

**DISEÑO E IMPLEMENTACION DE UN MICROSERVIDOR WEB PARA EL
MONITOREO DE VARIABLES FISICAS EN EL LABORATORIO DE CIENCIA Y
TECNOLOGIA DE MATERIALES CERAMICOS DE LA UNIVERSIDAD DEL
CAUCA**



**JUAN MANUEL CAICEDO GALLEGO
VICTOR MANUEL GOMEZ HOLGUÍN**

**UNIVERSIDAD DEL CAUCA
FACULTAD DE CIENCIAS NATURALES, EXACTAS Y DE LA EDUCACIÓN
GRUPO I + D EN INGENIERIA FÍSICA
INGENIERÍA FÍSICA
POPAYÁN
2009**

**DISEÑO E IMPLEMENTACION DE UN MICROSERVIDOR WEB PARA EL
MONITOREO DE VARIABLES FISICAS EN EL LABORATORIO DE CIENCIA Y
TECNOLOGIA DE MATERIALES CERAMICOS DE LA UNIVERSIDAD DEL
CAUCA**

**JUAN MANUEL CAICEDO GALLEGO
VICTOR MANUEL GOMEZ HOLGUÍN**

Proyecto de grado para optar al título de Ingeniero Físico

**Director
Ing. Diego Alberto Bravo M.**

**UNIVERSIDAD DEL CAUCA
FACULTAD DE CIENCIAS NATURALES, EXACTAS Y DE LA EDUCACIÓN
GRUPO I + D EN INGENIERIA FÍSICA
INGENIERIA FÍSICA
POPAYÁN
2009**

NOTA DE ACEPTACION

Director _____

Ing. Diego Alberto Bravo M.

Jurado _____

Ing. Germán A. Bacca

Jurado _____

Mg. Jorge W. Coronel G.

Fecha de sustentación: Popayán, 18 de Diciembre de 2009

AGRADECIMIENTOS

Los autores manifiestan sus agradecimientos

A mi FAMILIA, de manera muy especial a mi Madre ANA BEIBA HOLGUIN por su apoyo, preocupación, sobre todo mucho AMOR y que hace tres años desde el cielo me sigue acompañando. A mis hermanos y hermanas quienes con su consejo, amor y comprensión incondicional durante todo este tiempo me ayudaron a llevar este proyecto a feliz término. A mis compañeros de trabajo Juan Manuel Caicedo y José Cabrera, así como también al Ing. Diego Bravo y a todas aquellas personas que me apoyaron durante mi carrera.

Victor Manuel Gómez Holguín

A mi madre y a mi hijo y a todos aquellos que nos colaboraron desinteresadamente en este proyecto.

Juan Manuel Caicedo Gallego

TABLA DE CONTENIDO

	Pag
ÍNDICE DE GRÁFICAS	7
1. RESUMEN	8
2. INTRODUCCIÓN	9
3. SERVIDORES WEB	10
3.1 EL MODELO OSI	11
3.1.1. CAPA FÍSICA (CAPA 1)	13
3.1.2. CAPA DE ENLACE DE DATOS (CAPA 2)	16
3.1.3. CAPA DE RED (CAPA 3)	16
3.1.4. CAPA DE TRANSPORTE (CAPA 4)	16
3.1.5. CAPA DE SESIÓN (CAPA 5)	17
3.1.6. CAPA DE PRESENTACIÓN (CAPA 6)	18
3.1.7. CAPA DE APLICACIÓN (CAPA 7)	18
3.1.8. UNIDADES DE DATOS	19
3.1.9. TRANSMISIÓN DE LOS DATOS	20
3.1.10. FORMATO DE LOS DATOS	21
3.1.11. OPERACIONES SOBRE LOS DATOS	22
3.2. PROTOCOLO HTTP	22
3.2.1. TRANSACCIONES HTTP	23
3.2.2. VERSIONES	23
3.3. HTML	24
3.3.1. MARCADO HTML	24
3.3.2. ATRIBUTOS	25
3.4. JAVASCRIPT	26
3.5. JOOMLA!	27
4. MICROSERVIDOR WEB	28
4.1. EL PIC18F4520	30
4.2. EL ENC28J60	31
4.3. CONECTOR RJ45	33

5. SENSORES E INTERFACES EMPLEADAS.....	35
5.1. SENSOR DE TEMPERATURA LM35.....	35
5.2. SENSOR DE PRESIÓN MPXM2102	35
5.3. CÁMARA IPCAM SECURE 300R.....	36
6. DISEÑO DEL HARDWARE	38
6.1. TARJETA PRINCIPAL.....	38
6.2. TARJETA DE TEMPERATURA.....	40
6.3. TARJETA DE PRESIÓN	41
6.4. ETAPA DE POTENCIA	42
7. DISEÑO DEL FIRMWARE	44
7.1. SOFTWARE EMBEBIDO	44
7.1. DESARROLLO DEL SITIO WEB	46
8. ANÁLISIS Y RESULTADOS	49
9. APLICACIONES	51
10. CONCLUSIONES	52
11. BIBLIOGRAFÍA.....	54
12. REFERENCIAS BIBLIOGRÁFICAS	55

ÍNDICE DE GRÁFICAS

	Pag
Imagen 1 – Pila del Modelo OSI con sus Siete Capas	13
Imagen 2 - Transferencia de Información del Modelo OSI	20
Imagen 3 - Formato de datos en el modelo OSI.....	21
Imagen 4 - Diagrama de bloques del Sistema desarrollado dentro de la Red de Datos de UNICAUCA	29
Imagen 5 - Diagrama Circuitual del PIC18F4520	30
Imagen 6 - Diagrama Circuitual del ENC28J60	32
Imagen 7 - Board del Conector RJ45	34
Imagen 8 - Diagrama de Conexiones del Conector RJ45.....	34
Imagen 9 - Sensor de Temperatura LM35.....	35
Imagen 10 - Sensor de Presión MPXM2102.....	35
Imagen 11 - Gráfico de Presión Contra Voltaje de Salida del MPXM2102	36
Imagen 12 - Cámara IPCAM SECURE 300R.....	37
Imagen 13 - Tarjeta Principal del Microservidor	38
Imagen 14 - Diagrama Circuitual del Buffer 74HCT245.....	39
Imagen 15 - Fotografía de la Tarjeta Principal con sus componentes soldados	39
Imagen 16 - Detalle del Disipador de Calor Instalado.....	40
Imagen 17 - Tarjeta del Sensor de Temperatura	40
Imagen 18 - Diagrama Circuitual de la Tarjeta de Temperatuta	41
Imagen 19 - Diagrama Circuitual de la Tarjeta de Presión	42
Imagen 20 - Fotografía de la Etapa de Potencia	42
Imagen 21 - Diagrama de Flujo del Firmware	45
Imagen 22 - Detalle de la página Web con el microservidor web funcionando	47
Imagen 23 - Detalle de la cámara y la interfaz del microservidor.....	48

1. RESUMEN

Con base en el empleo de tecnología de microcontroladores PIC, se desarrolló una plataforma electrónica de monitoreo y control, que utiliza como medio de transmisión y recepción de datos y señales las redes Ethernet e Internet.

Se trata de un dispositivo denominado "Microservidor", que posee conectividad a redes cableadas 10/100 a través de su puerto RJ45. Su unidad central de procesamiento, es un microcontrolador, que alberga tanto al sistema operativo del dispositivo, como a la página web desarrollada en HTML y JavaScript que sirve de interfaz de usuario del sistema.

Inicialmente concebido para operar en el interior del laboratorio de Ciencia y Tecnología de Materiales Cerámicos de la Universidad del Cauca, el sistema desarrollado, es una poderosa y versátil herramienta, que sirve como base para el desarrollo de innumerables proyectos de monitoreo y control remoto de variables físicas, en y desde cualquier lugar del mundo donde haya acceso a internet.

2. INTRODUCCIÓN

Es una realidad ineludible, que la evolución de internet ha permitido interconectar al mundo, más allá de su concepción inicial en la que solo los computadores eran los protagonistas del más influyente invento tecnológico de la humanidad en todos los tiempos. La “interconectividad”ⁱ ha sobrepasado todas las barreras y se ha inmiscuido en la gran mayoría de los electrodomésticos, equipos de oficina e instrumentos industriales de uso diario. Desde los más avanzados sistemas de seguridad, con cámaras IP, pasando por dispositivos de metrología industrial, hasta llegar a los videojuegos, y los aparatos de la cocina, hoy en día pueden ser conectados a la “súper autopista de la información”ⁱⁱ para interactuar con el mundo entero.

El presente trabajo de grado, expone el diseño e implementación de una plataforma electrónica, que permite monitorear y controlar desde internet, prácticamenteⁱⁱⁱ cualquier sensor comercial conectado al sistema, a través de una página web.

Este dispositivo desarrollado con tecnología de punta en el campo de los microcontroladores, permitirá el desarrollo de gran cantidad de sistemas de monitoreo y control de variables físicas en cualquier campo donde se pueda garantizar el acceso a internet.

A pesar de que inicialmente el proyecto surgió como solución a una necesidad particular del laboratorio de Ciencia Y Tecnología de Materiales Cerámicos de la Universidad del Cauca, el desarrollo dio a luz, a un sistema de monitoreo y control multipropósito que desde ya, abre las puertas de la adquisición y manipulación remota de variables físicas de cualquier índole a través de Internet.

3. SERVIDORES WEB

Un Servidor Web, es un software que implementa el protocolo de comunicaciones HTTP^{iv}, el cual pertenece a la capa de aplicación del modelo OSI^v y está diseñado para transferir lo que llamamos hipertextos, páginas web o páginas HTML^{vi}: textos complejos con enlaces, figuras, formularios, botones y objetos incrustados como animaciones o reproductores de música.

Sin embargo, el hecho de que HTTP y HTML estén íntimamente ligados no debe dar lugar a confundir ambos términos. HTML es un lenguaje de marcas, que al compilarse genera un documento “navegable” y HTTP es un protocolo de comunicaciones.

El Servidor, es un programa que se ejecuta continuamente en un ordenador (aunque también se emplea el término *servidor* para referirse al ordenador que lo ejecuta), manteniéndose a la espera de peticiones por parte de un cliente (generalmente un *navegador web*) y que responde a estas peticiones adecuadamente, mediante una *página web* que se exhibirá en el navegador o mostrando el respectivo mensaje si se detectó algún error.

El servidor responde al cliente enviando el código HTML de la página; el cliente, una vez recibido el código, lo interpreta y lo exhibe en pantalla. El cliente es el encargado de interpretar el código HTML, es decir, de mostrar las fuentes, los colores y la disposición de los textos y objetos de la página; el servidor tan sólo se limita a transferir el código de la página sin llevar a cabo ninguna interpretación de la misma.

Sobre el servicio web *clásico* podemos disponer de aplicaciones web. Éstas son porciones de código que se ejecutan cuando se realizan ciertas peticiones o respuestas HTTP. Hay que distinguir entre:

- **Aplicaciones en el lado del cliente:** el cliente web es el encargado de ejecutarlas en la máquina del usuario. Son las aplicaciones tipo *Java* o *Javascript*. el servidor proporciona el código de las aplicaciones al cliente y éste, mediante el navegador, las ejecuta. Es necesario, por tanto, que el cliente disponga de un navegador con capacidad para ejecutar aplicaciones (también llamadas *scripts*). Comúnmente, los navegadores permiten ejecutar aplicaciones escritas en lenguaje *javascript* y *java*, aunque pueden añadirse más lenguajes mediante el uso de *plugins*.
- **Aplicaciones en el lado del servidor:** el servidor web ejecuta la aplicación; ésta, una vez ejecutada, genera cierto código HTML; el servidor

toma este código recién creado y lo envía al cliente por medio del protocolo HTTP.

Las aplicaciones de servidor muchas veces suelen ser la mejor opción para realizar aplicaciones web. La razón es que, al ejecutarse ésta en el servidor y no en la máquina del cliente, éste no necesita ninguna capacidad añadida, como sí ocurre en el caso de querer ejecutar aplicaciones *javascript* o *java*. Así pues, cualquier cliente dotado de un navegador web básico puede utilizar este tipo de aplicaciones.

3.1 EL MODELO OSI

El **modelo de referencia de Interconexión de Sistemas Abiertos** (*OSI, Open System Interconnection*) fue el modelo de red descriptivo creado por la ISO^{vii} lanzado en 1984. Es decir, fue un marco de referencia para la definición de arquitecturas de interconexión de sistemas de comunicaciones.

A principios de la década de 1980, el desarrollo de redes sucedió con desorden en muchos sentidos. Se produjo un enorme crecimiento en la cantidad y el tamaño de las redes. A medida que las empresas tomaron conciencia de las ventajas de usar tecnologías de conexión, las redes se agregaban o expandían a casi la misma velocidad a la que se introducían las nuevas tecnologías de red.

Para mediados de la década de 1980, estas empresas comenzaron a sufrir las consecuencias de la rápida expansión. De la misma forma en que las personas que no hablan un mismo idioma tienen dificultades para comunicarse, las redes que utilizaban diferentes especificaciones e implementaciones tenían dificultades para intercambiar información. El mismo problema surgía con las empresas que desarrollaban tecnologías de conexiones privadas o propietarias. "Propietario" significa que una sola empresa o un pequeño grupo de empresas controlan todo uso de la tecnología. Las tecnologías de conexión que respetaban reglas propietarias en forma estricta no podían comunicarse con tecnologías que usaban reglas propietarias diferentes.

Para enfrentar el problema de incompatibilidad de redes, la Organización Internacional para la Estandarización (ISO) investigó modelos de conexión como la red de *Digital Equipment Corporation* (DECnet), la Arquitectura de Sistemas de Red (SNA) y TCP/IP a fin de encontrar un conjunto de reglas aplicables de forma general a todas las redes. Con base en esta investigación, la ISO desarrolló un modelo de red que ayuda a los fabricantes a crear redes que sean compatibles con otras redes.

Siguiendo el esquema de este modelo se crearon numerosos protocolos, por ejemplo X.25^{viii}, que durante muchos años ocuparon el centro de la escena de las comunicaciones informáticas. El advenimiento de protocolos más flexibles donde las capas no están tan demarcadas y la correspondencia con los niveles no era tan clara puso a este esquema en un segundo plano. Sin embargo es muy usado

en la enseñanza como una manera de mostrar la forma en que puede estructurarse una "pila" de protocolos de comunicaciones.

El modelo en sí mismo no puede ser considerado una arquitectura, ya que no especifica el protocolo que debe ser usado en cada capa, sino que suele hablarse de modelo de referencia. Este modelo está dividido en siete capas:



Imagen 1 – Pila del Modelo OSI con sus Siete Capas

3.1.1. CAPA FÍSICA (CAPA 1)

La Capa Física del modelo de referencia OSI es la que se encarga de las conexiones físicas de un computador hacia la red, tanto en lo que se refiere al

medio físico (**medios guiados**: cable coaxial, cable de par trenzado, fibra óptica y otros tipos de cables; **medios no guiados**: radio, infrarrojos, microondas, láser y otras redes inalámbricas); características del medio (por ejemplo: tipo de cable o calidad del mismo; tipo de conectores normalizados o en su caso tipo de antena; etc.) y la forma en la que se transmite la información (codificación de señal, niveles de tensión/intensidad de corriente eléctrica, modulación, tasa binaria, etc.)

Esta capa es la encargada de transmitir los bits de información a través del medio utilizado para la transmisión. Se ocupa de las propiedades físicas y características eléctricas de los diversos componentes; de la velocidad de transmisión, si ésta es unidireccional o bidireccional (*síplex*, *dúplex* o *full-dúplex*). También de aspectos mecánicos de las conexiones y terminales, incluyendo la interpretación de las señales eléctricas/electromagnéticas.

Se encarga de transformar una trama de datos proveniente del “nivel de enlace” en una señal adecuada al medio físico utilizado en la transmisión. Estos impulsos pueden ser eléctricos (transmisión por cable) o electromagnéticos (transmisión inalámbrica). Estos últimos, dependiendo de la frecuencia y longitud de onda de la señal pueden ser: ópticos, de micro-ondas o de radio. Cuando actúa en modo recepción el trabajo es inverso; se encarga de transformar la señal transmitida en tramas de datos binarios que serán entregados al nivel de enlace.

Sus principales funciones se pueden resumir como:

- Definir el medio o medios físicos por los que va a viajar la comunicación: cable de pares trenzados (o no, como en RS232/EIA232), coaxial, guías de onda, aire, fibra óptica.
- Definir las características materiales (componentes y conectores mecánicos) y eléctricas (niveles de tensión) que se van a usar en la transmisión de los datos por los medios físicos.
- Definir las características funcionales de la interfaz (establecimiento, mantenimiento y liberación del enlace físico).
- Transmitir el flujo de bits a través del medio.
- Manejar las señales eléctricas/electromagnéticas
- Especificar cables, conectores y componentes de interfaz con el medio de transmisión, polos en un enchufe, etc.
- Garantizar la conexión (aunque no la fiabilidad de ésta).

3.1.1.1. CODIFICACIÓN DE LA SEÑAL

El nivel físico recibe una trama binaria que debe convertir a una señal eléctrica, electromagnética u otra dependiendo del medio, de tal forma que a pesar de la

degradación que pueda sufrir en el medio de transmisión, vuelva a ser interpretable correctamente en el receptor.

En el caso más sencillo, el medio es directamente digital, como en el caso de las fibras ópticas, dado que por ellas se transmiten pulsos de luz.

Cuando el medio no es digital hay que codificar la señal. En los casos más sencillos la codificación puede ser por pulsos de tensión PCM^{ix}. Otros medios se codifican mediante presencia o ausencia de corriente. En general estas codificaciones son muy simples y no usan bien la capacidad de medio. Cuando se quiere sacar más partido al medio se usan técnicas de modulación más complejas, y suelen ser muy dependientes de las características del medio concreto.

En los casos más complejos, como suelen ser las comunicaciones inalámbricas, se pueden dar modulaciones muy sofisticadas, este es el caso de los estándares Wi-Fi, en el que se utiliza codificación *OFDM*^x.

3.1.1.2. TOPOLOGÍA Y MEDIOS COMPARTIDOS

Indirectamente, el tipo de conexión que se haga en la capa física puede influir en el diseño de la capa de Enlace. Atendiendo al número de equipos que comparten un medio hay dos posibilidades:

- **Conexiones punto a punto:** que se establecen entre dos equipos y que no admiten ser compartidas por terceros.
- **Conexiones multipunto:** en la que más de dos equipos pueden usar el medio.

Así por ejemplo la fibra óptica no permite fácilmente conexiones multipunto y por el contrario las conexiones inalámbricas son inherentemente multipunto (a excepción de los enlaces infrarrojo). Hay topologías por ejemplo la de “anillo”, que permiten conectar muchas máquinas a partir de una serie de conexiones punto a punto (Directa entre dos máquinas).

3.1.1.3. EQUIPOS ADICIONALES

A la hora de diseñar una red hay equipos adicionales que pueden funcionar a nivel físico, se trata de los **repetidores**, que en esencia son equipos que amplifican la señal, pudiendo también regenerarla. En las redes Ethernet con la opción de cableado de par trenzado, se emplean equipos de interconexión llamados concentradores (repetidores en las redes 10Base-2) más conocidos por su nombre en inglés **hubs** que convierten una topología física en estrella en un “**bus lógico**” y que actúan exclusivamente a nivel físico, a diferencia de los conmutadores (**switches**) que actúan a nivel de enlace.

3.1.2. CAPA DE ENLACE DE DATOS (CAPA 2)

Cualquier medio de transmisión debe ser capaz de proporcionar una transmisión sin errores, es decir, un tránsito de datos fiable a través de un enlace físico. Debe crear y reconocer los límites de las tramas, así como resolver los problemas derivados del deterioro, pérdida o duplicidad de las tramas. También puede incluir algún mecanismo de regulación del tráfico que evite la saturación de un receptor que sea más lento que el emisor.

La capa de enlace de datos se ocupa del direccionamiento físico, de la topología de la red, del acceso a la red, de la notificación de errores, de la distribución ordenada de tramas y del control del flujo.

Se hace un direccionamiento de los datos en la red, ya sea en la distribución adecuada desde un emisor a un receptor, la notificación de errores, o de la topología de la red de cualquier tipo. La tarjeta NIC^{xi}, Tarjeta de Interfaz de Red en español o simplemente Tarjeta de Red, que se encarga de que exista conexión. Posee una dirección MAC^{xii} (control de acceso al medio) y la LLC (Control de Enlace Lógico).

Los *Switches* o conmutadores, realizan su función en esta capa.

3.1.3. CAPA DE RED (CAPA 3)

El cometido de la capa de red es hacer que los datos lleguen desde el origen al destino, aun cuando ambos no estén conectados directamente. Los dispositivos que facilitan tal tarea se denominan en “enrutadores”, aunque es más frecuente encontrar el nombre inglés “*routers*” y, en ocasiones “encaminadores”.

Adicionalmente la capa de red lleva un control de la “congestión de red”, que es el fenómeno que se produce cuando una saturación de un nodo tira abajo toda la red (similar a un atasco en un cruce importante en una ciudad grande). La Unidad de Datos de Protocolo (PDU) de la capa de red es el “paquete”.

Los *routers* trabajan en esta capa, aunque pueden actuar como *switch* de nivel 2 en determinados casos, dependiendo de la función que se le asigne. Los *firewalls* (cortafuegos) actúan sobre esta capa principalmente, para descartar direcciones de máquinas.

En este nivel se realiza el direccionamiento lógico y la determinación de la ruta de los datos hasta su receptor final.

3.1.4. CAPA DE TRANSPORTE (CAPA 4)

Su función básica es aceptar los datos enviados por las capas superiores, dividirlos en pequeñas partes si es necesario, y pasarlos a la capa de red. En el caso del modelo OSI, también se asegura que lleguen correctamente al otro lado de la comunicación. Otra característica a destacar es que debe aislar a las capas superiores de las distintas posibles implementaciones de tecnologías de red en las

capas inferiores, lo que la convierte en el corazón de la comunicación. En esta capa se proveen servicios de conexión para la capa de sesión, que serán utilizados finalmente por los usuarios de la red al enviar y recibir paquetes. Estos servicios estarán asociados al tipo de comunicación empleada, la cual puede ser diferente según el requerimiento que se le haga a la capa de transporte. Por ejemplo, la comunicación puede ser manejada para que los paquetes sean entregados en el orden exacto en que se enviaron, asegurando una comunicación punto a punto libre de errores, o sin tener en cuenta el orden de envío. Una de las dos modalidades debe establecerse antes de comenzar la comunicación para que una sesión determinada envíe paquetes, y ése será el tipo de servicio brindado por la capa de transporte hasta que la sesión finalice. De la explicación del funcionamiento de esta capa se desprende que no está tan encadenada a capas inferiores como en el caso de las capas 1 a 3, sino que el servicio a prestar se determina cada vez que una sesión desea establecer una comunicación. Todo el servicio que presta la capa está gestionado por las cabeceras que agrega al paquete a transmitir.

En resumen, podemos definir a la capa de transporte como la capa encargada de efectuar el transporte de los datos (que se encuentran dentro del paquete), de la máquina origen a la de destino, independizándolo del tipo de red física que se esté utilizando. La Unidad de Datos de Protocolo (PDU) de la capa 4 se denomina "Segmento". Sus protocolos son TCP^{xiii} y UDP^{xiv} el primero orientado a conexión y el otro sin conexión.

3.1.5 CAPA DE SESIÓN (CAPA 5)

Esta capa establece, gestiona y finaliza las conexiones entre usuarios (procesos o aplicaciones) finales. Ofrece varios servicios que son cruciales para la comunicación, como son:

- Control de la sesión a establecer entre el emisor y el receptor (quién transmite, quién escucha y seguimiento de ésta).
- Control de la concurrencia (que dos comunicaciones a la misma operación crítica no se efectúen al mismo tiempo).
- Mantener puntos de verificación (*checkpoints*), que sirven para que, ante una interrupción de transmisión por cualquier causa, la misma se pueda reanudar desde el último punto de verificación en lugar de repetirla desde el principio.

Por lo tanto, el servicio provisto por esta capa es la capacidad de asegurar que, dada una sesión establecida entre dos máquinas, la misma se pueda efectuar para las operaciones definidas de principio a fin, reanudándolas en caso de interrupción. En muchos casos, los servicios de la capa de sesión son parcial o totalmente prescindibles.

En conclusión esta capa es la que se encarga de mantener el enlace entre los dos computadores que estén transmitiendo datos de cualquier índole.

3.1.6. CAPA DE PRESENTACIÓN (CAPA 6)

El objetivo de la capa de presentación es encargarse de la representación de la información, de manera que aunque distintos equipos puedan tener diferentes representaciones internas de caracteres (*ASCII*, *Unicode*, *EBCDIC*), números (*little-endian* tipo Intel, *big-endian* tipo Motorola), sonido o imágenes, los datos lleguen de manera reconocible.

Esta capa es la primera en trabajar más el contenido de la comunicación que el cómo se establece la misma. En ella se tratan aspectos tales como la semántica y la sintaxis de los datos transmitidos, ya que distintas computadoras pueden tener diferentes formas de manejarlas.

Esta capa también permite cifrar los datos y comprimirlos. En pocas palabras es un traductor.

Por todo ello, se puede concluir la definición de esta capa, como: la encargada de manejar la estructura de datos abstracta y realizar las conversiones de representación de los datos necesarias para la correcta interpretación de los mismos.

3.1.7. CAPA DE APLICACIÓN (CAPA 7)

Ofrece a las aplicaciones (de usuario o de cliente), la posibilidad de acceder a los servicios de las demás capas y define los protocolos que utilizan las aplicaciones para intercambiar datos, como correo electrónico (POP y SMTP), gestores de bases de datos y servidor de ficheros (FTP). Hay tantos protocolos como aplicaciones distintas y puesto que continuamente se desarrollan nuevas aplicaciones, el número de protocolos crece sin parar.

Es en esta capa donde interactúan los servidores web.

Cabe aclarar que el usuario normalmente no interactúa de forma directa con el nivel de aplicación. Suele interactuar con programas que a su vez interactúan con el nivel de aplicación, pero ocultando la complejidad subyacente. Así por ejemplo un usuario no manda una petición "GET index.html HTTP/1.0" para conseguir una página en HTML, ni lee directamente el código HTML/XML.

Entre los protocolos (refiriéndose a protocolos genéricos, no a protocolos de la capa de aplicación de OSI) más conocidos se destacan:

- HTTP (*HyperText Transfer Protocol* = Protocolo de Transferencia de Hipertexto) El protocolo bajo la "WWW".
- FTP (*File Transfer Protocol* = Protocolo de Transferencia de Archivos) Transferencia de ficheros.

- SMTP (*Simple Mail Transfer Protocol* = Protocolo Simple de Correo) Envío y distribución de correo electrónico.
- POP (*Post Office Protocol* = Protocolo de Oficina de Correo) / IMAP (*Internet Message Access Protocol* = Protocolo de Acceso a Mensajes de Internet) Reparto de correo al usuario final.
- SSH (*Secure Shell* = Capa Segura) Principalmente terminal remoto, aunque en realidad cifra casi cualquier tipo de transmisión.
- Telnet (*TELEcommunication NETwork* = Red de Trabajo de Telecomunicaciones) Otro terminal remoto, ha caído en desuso por su inseguridad intrínseca, ya que las claves viajan sin cifrar por la red.

3.1.8. UNIDADES DE DATOS

El intercambio de información entre dos capas del modelo OSI, consiste en que cada capa en el sistema fuente le agrega información de control a los datos, y cada capa en el sistema de destino analiza y remueve la información de control de los datos como se muestra a continuación:

Si un ordenador “anfitrión” (host A) desea enviar datos a otro (host B), en primer término los datos deben empaquetarse a través de un proceso denominado encapsulamiento, es decir, a medida que los datos se desplazan a través de las capas del modelo OSI, reciben encabezados, información final y otros tipos de información.

N-PDU (*Unidad de datos de protocolo*)

Es la información intercambiada entre entidades pares, es decir, dos entidades pertenecientes a la misma capa pero en dos sistemas diferentes, utilizando una conexión (N-1). Está compuesta por:

N-SDU (Unidad de datos del servicio) Son los datos que se necesitan las entidades (N) para realizar funciones del servicio pedido por la entidad (N+1).

N-PCI (Información de control del protocolo) Información intercambiada entre entidades (N) utilizando una conexión (N-1) para coordinar su operación conjunta.

N-IDU (*Unidad de datos de interface*)

Es la información transferida entre dos niveles adyacentes, es decir, dos capas contiguas. Está compuesta por:

N-ICI (Información de control del interface) Información intercambiada entre una entidad (N+1) y una entidad (N) para coordinar su operación conjunta.

Datos de Interface-(N) Información transferida entre una entidad-(N+1) y una entidad-(N) y que normalmente coincide con la (N+1)-PDU.

3.1.9. TRANSMISIÓN DE LOS DATOS

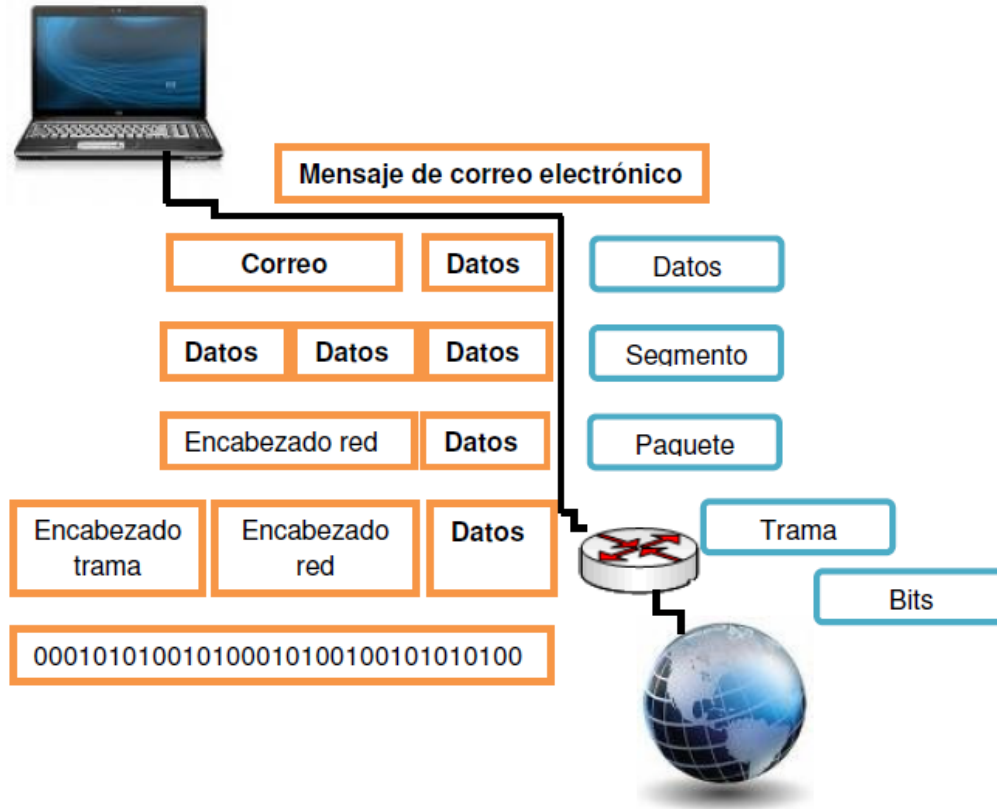


Imagen 2 - Transferencia de Información del Modelo OSI

La capa de aplicación recibe el mensaje del usuario y le añade una cabecera constituyendo así la PDU de la capa de aplicación. La PDU se transfiere a la capa de aplicación del nodo destino, éste elimina la cabecera y entrega el mensaje al usuario.

Para ello ha sido necesario este proceso:

1. Entregar la PDU a la capa de presentación, añadiendo la correspondiente cabecera ICI y transformándola así en una IDU, la cual se transmite a dicha capa.
2. La capa de presentación recibe la IDU, le quita la cabecera y extrae la información, es decir, la SDU, a esta le añade su propia cabecera (PCI) constituyendo así la PDU de la capa de presentación.

3. Esta PDU es transferida a su vez a la capa de sesión mediante el mismo proceso, repitiéndose así para todas las capas.
4. Al llegar al nivel físico, se envían los datos que son recibidos por la capa física del receptor.
5. Cada capa del receptor se ocupa de extraer la cabecera, que anteriormente había añadido su capa homóloga, interpretar la y entregar la PDU a la capa superior.
6. Finalmente llegará a la capa de aplicación la cual entregará el mensaje al usuario.

3.1.10. FORMATO DE LOS DATOS

Estos datos reciben una serie de nombres y formatos específicos en función de la capa en la que se encuentren, debido a como se describió anteriormente la adición de una serie de encabezados e información final. Los formatos de información son los que muestra el gráfico:

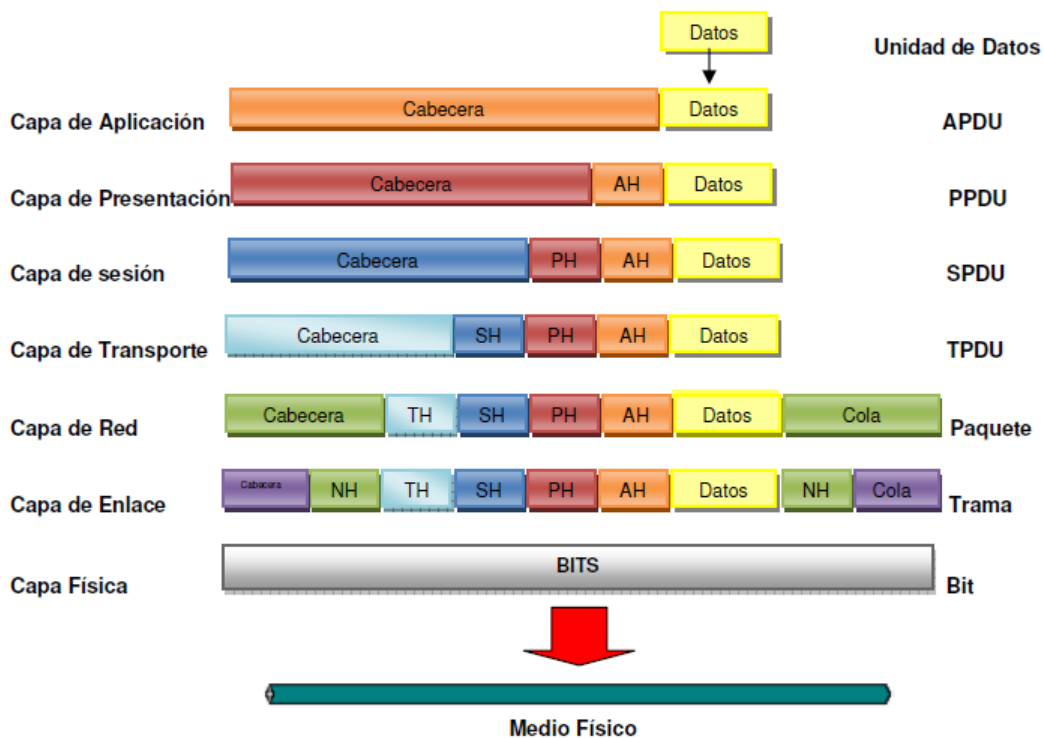


Imagen 3 - Formato de datos en el modelo OSI

APDU Unidad de datos en la capa de aplicación (*Capa 7*).

PPDU Unidad de datos en la capa de presentación (*Capa 6*).

SPDU Unidad de datos en la capa de sesión (*Capa 5*).

TPDU (segmento o datagrama) Unidad de datos en la capa de transporte (*Capa 4*).

Paquete Unidad de datos en el nivel de red (*Capa 3*).

Trama Unidad de datos en la capa de enlace (*Capa 2*).

Bits Unidad de datos en la capa física (*Capa 1*).

3.1.11. OPERACIONES SOBRE LOS DATOS

En determinadas situaciones es necesario realizar una serie de operaciones sobre las PDU para facilitar su transporte, debido a que son demasiado grandes o bien porque son demasiado pequeñas y estaríamos desaprovechando la capacidad del enlace.

Segmentación y re ensamblaje: Hace corresponder a una (N)-SDU sobre varias (N)-PDU. El re ensamblaje hace corresponder a varias (N)-PDU en una (N)-SDU.

Bloqueo y desbloqueo: El bloqueo hace corresponder varias (N)-SDUs en una (N)-PDU. El desbloqueo identifica varias (N)-SDUs que están contenidas en una (N)-PDU.

Concatenación y separación: La concatenación es una función-(N) que realiza el nivel-(N) y que hace corresponder varias (N)-PDUs en una sola (N-1)-SDU. La separación identifica varias (N)-PDUs que están contenidas en una sola (N-1)-SDU.

3.2. PROTOCOLO HTTP

El protocolo de transferencia de hipertexto (**HTTP**, *HyperText Transfer Protocol*) es el protocolo usado en cada transacción de la Web (WWW). HTTP fue desarrollado por el consorcio W3C^{xv} y la IETF^{xvi}, colaboración que culminó en 1999 con la publicación de una serie de RFC^{xvii}, siendo el más importante de ellos el RFC2616, que especifica la versión 1.1. HTTP define la sintaxis y la semántica que utilizan los elementos software de la arquitectura web (clientes, servidores, proxies) para comunicarse. Es un protocolo orientado a transacciones y sigue el esquema petición-respuesta entre un cliente y un servidor. Al cliente que efectúa la petición (un navegador) se lo conoce como "*user agent*" (agente del usuario). A la información transmitida se la llama "recurso" y se la identifica mediante un URL. Los recursos pueden ser archivos, el resultado de la ejecución de un programa, una consulta a una base de datos, la traducción automática de un documento, etc.

HTTP es un protocolo sin estado, es decir, que no guarda ninguna información sobre conexiones anteriores. El desarrollo de aplicaciones web necesita frecuentemente mantener estado. Para esto se usan las *cookies* (*Galletitas*), que es información que un servidor puede almacenar en el sistema cliente. Esto le permite a las aplicaciones web instituir la noción de "sesión", y también permite rastrear usuarios ya que las *cookies* pueden guardarse en el cliente por tiempo indeterminado.

3.2.1. TRANSACCIONES HTTP

Una transacción HTTP está formada por un encabezado, seguido opcionalmente, por una línea en blanco y algún dato. El encabezado especificará cosas como la acción requerida del servidor, o el tipo de dato retornado, o el código de estado.

El uso de campos de encabezados enviados en las transacciones HTTP le da gran flexibilidad al protocolo. Estos campos permiten que se envíe información descriptiva en la transacción, permitiendo así la autenticación, cifrado e identificación de usuario.

Un encabezado es un bloque de datos que precede a la información propiamente dicha, por lo que muchas veces se hace referencia a él como *metadato* (porque tiene datos sobre los datos).

Si se reciben líneas de encabezado del cliente, el servidor las coloca en las variables de ambiente de CGI^{xviii} con el prefijo HTTP_ seguido del nombre del encabezado. Cualquier carácter guión (-) del nombre del encabezado se convierte a caracteres "_".

El servidor envía al cliente:

- Un código de estado que indica si la petición fue correcta o no. Los códigos de error típicos indican que el archivo solicitado no se encontró, que la petición no se realizó de forma correcta o que se requiere autenticación para acceder al archivo.
- La información propiamente dicha. Como HTTP permite enviar documentos de todo tipo y formato, es ideal para transmitir multimedia, como gráficos, audio y video. Esta libertad es una de las mayores ventajas de HTTP.
- Información sobre el objeto que se retorna.

3.2.2. VERSIONES

HTTP ha pasado por múltiples versiones del protocolo, muchas de las cuales son compatibles con las anteriores. El RFC 2145 describe el uso de los números de versión de HTTP. El cliente le dice al servidor al principio de la petición la versión que usa, y el servidor usa la misma o una anterior en su respuesta.

0.9 Obsoleta. Soporta sólo un comando, *GET*, y además no especifica el número de versión HTTP. No soporta cabeceras. Como esta versión no soporta *POST*, el cliente no puede enviarle mucha información al servidor.

HTTP/1.0 (mayo 1996) Esta es la primera revisión del protocolo que especifica su versión en las comunicaciones, y todavía se usa ampliamente, sobre todo en servidores “proxy”^{xix}.

HTTP/1.1 (junio 1999) Versión actual; las conexiones persistentes están activadas por defecto y funcionan bien con los proxies. También permite al cliente enviar múltiples peticiones a la vez (*pipelining*) lo que hace posible eliminar el tiempo de *Round-Trip delay* (tiempo que tarda un paquete enviado desde un emisor en volver a este mismo emisor habiendo pasado por el receptor de destino) por cada petición.

HTTP/1.2 Los primeros borradores de 1995 del documento *PEP — an Extension Mechanism for HTTP* (el cuál propone el Protocolo de Extensión de Protocolo, abreviado PEP) los hizo el *World Wide Web Consortium W3C*, y se envió al *Internet Engineering Task Force IETF*. El PEP inicialmente estaba destinado a convertirse en un rango distintivo de HTTP/1.2. En borradores posteriores, sin embargo, se eliminó la referencia a HTTP/1.2. El RFC 2774 (experimental), *HTTP Extension Framework*, incluye en gran medida a PEP. Se publicó en febrero de 2000.

3.3. HTML

HTML, siglas de *HyperText Markup Language (Lenguaje de Marcas de Hipertexto)*, es el lenguaje de marcado predominante dentro del proceso de construcción de páginas web. Es usado para describir la estructura y el contenido en forma de texto, así como para complementar el texto con objetos tales como imágenes. HTML se escribe en forma de "etiquetas", rodeadas por corchetes angulares (<,>). HTML también puede describir, hasta un cierto punto, la apariencia de un documento, y puede incluir un *script* (por ejemplo *Javascript*), el cual puede afectar el comportamiento de navegadores web y otros procesadores de HTML.

Por convención, los archivos de formato HTML usan la extensión .htm o .html.

3.3.1. MARCADO HTML

HTML consiste de varios componentes vitales, incluyendo los *elementos* y sus *atributos*, *tipos de datos*, y la *declaración de tipo de documento*.

Elementos: Los elementos son la estructura básica de HTML. Tienen dos propiedades básicas: atributos y contenido. Cada atributo y contenido tiene ciertas restricciones para que se considere válido al documento HTML. Un elemento generalmente tiene una etiqueta de inicio (por ejemplo <nombre-de-elemento>) y una etiqueta de cierre (por ejemplo </nombre-de-elemento>). Los atributos del

elemento están contenidos en la etiqueta de inicio y el contenido está ubicado entre las dos etiquetas (por ejemplo `<nombre-de-elemento atributo="valor">Contenido</nombre-de-elemento>`). Algunos elementos, tales como `
`, no tienen contenido ni llevan una etiqueta de cierre.

El marcado “estructural” describe el propósito del texto. El marcado estructural no define cómo se verá el elemento, pero la mayoría de los navegadores web han estandarizado el formato de los elementos. Un formato específico puede ser aplicado al texto por medio de hojas de estilo en cascada (CSS).

El marcado “presentacional” describe la apariencia del texto, sin importar su función. Por ejemplo, `negrita` indica que los navegadores web visuales deben mostrar el texto en **negrita**, pero no indica qué deben hacer los navegadores web que muestran el contenido de otra manera (por ejemplo, los que leen el texto en voz alta). En el caso de `negrita` e `<i>itálica</i>`, existen elementos que se ven de la misma manera pero tienen una naturaleza más semántica: `énfasis fuerte` y `énfasis`. Es fácil ver cómo un lector de pantalla debería interpretar estos dos elementos. Sin embargo, son equivalentes a sus correspondientes elementos presentacionales: un lector de pantalla no debería decir más fuerte el nombre de un libro, aunque éste esté en *itálicas* en una pantalla. La mayoría del marcado presentacional ha sido desechada con HTML 4.0, en favor de las Hojas de estilo en cascada.

El marcado “hipertextual” se utiliza para enlazar partes del documento con otros documentos o con otras partes del mismo documento. Para crear un enlace es necesario utilizar la etiqueta de ancla `<a>` junto con el atributo *href*, que establecerá la dirección URL a la que apunta el enlace. Por ejemplo, un enlace a la página de CYTEMAC sería de la forma `CYTEMAC`. También se pueden crear enlaces sobre otros objetos, tales como imágenes ``.

3.3.2. ATRIBUTOS

La mayoría de los atributos de un elemento son pares nombre-valor, separados por un signo de igual "=" y escritos en la etiqueta de comienzo de un elemento, después del nombre de éste. El valor puede estar rodeado por comillas dobles o simples, aunque ciertos tipos de valores pueden estar sin comillas en HTML (pero no en XHTML). De todas maneras, dejar los valores sin comillas es considerado poco seguro.

El microservidor web, objeto del presente trabajo, se basa en la implementación del protocolo HTTP, a través del código HTML que al compilarse por un navegador, presenta una página web a modo de interfaz de usuario para el monitoreo y control de las variables

3.4. JAVASCRIPT

JavaScript es un lenguaje de programación interpretado, es decir, que no requiere compilación, utilizado principalmente en páginas web, con una sintaxis semejante a la del lenguaje *Java* y el lenguaje *C*.

Todos los navegadores modernos interpretan el código JavaScript integrado dentro de las páginas web. Para interactuar con una página web se provee al lenguaje JavaScript de una implementación del DOM^{xx}.

El lenguaje fue inventado por Brendan Eich en la empresa *Netscape Communications*, que es la que desarrolló los primeros navegadores web comerciales. Apareció por primera vez en el producto de Netscape llamado Netscape Navigator 2.0.

Tradicionalmente, se venía utilizando en páginas web HTML, para realizar tareas y operaciones en el marco de la aplicación únicamente cliente, sin acceso a funciones del servidor. JavaScript se ejecuta en el agente de usuario al mismo tiempo que las sentencias van descargándose junto con el código HTML.

Los autores inicialmente lo llamaron Mocha y más tarde LiveScript pero fue rebautizado como JavaScript en un anuncio conjunto entre Sun Microsystems y Netscape, el 4 de diciembre de 1995.

En 1997 los autores propusieron JavaScript para que fuera adoptado como estándar de la *European Computer Manufacturers' Association* ECMA, que a pesar de su nombre no es europeo sino internacional, con sede en Ginebra. En junio de 1997 fue adoptado como un estándar ECMA, con el nombre de ECMAScript. Poco después también lo fue como un estándar ISO.

JScript es la implementación de ECMAScript de Microsoft, muy similar al JavaScript de Netscape, pero con ciertas diferencias en el modelo de objetos del navegador que hacen a ambas versiones con frecuencia incompatibles.

Para evitar estas incompatibilidades, *el World Wide Web Consortium* diseñó el estándar *Document Object Model* (DOM, ó Modelo de Objetos del Documento en castellano), que incorporan la mayoría de los navegadores de la actualidad, como *Internet Explorer, Mozilla Firefox, Opera, Google Chrome, etc.*

Javascript se puede incluir en cualquier documento y es compatible con cualquier sistema operativo HTML, o todo aquel que termine traducándose en HTML en el navegador del cliente; ya sea PHP, ASP, JSP, SVG. Incluir código directamente en una estructura HTML es una práctica invasiva, y no recomendada. El método correcto que define la W3C es incluir javascript como un archivo externo, tanto por cuestiones de accesibilidad, como practicidad y velocidad en la navegación.

En el caso del microservidor desarrollado, el javascript se emplea como herramienta para publicar sobre la página HTML, el valor de las variables ubicadas

en la memoria de datos del microcontrolador, e inversamente, para tomar el estado de variables booleanas en el código HTML y leerlo por la memoria de programas del dispositivo para su posterior interpretación. Sin embargo, por el hecho de que la única ubicación disponible para los códigos es la memoria ROM del PIC, solo es posible la inserción del javascript directamente sobre el código HTML.

3.5. JOOMLA!

Joomla!^{xxi}, es un Sistema de Administración de Contenidos (CMS por su sigla en inglés *Content Management System*). Consiste en una aplicación de código abierto, creada en su mayoría en PHP, que independiza el diseño gráfico de un sitio web, de su contenido, a través del empleo de hojas de estilo en cascada (CCS) y una base de datos SQL.

Entre las principales características de los CMS, se encuentran: la facilidad de creación y edición del contenido y el manejo de roles para los usuarios del sitio.

Joomla! Es un CMS que permite la fácil incorporación y manejo de módulos al sitio web, que permiten emplear elementos como: foros, chats, centrales de noticias, blogs, registro y control de usuarios, contenido rotatorio, etc.

La versión actual de Joomla! Es la 1.5X y se instala fácilmente sobre un servidor Windows o Linux, sobre una base de datos preconfigurada en el mismo.

Joomla!, es un software de código abierto, soportado por una comunidad mundial que permanentemente desarrolla nuevas características, corrige fallos y lo mantiene en la punta del manejo de las nuevas tecnologías para la información y la comunicación.

4. MICROSERVIDOR WEB

Un microservidor web, es un dispositivo hardware, basado en un microcontrolador, cuyo *firmware* (o software embebido), es una implementación básica del protocolo HTTP desde el lado del anfitrión, que incluye el código HTML necesario para generar una página web, con comandos de JavaScript incrustados a lo largo de su código, para interactuar directamente con los distintos elementos del microcontrolador.

La página web, alojada en la memoria de programa del dispositivo, se emplea como interfaz gráfica de usuario, desde la cual se puede conocer el estado de las variables referenciadas e interactuar con los elementos de control del sistema.

El código del protocolo HTTP, está implementado en HTML Versión 1.1, en dos constantes tipo *char* en la memoria de datos del microcontrolador, y responde a peticiones de tipo *GET* de cualquier navegador que soporte JavaScript.

El microservidor debe ser pre configurado con una dirección MAC y una IP para poder interactuar con los computadores de una red. Si se pretende que el dispositivo responda a peticiones por fuera de la intranet (desde y hacia internet), entonces la IP que se le configure al dispositivo debe ser de tipo “real”.

El sistema permite monitorear todas las entradas analógicas que el microcontrolador sea capaz de procesar, y puede conmutar el estado lógico de las salidas digitales de todos los puertos del chip. Para el caso del presente proyecto, se determinó emplear tres de los canales de conversión análogo-digital para el monitoreo de tres sensores desde la página web y además controlar desde la misma, dos salidas digitales, conectadas a sendas etapas de potencia.

Cada entrada analógica, es acondicionada electrónicamente de acuerdo a las características del sensor empleado, para llevar su rango de salida al intervalo de 0V a 5V, correspondiente al rango total de las entradas analógicas del microcontrolador. De igual forma, la función de transferencia de cada sensor, debe ser programada en el firmware para que la interfaz gráfica de usuario, exhiba correctamente las unidades de cada medida.

Un intercambio de información desde la interfaz gráfica de usuario, con la posibilidad de escoger el tipo de sensor conectado y su respectiva configuración directamente desde la página web y sin alterar la programación del microcontrolador, es el siguiente paso en el proceso de desarrollo, y puede ser tema de un nuevo trabajo de implementación por parte de un nuevo grupo de estudiantes.

Como se puede analizar, las entradas del sistema, son las señales analógicas provenientes de los sensores. Una termocupla y dos sensores de temperatura LM35, se emplearon para el diseño base del sistema, pero como se comentó anteriormente, cualquier sensor cuya salida sea acondicionada al rango de entrada del ADC^{xxii} del microcontrolador, puede ser monitoreado.

El NIC (*Network Interface Connector*), es la “tarjeta de red” que permitirá al microcontrolador conectarse a cualquier red a través de una conexión Ethernet, se empleó para tal fin un ENC28J60 de la empresa Microchip. Este dispositivo por sí mismo, no maneja los niveles de tensión y corriente necesarios para la conexión 10/100 por cable UTP, por lo que es necesario implementar un arreglo electromagnético para realizar el acople de las señales. Este dispositivo es el “conector” RJ45 presentado en el esquema.

El microcontrolador que gobierna el sistema es un PIC18F4520 de Microchip, que es una versión mejorada de su sucesor, el PIC18F452.

La plataforma se instaló y se puso a prueba al interior de la red de datos de la Universidad del Cauca, con salida a Internet a través de una dirección IP pública. El siguiente esquema, muestra el diagrama de bloques del sistema instalado:

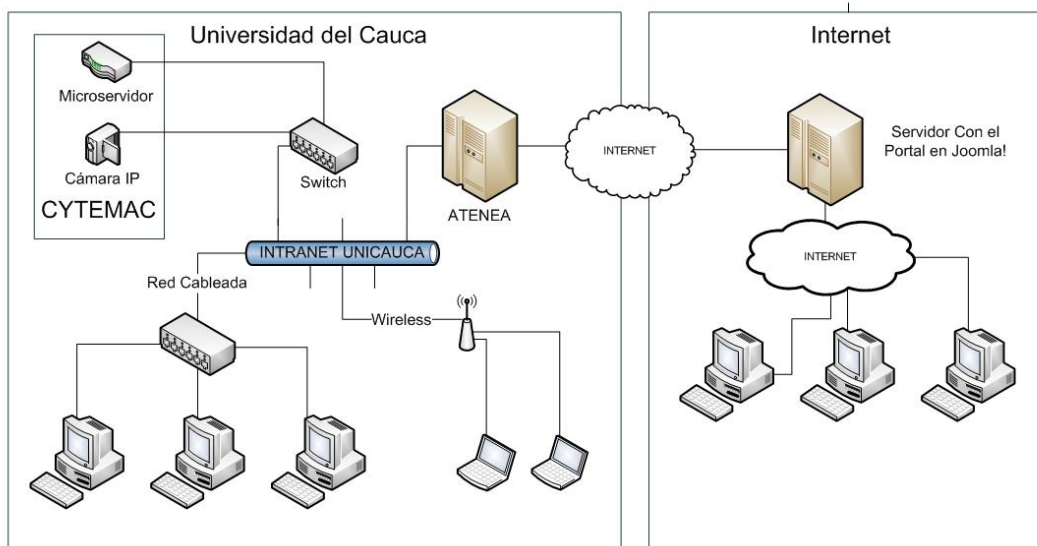


Imagen 4 - Diagrama de bloques del Sistema desarrollado dentro de la Red de Datos de UNICAUCA

4.1. EL PIC18F4520

Este circuito integrado fabricado por *Microchip Technology Inc.* es un dispositivo programable de gama alta, compatible con código de ensamblador, pero optimizado para ser programado en lenguaje C. Su versatilidad, producto de sus periféricos e interfaces de comunicación serial y paralela, en conjunto con sus características de velocidad y memoria, lo convierten en una excelente opción para diseñar instrumentos de control de mediana y alta complejidad, razones por las cuales, fue incorporado al proyecto como unidad central de procesamiento.

Esta versión, es la mejora del difundido PIC18F452, que incrementa frente a su predecesor, la memoria de programa, el número de canales del Conversor Análogo Digital y las salidas PWM, entre otras características.

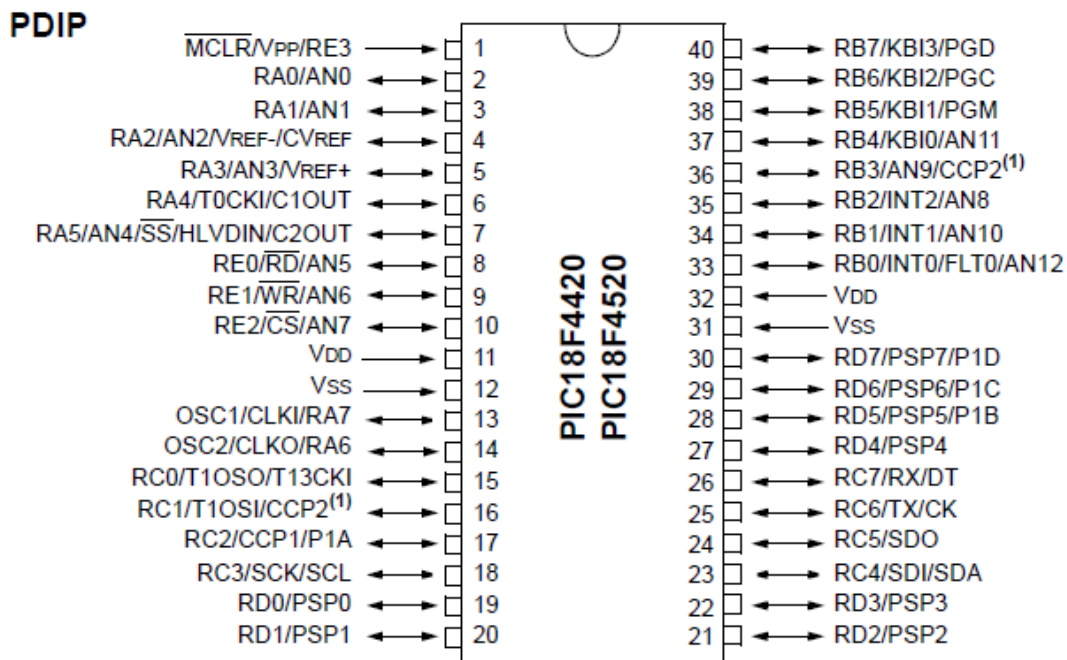


Imagen 5 - Diagrama Circuitual del PIC18F4520

Cuenta con cinco puertos digitales paralelos de entrada y salida, tres de ellos de 8 bits, uno de 5 bits y el último de 3 bits. Entre sus principales módulos adicionales, se destacan: El conversor análogo digital de 13 canales, las unidades de control de velocidad por ancho de pulso PWM, temporizadores, el manejo de interrupciones, entre otros.

Principales Características Técnicas:

- Memoria de programa tipo Flash Embebida de 32 Kb
- Memoria RAM de 1536 Bytes
- Memoria EEPROM de 256 bytes
- Hasta 10 millones de instrucciones por segundo (MIP's)
- Hasta 40 MHz de frecuencia de operación
- Corriente de Fuente y Sumidero para los pines digitales de hasta 25 mA
- Tres pines externos de interrupciones
- Cuatro módulos temporizadores/contadores
- Dos unidades de captura/comparación/PWM
- Módulo de puerto serial maestro sincrónico (MSSP), con dos modos de operación:
 - 3-wire SPI™ (Soporta los cuatro módulos SPI)
 - I2C™ En modo maestro y esclavo
- Módulo direccionable USART con soporte para los protocolos RS-485 y RS-232
- Conversor Análogo Digital de 10 bits, de 13 canales.
- Voltaje de operación de 2.2 a 5 voltios
- Encapsulado DIP de 40 pines

4.2. EL ENC28J60

Es un controlador NIC (*Network Interface Connector*) para interconectar los dispositivos programables de Microchip Technology como PIC's y DSPIC's a redes Ethernet con una interfaz serial SPI^{xxiii} entre el NIC y el microcontrolador. Viene en diferentes encapsulados y el empleado en el desarrollo del presente proyecto, se adquirió del tipo DIP (*Dual In Line Package*).

A pesar de que como tal, el ENC28J60 no es un dispositivo programable, sus funciones si son configurables desde el microcontrolador, y es una excelente base para la implementación de la capa física del modelo OSI.

Es una alternativa a la que se llegó tras el descarte de tarjetas de red 10/100 para PC, de puerto PCI, que se planeaban manejar desde el microcontrolador.

28-Pin SPDIP, SSOP, SOIC

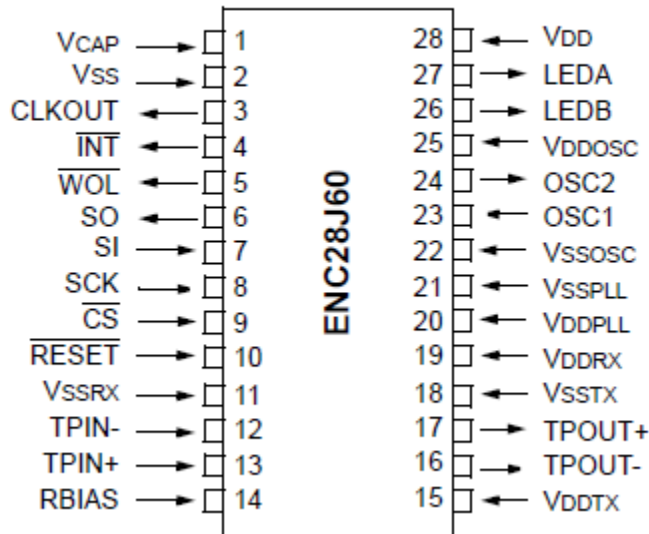


Imagen 6 - Diagrama Circuital del ENC28J60

El ENC28J60 implementa las funciones de las dos capas más bajas del modelo OSI (MAC & PHY). Permitiendo que cualquier microcontrolador pueda enviar tramas a través del cableado Ethernet existente en redes LAN de cualquier envergadura, e incluso desde internet, a través de servidores y enrutadores.

El ENC28J60 consta de siete grandes bloques funcionales:

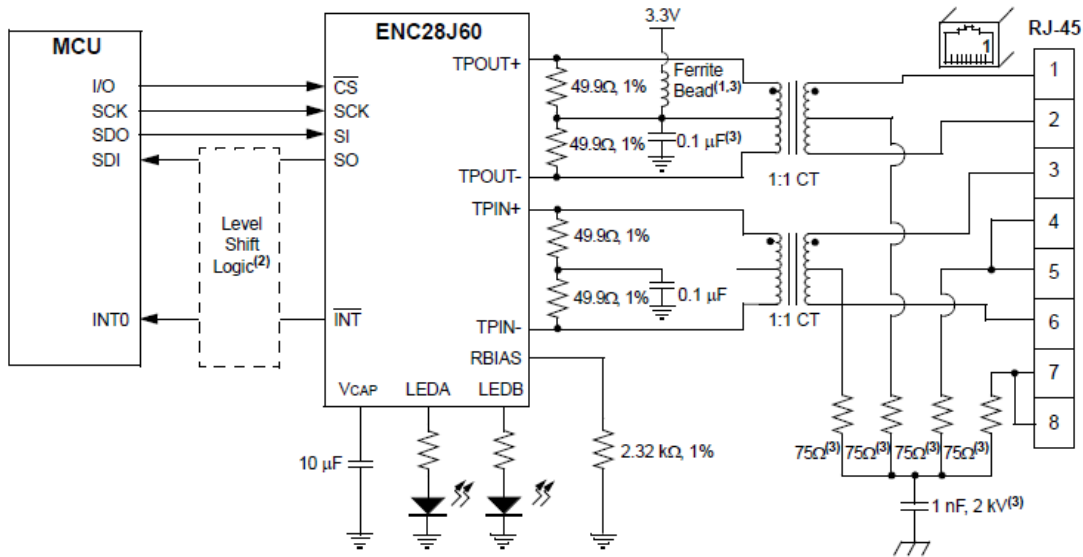
1. Una interfaz de SPI que sirve como canal de comunicación entre el microcontrolador maestro y el ENC28J60.
2. Un Sistema de Registros que se utilizan para el monitoreo y control del ENC28J60.
3. Un buffer RAM de doble puerto para la recepción y transmisión de datos.
4. Un sistema de chequeo para controlar el acceso a la memoria RAM cuando las solicitudes se hacen desde DMA, transmitir y recibir por bloques.
5. Un sistema de Comunicación que interpreta los datos y comandos recibidos a través de la interfaz SPI.
6. El MAC (Media Access Control) que implementa IEEE 802.3 para la lógica compatible su protocolo.
7. El PHY (capa física) que codifica y decodifica los datos analógicos que están presentes en la interfaz ETHERNET a través del cable UTP.

El dispositivo también contiene bloques de apoyo, tales como el oscilador, el regulador de voltaje, el sistema de adaptación para proporcionar 5V de tolerancia de las entradas y salidas, y un sistema de control lógico.

4.3. CONECTOR RJ45

Debido a las especificaciones del manual de usuario del ENC28J60, fue necesario implementar una etapa de acople electromagnético para las señales provenientes y emigrantes desde y hacia la red Ethernet a través del cable de par trenzado.

El manual expresa la necesidad de conectar una “*ferrite bead*”, o “fragmento de ferrita” en español, similar a una bobina, con una inductancia $\geq 100\text{mH}$ y un soporte para corrientes de hasta 80mA .



La imposibilidad de conseguir este dispositivo en el mercado nacional, llevó al grupo desarrollador a comprar una tarjeta prefabricada en el mercado de los Estados Unidos, que posee el conector RJ45, los transformadores y la “*ferrite bead*”, en un arreglo listo para su uso.

El propósito de este dispositivo inductor, es aislar al instrumento desarrollado, de la Interferencia Electromagnética (EMI por su sigla en inglés)

El conector RJ45 que se utilizó para conectar el microservidor a la red una tarjeta fabricada por la empresa *mikroElektronika*:



Imagen 7 - Board del Conector RJ45

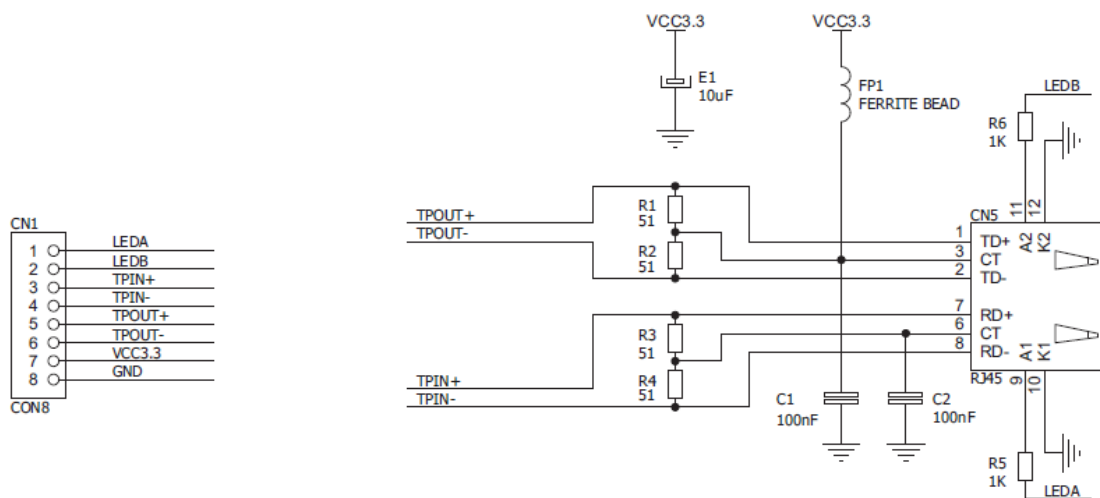


Imagen 8 - Diagrama de Conexiones del Conector RJ45

Pruebas realizadas al microservidor, con elementos similares extraídos de motherboard's de computadores fuera de servicio, demostraron la inestabilidad del empleo del ENC28J60 sin el elemento de ferrita y obligaron al empleo de la descrita solución.

5. SENSORES E INTERFACES EMPLEADAS

5.1. SENSOR DE TEMPERATURA LM35

El LM35 es un sensor de temperatura con una precisión calibrada de 1°C y un rango que abarca desde -55° a +150°C.



Imagen 9 - Sensor de Temperatura LM35

El sensor se presenta en diferentes encapsulados pero el más común es el T0-92 de igual forma que un típico transistor con 3 patas, dos de ellas para alimentarlo y la tercera entrega un valor de tensión proporcional a la temperatura medida por el dispositivo.

La función de transferencia para este sensor es:

$$V_{out}(T) = 10mV * T,$$

donde el voltaje de salida es una respuesta lineal a la temperatura de entrada a una tasa de 10mV por cada °C.

5.2. SENSOR DE PRESIÓN MPXM2102

Este es un sensor piezoresistivo de presión de alta precisión, diseñado para trabajar como altímetro o barómetro. Tiene una salida lineal de de 0.2V a 4.8V y un rango de 0 a 100 KPa, compensado en temperatura.



Imagen 10 - Sensor de Presión MPXM2102

El sensor es un sencillo diafragma de silicio monolítico, con una celda de carga y una delgada película que contiene una red de resistencias integrada en el chip. El sensor es calibrado por laser, para garantizar la precisión de salida y su compensación en temperatura.

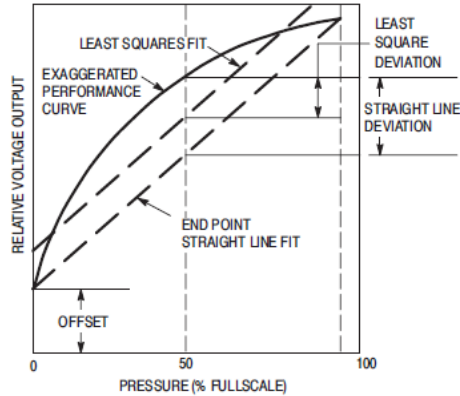


Imagen 11 - Gráfico de Presión Contra Voltaje de Salida del MPXM2102

5.3. CÁMARA IPCAM SECURE 300R

La cámara empleada como complemento visual para las pruebas en la implementación del microservidor, fue la *IPCAM SECURE 300R* de la empresa *GENIUS*.

Esta es una cámara IP, con un servidor Web embebido para visualización y manejo remoto y conexión directa a la red a través de su puerto RJ45. La IPCAM tiene seis diodos LED infrarrojos para visión en completa oscuridad. Está diseñada para usuarios del hogar y la oficina que deseen obtener una completa vigilancia remota, con imagen de alta calidad, a través de una red LAN o desde Internet. La cámara ofrece nitidez y claridad en las imágenes, gracias al empleo de un sensor CMOS de escaneo progresivo y avanzadas técnicas de procesamiento de señales.

Además del acceso a su servidor embebido, la cámara también es compatible con el software IDCS04i, que permite la visualización de múltiples cámaras IP de forma simultánea, agregando capacidad de detección de movimiento, alarmas por mensajes de correo electrónico, acceso vía ftp, entre otras.

Tipo de Sensor de Imagen	VGA (640 x 480) CMOS
Tipo de Lente	Lente Manual / Foco: 20cm ~ infinito
Interfaz de Red	802.3/802.3u NWAY Auto sensing 10/ 100Mbps (RJ-45 Jack)
Resolución de Video	640 x 480, 320 x 240

Cuadros por Segundo	1, 10, 20, 30, o Auto (Depende del formato de video)
Fuente de Poder	DC 5V/2ª
Requerimientos de Potencia	3.5 Vatios (700mA x 5V)



Imagen 12 - Cámara IPCAM SECURE 300R

6. DISEÑO DEL HARDWARE

6.1. TARJETA PRINCIPAL

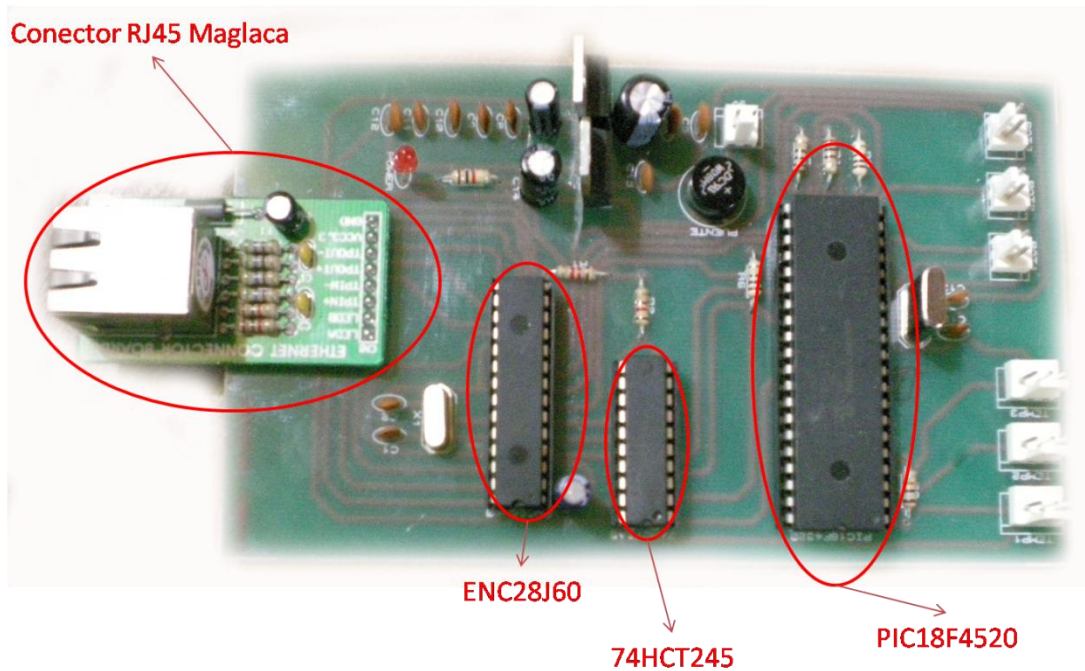


Imagen 13 - Tarjeta Principal del Microservidor

El ENC28J60, trabaja a 3.3V, como se expuso anteriormente junto con sus demás características eléctricas. Por lo tanto es necesario acoplar los niveles de tensión y corriente entre el microcontrolador y el NIC. Para este fin, se emplea el buffer 74HCT245. Que es un *buffer* transmisor y receptor de ocho canales, de tres estados.

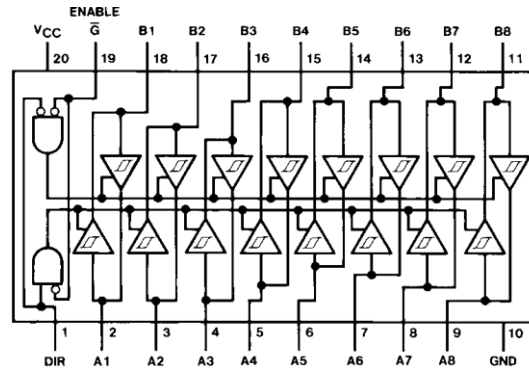


Imagen 14 - Diagrama Circuital del Buffer 74HCT245

Para obtener los voltajes de 5.0V del microcontrolador y los 3.3V del ENC28J60 se emplearon los reguladores LM7805 y LM1117 respectivamente. Ambos, en encapsulado T0-90 de tres pines.

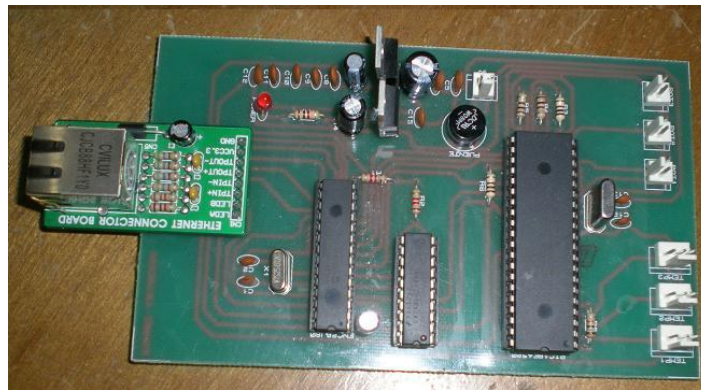


Imagen 15 - Fotografía de la Tarjeta Principal con sus componentes soldados

El sistema se conecta a la red, a través de un cable UTP, en configuración *patch cord* o *cable de conexión intermedia* (con una configuración de hilos específica clase A).

Debido a la diferencia de tensión a la entrada y salida del regulador de voltaje, y a la elevada frecuencia de operación del microcontrolador (32 MHz), el LM7805 se calentaba excesivamente (67°C), hasta el punto de colapsar el funcionamiento del dispositivo. De acuerdo al *datasheet* del Microcontrolador, se decidió instalar un disipador de temperatura, para garantizar la operabilidad del regulador a la frecuencia requerida para el sostenimiento de los protocolos de comunicación.

Con el disipador instalado, el integrado bajó su temperatura hasta los 27°C y el microservidor pudo operar de forma ininterrumpida sin alterar este valor.



Imagen 16 - Detalle del Disipador de Calor Instalado

6.2. TARJETA DE TEMPERATURA

Con el empleo de un LM35 y sus dispositivos de polarización, se elaboró una tarjeta de adecuación de señales de temperatura y entrega de datos análogos al microcontrolador en una escala apropiada de 0 a 5 Voltios.

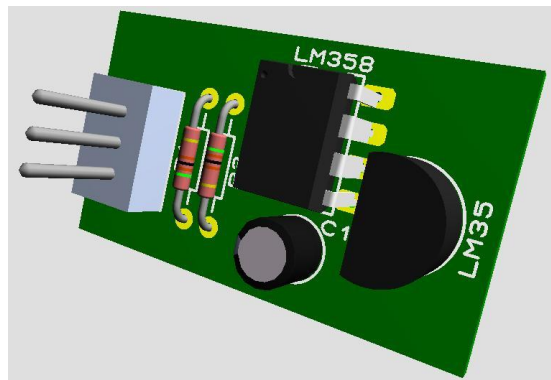


Imagen 17 - Tarjeta del Sensor de Temperatura

El circuito fue construido con base en el empleo de amplificadores operacionales en cascada: el primero como seguidor de tensión para el acople de impedancias y el segundo como amplificador no inversor.

El integrado LM358 fue el escogido por contener dos amplificadores operacionales en su interior.

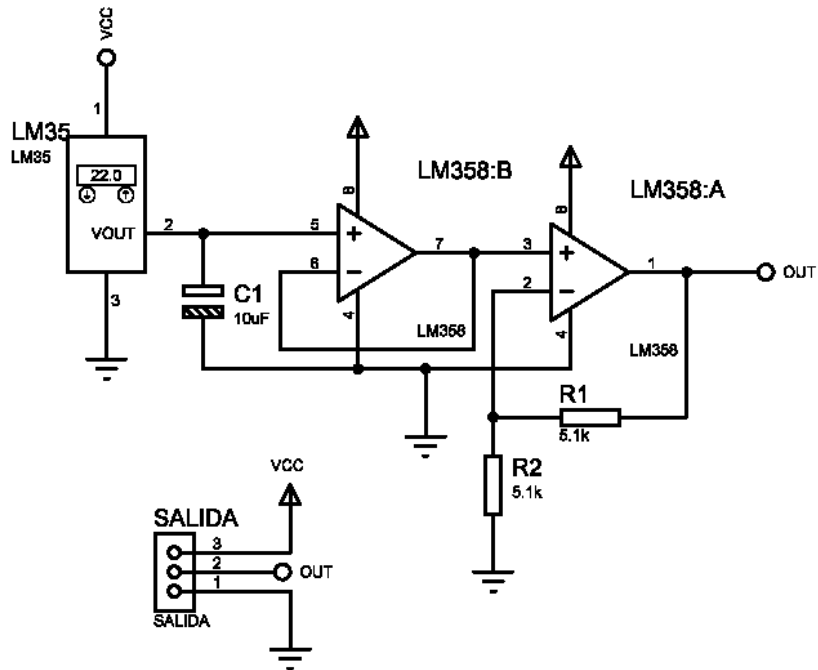


Imagen 18 - Diagrama Circuital de la Tarjeta de Temperatuta

6.3. TARJETA DE PRESIÓN

Con la ayuda del amplificador operacional LF353, y los respectivos resistores de polarización, se diseñó e implementó un circuito que garantiza la linealidad entre la presión aplicada sobre el cilindro del sensor, y el voltaje analógico de salida.

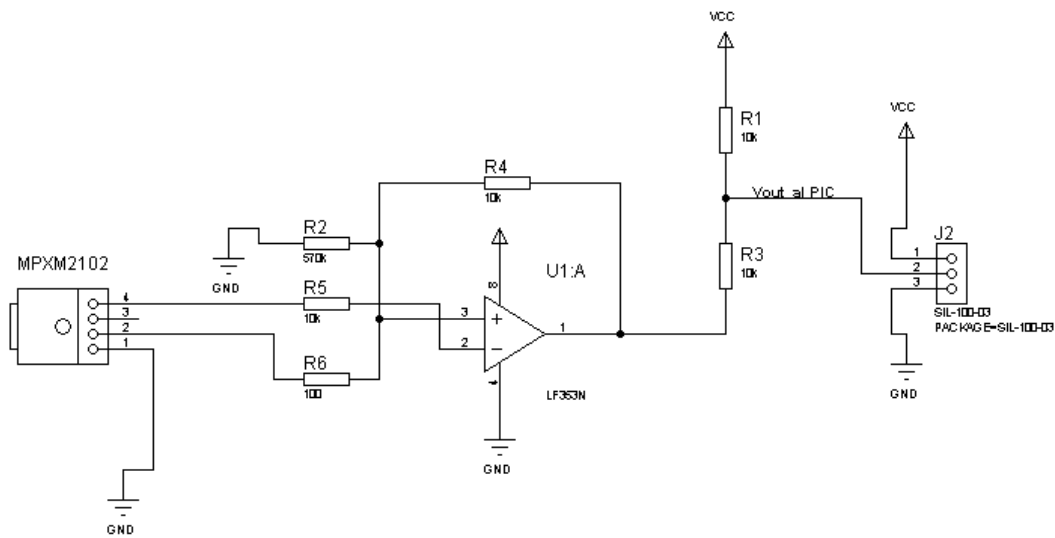


Imagen 19 - Diagrama Circuital de la Tarjeta de Presión

6.4. ETAPA DE POTENCIA

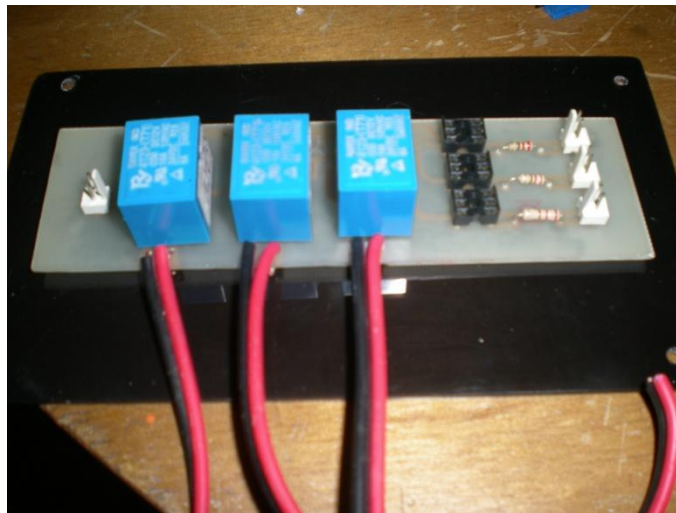


Imagen 20 - Fotografía de la Etapa de Potencia

A pesar de no ser un objetivo trazado en el anteproyecto, se habilitaron tres canales de potencia a partir de tres pines del puerto D (PORTD) del PIC18F4520 y se enlazaron con el código HTML/JavaScript a través de “Pulsadores” que se

cargan en la página web de la interfaz de usuario. El estado de estas salidas (Alto y Bajo) se presenta al usuario a través de colores “Rojo” para “Alto” y “Gris” para “Bajo”, además de acompañarse por los textos ON y OFF de acuerdo al estado del pin.



Para el acople de dispositivos de potencia a partir de las señales digitales TTL a la salida de los puertos del microcontrolador, se estableció el empleo de arreglos relevo-optoacoplador, con el fin de garantizar el mayor aislamiento posible a la red de salida del sistema.

7. DISEÑO DEL FIRMWARE

7.1. SOFTWARE EMBEBIDO

El software embebido en el microcontrolador, se desarrolló en lenguaje C, sobre el compilador MikroC de la empresa Mikroelectrónica. Este software, además de controlar el funcionamiento de la plataforma, contiene el código HTML/JavaScript para generar la página web que hace las veces de interfaz gráfica de usuario.

El siguiente diagrama de flujo ilustra el funcionamiento general de la aplicación:

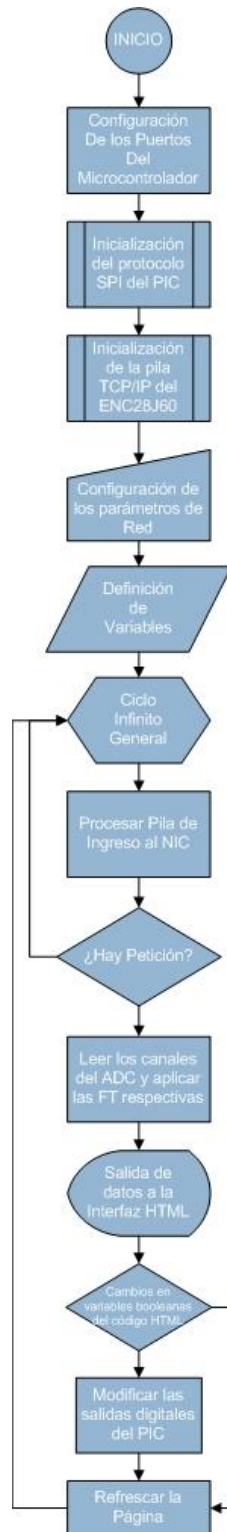


Imagen 21 - Diagrama de Flujo del Firmware

7.1. DESARROLLO DEL SITIO WEB

A pesar de que el código HTML/JavaScript embebido en el microcontrolador, genera una página web accesible a través de los navegadores, la imposibilidad de segmentar los paquetes de datos en esta implementación de la pila TCP/IP, requiere de una interfaz demasiado sencilla, y carente de detalles adicionales, como seguridad, control de acceso, y herramientas web adicionales. Por esta razón, se decidió desarrollar un portal web, bajo una herramienta de código abierto, de tipo CMS^{xxiv}, en el cual, se direcciona un contenedor a la IP real asignada al sistema.

De esta forma, el portal adquiere un carácter multipropósito: Servir como sitio web institucional ó corporativo a la entidad que emplee al microservidor en sus propósitos y agregar a la interfaz gráfica, las características de seguridad y privacidad de las que carece en su forma básica.

Como herramienta para el desarrollo del portal, se empleó la plataforma de administración de contenido JOOMLA, versión 1.5x.

Se modificó la apariencia de la plantilla, con un diseño gráfico básico y se subió la información relevante del laboratorio de Ciencia y Tecnología de Materiales Cerámicos de la Universidad del Cauca.

Se creó una sección de acceso al microservidor, sólo para los usuarios que se registren con un nombre de usuario y contraseña pre configurados por el administrador del sitio.

La siguiente imagen, muestra la página web del microservidor, embebida en el portal desarrollado, dentro de una zona protegida por acceso.



Imagen 22 - Detalle de la página Web con el microservidor web funcionando

El conjunto fué subido en un servidor privado, y se adquirió el nombre de dominio www.cytemac.com para direccionar el sitio y facilitar el acceso a sus usuarios.

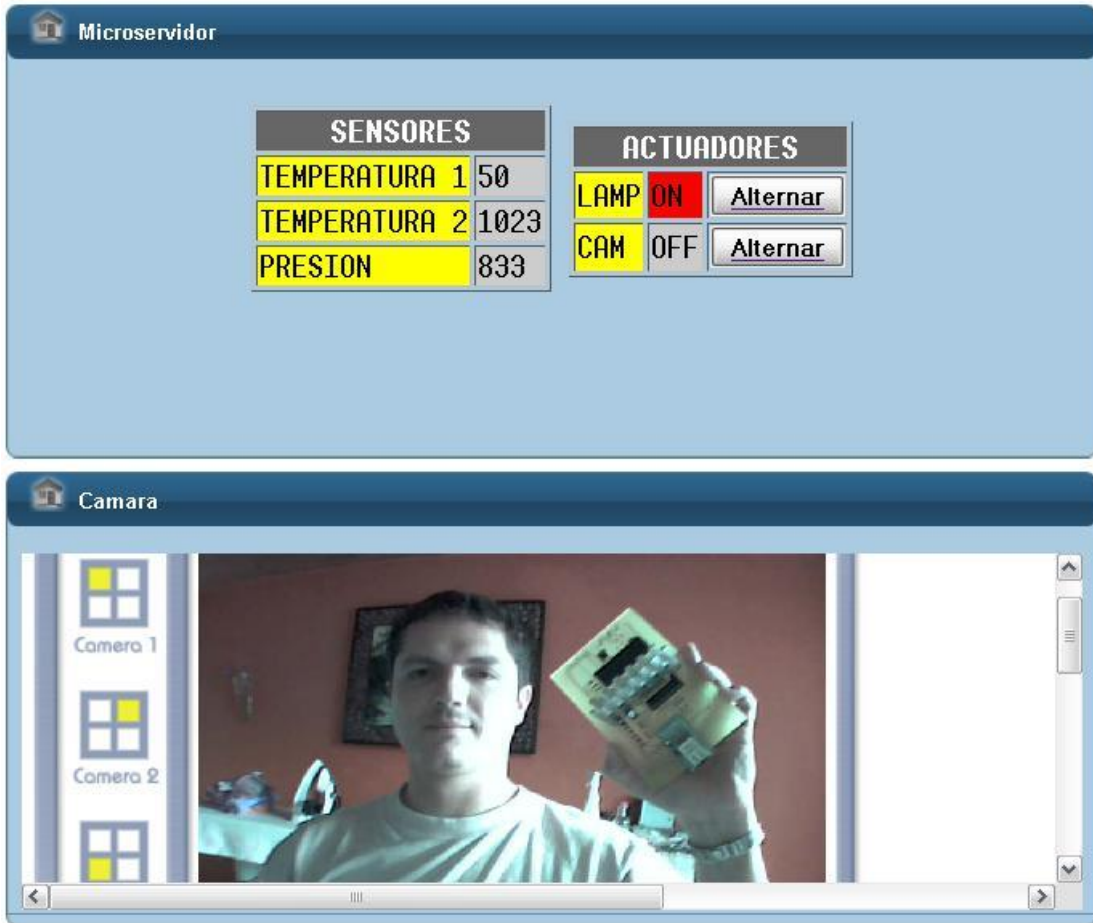


Imagen 23 - Detalle de la cámara y la interfaz del microservidor

8. ANÁLISIS Y RESULTADOS

Las primeras pruebas del sistema, se realizaron sin el empleo del portal desarrollado en Joomla!, y se montó un servidor PHP/SQL en una máquina local. Por lo tanto, los primeros accesos al servidor se hicieron punto a punto desde el equipo local al microservidor.

Se estableció una IP local, de la misma red para el PC y para el servidor: 192.168.1.1 y 192.168.1.10 respectivamente.

Después de unos segundos, en el computador se apreció el establecimiento de la comunicación bidireccional, tras la negociación ejercida por los protocolos de cada parte.

Se intentó el acceso desde varios navegadores diferentes, y se comprobó su funcionalidad desde: *Mozilla Firefox*, *Safari*, *Google Chrome* y *Opera*. Desde *Internet Explorer 7* no hay acceso al microservidor, y desde las versiones 8 y 9 de este navegador, se da el acceso, pero no se ejecutan adecuadamente los comandos JavaScript, por lo que no es posible desde *Internet Explorer*, controlar el estado de las salidas digitales del microcontrolador.

Se mantuvo el funcionamiento por periodos de tiempo superiores a una hora, y se comprobó el correcto desempeño de la totalidad del sistema.

La siguiente fase de pruebas, fue interconectar el microservidor a una pequeña red Ethernet de 3 computadores conectados a un *switch* 10/100.

Las pruebas realizadas en la red, mostraron el acceso simultáneo de los tres computadores al microservidor, sin problemas de congestión, empleando en todos ellos a *Mozilla Firefox* como navegador.

Finalmente, se estableció la salida a internet, desde una conexión privada, a través de un *router*, cuyo puerto 80 fue habilitado para el acceso público.

Con la opción de “servidor virtual” de dicho *router*, se habilitó el acceso al microservidor a través de su IP Dinámica y temporal.

Empleando los servicios de “Servidores de Nombres de Dominio Dinámicos”, que se encargan de mantener actualizada una dirección web, con la dirección dinámica de un *router*, a pesar de que esta cambie por cualquier motivo, se creó una cuenta y se empleó la dirección virtual para acceder al dispositivo a través de internet.

Las pruebas realizadas fueron satisfactorias, ya que los resultados obtenidos desde la Internet, son los mismos que se obtuvieron en las pruebas anteriores sobre las redes LAN.

Se solicitó a la Universidad del Cauca, la asignación de dos direcciones IP Reales, con el fin de poder acceder desde internet al microservidor web y a la cámara IP, cuyas interfaces fueron embebidas en el mismo portal Joomla!.

La División de Sistemas, asignó solo una dirección IP real al proyecto, y una temporal, para la cámara IP, con propósitos únicamente demostrativos.

Al momento de trasladar el sistema a la planta física del laboratorio de CYTEMAC, se descubrió la ausencia de un punto de red en el mismo, y se inició un nuevo trámite con la División de Sistemas para que fuese instalado.

Ante la tardanza en la respuesta, el grupo de desarrolladores decidió demostrar el perfecto funcionamiento del sistema desarrollado, empleando cualquier punto de la red de datos en el proceso de sustentación.

9. APLICACIONES

Es inevitable, ante el desarrollo de una plataforma multipropósito, a pesar de su misión inicial, el cuestionamiento de sus posibles aplicaciones en distintas áreas, para el monitoreo y control. Las siguientes son algunas de aplicaciones más relevantes para los autores del proyecto, aunque las posibilidades son mucho más extensas y algunas inimaginables sin el punto de vista de algún requerimiento particular:

- **Sistema de Monitoreo de Signos Vitales.** Los autores visualizan una plataforma operando en cada una de las habitaciones de una clínica, recibiendo como señales de entrada, signos vitales del paciente: ritmo cardíaco, temperatura corporal, etc. Estos microservidores interconectados a través de una red LAN, entregarían sus datos a un *host* o computador principal, que se encargaría de emitir alarmas y datos de una condición anormal en cualquiera de los pacientes monitoreados.
- **Sistema de Monitoreo y Control para el Hogar.** El sistema podría conectarse a diferentes controles de encendido de dispositivos eléctricos del hogar, y en compañía de la cámara IP, podría permitir el monitoreo integral de diversos factores: Humo, temperatura, luz, etc.
- **Sistema Paralelo de Seguimiento Evolutivo de Variables.** Algunos procesos sobre diferentes materiales, o experimentos de cualquier índole, requieren el análisis permanente de la evolución del proceso, en prolongados periodos de tiempo. La instalación de la plataforma, permite a diferentes espectadores, seguir el desarrollo del proceso en el tiempo, y desde cualquier lugar del mundo en el que puedan tener acceso a internet.
- **Sistemas de Control Distribuido.** Cada estación conformada por un microservidor, permite la adquisición de datos en distintas etapas de un sistema, su procesamiento y control, y la entrega de datos a una central para la consolidación, monitoreo y control general de la planta.

10. CONCLUSIONES

- El microservidor desarrollado, es una completa y robusta plataforma, para el desarrollo de sistemas de monitoreo y control a través de internet.
- La versatilidad del dispositivo desarrollado, permite adaptarle cualquier tipo de sensor de salida analógica, con lo que prácticamente, cualquier variable física puede ser monitoreada a través de la web, o de una red LAN.
- A pesar de que el sistema fue concebido inicialmente como un dispositivo de monitoreo de variables físicas, las salidas digitales y las PWM del microcontrolador, permiten el desarrollo de eficientes rutinas de control para las variables que monitorea.
- La combinación del microservidor, con elementos de seguridad doméstica, como cámaras, sensores de presencia, calor, contactores, marcadores telefónicos, etc., lo convierten en una excelente alternativa, como corazón de una alarma para el cuidado y vigilancia de hogares y oficinas.
- Las pruebas de esfuerzo, realizadas sobre el dispositivo, demostraron su robustez en periodos extensos de conexión permanente, lo que garantiza su operatividad ininterrumpida y su bajo consumo eléctrico, además de su inmunidad al ruido eléctrico en la red AC.
- Un factor en contra de la plataforma, es la inmensa dificultad para conseguir en el mercado nacional al ENC28J60 y el Jack RJ45 con los transformadores incorporados.
- La interacción con cualquier computador conectado a la red o con acceso a internet, no solo permite el acceso al dispositivo a través de la página alojada en su interior, sino que potencializa el desarrollo de aplicaciones complejas en lenguajes de instrumentación como *LabView* o *Matlab*, para incidir en las funciones de transferencia de cada sensor

conectado al microservidor, para adaptarlo fácilmente a una necesidad particular, sin tener que reprogramar el microcontrolador.

- Finalmente, la plataforma puede ser tomada como el punto de partida en el desarrollo de innumerables aplicaciones en el área de instrumentación y control electrónico.

11. BIBLIOGRAFÍA

- 1) **AXELSON**, Jan. Embedded Ethernet and Internet Complete: Designing and Programing Small Devices for Networking. LAKEVIEW RESEARCH LLC. 2003.
- 2) **BENTHAM**, Jeremy. TCP/IP Lean: Web Servers For Embedded Systems. Segunda Edición. CMP BOOKS. 2002.
- 3) **SPURGEON**, Charles E. ETHERNET: The Definitive Guide. O'Reilly. 2000.
- 4) **LEWIS**, Steven. An Introduction To TCP/IP for Embedded System Designers. Z-World. 2001.
- 5) **CATANIA**, Carlos Adrián. TCP/IP Sobre Sistemas Embebidos y su implementación sobre la arquitectura PIC16Fxxxx de Microchip. Universidad Marcelino Champagnat. 2005.
- 6) **LOEWEN**, Myron. Internet Appliance Interface : Circuit Cellar Magazine. Número 108, Julio de 1999, pp 24-34.
- 7) **LOEWEN**, Myron. Application Note AN724. "Using PICmicro MCU's to connect to Internet via PPP". Microchip Technology, [online], <http://www.microchip.com>. Julio 29, 1999.
- 8) **MICROCHIP TECHNOLOGY INC.** USA. 2008, [online], <http://www.microchip.com>.
- 9) **BALL**, Stuart R. Analog Interfacing to Embedded Microprocessors. Newnes. 2001.
- 10) **MIKROELEKTRONIKA INC.** USA. 2009, [online], <http://www.mikroe.com>.

12. REFERENCIAS BIBLIOGRÁFICAS

-
- ⁱ Término adoptado como la traducción al castellano del término del inglés “*internetworking*”
- ⁱⁱ es un término que se utilizó principalmente durante la década de 1990 para describir Internet. Al proyecto oficial se le bautizó como la *Infraestructura de Información Nacional* (NII son sus siglas en inglés) y buscaba la interconexión de ordenadores.
- ⁱⁱⁱ El sistema es capaz de interactuar con cualquier sensor electrónico cuya salida sea: una señal analógica, una señal digital, un valor de frecuencia, una salida PWM, y con adecuaciones electrónicas adicionales, sensores capacitivos, inductivos, piezoeléctricos, etc.
- ^{iv} Abreviatura de *HYPER TEXT TRANSFER PROTOCOL* (Protocolo de transferencia de Hipertexto)
- ^v Abreviatura de *Open System Interconnection*, Modelo de Referencia de Interconexión de Sistemas Abiertos
- ^{vi} Abreviatura de *HyperText Markup Language*
- ^{vii} Abreviatura de Organización Internacional para la Estandarización
- ^{viii} X.25 es un estándar UIT-T para redes de área amplia de conmutación de paquetes
- ^{ix} Abreviatura de *Pulse Code Modulation* en el que se emplean 5V para niveles lógicos de 1 y 0V para niveles lógicos de 0
- ^x La Multiplexación por División de Frecuencias Ortogonales, en inglés *Orthogonal Frequency Division Multiplexing (OFDM)*, también llamada modulación por multitono discreto, en inglés *Discrete Multitone Modulation (DMT)*, es una modulación que consiste en enviar un conjunto de ondas portadoras de diferentes frecuencias donde cada una transporta información, la cual es modulada en QAM o en PSK
- ^{xi} Abreviatura de *Network Interface Card* (Tarjeta de Interfaz de Red)
- ^{xii} Abreviatura de *Mac Address Control* (Control de Acceso al Medio)
- ^{xiii} Abreviatura de *Transfer Control Protocol* (Protocolo de Transferencia de Control)
- ^{xiv} Abreviatura de *User Datagram Protocol*, es un protocolo del nivel de transporte basado en el intercambio de datagramas.
- ^{xv} El *World Wide Web Consortium*, abreviado W3C, es un consorcio internacional que produce recomendaciones para la (www) *World Wide Web*. Está dirigida por Tim Berners-Lee, el creador original de URL (*Uniform Resource Locator*, Localizador Uniforme de Recursos), HTTP (*HyperText Transfer Protocol*, Protocolo de Transferencia de Hipertexto) y HTML (Lenguaje de Marcado de Hipertexto) que son las principales tecnologías sobre las que se basa la Web.
- ^{xvi} El IETF (*Internet Engineering Task Force*, Grupo de Trabajo en Ingeniería de Internet) es una organización internacional abierta de normalización, que tiene como objetivos el contribuir a la ingeniería de Internet, actuando en diversas áreas, tales como transporte, encaminamiento, seguridad. Fue creada en Estados Unidos en 1986.
- ^{xvii} Las *Request For Comments* —petición de comentarios— son una serie de notas sobre Internet que comenzaron a publicarse en 1969. Se abrevian como RFC.
- ^{xviii} Interfaz de entrada común (en inglés *Common Gateway Interface*, abreviado CGI) es una importante tecnología de la *World Wide Web* que permite a un cliente (navegador web) solicitar datos de un programa ejecutado en un servidor web.
- ^{xix} En el contexto de las redes informáticas, el término hace referencia a un programa o dispositivo que realiza una acción en representación de otro.
- ^{xx} El *Document Object Model* (una traducción al español no literal, pero apropiada, podría ser *Modelo en Objetos para la representación de Documentos* o también *Modelo de Objetos del Documento*), abreviado

DOM, es esencialmente una interfaz de programación de aplicaciones que proporciona un conjunto estándar de objetos para representar documentos HTML y XML, un modelo estándar sobre cómo pueden combinarse dichos objetos, y una interfaz estándar para acceder a ellos y manipularlos

^{xxi} www.joomla.org, página oficial de Joomla!

^{xxii} Abreviatura de *Analog To Digital Converter* (Conversor Análogo – Digital)

^{xxiii} El SPI, de *Serial Pheripheral Interface*, es un protocolo de comunicación serial asíncrona que opera en modo “full dúplex”

^{xxiv} Abreviatura de CONTENT MANAGEMENT SYSTEM (Sistema de Administración de Contenidos)