



**FACULTAD DE CIENCIAS DEL HOMBRE Y SU NATURALEZA
LIC. EN CIENCIAS DE LA COMPUTACIÓN**

MATERIA

Introducción al Software Libre

CATEDRÁTICO

Pedro Antonio Trejo Noble

TEMA: Implementar un Servidor Web de Arduino

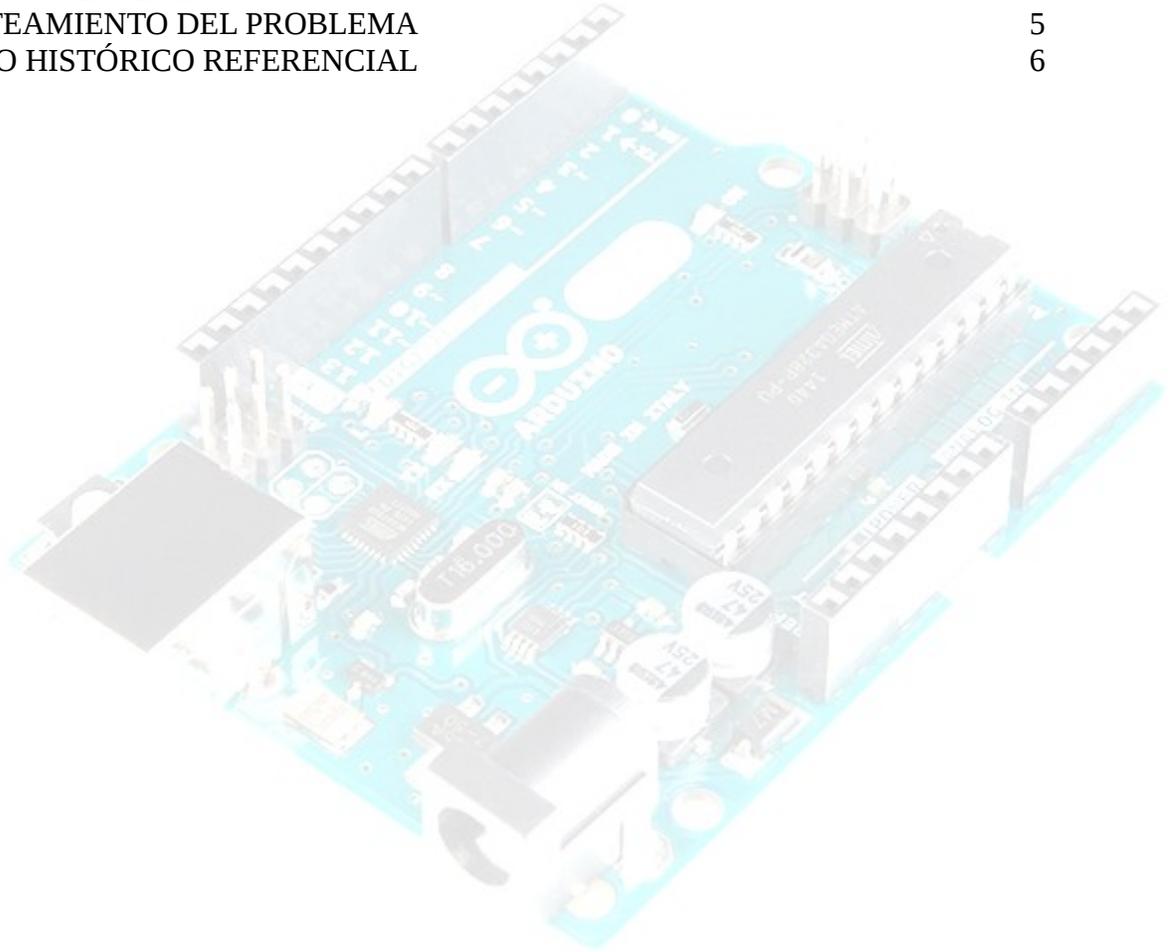
Integrantes:

1. Cesia Beatriz Garcia Diaz
2. Wilber Fernando Ramos
3. Elvis Henry Fabian Martinez

San Salvador 4 de Octubre del 2018

ÍNDICE

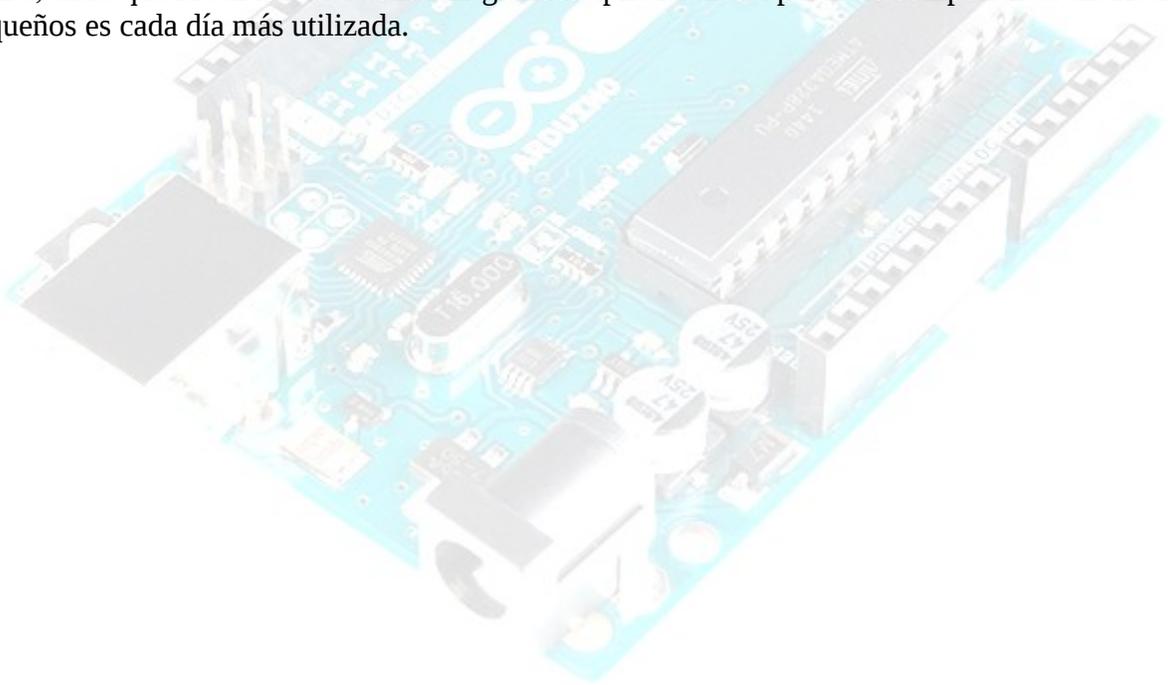
INTRODUCCIÓN	3
OBJETIVOS	4
OBJETIVOS GENERALES	4
OBJETIVOS ESPECÍFICOS	4
PLANTEAMIENTO DEL PROBLEMA	5
MARCO HISTÓRICO REFERENCIAL	6



Introducción

El presente proyecto que se pretende implementar es un Servidor Web con Arduino la idea principal es crear el servidor web pero que este pueda tener un enlace o una conexión con Arduino.

Antes que nada hay que saber que es un servidor web: Un servidor web es una computadora que guarda y transmite datos vía Internet. La información que almacenan es en forma de página Web. Un servicio web es un sistema software diseñado para soportar la interacción máquina a máquina, a través de una red, de forma interpretable. Cuenta con una interfaz descrita en un formato procesable por un equipo informático, a través de la que es posible interactuar con el mismo mediante el intercambio de mensajes. Una tercera razón por la que los servicios Web son muy prácticos es que pueden aportar gran independencia entre la aplicación que usa el servicio Web y el propio servicio. De esta forma, los cambios a lo largo del tiempo en uno no deben afectar al otro. Esta flexibilidad será cada vez más importante, dado que la tendencia a construir grandes aplicaciones a partir de componentes distribuidos más pequeños es cada día más utilizada.



OBJETIVOS GENERALES

- Conocer el lenguaje de programación, aplicando un servidor web y su función en Arduino.

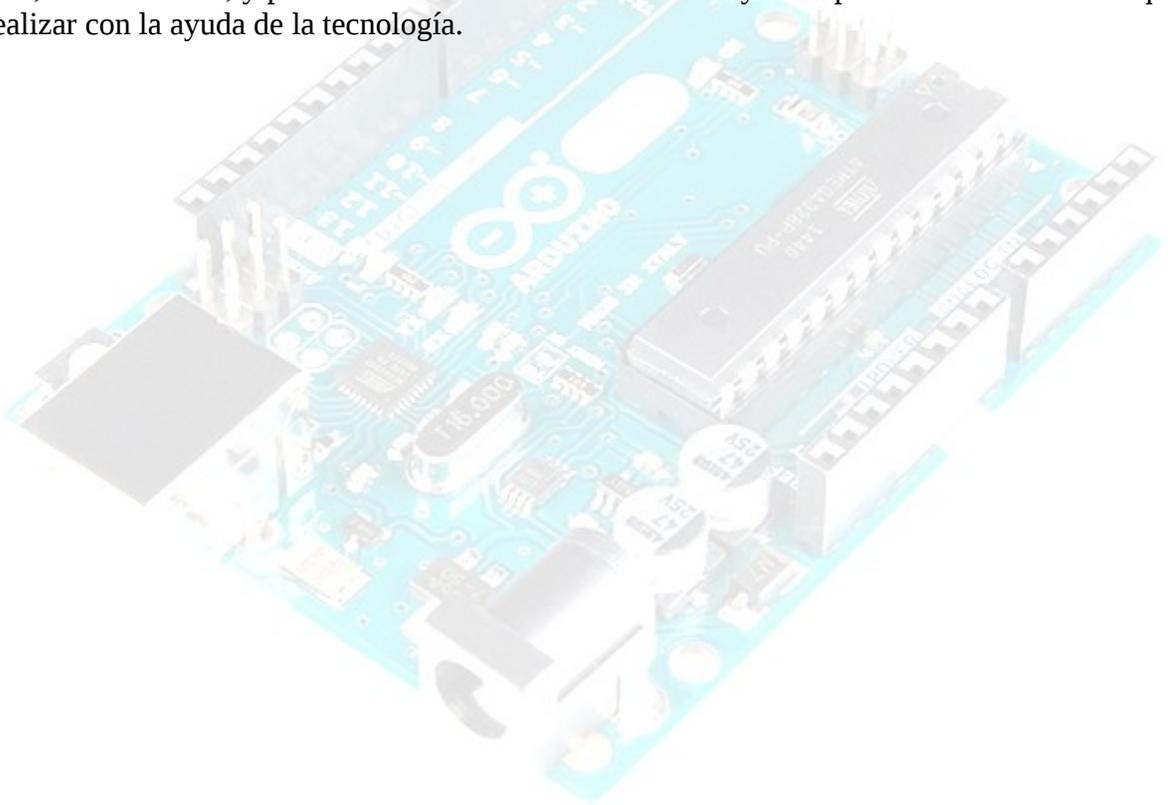
OBJETIVOS ESPECÍFICOS

- Diseñar y Ejecutar un servidor web enlazado con Arduino.
- Implementar los métodos y recursos para programar por medio de un Servidor Web y un Arduino.



PLANTEAMIENTO DEL PROBLEMA

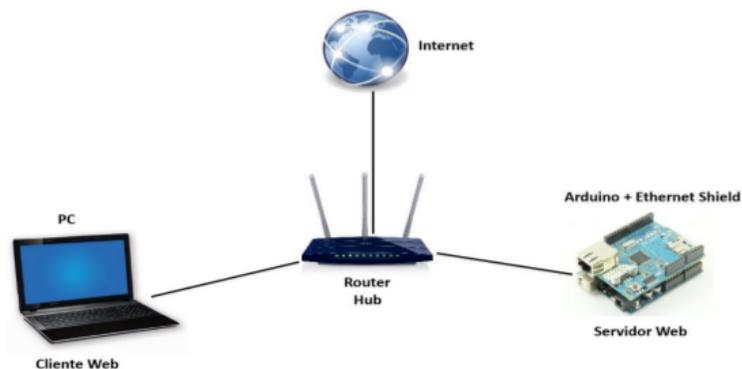
Nosotros como estudiantes elegimos el tema de un servidor web con Arduino porque se requiere de un conocimiento más y una prácticas en la cual aprenderemos a desarrollar un proyecto usando un servidor web que indique instrucciones a un Arduino en la cual estos puedan comunicarse entre sí. El código a aplicar debe contener pasos lógicos y poderse ejecutar sin ningún problema, se ejecutará por medio del lenguaje de programación utilizando los conocimientos adquiridos en clase, poniendo en práctica la lógica y la investigación y condiciones del código a implementar. Ya que para nosotros es necesario poder desarrollar el uso de un servidor para que nuestra comunidad tenga así un mejor estilo de vida para aquellas personas que tienen acceso a estas tecnologías ya sea para las empresas, compañías, universidades, y permitir un correcto funcionamiento y ellos pueda ir conociendo lo que se puede realizar con la ayuda de la tecnología.



MARCO HISTÓRICO REFERENCIAL

Con el crecimiento que han tenido los servicios de internet y su gran expansión, hoy en día, una gran cantidad de hogares de clase media y alta cuentan con una conexión 24/7 para navegar y hacer consultas en muy diversos sitios, según las preferencias de los cibernautas. Por otro lado, los proveedores de equipo cada día van incorporando nuevos dispositivos, que de una u otra forma, mediante diferentes protocolos, pueden comunicarse entre ellos o enlazarse a internet para poner a disposición algún tipo de servicio o información. así es que vemos surgir equipo de marcas como: Arduino, y tecnologías o protocolos. Arduino es una plataforma de hardware libre basada en una sencilla placa de entradas y salidas y un entorno de desarrollo, que implementa el lenguaje de programación Processing/Wiring. Se puede utilizar para desarrollar objetos interactivos autónomos o puede ser conectado a software de la computadora (por ejemplo, macro-media Flash, Processing, max/msP, Puredata). Las placas se pueden montar a mano o adquirirse. el entorno libre de desarrollo integrado, se puede descargar gratuitamente. Las plataformas Arduino están basadas en los microcontroladores atmega328, atmega168 o en el atmega8, chips sencillos y de bajo costo que permiten el desarrollo de múltiples diseños. Al ser open-hardware, tanto su diseño como su distribución son libres, es decir, se puede utilizar libremente para el desarrollo de cualquier tipo de proyecto sin haberse adquirido ninguna licencia.

Diseño de un Servidor



MARCO CONTEXTUAL

La universidad Luterana Salvadoreña y los estudiantes de la materia Introducción de Software Libre queremos desarrollar nos proponemos a desarrollar un proyecto que consiste en la implementando un servidor web a la vez dar a conocer como es la forma de diseñarlo y ejecutarlo para que puedan ser de gran utilidad para todo aquel que quiera aprender a desarrollar un servidor web y puedan realizarlo aun mejor teniendo la libertad de mejorarlo según como lo queremos diseñar.

Para el diseño de la implementación de un servidor Linux deben conocerse todos los aspectos involucrados como la versión más conveniente a instalar en el servidor, es decir aquella versión que brinde al administrador de la red las ventajas, herramientas y aplicaciones que sirvan en la resolución de problemas.

Un sistema Operativo: son es un conjunto de programas que funcionan tras encender una computadora, su función es gestionar programas u aplicaciones, es decir administrar de forma eficiente los recursos de la computadora, a la vez crean la interfaz entre hardware y usuario, lo cual facilita el trabajo del mismo.

Un Servidor: Permite a los usuarios almacenar y acceder a los archivos de un ordenador y los servicios de aplicaciones, que realizan tareas en beneficio directo del usuario final. Es posible que un ordenador cumpla simultáneamente las funciones de cliente y de servidor.

Arduino: Es una plataforma de prototipos electrónica de código abierto (open-source) basada en hardware y software flexibles y fáciles de usar. Está pensado para artistas, diseñadores, como hobby y para cualquiera interesado en crear objetos o entornos interactivos. Arduino puede “sentir” el entorno mediante la recepción de entradas desde una variedad de sensores y puede afectar a su alrededor mediante el control de luces, motores y otros artefactos. El microcontrolador de la placa se programa usando el “*Arduino Programming Language*” (basado en Wiring) y el “*Arduino Development Environment*” (basado en Processing). Los proyectos de Arduino pueden ser autonomos o se pueden comunicar con software en ejecución en un ordenador (por ejemplo con *Flash*, *Processing*, *MaxMSP*, etc.). Las placas se pueden ensamblar a mano o encargarlas preensambladas; el software se puede descargar gratuitamente. Los diseños de referencia del hardware (archivos CAD) están disponibles bajo licencia open-source, por lo que eres libre de adaptarlas a tus necesidades. Arduino recibió una mención honorífica en la sección *Digital Communities* del *Ars Electronica Prix* en 2006

Servidores Web

Los servidores web son aquellos cuya tarea es alojar sitios y/o aplicaciones, las cuales son accedidas por los clientes utilizando un navegador que se comunica con el servidor utilizando el [protocolo HTTP](#) (hypertext markup language).

Básicamente un servidor WEB consta de un interprete **HTTP** el cual se mantiene a la espera de peticiones de clientes y le responde con el contenido según sea solicitado. El cliente, una vez recibido el [código](#), lo interpreta y lo exhibe en pantalla.

Además los servidores pueden disponer de un intérprete de otros lenguajes de [programación](#) que ejecutan código embebido dentro del código **HTML** de las páginas que contiene el sitio antes de enviar el resultado al cliente. Esto se conoce como programación de lado del servidor y utiliza lenguajes como **ASP, PHP, Perl** y **Ajax**. Las ventajas de utilizar estos lenguajes radica en la [potencia](#) de los mismos ejecutando tareas mas complejas como, por ejemplo acceder a [bases de datos](#) abstrayendo al cliente de toda la operación.

Características de un servidor

Servicio de ficheros estáticos: Todos los servidores web deben incluir, al menos, la capacidad para servir los ficheros estáticos que se hallen en alguna parte del disco.

Seguridad y autenticación:

La mayoría de los servidores web actuales permiten controlar desde el programa servidor los aspectos relacionados con la seguridad y la autenticación de los usuarios.

Contenido dinámico:

Uno de los aspectos fundamentales del servidor web elegido es el nivel de soporte que ofrece para servir contenido dinámico. Puesto que la mayor parte del contenido web que se sirve no viene de páginas estáticas, sino que se genera de forma dinámica, y esta tendencia se mueve claramente al alza, el soporte para contenido de tipo dinámico que ofrece un servidor web es uno de los puntos críticos en la elección. La mayor parte de los servidores web ofrecen soporte para CGI (se debe recordar que los CGI son el método más antiguo y sencillo para generar contenido dinámico).

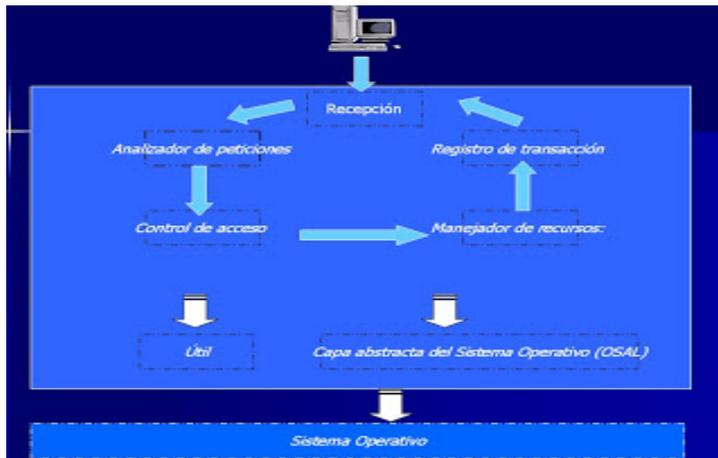
Servidores virtuales:

Una prestación que gana aceptación y usuarios rápidamente, muy especialmente entre los proveedores de servicios de Internet y las empresas de alojamiento de dominios, es la capacidad de algunos servidores web de facilitar múltiples dominios con una única dirección IP, discriminando entre los diferentes dominios alojados en función del nombre de dominio enviado en la cabecera HTTP.

Actuación como representantes:

Algunos servidores permiten su uso como servidores intermedios (proxy servers). Se pueden usar los servidores intermedios para diferentes propósitos: Servir de aceleradores de navegación (uso como proxy-caché). Servir como aceleradores de acceso frontal para un servidor web, instalando diferentes servidores web que repliquen los distintos accesos a un servidor maestro (reverse-proxy o HTTP server acceleration).

ARQUITECTURA



La arquitectura del servidor esta dividido en dos capas:

Capa servidor:

Esta capa contiene *cinco subsistemas*, que son los responsables de implementar la funcionalidad de un servidor Web. Subsistemas:

Subsistema de recepción: representa la primera “línea de ataque” y su labor consiste en esperar las peticiones HTTP de los clientes que llegan por la red. También, analiza las peticiones y determina las capacidades de los navegadores (tipo de navegador, compatibilidad, etc.). Este subsistema contiene la lógica necesaria para manejar múltiples peticiones.

Analizador de peticiones: encargado de traducir la localización del recurso de la red al nombre del archivo local. Por ejemplo, la solicitud del recurso <http://www.mec.es/> se traduce al fichero local `/var/www/webfiles/index.html`.

Control de acceso: sirve para autenticar y permitir el acceso.

Manejador de recursos: este subsistema es el responsable de determinar el tipo de recurso solicitado; lo ejecuta y genera la respuesta.

Registro de transacción: se encarga de registrar todas las peticiones y su resultado.

Capa soporte: Esta capa actúa como una interfaces entre el sistema operativo y el servidor Web y, entre los propios subsistemas de la capa superior.

Subsistemas:

Útil: contiene funciones que son utilizadas por el resto de subsistemas.

Capa abstracta del Sistema Operativo (OSAL): Este subsistema encapsula el funcionamiento específico del sistema operativo para facilitar la portabilidad del servidor Web a diferentes plataformas.

Ventajas de contratar un servidor web dedicado

ApacMayores recursos: si bien se trata de un servidor virtual, al ser dedicado permite utilizar todos los recursos exclusivamente para el proyecto que la empresa necesita. Esto evita caídas por una sobrecarga o falta de espacio para la demanda de usuarios que el sitio web requiera.

- **Escalabilidad:** al trabajar con un servidor web se facilita la incorporación de nuevas herramientas en la medida en que los proyectos lo requieran. Por ejemplo se puede ampliar la capacidad del disco o los recursos disponibles sin tener que contratar otro servidor.
- **Personalización:** contar con un servidor web dedicado permite un mayor control de las configuraciones y la personalización del mismo. Si bien esto puede ser una desventaja para aquellos que no estén familiarizados con el manejo de un servidor, es de mucha utilidad para utilizar al máximo todas las utilidades que posee.

Desventajas de los servidores web dedicados

- **Valor:** el costo que tiene este tipo de servidores es mucho más alto que los servidores compartidos debido a las ventajas y posibilidades que brindan. Sin embargo es necesario hacer una evaluación previa antes de contratar un servidor web dedicado para determinar si es necesario o si alcanza con un servidor compartido.

Conocimientos para configurar el servidor: Al igual que sucede con un servidor dedicado físico, el servidor web requiere de conocimientos para poder configurarlo y utilizarlo. Es un servidor Web de código abierto. Su desarrollo comenzó en febrero de 1995, por Rob McCool. La primera versión apareció en enero de 1996, el apache 1.0. Hacia el 2000, el servidor Web apache era el más extendido en el mundo. El nombre «apache» es un acrónimo de “a patchy Server” (un servidor de remiendos), es decir un servidor construido con código preexistente, piezas y parches de código. Apache 2.0 es más seguro, ya que su configuración predeterminada viene de tal modo que deshabilita muchas cosas que podrán considerarse de cierto riesgo en su configuración.

Servidores

- Procesador
- Intel Celeron 3.20 mhz (Bion student 3200)
- Memoria RAM 1 GB DDR
- Disco duro 160GB

Nginx: (pronunciado en inglés “engine X”) Es un servidor web proxy inverso ligero de alto rendimiento y un proxy para protocolos de correo electrónico IMAP/POP3.

Es un software libre y de código abierto, licenciado bajo la Licencia BSD significada; también existe una versión comercial distribuida bajo el nombre de Nginx Plus. Es multiplataforma, por lo que corre en sistemas tipo Unix (GNUL/Linux, BSD, Solaris, Mac OS X y Wimdown).

El sistema es usado por una larga lista de sitios web conocidos, como: WordPress, Netflix, Hulu, GitHub, Ohloh, SourceForge y partes de Facebook (como el servidor de descarga de archivos Zip).

Nombre



```
Terminal: Archivo Editar Ver Borrar Terminal Ayuda
jimmy@k1v2k1:~$ curl -I http://192.168.1.60
HTTP/1.1 200 OK
Server: nginx/1.14.0 (Ubuntu)
Date: Fri, 13 Jul 2018 16:28:23 GMT
Content-Type: text/html
Content-Length: 622
Last-Modified: Fri, 13 Jul 2018 15:17:19 GMT
Connection: keep-alive
ETag: "5b48c27f264"
Accept-Ranges: bytes
jimmy@k1v2k1:~$
```

Encabezado hTTP devuelto por nginx con curl ejecutado desde ubuntu.

Su creador, el ruso [Igor Sysoev](#), en su página personal desde 2009 escribe el nombre totalmente en minúsculas, mientras que el nombre de la empresa propietaria desde 2011 lo escribe totalmente en mayúsculas, lo cual se corresponde con el nombre que devuelve el encabezado HTTP en todas y cada una de las solicitudes de conexión con que inicia la visita de cada página web.

Para complicar más el asunto el logotipo tiene caracteres tanto en mayúsculas y minúsculas del alfabeto cirílico, no obstante se ha logrado un consenso en denominar *nginx* al servidor web, *NGINX* a los productos y servicios derivados que maneja la empresa y *Nginx* para referirse a ambos en conjunto.

Uso

Originalmente, Nginx fue desarrollado para satisfacer las necesidades de varios [sitios web](#) de [Rambler](#) que recibían unas 500 millones de peticiones al día en septiembre de 2008.

De acuerdo con el estudio de [Netcraft](#), *Netcraft's Jul 2014 Web Server Survey*, nginx es el segundo servidor web más usado en dominios activos (14,35%) superando a Internet Information Server de Microsoft. Además, pasó la marca de ser usado en más de 100 millones de sitios. Para el 29 de mayo de 2018 en el informe actualizado para este mismo estudio, Nginx alcanzó los 359 millones de dominios servidos, a pesar de haber perdido 44 millones con respecto al mes anterior.

1. En febrero de 2017, la adopción de Nginx fue:
2. Argentina: 24,94% del total de dominios.
3. España: 11,51% del total de dominios.
4. México: 13.10% del total de dominios.
5. Chile: 20,44% del total de dominios.
6. Colombia: 16,03% del total de dominios

Nginx vs Nginx Plus

Hay dos versiones de Nginx, OSS Nginx y Nginx Plus. 2014 Nginx Plus ofrece funcionalidades adicionales que no son incluidas en OSS Nginx, como por ejemplo Active Health Checks, persistencia de sesión basada en cookies, integración del servicio de descubrimiento DNS, Api de Purgación de Cache, AppDynamic, Datalog, plug-ins de Dynatrace y New Relic, almacen clave-valor, entre otras.

Comparación con Apache

Nginx fue inicialmente desarrollado con el fin explícito de superar el rendimiento ofrecido por el servidor web Apache. Sirviendo archivos estáticos, Nginx usa dramáticamente menos memoria que Apache, y puede manejar aproximadamente cuatro veces más solicitudes por segundo. Este aumento de rendimiento viene con un costo de disminuida flexibilidad, como por ejemplo la capacidad de anular las configuraciones de acceso del sistema por archivo (Apache logra esto con un archivo. [htaccess](#), mientras que Nginx no tiene desarrollada tal funcionalidad). Anteriormente, incorporar módulos de terceros en Nginx requería recompilar la aplicación fuente con los módulos enlazados estáticamente. Esto fue parcialmente superado en la versión 1.9.11 de Febrero de 2016, con la adición de carga dinámica de módulos. Sin embargo, los módulos aun deben ser compilados al mismo tiempo que Nginx, y no todos los módulos son compatibles con este sistema; algunos requieren el antiguo proceso de enlazado estático.

Características básicas del servidor web

- Servidor de archivos estáticos, índices y autoindexado.
- Proxy Inversión con opciones de caché.
- Balance de carga.
- Tolerancia a Fallos.
- Soporte de HTTP y HTTP2 sobre ssl.
- Soporte para [FastCGI](#) con opciones de caché.
- Servidores virtuales basados en nombre y/o en dirección IP.
- [Streaming](#) de archivos [FLV](#) y [MP4](#).
- Soporte para [autenticación](#).
- Compatible con [IPv6](#)
- Soporte para protocolo [SPDY](#)
- Compresión [gzip](#).
- Habilitado para soportar más de 10.000 conexiones simultáneas

Cómo Funciona NGINX

NGINX está diseñado para ofrecer un bajo uso de memoria y alta concurrencia.

En lugar de crear nuevos procesos para cada solicitud web, NGINX usa un enfoque asíncrono basado en eventos donde las solicitudes se manejan en un solo hilo (single-thread). Por otro lado, una alternativa como el servidor web Apache crearía un hilo separado para cada proceso.

Con NGINX, un proceso maestro puede controlar múltiples procesos de trabajo. El proceso maestro mantiene los procesos de trabajo, y son estos lo que hacen el procesamiento real.

Debido a que NGINX es asíncrono, cada solicitud se ejecuta por el proceso de trabajo de forma concurrente sin bloquear otras solicitudes.

Cómo Funciona NGINX

NGINX está diseñado para ofrecer un bajo uso de memoria y alta concurrencia.

En lugar de crear nuevos procesos para cada solicitud web, NGINX usa un enfoque asíncrono basado en eventos donde las solicitudes se manejan en un solo hilo (single-thread). Por otro lado, una alternativa como el servidor web Apache crearía un hilo separado para cada proceso.

Con NGINX, un proceso maestro puede controlar múltiples procesos de trabajo. El proceso maestro mantiene los procesos de trabajo, y son estos lo que hacen el procesamiento real.

Debido a que NGINX es asíncrono, cada solicitud se ejecuta por el proceso de trabajo de forma concurrente sin bloquear otras solicitudes.

Estadísticas de Uso de NGINX vs Apache

[Apache](#): Es otro popular servidor web de código abierto. En términos de números en bruto, Apache es el servidor más popular que existe y es usado por el 47% de todos los sitios web con un servidor web conocido, según [W3Teach](#).

Si bien Apache es la opción general más popular, **NGINX es, en realidad, el servidor web más popular entre los sitios web con mucho tráfico.**

Cuando se analizan las tasas de uso por tráfico, **NGINX impulsa** el:

- 56.1% de los 100,000 sitios más populares
- 63.2% de los 10,000 sitios más populares
- 57% de los 1,000 sitios más populares

De hecho, NGINX se usan los sitios con los usos más intensivos de recursos que existen, [incluyendo Netflix](#), NASA, e incluso WordPress.com.

El uso de Apache, por otro lado, se mueve en la dirección opuesta a medida que aumente el tráfico web del sitio. Apache impulsa el:

- 27.1% de los 100,000s sitios más populares

- 21.5% de los 10,000 sitios más populares
- 16.2% de los 1,000 sitios más populares

Si vemos los [términos buscados en Google Search](#) desde 2004 podemos ver que Apache ha estado disminuyendo de forma constante, mientras que NGINX ha experimentado un ligero crecimiento.

Aquí podemos ver como esta estructurado el código de la página Web.

```
ServidorWeb
#include <SPI.h>
#include <Ethernet.h>
#include <SD.h>

//-----
//Declaración de la direcciones MAC e IP. También del puerto 80
byte mac[] = {0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED}; //MAC
IPAddress ip(192, 168, 0, 100); //IP
EthernetServer servidor(80);
//-----

void setup()
{
    //Inicializamos el servidor
    Ethernet.begin(mac, ip);
    servidor.begin();
}

void loop()
{
    EthernetClient cliente = servidor.available();

    if (cliente.available())
    {
```

Cliente

ServidorWeb

```
cliente.print("<!DOCTYPE html>");
cliente.print("<html lang='es'>");

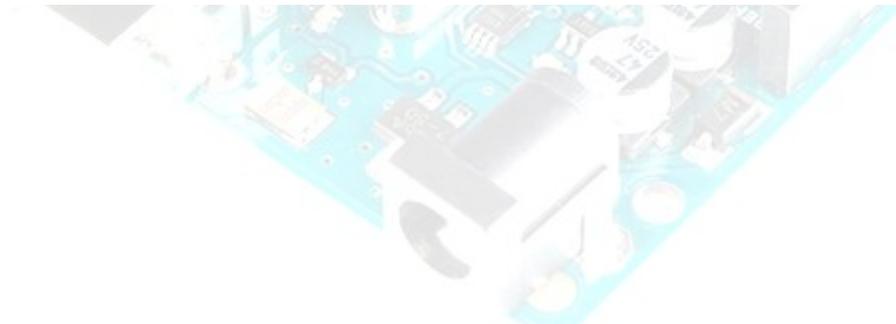
cliente.println("<head>");
cliente.println("<title>Server|Arduino</title>");
cliente.println("<meta charset='utf-8'>");
cliente.println("</head>");

cliente.println("<body style=background-color:powderblue>");
cliente.println("<img src= /softwarelibre.png />");

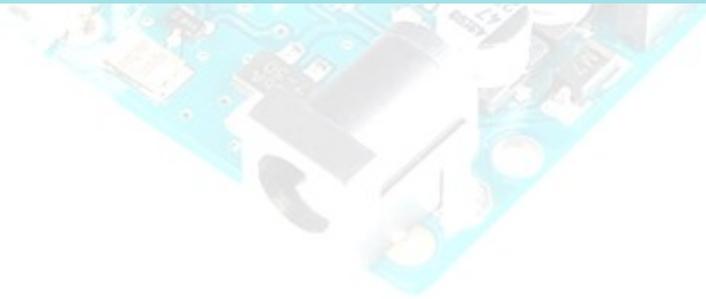
cliente.print("<center>");
cliente.print("<h1>Sevidor WEB con Arduino</h1>");
cliente.print("<br><h3>Introduccion al software libre</h3><br>");
cliente.print("<br>IMPLEMENTADO POR<br>");
cliente.print("<br>Wilber Ramos<br>");
cliente.print("<br>Cesia Garcia<br>");
cliente.print("<br>Elvis Henry Fabian<br>");
cliente.print("</center>");
```

```
cliente.println("</body>");

cliente.println("</html>");
cliente.stop();//Cierro conexión con el cliente
}
}
```



Aqui vemos como nos muestra la pantalla de Bienvenida



GUIA DE INSTALACIÓN DE UN SERVIDOR WEB CON ARDUINO

Para poder realizar el proyecto de servidor web con arduino es necesario contar con algunas herramientas las cuales se detallan a continuación.

1. Una computadora
 - a) Tener instalado en la computador “Arduino IDE”
2. Un Arduino uno
3. Una tarjeta de red para el arduino
4. cable de red para la tarjeta de red del arduino
5. Un router

Teniendo todo lo anterior comenzamos a trabajar:

Para poder instalar Arduino IDE en nuestras computadoras necesitamos descargar el paquete de instalacion desde internet cuando ya lo tengamos descargado lo descomprimimos luego ingresamos a la carpeta que se descomprimio y ejecutamos el siguiente comando

```
./install.sh
```

En nuestra computadora iniciamos el programa Arduino IDE, el cual nos ayudara para digitar nuestro código y a la misma ves poderlo enviar al dispositivo (Arduino).



```
ServidorWeb Arduino 1.8.7
Archivo Editar Programa Herramientas Ayuda
ServidorWeb
#include <SPI.h>
#include <Ethernet.h>
#include <SD.h>

//-----
//Declaración de la direcciones MAC e IP. También del puerto 80
byte mac[] = {0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED}; //MAC
IPAddress ip(192, 168, 0, 100); //IP
EthernetServer servidor(80);
//-----

void setup()
{
  //Inicializamos el servidor
  Ethernet.begin(mac, ip);
  servidor.begin();
}
```

Podemos usar los códigos de ejemplo que ya trae el Arduino IDE siempre y cuando lo que vamos hacer no es muy complejo.



Nuevo	Ctrl+N
Abrir...	Ctrl+O
Abrir Reciente	▶
Proyecto	▶
Ejemplos	▶
Cerrar	Ctrl+W
Salvar	Ctrl+S
Guardar Como...	Ctrl+Mayús+S
Configurar Página	Ctrl+Mayús+P
Imprimir	Ctrl+P
Preferencias	Ctrl+Coma
Salir	Ctrl+Q

En la pestaña archivo podemos ver la opción de Ejemplos ahí podemos encontrar los códigos

```
#include <SPI.h>
#include <Ethernet.h>
#include <SD.h>

//-----
//Declaración de la direcciones MAC e IP. También del puerto 80
byte mac[] = {0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED}; //MAC
IPAddress ip(192, 168, 0, 100); //IP
EthernetServer servidor(80);
//-----

void setup()
{

  //Inicializamos el servidor
  Ethernet.begin(mac, ip);
  servidor.begin();

}
```

```
void loop()
{

EthernetClient cliente = servidor.available();

if (cliente.available())
{

//-----WEB-----

cliente.print("<!DOCTYPE html>");
cliente.print("<html lang='es'>");

cliente.println("<head>");
cliente.println("<title>Server|Arduino</title>");
cliente.println("<meta charset='utf-8'>");
cliente.println("</head>");

cliente.println("<body style=background-color:powderblue>");
cliente.println ("<img src= /softwarelibre.png />");

cliente.print("<center>");
cliente.print("<h1>Sevidor WEB con Arduino</h1>");
cliente.print("<br><h3>Introduccion al software libre</h3><br>");
cliente.print("<br>IMPLEMENTADO POR<br>");
cliente.print("<br>Wilber Ramos<br>");
cliente.print("<br>Cesia Garcia<br>");
cliente.print("<br>Elvis Henry Fabian<br>");
cliente.print("</center>");
```

```
cliente.println("</body>");  
cliente.println("</html>");  
cliente.stop();//Cierro conexión con el cliente  
}  
}
```

Conectamos nuestro arduino al router y de igual forma conectamos nuestra maquina al mismo router en el codigo se establece la MAC y la IP.

Ahora desde un navegador accedamos a la ip que le establecimos al arduino y nos mostrara la pagina que hemos editado en nuestro codigo

