*

- **Otros**

- **Cableado**

Cables de tipo “female jumper wire” han sido utilizados para el conexionado de la pantalla, led y pulsador.



*Figura 37. Faja de cables tipo “female jumper wire”*

- **Resistencias**

Tres resistores individuales han sido empleados en el montaje de la RTU: 2 unidades en el cableado de la pantalla LCD para el ajuste del contraste de ésta, y una más para la polarización del LED de estado.

- **GPIO Breakout**

Un conector externo para la interfaz GPIO ha sido incluido en la carcasa. Nótese que este no se encuentra habilitado.





*Fig 38. Conector externo GPIO*

– **Separadores**

Separadores hexagonales metálicos roscados macho/hembra de 5mm de altura M3 x M3, han sido empleados para elevar la placa Raspberry sobre la tapa de la carcasa.



*Fig 39. Separadores hexagonales metálicos*

– **Patas de goma**

Cuatro unidades mantienen la carcasa elevada de la superficie donde se apoya evitando además que esta se deslice.



*Fig 40. Patas de goma*

– **Tornillos**

Tornillos zincados M3 x 10mm de cabeza cilíndrica han sido utilizados para fijar la placa, pantalla, conector GPIO externo y patas de goma a la caja.

## 5.8 Funcionamiento del terminal

La sencillez a la hora de comenzar a obtener y publicar datos del instrumento de medida hacia el Servidor Principal, ha sido uno de los factores que se han primado en el desarrollo de esta herramienta.

Basta con disponer de una toma de corriente y de conexión a internet en el lugar donde se deseen capturar los datos publicados por el dispositivo de medida. El único requisito que debe cumplir para la primera conexión a internet es tener activado el servicio DHCP el cual permitirá asignar una IP de forma dinámica al equipo.

Una vez conectado a internet a través de cable ethernet, de forma automática el dispositivo cargará utilizando el protocolo de transferencia de archivos SCP, el fichero de configuración previamente definido y almacenado en el Servidor Principal el cual contiene los parámetros de configuración que permitirán al dispositivo, por ejemplo, conectarse a una red WiFi móvil creada expresamente para permitir la transferencia de los datos en lugares donde no se dispone de una conexión fija a internet. Esta nueva configuración se aplicará tras reiniciar la RTU.

Una vez conectado a internet, sólo será necesario iniciar la publicación de datos por parte del instrumento de medida. De forma autónoma, La RTU comenzará a transferir de forma periódica estos datos al servidor central.

### 5.8.1. Pantalla de la RTU

La pantalla del terminal ha sido implementada de forma que pueda conocerse en todo momento el estado de funcionamiento del dispositivo. De forma cíclica se mostrará la siguiente información:

- **Hora y fecha**

Debido a que la placa Raspberry Pi no cuenta de serie con un reloj hardware interno, una vez conectado de forma satisfactoria a internet, el sistema operativo del terminal obtendrá la hora y fecha exacta utilizando el servicio NTP (Network Time Protocol). Dado que el sistema basa su funcionamiento en el

acceso a internet, no supone ningún problema la ausencia de reloj hardware. El servidor de hora utilizado es [debian.pool.ntp.org](http://debian.pool.ntp.org).



*Figura 41. Fecha y hora mostradas por pantalla*

La fecha y hora mostrada en pantalla es por tanto la misma que utiliza el sistema para cualquier otro proceso que ejecute. Es por esta razón que es posible detectar incidencias en la actualización de datos en el servidor principal atendiendo a la hora mostrada por el terminal.

- **Estado de conexión**

Gracias a la capacidad DHCP (Dynamic Host Configuration Protocol) del dispositivo, y si el punto de acceso a internet (Router, Wi-Fi móvil 3G, etc.) permite asignar direcciones IP LAN (Local Area Network) de forma dinámica, este obtendrá de forma automática una IP local la cual se mostrará en pantalla indicando al usuario que la conexión a la red local (LAN) se ha realizado de forma satisfactoria.

Si el acceso a internet se encuentra habilitado, mostrará asimismo en pantalla la IP WAN pública asignada por el ISP (Internet Service Provider).



*Figura 42. Ejemplo de pantalla cuando la conexión a internet ha sido establecida satisfactoriamente*

Si la conexión local se encuentra establecida, pero la RTU no dispone de acceso a internet, el campo WAN (Wide Area Network) aparecerá vacío.



*Figura 43. Ejemplo de pantalla cuando la conexión local ha sido establecida satisfactoriamente, pero la RTU no tiene acceso a internet*

En el caso de que no se haya establecido conexión LAN, y por tanto no haya acceso a una red WAN, estos campos (WAN y LAN) aparecerán vacíos en la pantalla.



*Figura 44. Ejemplo de pantalla cuando no ha sido posible establecer una conexión LAN.*

En el caso de que no se establezca la conexión, será necesario revisar el estado de la red local, verificar que el servicio de internet está disponible y por último comprobar los parámetros de conexión a la red establecidos en la RTU.

- **Ultimos datos registrados**

De forma cíclica la pantalla mostrará los últimos datos adquiridos desde el instrumento de medida. De esta forma, el usuario puede comprobar fácilmente si la transferencia de datos se ha iniciado o si existe algún problema en la comunicación con el instrumento de medida.

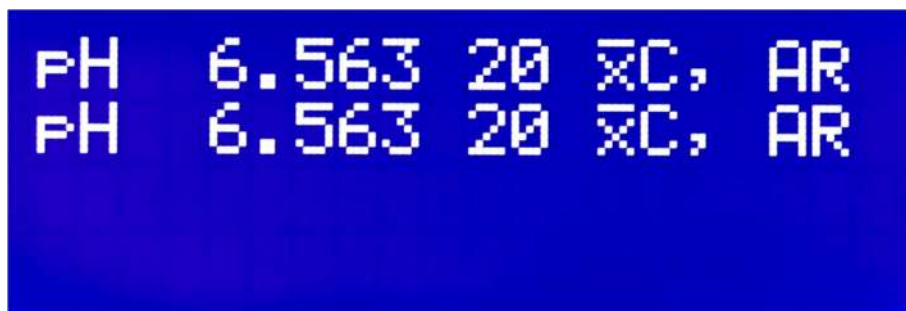


Figura 45. Ejemplo de pantalla en la que aparecen los últimos datos publicados por el instrumento de medida.

Al arrancar el sistema, este de forma automática elimina todos los registros de datos recibidos del instrumento de medida en la sesión anterior. Por ello, si desde la última vez que se inició la RTU no se ha recibido ningún dato desde el instrumento de medida, durante el ciclo de la pantalla correspondiente a “Últimos datos registrados” aparecerá un mensaje con el formato “INICIO: Fecha-hora” indicando el momento en el que la RTU arrancó el módulo de comunicación con el instrumento de medida. En este caso, debe verificarse que la conexión física con el instrumento de medida es correcta y que la publicación de datos por parte del mismo se encuentra habilitada.

- **Botón y led de apagado**

Prosiguiendo con el enfoque de desarrollo que ha llevado a buscar la mayor simplicidad en el uso del dispositivo, se ha dotado a éste de un único botón. Si se pulsa este botón durante 5 segundos, el led de estado parpadeará rápidamente indicando que el dispositivo se está apagando y puede ser desconectado de la alimentación pasados unos instantes. Para encender el dispositivo, basta con reponer la alimentación.

### 5.8.2. Administración local de la RTU

Si fuese necesario acceder a la RTU con el fin de resolver alguna incidencia o si se desea administrar el dispositivo de forma manual, puede realizarse una conexión directa a la unidad remota desde un equipo situado en la misma red local (LAN) mediante protocolo SSH.

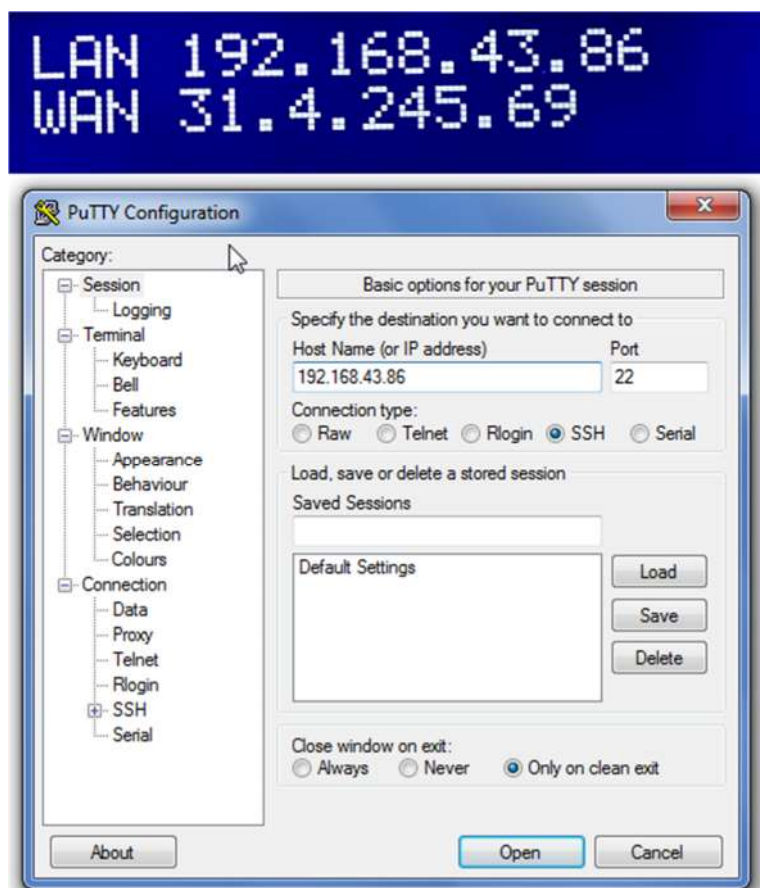
Nótese que también es posible acceder a la RTU desde fuera de su misma red local, pero en ese caso sería necesario realizar ciertas configuraciones en el punto de acceso a internet (p. ej. un router Wi-Fi) de la unidad. Dicha configuración no se encuentra contemplada en el alcance de este proyecto.

- **Conexión al ordenador de la RTU mediante protocolo SSH**

Para acceder desde un PC a la consola Linux de la unidad remota ambos equipos deben estar conectados a la misma red LAN.

Una vez conectados a la misma red LAN, basta con iniciar una sesión con un cliente SSH introduciendo en “Port” el puerto 22, en el campo de dirección del servidor (Host Name), la IP LAN que aparece en la pantalla de la RTU y pulsar “Open”. Una vez se haya accedido a la unidad remota, el administrador tendrá acceso completo a toda la configuración de la RTU.

En el ejemplo expuesto a continuación se ha utilizado el cliente de código libre Putty para Windows.



*Figura 46. Acceso a la RTU mediante el cliente SSH Putty*

## 6. El Servidor Principal

### 6.1. Concepto

La finalidad fundamental del Servidor Principal es la de ejercer como punto de centralización de las medidas en tiempo real e históricas publicadas por los instrumentos de medida a través de las RTUs. Sin embargo, aparte de actuar como almacenamiento pasivo de información, el Servidor Principal también debe cumplir otros roles en el funcionamiento del sistema presentado en este proyecto. Estos son: dar servicio a una plataforma web que permita supervisar los datos desde cualquier equipo conectado a internet, y por otro lado, actuar como servidor de configuración de las unidades remotas.

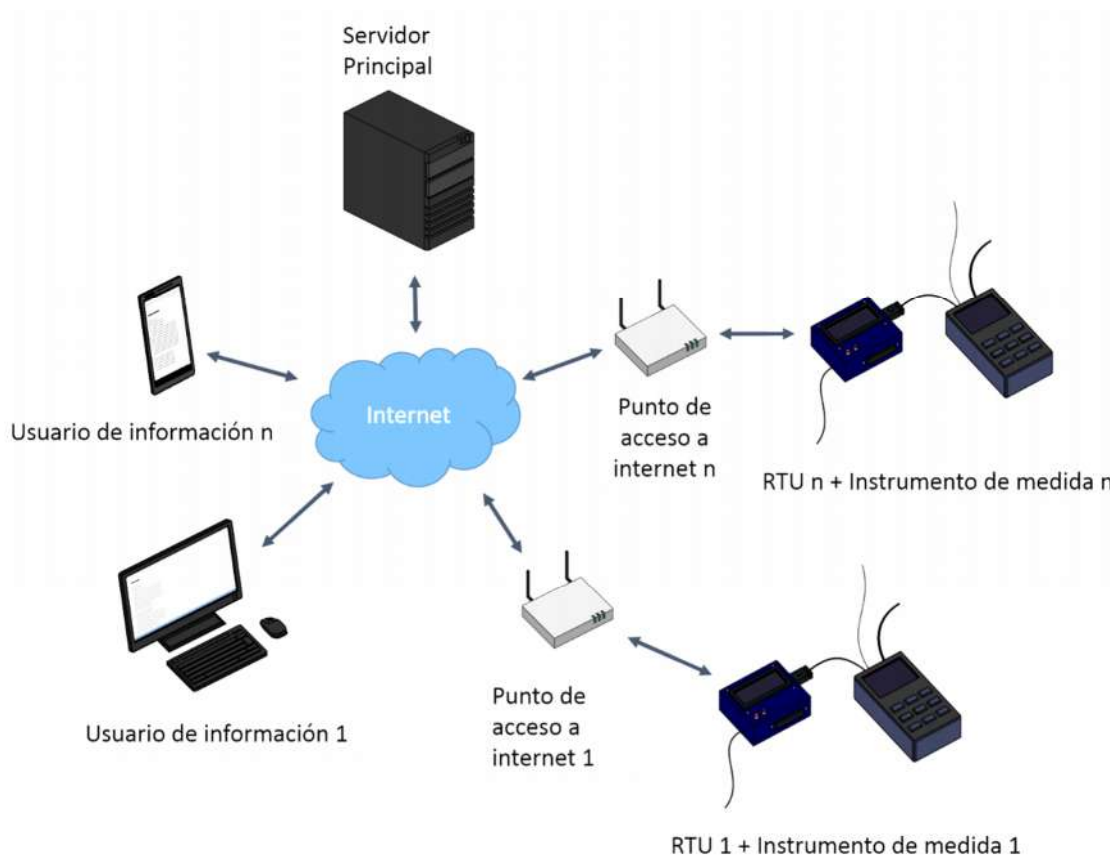


Figura 47. Diagrama conceptual de la plataforma

## 6.2. Requisitos del Servidor Principal

Gracias a las características de la plataforma presentada en este proyecto, prácticamente cualquier equipo PC puede ser apto para ejercer como Servidor Principal del sistema. Los requisitos básicos para este equipo son los siguientes:

### 6.2.1. Sistema operativo Linux

Continuando con el enfoque aplicado a la totalidad del proyecto, y por las mismas razones que ya se expusieron en el punto “5.2.1. Requisitos del ordenador de la RTU”, Linux ha sido el sistema operativo elegido para el Servidor Principal.

Concretamente, en esta prueba de concepto se utiliza la distribución de Linux “Ubuntu 14.04.2 LTS”. Las siglas LTS aluden a “Long Term Support”, indicando que se trata de una versión del S.O. a la cual los desarrolladores darán soporte a largo plazo. De esta forma, puede plantearse el funcionamiento futuro del Servidor Principal sobre esta misma versión del sistema operativo.

### 6.2.2. Hardware

Gracias al uso del sistema operativo Ubuntu, resulta posible que las necesidades hardware que debe cumplir el equipo Servidor Principal sean muy discretas.

Según documentación oficial de Ubuntu, los requisitos mínimos de la última versión de este S.O. son: procesador a 700 MHz, 512 MB de memoria RAM, y 5 GB de espacio libre en disco duro. Es por ello que el equipo instalado para actuar como Servidor Principal, supera con creces estos requisitos con la siguiente configuración: procesador Intel® Core™2 Duo 2 a 2.20 GHz, 3 GB de RAM y disco duro de 500 GB de espacio. De esta forma, será posible aumentar el número de unidades RTU sin afectar al rendimiento general del equipo; únicamente será necesario plantear el aumento del disco duro en función de las necesidades de almacenar la información a largo plazo.

### 6.2.3. Software

Contar con un servidor web HTTP es el único requisito software que debe cumplir el Servidor Principal. Para ello, y continuando con el enfoque aplicado a la totalidad del proyecto, se ha optado por utilizar un servidor web de código



abierto, y con una comunidad de usuarios amplia y activa. La opción seleccionada ha sido el servidor web Apache Tomcat 7.

### 6.2.4. Características de red

La necesidad de que el equipo Servidor Principal cuente con una IP pública estática es un requisito indispensable para el funcionamiento de la plataforma.

La red local a la que se conecte el equipo que actuará como Servidor Principal debe permitir, al menos, el re-direccionamiento (Port Forwarding) hacia este equipo de una conexión realizada a la IP pública estática del mismo sobre el puerto TCP 22 (Puerto SSH/SCP). De esta forma, la RTU será capaz de ejecutar contra el Servidor Principal los comandos SCP necesarios para recibir y enviar información.

Si no fuese posible disponer de una IP pública estática para el Servidor Principal, puede optarse por el uso de un servicio de redirección dinámico como DynDNS.

En el caso del Servidor Principal utilizado en el desarrollo de este proyecto, el cual se encuentra ubicado en el departamento TAR de la Escuela Politécnica Superior, ha sido posible asignar una IP pública estática proporcionada por la Universidad de Sevilla. Dicha IP ha sido incluida en el fichero */etc/hosts* de la RTU como “tar” (véase el punto “5.6. Gestión de usuarios del sistema de la RTU”).

## 6.3. Módulos del Servidor Principal

Únicamente son necesarios dos módulos en el Servidor Principal para proporcionar el servicio de consulta de datos en tiempo real a través de un explorador web. Estos son:

- **Módulo de copia de datos para la web “web\_tr.sh”**

Como se indicaba en el capítulo “5.6. Gestión de usuarios del sistema de la RTU”, toda la información referente a la RTU “juan” será almacenada en el Servidor Principal en la ruta */home/juan*. De esta manera, los datos en tiempo real publicados por esta RTU se encontrarán en el fichero */home/juan/RT\_DATA*.

El Script Linux “web\_tr.sh” lee mediante un comando *tail* los últimos registros almacenados en el fichero */home/juan/RT\_DATA*, y los copia en un

nuevo fichero llamado *data\_juan* en el directorio */tomcat7/webapps/ROOT*. De esta forma, a la hora de mostrar los datos a través de un explorador, la aplicación web destinada a mostrar los datos de la RTU “juan”, cargará este fichero mostrando únicamente los datos de esta unidad remota.

En el caso de que se integrasen más unidades remotas, cada una de ellas en su correspondiente usuario en la ruta */home/usuario*, contará con su propio Script “web\_tr.sh” que transferirá los datos pertinentes hasta el directorio del proyecto web.

- **Módulo web “juan.html”**

Se trata de una aplicación web sencilla escrita en lenguaje html. Cuando se accede a la aplicación mediante un explorador web, esta lee periódicamente el fichero */tomcat7/webapps/ROOT/data\_juan*, mostrando en tiempo real los datos enviados desde la RTU denominada “juan”.

Al igual que en el punto anterior, por cada remota integrada en la plataforma, existirá una aplicación web del tipo “usuario.html”. De esta forma, un cliente de información podrá acceder a los datos de la remota que desee simplemente escribiendo en su navegador web la dirección “ip\_servidor\_principal:8080/usuario”.

## 6.4. Funcionamiento del Servidor Principal

El Servidor Principal tal y como ha sido implementado en este proyecto, ofrece tres servicios: web de datos en tiempo real, acceso a datos históricos y configuración de la remota.

- **Web de datos en tiempo real**

Consiste en una web accesible desde cualquier explorador web, PC o móvil, que permite monitorizar en tiempo real los datos obtenidos del instrumento de medida conectado una RTU.



Figura 48. Web de datos en tiempo real

- **Acceso a los datos históricos de medidas**

Mediante un cliente SCP de código libre como WinSCP para Windows, es posible de manera sencilla conectarse al Servidor Principal, y acceder a la ruta */home/usuario/STORE\_DATA/*. En este directorio se encontrarán los ficheros históricos almacenados según el patrón *datos\_fecha\_hora*.

- **Servidor de configuración**

Al igual que en el apartado anterior, basta con utilizar WinSCP para cargar en el directorio */home/usuario/CONF/* los archivos de configuración que el módulo “conf.sh” de la RTU debe gestionar.

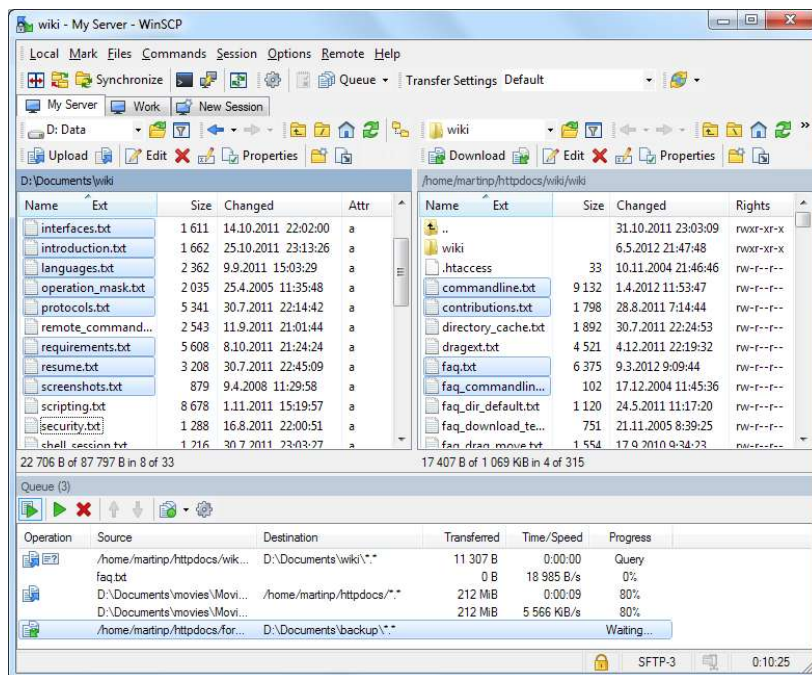


Figura 49. Captura de pantalla de la aplicación WinSCP

## 7. Mejoras Futuras

En próximas versiones de la RTU, será posible emplear la nueva placa presentada recientemente por Raspberry Pi Foundation: la Raspberry Pi 2 Model B. Esta nueva placa aparte de incorporar un procesador más potente, y una memoria RAM mayor, cuenta con 40 pines GPIO. Gracias a esta última característica, será posible conectar una placa controladora de relés que permitirá mediante el desarrollo de un nuevo módulo, controlar de forma remota bombas, calentadores, válvulas, etc. aportando un gran valor añadido a la plataforma presentada en este proyecto.

## 8. Bibliografía

[www.wtw.de](http://www.wtw.de)

[www.ubuntu.com](http://www.ubuntu.com)

[www.onsetcomp.com](http://www.onsetcomp.com)

[www.raspberrypi.org](http://www.raspberrypi.org)

[www.rtaautomation.com](http://www.rtaautomation.com)

[www.nullmodem.com](http://www.nullmodem.com)

[www.linux.org](http://www.linux.org)

[www.stackoverflow.com](http://www.stackoverflow.com)

[www.opcfoundation](http://www.opcfoundation)

[www.adafruit.com](http://www.adafruit.com)

[www.beagleboard.org](http://www.beagleboard.org)

[www.python.org](http://www.python.org)

[www.modbus.org](http://www.modbus.org)

[www.w3schools.com](http://www.w3schools.com)

[www.symplymodbus.ca](http://www.symplymodbus.ca)

[www.modbusdriver.com](http://www.modbusdriver.com)

[www.mayor.de](http://www.mayor.de)

[www.sparkfun.com](http://www.sparkfun.com)

[www.lammertbies.nl](http://www.lammertbies.nl)

[www.ti.com](http://www.ti.com)

[www.tomcat.apache.org](http://www.tomcat.apache.org)