



CENTRO UNIVERSITARIO DE CIENCIAS EXACTAS E INGENIERÍAS

INGENIERIA EN COMUNICACIONES Y ELECTRONICA

Diseño de interfaces I7262 D-02

Dr. Rubén Estrada Marmolejo
Alumno: Aguilar Rodríguez Carlos Adolfo
Código: 215860049

Proyecto Final
Control de Asistencia con RFID

Jueves 30 de Mayo del 2019

Contenido

-Resumen	(3)
-Introducción	(3)
-Palabras Clave	(4)
Diagrama De Flujo	(4)
Diagrama Esquemático	(5)
Desarrollo	(6-19)
-Instalar Nginx	(6)
-Instalar MySQL	(7)
-Instalar PHP	(8)
-Instalar phpMyAdmin	(9)
-Crear HTML	(13)
-Crear AgregarAlumno.php	(15)
-Crear CapturarAsistencia.php	(17)
-Crear verAsistencia.php	(18)
Programar ESP8266	(19)
Código Arduino Lectura-Escritura RFID	(21)
Código Arduino Practica 4 RFID	(23)
Implementación	(27)
Características	(27)
Conclusión	(27)
Referencias	(28)

ACTUALIZACION DE CODIGO PHP PARA CAPTURAR SOLO UNA ASISTENCIA
Ultima pagina



Resumen

Este reporte contiene el proceso para realizar una aplicación con arduino de control de asistencia, conectada a internet con un servidor NGINX, con una página web en HTML y una base de datos mysql en php myadmin.

Para realizar el sistema se requiere los siguientes materiales:

- Sistema Operativo linux ubuntu 18.04.1 LTS
- Servidor Nginx
- Base de datos php myadmin
- Modulo wifi ESP 8266
- Modulo RFID RC522
- Arduino UNO R3 (descargar librerías JSON e instalar controlador del ESP8266)
- Celular con datos y punto de acceso wifi (para usarlo como modem)
- Protoboard

Para comprender mejor como funciona este sistema es necesario tener conocimientos previos en:

- Arduino
- Linux
- C++
- HTML
- JavaScript
- CSS
- PHP
- MySQL
- Comandos AT
- Mensajes JSON
- Comunicación Serial

Introducción

Por medio de un formulario web, será posible dar de alta el registro de alumnos a una base de datos para poder capturar la asistencia de los alumnos registrados por medio de Tags RFID, se contara con una página adicional para mostrar las asistencias registradas con fecha y hora de la asistencia.

Arduino, se comunicara mediante un módulo wifi ESP8266 con la base de datos por medio de mensajes tipo JSON, los mensajes JSON contendrán el código del alumno previamente almacenado en la tarjeta RFID y la estructura para realizar la conexión entre arduino y el servidor web. Si el programa reconoce el UID de la tarjeta y el código de alumno, se realizara una conexión con una base de datos para que se tome una captura del momento en que se detectó y aprobó la tarjeta leída.

Palabras Clave

- Servidor
- Cliente
- JSON
- Nginx
- mysql
- RFID – Tags - Hertz
- Comandos AT
- HTML5
- JavaScript,
- CSS3
- SerialPort
- PHP
- Serialportinfo
- Mapa pixeles
- QString
- Timer
- Sudo

Diagrama de Flujo

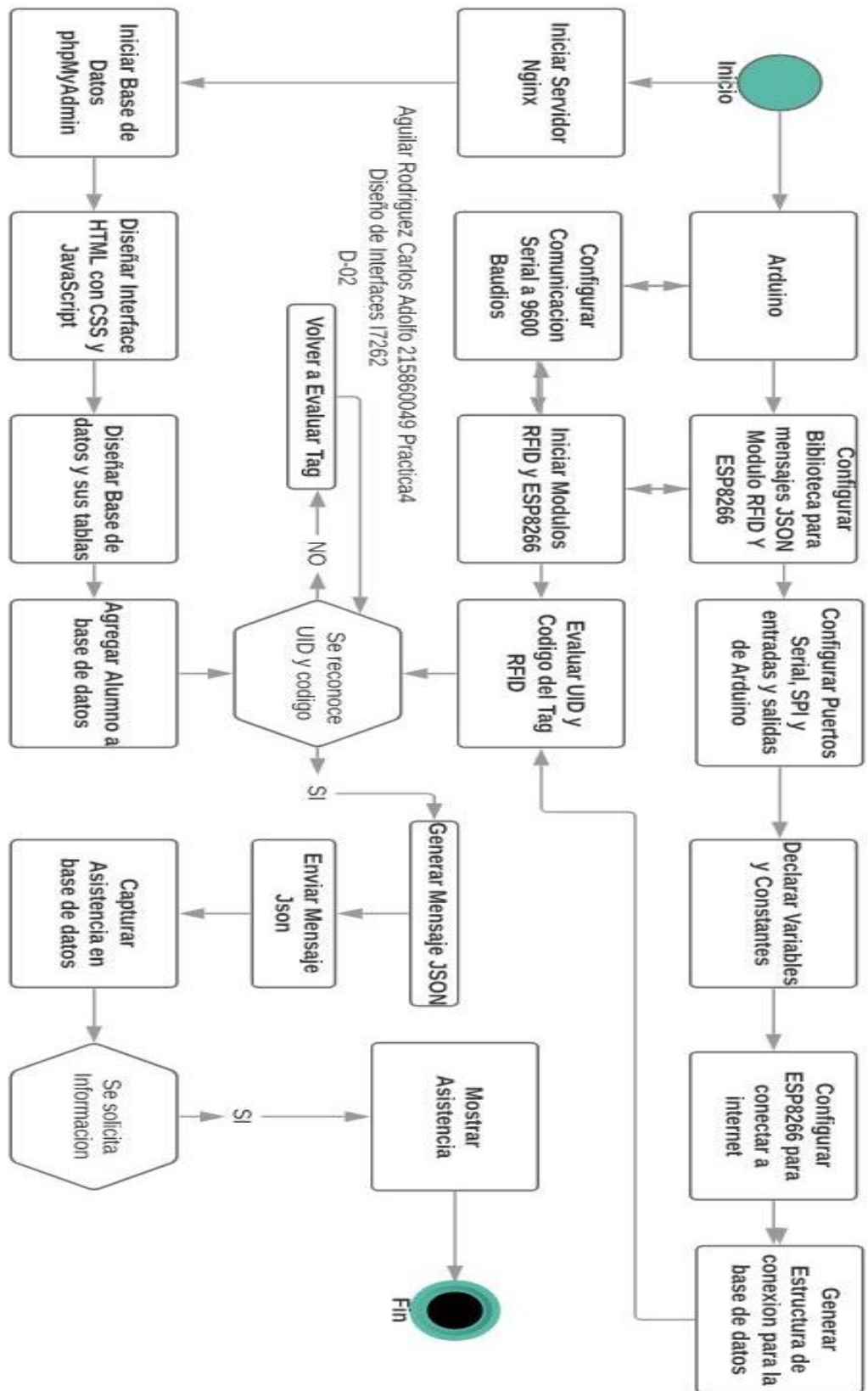
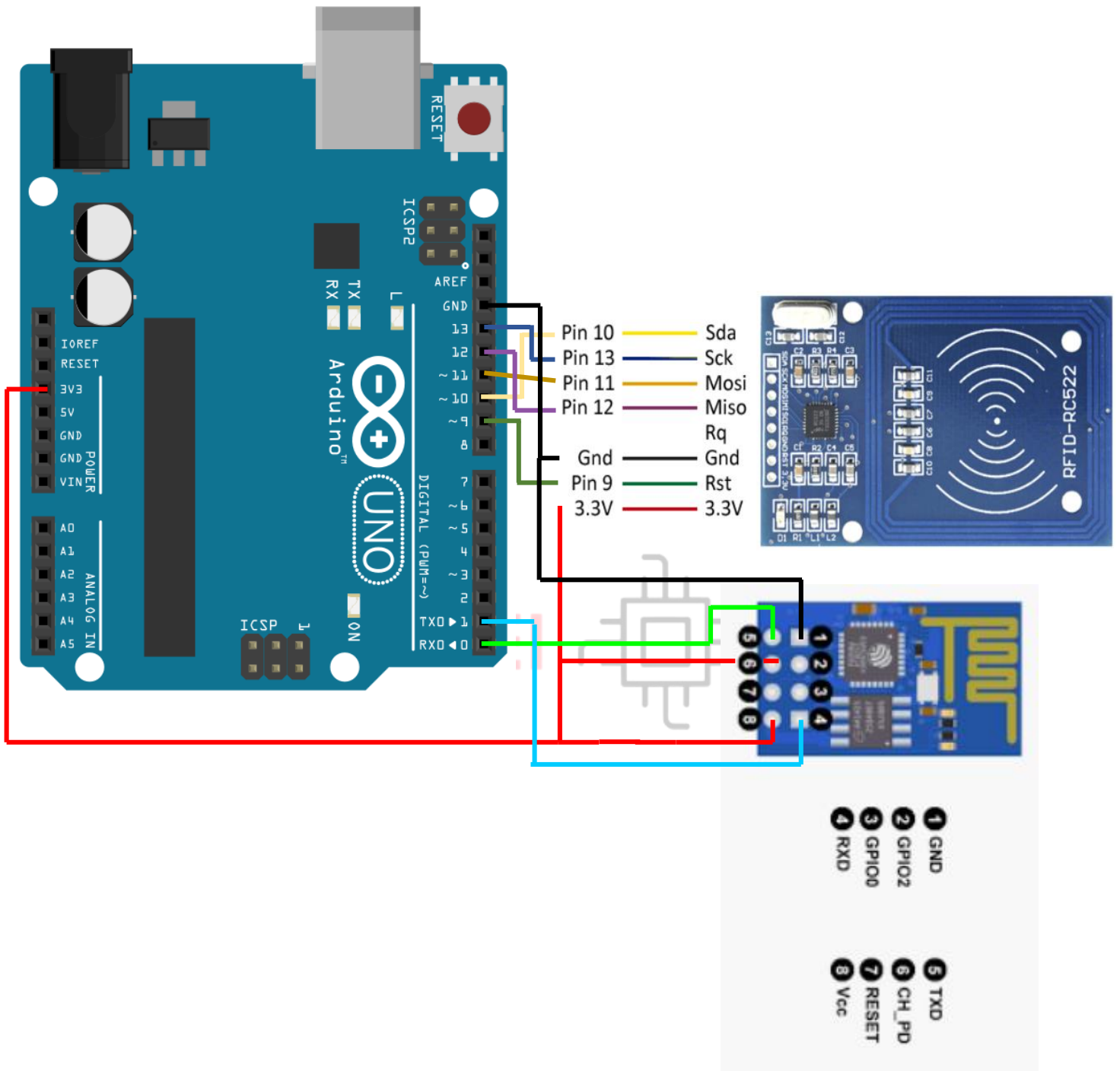


Diagrama Esquemático



Desarrollo

→ Instalar **NGINX**

Nginx es un servidor web de código abierto, actualmente existen varios servidores web similares a nginx los más usados son: apache, lighttpd, litespeed, cherokee, caddy, haProxy

Algunas de las características de nginx son:

Manejo de archivos estáticos, archivos índice y auto-indexación
Balanceo de carga
IPv6
Proxy inverso con cache
Soporte Fast Common Gateway Interface con almacenamiento en cache
Transport Layer Security/Secure Sockets Layer con Server Name Indication

1.- Para instalar nginx usaremos los siguientes comandos.

Ejecutar comandos:

- sudo apt update && sudo apt upgrade
- sudo apt install nginx

2.- Verificar el estado del servidor

Ejecutar el comando:

- sudo service nginx status

3.- Habilitar el firewall

Ejecutar comandos:

- sudo ufw enable
- Reiniciar la computadora
- sudo ufw allow 'Nginx HTTP'

4.- Comprobar que IP está asignada a la conexión de internet de la computadora

Ejecutar el comando:

→ ifconfig (Interface Configuration)

Se mostraran las interfaces de red disponibles del equipo, dependiendo de nuestra conexión a internet buscaremos la IP asignada, como la conexión será por wifi identificamos la dirección IP del bloque wlp2s0 (192.168.42.233)

```
muthe@shiba:~$ ifconfig
enp0s25: flags=4099<UP,BROADCAST,MULTICAST> mtu 1500
    ether 00:23:18:0c:a0:8e txqueuelen 1000 (Ethernet)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 0 bytes 0 (0.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
    device interrupt 20 memory 0xd4600000-d4620000

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Bucle local)
    RX packets 624 bytes 54951 (54.9 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 624 bytes 54951 (54.9 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

wlp2s0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.43.233 netmask 255.255.255.0 broadcast 192.168.43.255
    inet6 fe80::5d7d:ac5a:5fb8:6d58 prefixlen 64 scopeid 0x20<link>
    ether 00:27:10:8a:32:a4 txqueuelen 1000 (Ethernet)
    RX packets 86703 bytes 118556566 (118.5 MB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 43332 bytes 4932852 (4.9 MB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

Proyecto Final

5.-Ingresar al servidor NGINX

- En un navegador Web ingresar la dirección IP del paso anterior debe aparecer lo siguiente

Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to nginx.org. Commercial support is available at nginx.com.

Thank you for using nginx.

6.- Instalar

MySQL es un sistema de administración de base de datos fue escrito en C y C++ compatible con JSON php, perl, java, Linux, Unix y Windows su uso es gratuito y de código abierto por lo que se permite descargar su código fuente para poder realizar modificaciones y contribuir a las actualizaciones de la comunidad, mySQL es una herramienta muy importante para el desarrollo web.

Algunas de las empresas que manejan mySQL son: Google, Adobe, Alcatel Lucent, Zappos Facebook entre otros

Ejecutar comandos:

- sudo apt install mysql-server
Indicar contraseña de mySQL (51 es mi contraseña)

Configurar mySQL

- sudo mysql_secure_installation
Contestar las configuraciones con un Yes o un No de la siguiente manera

¿Quieres instalar el validador de contraseñas?

No

¿Quieres cambiar la contraseña asignada al superusuario?

No

¿Quieres remover a los usuarios anonimos?

Yes

¿Quieres deshabilitar el logeo remoto del superusuario root?

Yes

¿Quieres remover la base de datos de prueba y su acceso?

Yes

¿Quieres actualizar los privilegios de las tablas ahora?

Yes

Verificar la instalación

- sudo mysqladmin -p -u root versión

```
muthe@shiba:~$ sudo mysqladmin -p -u root version
[sudo] contraseña para muthe:
Enter password:
mysqladmin Ver 8.42 Distrib 5.7.26, for Linux on x86_64
Copyright (c) 2000, 2019, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Server version          5.7.26-0ubuntu0.18.04.1
Protocol version        10
Connection              Localhost via UNIX socket
UNIX socket             /var/run/mysqld/mysqld.sock
Uptime:                 7 hours 9 min 56 sec

Threads: 1 Questions: 10 Slow queries: 0 Opens: 105 Flush tables: 1 Open tables: 98 Queries
per second avg: 0.000
```

Proyecto Final

7.- Instalar

PHP (Hypertext Preprocessor) es un lenguaje de código abierto para el desarrollo web y puede ser combinado con HTML, php se ejecuta en el servidor generando HTML y enviándolo al cliente, php se enfoca en recopilar datos de formularios, enviar y recibir cookies, scripts del lado del servidor, scripts desde la línea de comandos, generar paginas con contenidos dinámicos, php admite la mayoría de servidores web, php funciona tanto como modulo como procesador de Common Gateway Interface (Procesador de entrada comun), php también cuenta con soporte para comunicarse con otros servicios usando protocolos como LDAP, IMAP, SNMP, NNTP, POP3, HTTP, COM.

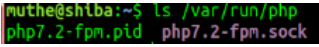
Ejecutar el comando:

→ sudo apt install php-fpm php-mysql

8.- Configurar PHP

Buscar el socket de php

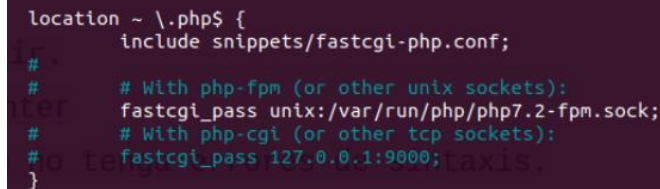
Ejecutar comandos:

→ ls /var/run/php 
→ sudo nano /etc/nginx/sites-available/default

Agregar index.php después de index

```
location /phpmyadmin {  
    root /usr/share/  
    index index.php index.html index.htm;  
    location ~ ^/phpmyadmin/(.+\.php)$ {  
        try_files $uri =404;  
        root /usr/share/  
        fastcgi_pass unix:/run/php/php7.2-fpm.sock;  
        fastcgi_index index.php;  
        fastcgi_param SCRIPT_FILENAME $document_root$fastcgi_script_name;  
        include /etc/nginx/fastcgi_params;  
    }  
}
```

Buscar fastcgi_pass unix:/var/run/php/php7.2-fpm.sock; y quitarle el símbolo #



```
location ~ \.php$ {  
    include snippets/fastcgi-php.conf;  
    #  
    # With php-fpm (or other unix sockets):  
    fastcgi_pass unix:/var/run/php/php7.2-fpm.sock;  
    #  
    # With php-cgi (or other tcp sockets):  
    # fastcgi_pass 127.0.0.1:9000;  
}
```

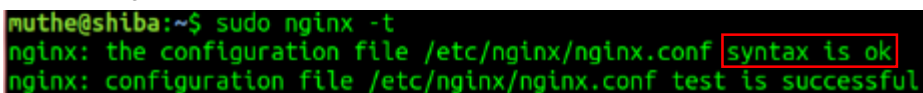
Guardar con Control +

X, dar la opción si/yes y presionar enter.

Verificar que el archivo no tenga error de sintaxis

Ejecutar el comando:

→ sudo nginx -t



```
muthe@shiba:~$ sudo nginx -t  
nginx: the configuration file /etc/nginx/nginx.conf syntax is ok  
nginx: configuration file /etc/nginx/nginx.conf test is successful
```

Si no tenemos errores, volver a cargar la configuración:

Ejecutar el comando:

→ sudo service nginx reload

Proyecto Final

Crear el siguiente archivo

Ejecutar el comando:

→ `sudo nano /var/www/html/info.php`

Agregar:

```
<?php
    php.info();
?>
```

9.-Instalar



Es una herramienta de software gratuita con interfaz grafica escrita en php para mejorar la administración de mySQL a través de paginas web, permite crear, editar, añadir exportar e importar y eliminar base de datos. Phpmyadmin es una alternativa a Microsoft SQL server phpPgAdmin o PostgreSQL, Adminer entre otros.

Adminer es conocido como *phpMinAdmin* y contiene las características mas importantes de *phpmyadmin* pero consiste en un solo archivo *php*

Características:

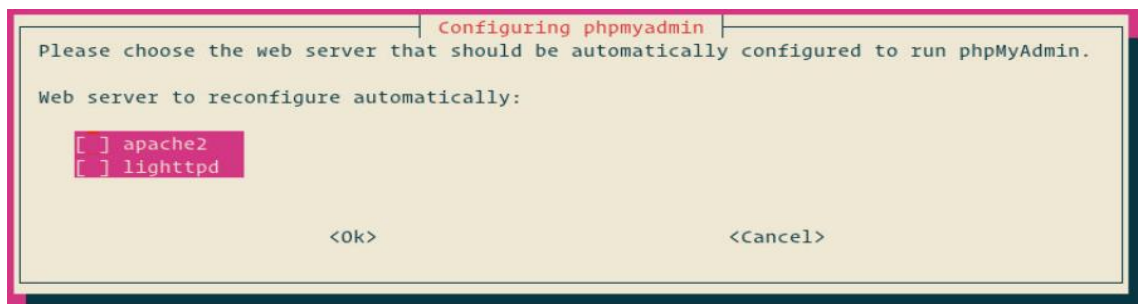
- Navega y suelta bases de datos, tablas, vistas, campos e índices.
- Crea, copia, suelta, renombra y modifica bases de datos, tablas, campos e índices.
- Servidor de mantenimiento, bases de datos y tablas, con propuestas de configuración del servidor.
- Ejecute, edite y marque cualquier declaración SQL , incluso consultas por lotes
- Administrar las cuentas de usuario y los privilegios de MySQL
- Gestionar procedimientos almacenados y disparadores.
- Importar datos desde CSV y SQL
- Exportación de datos a varios formatos: CSV , SQL , XML , PDF , ISO / IEC 26300 Texto y hoja de cálculo de OpenDocument, Word , L^AT_EX
- Administrar múltiples servidores
- Creando gráficos del diseño de su base de datos en varios formatos.
- Creación de consultas complejas mediante consulta por ejemplo (QBE)
- Búsqueda global en una base de datos o un subconjunto de ella
- Transformar datos almacenados en cualquier formato usando un conjunto de funciones predefinidas, como mostrar datos BLOB como imagen o enlace de descarga

Para instalar phpmyadmin

Ejecutar el comando

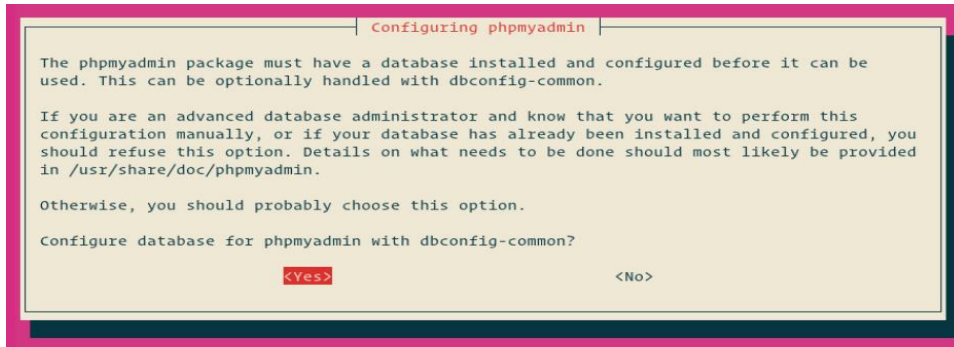
```
sudo apt install phpmyadmin
```

Aparece una ventana como la siguiente para seleccionar servidor que vamos a usar, como usaremos Nginx y no aparece en la opciones, presionar la tecla de Tabulacion y selecciona OK.



Proyecto Final

En la siguiente ventana que se muestra seleccionar No



10.- Crear un usuario para la base de datos mysql
Ejecutar los siguientes comandos:

→ sudo mysql

El siguiente comando lleva sin comillas dobles los datos del usuario

→ CREATE USER 'Nombre de usuario' IDENTIFIED BY 'contraseña';

→ GRANT ALL PRIVILEGES ON *.* TO 'Nombre de usuario' WITH GRANT OPTION;

→ exit (Presionar enter)

11.- Crear archivo de configuración para phpmyAdmin

→ sudo gedit /etc/nginx/snippets/phpmyadmin.conf

Agregar el siguiente texto

```
location /phpmyadmin {
    root /usr/share/;
    index index.php index.html index.htm;
    location ~ ^/phpmyadmin/(.+\.php)$ {
        try_files $uri =404;
        root /usr/share/;
        fastcgi_pass unix:/run/php/php7.2-fpm.sock;
        fastcgi_index index.php;
        fastcgi_param SCRIPT_FILENAME $document_root$fastcgi_script_name;
        include /etc/nginx/fastcgi_params;
    }
    location ~* ^/phpmyadmin/(.+\. (jpg|jpeg|gif|css|png|js|ico|html|xml|txt))$ {
        root /usr/share/;
    }
}
```

Guardar y salir

Proyecto Final

12.- Modificar el siguiente archivo

→ sudo nano /etc/nginx/sites-available/default

Agregar: `include snippets/phpmyadmin.conf;`

después de default_server y antes de #SSL configuration

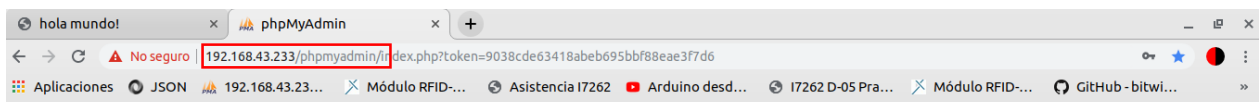
```
#  
server {  
    listen 80 default server;  
    listen [::]:80 default server;  
    include snippets/phpmyadmin.conf;  
    # SSL configuration
```

Ahora recargamos el servidor nginx

→ sudo nginx -t

→ sudo service nginx reload

13.- Ingresar a phpmyadmin (`DireccionIP/phpmyadmin`)



phpMyAdmin
Bienvenido a phpMyAdmin

Idioma - Language
Español - Spanish

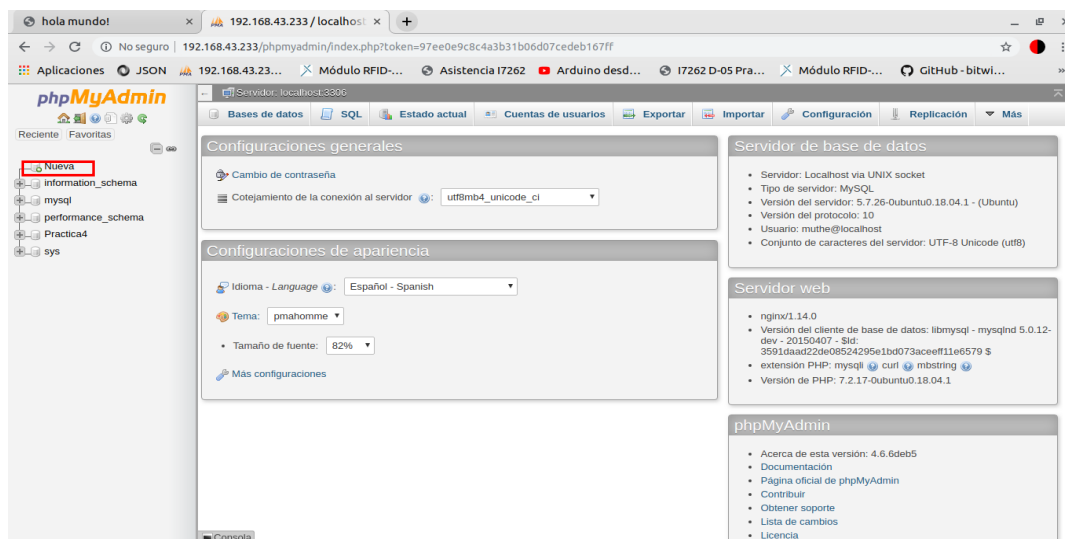
Iniciar sesión

Usuario:

Contraseña:

Continuar

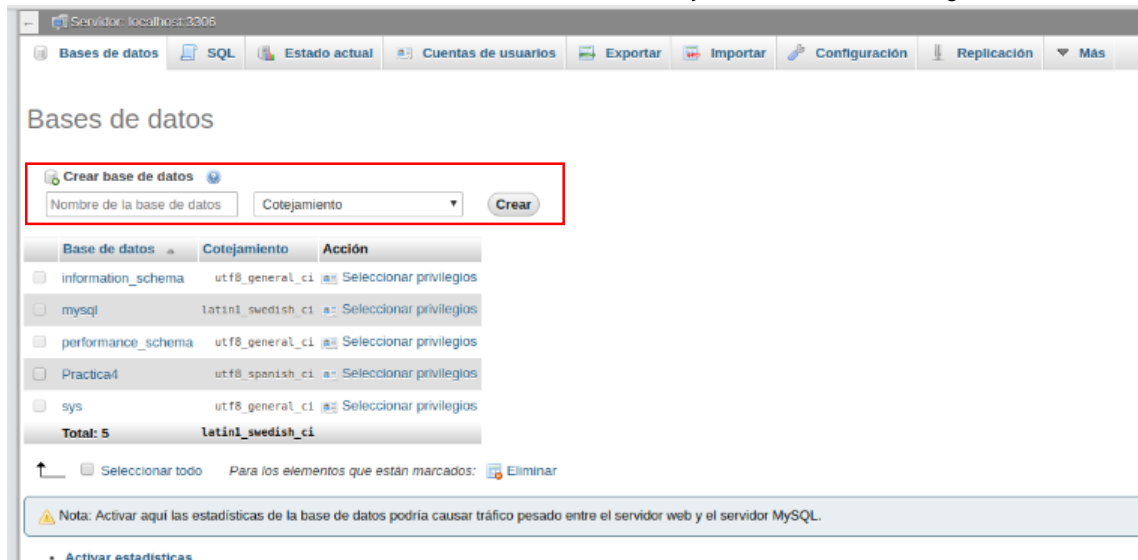
Ingresar usuario y contraseña del paso 10, si todo fue realizado correctamente debe aparecer una ventana como esta:



Para crear una base de datos Nueva, seleccionas la pestaña Nueva

Proyecto Final

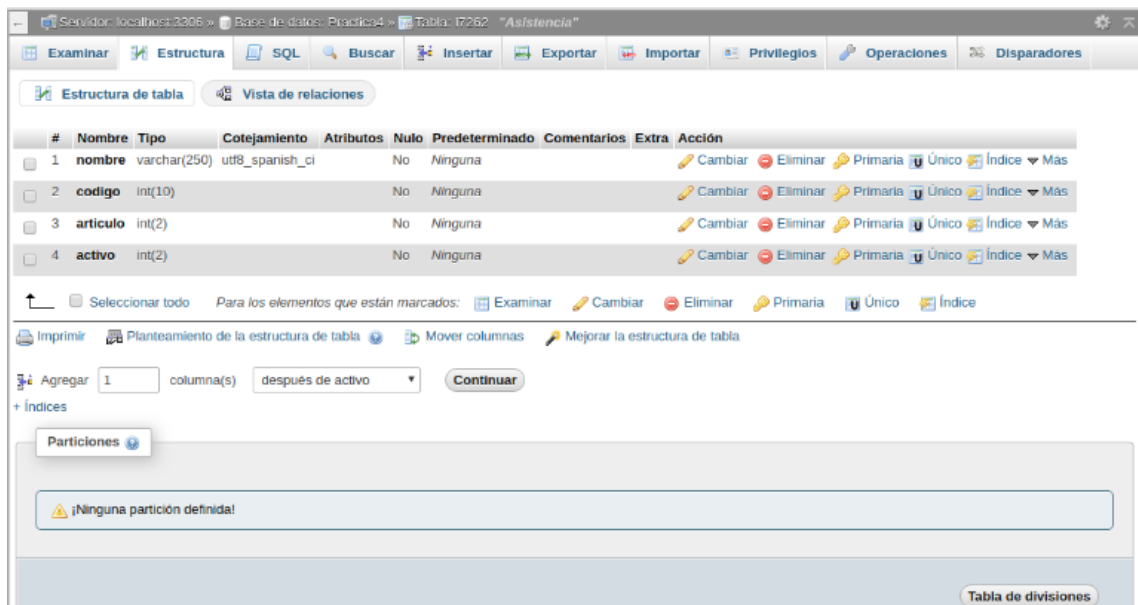
Al seleccionar una nueva base de datos la interface de myadmin debería ser la siguiente



Para crear la nueva base de datos introducimos el nombre de la nueva base de datos y seleccionamos el tipo de cotejamiento, el cotejamiento es el tipo de codificación que queremos darle a nuestra base de datos, esto quiere decir que depende la lengua y simbología de la misma; debemos especificar el idioma y codificación adecuada a la zona del usuario aquí el cotejamiento se aplica a toda la estructura de la base de datos.

Una vez creada la base de datos ingresamos lo datos necesarios para el registro de los alumnos, (Nombre, Código, Artículo, Activo) por medio de un formulario en una página web.

Aquí podemos editar el tipo de variable, el cotejamiento por separado, atributos, comentarios, Acciones, eliminar datos, otorgar permisos, y mas



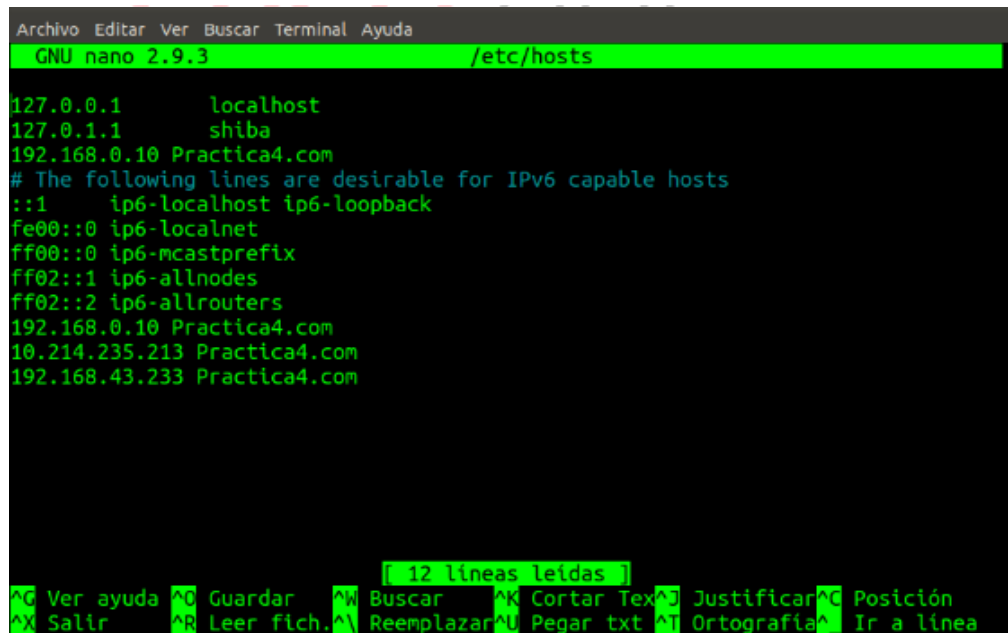
14.- Crear bloque HTML

Ejecutar los siguientes comandos:

- Crear archivo en el siguiente directorio
`sudo mkdir -p /var/www/NombreDeLaPagina.com/public_html`
- Otorgar permisos al archivo
`sudo chown -R $(whoami):$(whoami)/var/www/NombreDeLaPagina.com/public_html`
- Otorgar permisos a la carpeta
`sudo chmod -R 755 /var/www`

Proyecto Final

- Escribir "Hola mundo" en el archivo creado con el siguiente comando
sudo echo "Hola mundo!" > /var/www/**NombreDeLaPagina.com**/public_html/index.html
- Copiar la página creada al siguiente directorio
sudo cp /etc/nginx/sites-available/default/etc/nginx/sites-available/ **NombreDeLaPagina.com**
- Modificar el siguiente archivo
sudo nano /etc/nginx/sites-available/**NombreDeLaPagina.com**
- Buscar entre las líneas de código cambiar lo siguiente:
root /var/www/**NombreDeLaPagina.com**/public_html;
server_name **NombreDeLaPagina.com** www.**NombreDeLaPagina.com**
- Salir.
- Ejecutar:
sudo nginx -t
- Habilitar el uso de la pagina con el servidor
sudo ln -s /etc/nginx/sites-available/**NombreDeLaPagina.com**/etc/nginx/sites-enabled/
- Remover los sitios por default
sudo rm /etc/nginx/sites-enabled/default
- Agregar al archivo hosts lo siguiente
Ejecutar el comando
sudo nano /etc/hosts
- Ahora ingresar: **DireccionIP NombreDeLaPagina.com**
- Por ejemplo:
Se deben agregar los posibles dirección IP del servidor al conectarse desde redes distintas



```
Archivo  Editar  Ver  Buscar  Terminal  Ayuda
GNU nano 2.9.3 /etc/hosts
127.0.0.1    localhost
127.0.1.1    shiba
192.168.0.10 Practica4.com
# The following lines are desirable for IPv6 capable hosts
::1        ip6-localhost ip6-loopback
fe00::0    ip6-localnet
ff00::0    ip6-mcastprefix
ff02::1    ip6-allnodes
ff02::2    ip6-allrouters
192.168.0.10 Practica4.com
10.214.235.213 Practica4.com
192.168.43.233 Practica4.com

12 líneas leídas
Ver ayuda  Guardar  Buscar  Cortar Text  Justificar  Posición
Salir  Leer fich.  Reemplazar  Pegar txt  Ortografía  Ir a línea
```

Presionar Control+X (Para salir), Como realizamos cambios, nos preguntara si deseamos guardar dichos cambios seleccionar Si o Yes. A continuación nos permite modificar el nombre, lo conservamos y Presionar enter para confirmar.

- Reiniciar el servidor nginx
sudo service nginx restart

CREAR ARCHIVO HTML



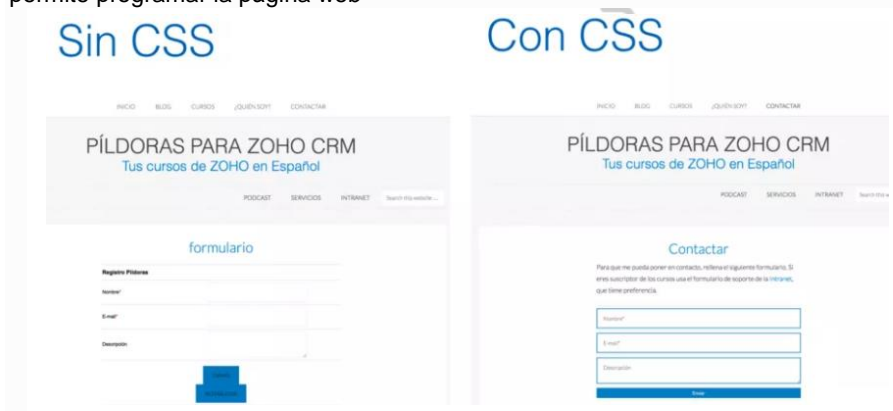
HTML (Hyper Text Markup Language)

Estructura básica de una página web

```
<HTML>
<HEAD>
  <TITLE>
    Título de la página
  </TITLE>
</HEAD>
<BODY>
  Aquí iría el contenido de la página
</BODY>
</HTML>
```

HTML nos permite construir la página web y es muy parecido a otros lenguajes de programación. HTML funciona con bloques de instrucciones dentro de las etiquetas, las etiquetas tienen una función parecida a las llaves de inicio y cierre de instrucción. Solo que aquí el nombre de la etiqueta determina el nombre de la instrucción que se desea ejecutar.

HTML puede funcionar por sí mismo, pero es mejor combinarlo con otros lenguajes complementarios como CSS (Cascading Style Sheets "Hojas en estilo cascada") el cual es un lenguaje de diseño gráfico que nos permite darle mayor estética y una visualización más llamativa a nuestra página web y también JAVASCRIPT, javascript se ejecuta en el cliente es decir en el ordenador donde se accede a la web no como en php que se ejecutaba en el servidor del host, javascript es un lenguaje orientado a objetos, específicamente para desarrollo web nos permite programar la página web



15.-Abrir el siguiente archivo ejecutando el comando:

→ `sudo gedit /var/www/NombreDeLaPagina.com/public_html/index.html`

Copiar y Pegar el siguiente texto:

```
<!DOCTYPE html>
<html lang="en" class="no-js">
  <head>
    <meta charset="UTF-8" />
    <meta http-equiv="X-UA-Compatible" content="IE=edge,chrome=1">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link rel="stylesheet" type="text/css" href="vendor/bootstrap/css/bootstrap.min.css">
    <link rel="stylesheet" type="text/css" href="fonts/font-awesome-4.7.0/css/font-awesome.min.css">
    <link rel="stylesheet" type="text/css" href="vendor/animate/animate.css">
    <link rel="stylesheet" type="text/css" href="vendor/css-hamburgers/hamburgers.min.css">
    <link rel="stylesheet" type="text/css" href="vendor/animation/css/animation.min.css">
    <link rel="stylesheet" type="text/css" href="vendor/select2/select2.min.css">
    <link rel="stylesheet" type="text/css" href="vendor/daterangepicker/daterangepicker.css">
    <link rel="stylesheet" type="text/css" href="css/util.css">
```

Proyecto Final

```
<link rel="stylesheet" type="text/css" href="css/main.css">
<title>I7262 D-05 Practica 4</title>
<meta name="description" content="Control de Acceso con RFID" />
<link rel="stylesheet" type="text/css" href="css/normalize.css" />
<link rel="stylesheet" type="text/css" href="css/demo.css" />
<link rel="stylesheet" type="text/css" href="css/component.css" />
<script src="js/modernizr.custom.js"></script>
</head>
<body>
<div class="container">
<ul id="gn-menu" class="gn-menu-main">
<li class="gn-trigger">
<a class="gn-icon gn-icon-menu"><span>Menu</span></a>
<nav class="gn-menu-wrapper">
<div class="gn-scroller">
<ul class="gn-menu">

<li>
<a href="/agregarAlumnos.php" class="gn-icon gn-icon-download">Agregar alumnos</a>
</li>
<li>
<a href="/capturaAsistencia.php" class="gn-icon gn-icon-download">Capturar
Asistencia</a>
</li>
<li>
<a href="/verAsistencia.php" class="gn-icon gn-icon-download">Mostrar Asistencia</a>
</li>
</ul>
</div><!-- /gn-scroller -->
</nav>
</li>
<li><a href="http://IP_Servidor">Inicio</a></li>
</ul><header>
<h1>Practica 4 <span> Control de Acceso con RFID y <a href="https://www.arduino.cc/">
Arduino</a> </span></h1>
<form class="contact100-form validate-form">
<span class="contact100-form-title">
</span>
</form>
</header>
</div>
<div id="dropDownSelect1"></div>
<script src="vendor/jquery/jquery-3.2.1.min.js"></script>
<script src="vendor/animation/js/animation.min.js"></script>
<script src="vendor/bootstrap/js/popper.js"></script>
<script src="vendor/bootstrap/js/bootstrap.min.js"></script>
<script src="vendor/select2/select2.min.js"></script>
<script src="vendor/daterangepicker/moment.min.js"></script>
<script src="vendor/daterangepicker/daterangepicker.js"></script>
<script src="vendor/countdown/countdown.js"></script>
<script src="js/main.js"></script>
<script src="js/classie.js"></script>
<script src="js/gnmenu.js"></script>
<script>
window.dataLayer = window.dataLayer || [];
function gtag(){dataLayer.push(arguments);}
gtag('js', new Date());
gtag('config', 'UA-23581568-13');
new gnMenu( document.getElementById( 'gn-menu' ) );
</script>
</body>
</html>
```

Proyecto Final

CREAR ARCHIVO AgregarAlumno.php

16.-Abrir el siguiente archivo ejecutando el comando:

→ sudo gedit /var/www/**NombreDeLaPagina.com**/public_html/agregarAlumnos.php

Copiar y pegar el siguiente texto:

```
<!DOCTYPE html>
<html lang="en" class="no-js">
<head>
<meta charset="UTF-8" />
<meta http-equiv="X-UA-Compatible" content="IE=edge,chrome=1">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Practica 4 I7262 Arduino con RFID 215860049</title>
<meta name="description" content="A sidebar menu as seen on the Google Nexus 7
website" />
<meta name="keywords" content="google nexus 7 menu, css transitions, sidebar, side
menu, slide out menu" />
<meta name="author" content="Codrops" />
<link rel="shortcut icon" href=" ../favicon.ico">
<link rel="stylesheet" type="text/css" href="css/normalize.css" />
<link rel="stylesheet" type="text/css" href="css/demo.css" />
<link rel="stylesheet" type="text/css" href="css/component.css" />
<link rel="stylesheet" type="text/css" href="css/util.css">
<link rel="stylesheet" type="text/css" href="css/main.css">
<script src="js/modernizr.custom.js"></script>
</head>
<body>
<div class="container">
<ul id="gn-menu" class="gn-menu-main">
<li class="gn-trigger">
<a class="gn-icon gn-icon-menu"><span>Menu</span></a>
<nav class="gn-menu-wrapper">
<div class="gn-scroller">
<ul class="gn-menu">
<li>
<a class="gn-icon gn-icon-download">Agregar Alumno</a>
</li>
<li>
<a href="/capturaAsistencia.php" class="gn-icon gn-icon-download">Asistencia</a>
</li>
<li>
<a href="/verAsistencia.php" class="gn-icon gn-icon-download">Mostrar Asistencia</a>
</li>
</ul>
</div><!-- /gn-scroller -->
</nav>
</li>
<li><a href="http:// IP_Servidor ">Inicio</a></li>
</ul>
<header>
<?php
echo "<h1>Ingresa tus datos</h1>";
if(isset($_POST["submit"])){
echo "<h2>Vengo de una conexion POST</h2>";
$nombre = $_POST["name"];
$codigo = $_POST["codigo"];
$articulo = $_POST["articulo"];
$activo = $_POST["activo"];
echo "<h2>El nombre que llego en el POST: $nombre</h2>";
echo "<h2>El codigo que llego en el POST: $codigo</h2>";
echo "<h2>El articulo que llego en el POST: $articulo</h2>";
echo "<h2>El activo que llego en el POST: $activo</h2>";
```


Proyecto Final

```
$conexion = new PDO('mysql:host=localhost;dbname=NombreBaseDatos;charset=utf8',
'NombreUsuario', 'ContraseñaUsuario');
$stmt = $conexion->prepare("INSERT INTO NombreTabla (nombre, codigo, articulo, activo)
VALUES (?, ?, ?, ?)");
if($stmt->execute([$nombre, $codigo, $articulo, $activo])){
echo "<h2>Los campos se insertaron correctamente en la base de datos</h2>";
}}
?>
<form class="contact100-form validate-form" action="" method="post">
<span class="contact100-form-title">Envia tus datos</span>
<div class="wrap-input100 validate-input" data-validate="Ingresa tu nombre">
<input class="input100" type="text" name="name" placeholder="Nombre completo...">
<span class="focus-input100"></span>
</div>
<div class="wrap-input100 validate-input" data-validate="Ingresa tu codigo">
<input class="input100" type="text" name="codigo" placeholder="Codigo">
<span class="focus-input100"></span>
</div>
<div class="wrap-input100 validate-input" data-validate="Indica si estas en articulo 34">
<input class="input100" type="text" name="articulo" placeholder="Articulo 34, 1=Si, 0=No">
<span class="focus-input100"></span>
</div>
<div class="wrap-input100 validate-input" data-validate="Indica si estas Activo">
<input class="input100" type="text" name="activo" placeholder="Activo, 1 = Si, 0 = No">
<span class="focus-input100"></span>
</div>
<div class="container-contact100-form-btn">
<button class="contact100-form-btn" type="submit" class="btn btn-primary" name="submit"
value="Submit" >
<span><i class="fa fa-paper-plane-o m-r-6" aria-hidden="true"></i>Enviar</span>
</button>
</div>
</form>
</header>
</div><!-- /container -->
<script src="js/classie.js"></script>
<script src="js/gnmenu.js"></script>
<script>new gnMenu( document.getElementById( 'gn-menu' ) );</script>
</body>
</html>
```

CREAR ARCHIVO capturaAsistencia.php

17.-Abrir el siguiente archivo ejecutando el comando:

→ sudo gedit /var/www/NombreDeLaPagina.com/public_html/capturaAlumnos.php

Copiar y pegar el siguiente texto

```
<?php
date_default_timezone_set('America/Mexico_City');
setlocale(LC_ALL,"es_ES");
$quienB = $current_user->user_firstname;
$postdata = file_get_contents("php://input");
$obj = json_decode($postdata);
$pass = $obj->{'key'};
if($pass == 'x9xyLx56v9G7b3WC'){
$fecha222 = new DateTime();
$fechaString2 = $fecha222->format('Y-m-d H:i:s');
$fechaNumero222 = $fecha222->getTimestamp();
$codigo = $obj->{'var'};
$connect = new PDO('mysql:host=localhost;dbname=NombreBaseDeDatos;charset=utf8',
'NombreUsuario', 'ContraseñaUsuario');
$stmt = $connect->prepare("INSERT INTO NombreTabla (fecha, codigo) VALUES (?, ?)");
if($stmt->execute([$fechaNumero222, $codigo])){
$NumCot = $connect->lastInsertId();
```

```
$stmt = null;
echo "{OK}-{ $pass}-{ $codigo}-{ $fechaString2}";
}else{
//echo "{NOK2}"; }
}
else{
echo "{NOK}-{ $pass}-{ $codigo}-{ $fechaString2}";
}
?>
```

CREAR ARCHIVO verAsistencia.php

18.-Abrir el siguiente archivo ejecutando el comando:

→ sudo gedit /var/www/**NombreDeLaPagina.com**/public_html/verAlumnos.php

Copiar y pegar el siguiente texto

```
<!DOCTYPE html>
<html lang="en" class="no-js">
<head>
<meta charset="UTF-8" />
<meta http-equiv="X-UA-Compatible" content="IE=edge,chrome=1">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Asistencia 17262</title>
<meta name="description" content="A sidebar menu as seen on the Google Nexus 7
website" />
<meta name="keywords" content="google nexus 7 menu, css transitions, sidebar, side
menu, slide out menu" />
<meta name="author" content="Codrops" />
<link rel="shortcut icon" href=".." />
<link rel="stylesheet" type="text/css" href="css/normalize.css" />
<link rel="stylesheet" type="text/css" href="css/demo.css" />
<link rel="stylesheet" type="text/css" href="css/component.css" />
<script type="text/javascript" src="https://www.gstatic.com/charts/loader.js"></script>
<script type="text/javascript">google.charts.load('current', {'packages':['table']});
google.charts.setOnLoadCallback(drawTable);
function drawTable() {
var data = new google.visualization.DataTable();
data.addColumn('string', 'fecha');
data.addColumn('string', 'codigo');
data.addRows([
<?php
$fechaString = 'Mayo';
$codigoVar = 215860049;
//Conectar con mysql
$connect = new PDO('mysql:host=localhost;dbname=NombreBaseDeDatos;charset=utf8',
'NombreUsuario', 'ContraseñaUsuario');
$stmt = $connect->prepare("SELECT * FROM NombreDeLaTabla ORDER BY codigo DESC
LIMIT 50");
if($stmt->execute()){
while ($row = $stmt->fetch()){
$fechaNumero = $row['fecha'];
$codigoVar = $row['codigo'];
$fechaString = date('Y-m-d H:i:s', $fechaNumero);
echo "['$fechaString' , '$codigoVar'],";}}?>];
var table = new google.visualization.Table(document.getElementById('table_div'));
table.draw(data, {showRowNumber: true, width: '80%', height: '80%'});
</script>
<script src="js/modernizr.custom.js"></script>
</head>
<body>
<div class="container">
<ul id="gn-menu" class="gn-menu-main">
<li class="gn-trigger">
<a class="gn-icon gn-icon-menu"><span>Menu</span></a>
```

Proyecto Final

```
<nav class="gn-menu-wrapper">
<div class="gn-scroller">
<ul class="gn-menu">
<li>
<a href="/agregarAlumnos.php" class="gn-icon gn-icon-download">Agregar alumnos</a>
</li>
<li>
<a href="/capturaAsistencia.php" class="gn-icon gn-icon-download">Capturar Asistencia</a>
</li>
<li>
<a href="/verAsistencia.php" class="gn-icon gn-icon-download">Mostrar Asistencia</a>
</li>
</ul>
</div><!-- /gn-scroller -->
</nav>
</li>
<li><a class="codrops-icon codrops-icon-prev" href="http:// DireccionIPServidor ">
<span>Inicio</span></a></li>
<li><a class="codrops-icon codrops-icon-drop" href="http://www.cucei.udg.mx/">
<span>CUCEI</span></a></li>
</ul>
<header>
<h1>Practica 4 <span> Asistencia <a href="http://<span>DireccionIPServidor</span>/verAsistencia.php">
I7262</a> </span></h1>
<div id="table_div"></div>
</header>
</div><!-- /container -->
<script src="js/classie.js"></script>
<script src="js/gnmenu.js"></script>
<script>new gnMenu( document.getElementById( 'gn-menu' ) );</script>
</body>
</html>
```

Programar ESP8266 con IDE de Arduino

Es recomendable realizar las configuraciones previas al ESP8266



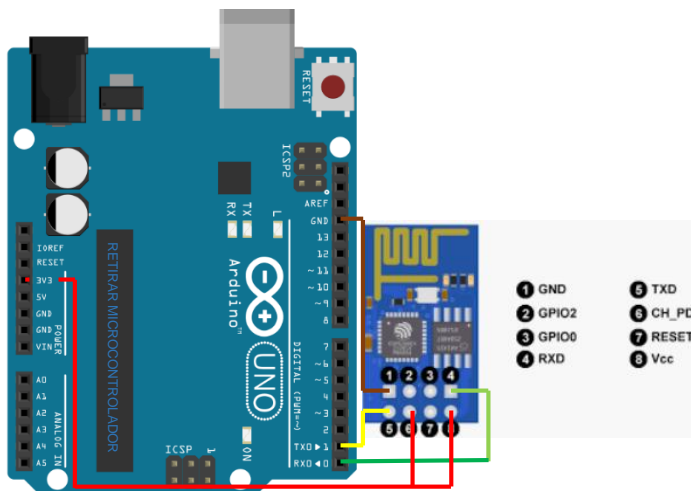
USB a ESP8266

para que se conecte a una red deseada automáticamente sin estar repitiendo los pasos de la conexión. Es posible programar el ESP8266 con un módulo USB a TTL, o también un USB a ESP8266, arduino cuenta con su propio modulo, pero para poder programarlo con mayor precisión es necesario retirar el microcontrolador del arduino.



USB a TTL

Ya que se retiró el microcontrolador realizamos las conexiones del siguiente diagrama, nótese que en esta configuración el puerto TX va conectado con el puerto TX del ESP8266 al igual que el puerto RX del ESP8266 va al puerto RX de arduino el pin 6(EN) y 8() a 3.3v y el pin 1 a tierra.



Ahora abrimos el IDE de arduino y su monitor serial.

Proyecto Final

19.-Programar ESP8266 con comandos AT

Generalmente los ESP8266 vienen configurados de fábrica a una velocidad de 9600 o 115200 baudios. Lo primero es seleccionar en la terminal del arduino una de esas dos velocidades y en la pestaña de la izquierda seleccionar [Ambos NL&CR](#)

```
AT
17:31:00.351 ->
17:31:00.351 -> OK
17:31:13.223 -> AT+GMR
17:31:13.223 -> AT version:1.2.0.0(Jul 1 2016 20:04:45)
17:31:13.256 -> SDK version:1.5.4.1(39cb9a32)
17:31:13.256 -> Ai-Thinker Technology Co. Ltd.
17:31:13.256 -> Dec 2 2016 14:21:16
17:31:13.256 -> OK
```

☒ Autoscroll ☒ Mostrar marca temporal Ambos NL & CR 115200 baudio Limpiar salida

Ahora probar con el primer comando

→ AT

Debe respondernos con un [OK](#)
comprobar la versión del ESP8266 ([Opcional](#))

→ AT+GMR

Verificar el modo de operación actual con el comando:

→ AT+CWMODE?

Debemos estar en el modo 3 de operación que es cliente + servidor en caso de estar en otro modo de operación usar el comando

→ AT+CWMODE=3

```
AT
17:31:00.351 ->
17:31:00.351 -> OK
17:31:13.223 -> AT+GMR
17:31:13.223 -> AT version:1.2.0.0(Jul 1 2016 20:04:45)
17:31:13.256 -> SDK version:1.5.4.1(39cb9a32)
17:31:13.256 -> Ai-Thinker Technology Co. Ltd.
17:31:13.256 -> Dec 2 2016 14:21:16
17:31:13.256 -> OK
17:38:56.057 -> AT+CWMODE?
17:38:56.057 -> +CWMODE:3
17:38:56.057 ->
17:38:56.057 -> OK
```

☒ Autoscroll ☒ Mostrar marca temporal Ambos NL & CR 115200 baudio Limpiar salida

Detectar redes wifi disponibles usando el comando

→ AT+CWLAP

```
17:41:24.386 -> +CWLAP:(4,"Socostroko2",-75,"b0:98:2b:54:3c:8b",11,-41,0)
17:41:24.386 ->
17:41:24.386 -> OK
17:45:02.319 -> AT+CWLAP
17:45:04.443 -> +CWLAP:(4,"Socostroko2",-71,"b0:b2:8f:49:52:dd",1,-21,0)
17:45:04.443 -> +CWLAP:(3,"Brothers 2474",-71,"f4:9e:ef:0e:0d:39",1,-21,0)
17:45:04.443 -> +CWLAP:(3,"UbeeDA9F",-80,"f8:da:0c:88:da:a3",11,-44,0)
17:45:04.476 -> +CWLAP:(4,"Socostroko2",-75,"b0:98:2b:54:3c:8b",11,-41,0)
17:45:04.476 -> +CWLAP:(3,"INFINITUM6715 2.4",-78,"f4:9e:ef:0c:4a:c9",6,-36,0)
17:45:04.476 -> +CWLAP:(3,"TOTALPLAY_52FA0A",-79,"b0:98:2b:52:fa:0a",6,-42,0)
17:45:04.476 -> +CWLAP:(2,"51",-36,"a2:39:f7:24:44:aa",7,-29,0)
17:45:04.476 ->
17:45:04.476 -> OK
```

☒ Autoscroll ☒ Mostrar marca temporal Ambos NL & CR 115200 baudio Limpiar salida

Proyecto Final

Conectarse a una red wifi usando el comando:

→ AT+CWJAP="**NombreDeLaRed**","**ContraseñaDeLaRed**"

```
17:50:48.988 -> ready
17:50:52.140 -> WIFI DISCONNECT
17:51:03.820 -> WIFI CONNECTED
17:51:07.138 -> WIFI GOT IP
17:52:06.505 -> AT+CIPMUX=1
17:52:06.505 ->
17:52:06.539 -> OK
17:55:08.478 -> AT+CIPSERVER=1,80
17:55:08.478 ->
17:55:08.478 -> OK
```

Habilitar múltiples conexiones con el comando

→ AT+CIPMUX=1

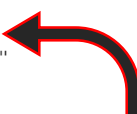
Habilitar un servidor por el puerto 80

→ AT+CIPSERVER=1,80

Encontrar la dirección IP asignada al módulo ESP8266 con el comando

→ AT+CIFSR

```
17:55:30.874 -> AT+CIFSR
17:55:30.874 -> +CIFSR:APIP,"192.168.4.1"
17:55:30.874 -> +CIFSR:APMAC,"86:f3:eb:93:1c:dc"
17:55:30.907 -> +CIFSR:STAIP,"192.168.43.164"
17:55:30.907 -> +CIFSR:STAMAC,"84:f3:eb:93:1c:dc"
17:55:30.907 ->
17:55:30.907 -> OK
```



Ahora podemos abrir a un navegador web e ingresar la dirección IP asignada para que en el monitor serial de arduino nos aparezca algo parecido a esto, presenta los registros de quien se conectó y desde donde se conectó

```
17:57:31.481 -> +IPD,0,391:GET / HTTP/1.1
17:57:31.481 -> Host: 192.168.43.164
17:57:31.481 -> Connection: keep-alive
17:57:31.481 -> Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
17:57:31.515 -> User-Agent: Mozilla/5.0 (Linux; Android 5.0.1; LG-H420 Build/LRX21Y) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/39.0.2171.93
17:57:31.515 -> Accept-Encoding: gzip, deflate, sdch
17:57:31.515 -> Accept-Language: es-MX,es;q=0.8,es-419;q=0.6,en;q=0.4
17:57:31.515 ->
```

Algunos paso no son necesarios para la práctica, pero nos sirven para verificar que el módulo ESP8266 funciona correctamente.

Por último, estamos programando el ESP8266 a 115200 baudios pero a esta velocidad, la tasa de error en el envío y recibo de bits es mayor por lo se recomienda cambiar la velocidad al módulo a una velocidad de 9600 baudios para reducir bits perdidos, esto será ejecutando el comando:

→ AT+CIOBAUD=9600

Con esta configuración la próxima vez que se inicie el ESP8266 se conectara a la red configurada siempre que este disponible, es importante mencionar que para programar el arduino es necesario desconectar los pines RX y TX del arduino para que no tenga errores al programarse.

CODIGO ARDUINO Leer y Escribir Tag RFID

```
#include <SPI.h>
#include <MFRC522.h>
constexpr uint8_t RST_PIN = 9; // Configurable, see typical pin layout above
constexpr uint8_t SS_PIN = 10; // Configurable, see typical pin layout above
MFRC522 mfr522(SS_PIN, RST_PIN); // Create MFRC522 instance
char c;

void setup() {
  Serial.begin(9600); // Initialize serial communications with the PC
  SPI.begin(); // Init SPI bus
  mfr522.PCD_Init(); // Init MFRC522 card
}

void loop()
{
  Serial.println("1. Escribir");
  Serial.println("2. Leer");
  while(!Serial.available()){
  c=Serial.read();
  switch(c)
  {
```

Proyecto Final

```
case '1':
    Escribir();
    break;
case '2':
    Leer();
    break;
}
}
void Escribir()
{
    // Prepare key - all keys are set to FFFFFFFFh at chip delivery from the factory.
    MFRC522::MIFARE_Key key;
    for (byte i = 0; i < 6; i++) key.keyByte[i] = 0xFF;
    // Look for new cards
    while ( ! mfrc522.PICC_IsNewCardPresent()) {
        //return;
    }
    // Select one of the cards
    while ( ! mfrc522.PICC_ReadCardSerial()) {
        //return;
    }
    Serial.print(F("Card UID:")); //Dump UID
    for (byte i = 0; i < mfrc522.uid.size; i++) {
        Serial.print(mfrc522.uid.uidByte[i] < 0x10 ? " 0" : " ");
        Serial.print(mfrc522.uid.uidByte[i], HEX);
    }
    Serial.print(F(" PICC type: ")); // Dump PICC type
    MFRC522::PICC_Type piccType = mfrc522.PICC_GetType(mfrc522.uid.sak);
    Serial.println(mfrc522.PICC_GetTypeName(piccType));
    byte buffer[34];
    byte block;
    MFRC522::StatusCode status;
    byte len;
    Serial.setTimeout(20000L); // wait until 20 seconds for input from serial
    // Ask personal data: Family name
    Serial.println(F("Entra codigo terminando con #"));
    len = Serial.readBytesUntil('#', (char *) buffer, 30); // read family name from serial
    for (byte i = len; i < 30; i++) buffer[i] = ' '; // pad with spaces
    block = 1;
    //Serial.println(F("Authenticating using key A..."));
    status = mfrc522.PCD_Authenticate(MFRC522::PICC_CMD_MF_AUTH_KEY_A, block, &key,
    &(mfrc522.uid));
    if (status != MFRC522::STATUS_OK) {
        Serial.print(F("PCD_Authenticate() failed: "));
        Serial.println(mfrc522.GetStatusCodeName(status));
        return;
    }
    else Serial.println(F("PCD_Authenticate() success: "));
    // Write block
    status = mfrc522.MIFARE_Write(block, buffer, 16);
    if (status != MFRC522::STATUS_OK) {
        Serial.print(F("MIFARE_Write() failed: "));
        Serial.println(mfrc522.GetStatusCodeName(status));
        return;
    }
    else Serial.println(F("MIFARE_Write() success: "));
    Serial.println(" ");
    mfrc522.PICC_HaltA(); // Halt PICC
    mfrc522.PCD_StopCrypto1(); // Stop encryption on PCD
}
void Leer()
{
    MFRC522::MIFARE_Key key;
    for (byte i = 0; i < 6; i++) key.keyByte[i] = 0xFF;
    //some variables we need
    byte block;
    byte len;
```

Proyecto Final

```
MFRC522::StatusCode status;
//-----
// Look for new cards
while ( ! mfrc522.PICC_IsNewCardPresent() ) {
    //return;
}
// Select one of the cards
while ( ! mfrc522.PICC_ReadCardSerial() ) {
    //return;
}
Serial.println(F("***Card Detected:***"));
mfrc522.PICC_DumpDetailsToSerial(&(mfrc522.uid)); //dump some details about the card
//mfrc522.PICC_DumpToSerial(&(mfrc522.uid)); //uncomment this to see all blocks in hex
len = 18;
byte buffer2[18];
block = 1;
status = mfrc522.PCD_Authenticate(MFRC522::PICC_CMD_MF_AUTH_KEY_A, 1, &key,
&(mfrc522.uid)); //line 834
if (status != MFRC522::STATUS_OK) {
    Serial.print(F("Authentication failed: "));
    Serial.println(mfrc522.GetStatusCodeName(status));
    return;
}
status = mfrc522.MIFARE_Read(block, buffer2, &len);
if (status != MFRC522::STATUS_OK) {
    Serial.print(F("Reading failed: "));
    Serial.println(mfrc522.GetStatusCodeName(status));
    return;
}
//PRINT LAST NAME
for (uint8_t i = 0; i < 16; i++) {
    Serial.write(buffer2[i] );
}
Serial.println(F("\n**End Reading**\n"));
delay(1000); //change value if you want to read cards faster
mfrc522.PICC_HaltA();
mfrc522.PCD_StopCrypto1();
}
```

CODIGO ARDUINO

Practica 4 Lector RFID con ESP8266 y Base de Datos

```
/* Practica 4 Control de Asistencia con Arduino y RFID, Base de datos y pagina web
I7262 Diseño de Interfaces
Dr. Ruben Estrada Marmolejo
Aguilar Rodriguez Carlos Adolfo
*/
```

```
//-----Arreglos globales-----
//Incluir la libreria bus SPI (Serial Peripheral Interface)
#include <SPI.h>
//Incluir la libreria para el modulo de RFID MFRC522
#include <MFRC522.h>
//Declaracion de una constante para el pin 9 de arduino como reset del modulo RFID
#define RST_PIN 9
//Declaracion de una constante para el pin 10 de arduino como el slave select del modulo RFID
#define SS_PIN 10
//Crea un objeto declarando los pines anteriores
MFRC522 mfrc522(SS_PIN, RST_PIN);
MFRC522::MIFARE_Key key;
//Variable para almacenar los datos leidos del tag
char *codigo=" ";
//Variable para almacenar la cadena de caracteres del mensaje JSON
char mensajeJSON[400];
```

Aguilar Rodríguez Carlos Adolfo 215860049

Proyecto Final

```
//Variable para almacenar la cadena de caracteres del mensaje HTTP
char mensajeHTTP[600];
//Tiempo que tarda en realizar la conexion
int conexionTiempo = 11;
//Tiempo que tarda en realizar la ejecucion de los comandos
byte conexionComando1 = 2;
int contadorTiempo = 0;
//Arreglo de caracteres constantes que continen quien sera el HOST
//El HOST sera la direccion de la pagina web principal del servidor
const char HOST[] = "Practica4.com";
//Arreglo de caracteres constantes que continen la contraseña que se enviara por medio de un mensaje
JSON
//Del arduino al modulo esp8266 y del modulo a la pagina de la base de datos capturarAsistencia.php
const char keyVar[] = "x9xyLx56v9G7b3WC";
//Direccion del subdirectorio de la pagina web en donde seran enviados los datos
const char urlVar[] = "/capturaAsistencia.php";
//Direccion IP del servidor web (esta IP es la IP del la computadora por donde se conecta a internet)
const char servidorVar[] = "192.168.43.233"; //(Esta es la dirección IP del servidor Nginx)
//Este es el nombre de usuario de la base de datos phpmyadmin
const char usuarioHTTP[] = "muthe";
long int t0 = 0;
long int t1 = 0;
//Variable logica para el estado del led declarada en falso
boolean statusLED = false;
//Variable logica para el estado del debug declarada en falso
boolean debug = false;
//-----

//-----Configuracion Principal-----
//Bloque principal para configurar al arduino
void setup() {
//Inicializacion de la comunicacion serial a una velocidad de 9600 baudios
Serial.begin(9600);
//Inicia la comunicacion SPI
SPI.begin();
//Inicia al modulo RFID MFRC522
mfrc522.PCD_Init();
}
//-----

//Bloque para crear el mensaje JSON que sera enviado del arduino hacia el modulo wifi y del modulo a la
pagina de la base de datos
char* crearMensajeJSON(char *bufferTX, int variable){
//Funcino que imprime la variable buffeTX y el mensaje JSON
//sprintf(bufferTX, "{\"key\":\"x9xyLx56v9G7b3WC\",\"var\":\"215860049\"}");
sprintf(bufferTX, "{\"key\":\"%s\",\"var\":\"%s\"}", keyVar, codigo);
return bufferTX;
}

//Bloque para crear la estructura del mensaje HTTP que contiene la URL el servidor el usuario y la
cantidad de bytes a transmitir-----
void crearMensajeHTTP(char *bufferTX, char *mensajeJSON, char *URL, char *Servidor, char *Usuario){
//Leer cuantos Bytes tiene el mensajeJSON y depositar en la variable cuantosBytes
int cuantosBytes = strlen(mensajeJSON);
//Vector de caracteres temporal de 50 caracteres
char temporal[50];
//Enviar por el puerto serial la variable temporal que contiene la informacion del metodo de conexion
POST de php
sprintf(temporal, "POST %s HTTP/1.0\r\n", URL);
//Realiza una concatenacion entre bufferTX y la variable temporal
```


Proyecto Final

```
strcat(bufferTX, temporal);
//Enviar por el puerto serial la variable temporal que contiene el servidor HOST
sprintf(temporal, "Host: %s \r\n", Servidor);
//Realiza una concatenacion entre bufferTX y la variable temporal
strcat(bufferTX, temporal);
//Enviar por el puerto serial la variable temporal que contiene el nombre de usuario de la base de datos
sprintf(temporal, "User-Agent: %s \r\n", Usuario);
//Realiza una concatenacion entre bufferTX y la variable temporal
strcat(bufferTX, temporal);
//Enviar por el puerto serial la variable temporal que contiene la cantidad de bytes que seran enviados
sprintf(temporal, "Content-Length: %i \r\n", cuantosBytes);
//Realiza una concatenacion entre bufferTX y la variable temporal
strcat(bufferTX, temporal);
//Realiza una concatenacion entre bufferTX y la codificacion de la pagina web
strcat(bufferTX, "Content-Type: application/x-www-form-urlencoded\r\n");
//Esta instruccion permite realizar un salto de linea
strcat(bufferTX, "\r\n");
//Realiza una concatenacion entre bufferTX y el mensaje JSON
strcat(bufferTX, mensajeJSON);
//Retorna la variable bufferTX
return bufferTX;
}
```

//-----Bloque para enviar informacion por comunicacion serial-----

```
void enviarSerial(char *bufferTX){
//Variable que almacenara la cantidad de bytes que contiene la variable bufferTX
int cuantosBytes = strlen(bufferTX);
//Declaracion de un contador iniciado en 0
int contador = 0;
//Funcion condicional que determina si debug entonces ...
if(debug){
//Imprime por la comunicacion serial el siguiente texto "Bytes del mensaje: "
Serial.print("Bytes del mensaje: ");
//Imprime por la comunicacion serial la cantidad de bytes del mensaje
Serial.println(cuantosBytes);
}
//Ciclo for para leer cada uno de los caracteres que son recibidos del bufferTX
for(int j=0; j<=(cuantosBytes-1); j++){
//Leer del caracter bufferTX por la comunicacion serial
Serial.write((char)bufferTX[j]);
//Incrementar contador en 1
contador++;
}
//Funcion condicional que determina si debug entonces ...
if(debug){
//Si se cumple la funcion anterior imprimir el siguiente texto "Bytes enviados"
Serial.print("Bytes enviados: ");
//Imprimir por la comunicacion serial la variable contador
Serial.println(contador);
}
}
```

//-----Bloque de arduino que se repetira una y otra vez-----

```
void loop() {
Leer();
Serial.println(codigo);
if(debug){ //Enviar por la comunicacion serial el mensaje "Se creara el mensaje JSON"
Serial.println("Se creara el mensaje JSON");
crearMensajeJSON(mensajeJSON, 1);
enviarSerial(mensajeJSON);
//void crearMensajeHTTP(char *bufferTX, char *mensajeJSON, char *URL, char *Servidor, char
*Usuario){
crearMensajeHTTP(mensajeHTTP, mensajeJSON, urlVar, servidorVar, usuarioHTTP);
}
```

Proyecto Final

```
    enviarSerial(mensajeHTTP);
}else{
    crearMensajeJSON(mensajeJSON, 1);
    crearMensajeHTTP(mensajeHTTP, mensajeJSON, urlVar, servidorVar, usuarioHTTP);
    char temporal[50];
    //Realiza una conexion con el servidorVar por medio del puerto 80
    sprintf(temporal, "AT+CIPSTART=\"TCP\", \"%s\", 80\r\n", servidorVar);
    enviarSerial(temporal);
    delay(conexionComando1*1000);
    //Declaracion de una variable llamada cuantosbytesMJ donde será depositado la cantidad de bytes en
mensajeHTTP
    int cuantosBytesMJ = strlen(mensajeHTTP);
    //Enviar por la comunicacion serial la variable temporal y con el comando AT+CIPSEND imprimir el
numero de bytes del mensaje JSON
    sprintf(temporal, "AT+CIPSEND=%d\r\n", cuantosBytesMJ);

    enviarSerial(temporal);
    delay(conexionComando1*2000);

    enviarSerial(mensajeHTTP);

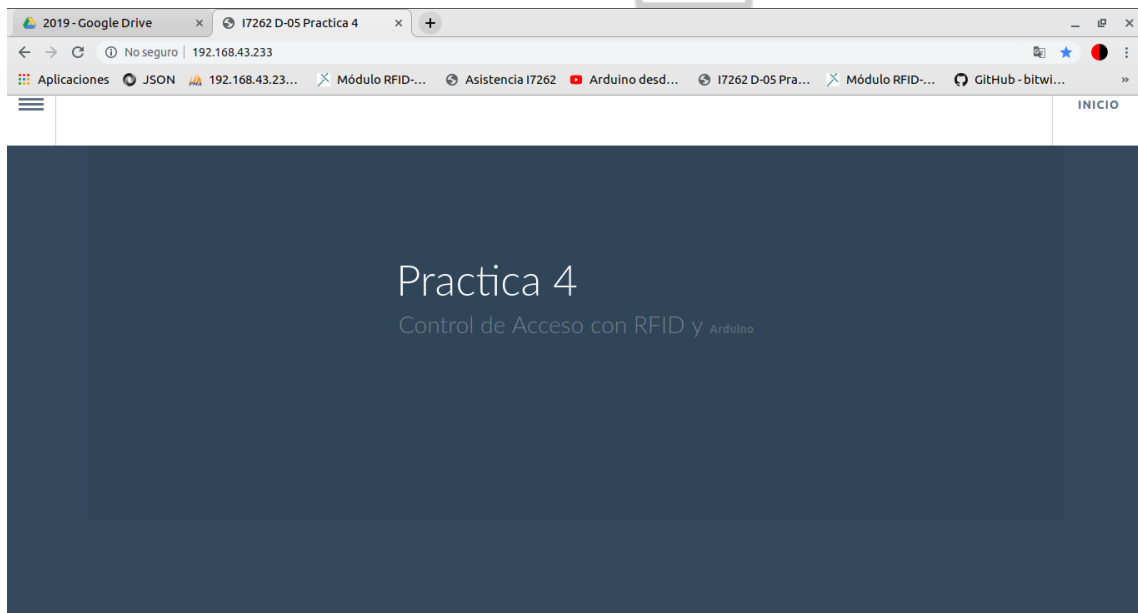
    delay(conexionComando1*2000);
}
}

//----- Funcion para leer la UID de los tags-----
void Leer(){
for (byte i = 0; i < 6; i++) {
    key.keyByte[i] = 0xFF;
}
byte block;
byte len;
//Funcion para indicar el estado de la tarjeta
MFRC522::StatusCode status;
//Si no esta presente una tarjeta retorna a la funcion
while( ! mfrc522.PICC_IsNewCardPresent()){
//Si no se puede leer la tarjeta retorna a la funcion
while( ! mfrc522.PICC_ReadCardSerial()){
len = 18;
//Arreglo de bytes para crear un buffer
char buffer2[9];
//Variable tipo byte llamada size que almacena el arreglo buffer
block = 1;
status = mfrc522.PCD_Authenticate(MFRC522::PICC_CMD_MF_AUTH_KEY_A, 1, & key,
&(mfrc522.uid));
if(status != MFRC522::STATUS_OK){
    return;
}
status = mfrc522.MIFARE_Read(block, buffer2, &len);
if(status!= MFRC522::STATUS_OK){
    return;
}
}
for(uint8_t i=0; i<10; i++){
}
codigo=buffer2;
// Vuelve a iniciar el modulo MFRC522
mfrc522.PICC_HaltA();
mfrc522.PCD_StopCrypto1();
}
```

IMPLEMENTACION



Montaje de la practica



Página principal Practica4

Proyecto Final

The screenshot shows a web browser window with the address bar displaying '192.168.43.233/agregarAlumnos.php'. The page has a dark blue header with a hamburger menu on the left and the word 'INICIO' on the right. The main content area is a dark blue rectangle with the title 'Ingresa tus datos' in white. Below the title is a small 'ENVIAR TUS DATOS' link. There are four white input fields with rounded corners, each containing a label in blue: 'Nombre completo...', 'Codigo', 'Articulo 34, 1 = Si, 0 = No', and 'Activo, 1 = Si, 0 = No'. At the bottom center is a purple 'ENVIAR' button.

Ingresa tus datos

ENVIAR TUS DATOS

Nombre completo...

Codigo

Articulo 34, 1 = Si, 0 = No

Activo, 1 = Si, 0 = No

ENVIAR

Página para capturar Alumnos

The screenshot shows a web browser window with the address bar displaying '192.168.43.233/verAsistencia.php'. The page has a dark blue header with a hamburger menu on the left and the word 'INICIO' on the right. The main content area is a dark blue rectangle with the title 'Practica 4' in white, followed by 'Asistencia I7262' in blue. Below the title is a table with 5 rows and 3 columns. The columns are labeled 'id', 'fecha', and 'codigo'. The table contains the following data:

	fecha	codigo
1	2019-05-28 04:24:51	215860049
2	2019-05-28 04:25:06	215860049
3	2019-05-28 04:25:30	215860049
4	2019-05-28 04:26:14	215860049
5	2019-05-28 04:26:27	215860049

Página para mostrar Asistencia

Proyecto Final

Características Modulo RFID

-2.5 a 3.3 Volts de consumo corriente de operación 13-26mA 10-13mA en Stanby <80uA en sleep mode
Frecuencia de operación 13.56MHz
Velocidad máxima de comunicación de 10Mbps/s
Soporta SPI, UART, I2C, ISO/IEC 14443 A/MIFARE y NTAG
Una distancia máxima de comunicación de 5cm
buffer FIFO 64 byte para enviar y recibir
Interrupciones
Apagado por software
Temporizador programable
oscilador interno de 27.12MHz
Coprocesador CRC

Características Modulo ESP8266

Protocolo 802.11 b/g/n
Antena de 2.4GHz
Comunicación SPI,UART,SDIO 2.0
MAC integrada
Motores WEP, AES, TKIP y WAPI
TCP/IP Integrado
sensor de temperatura integrado
+19.5dBm poder de salida en modo 802.11b
Consumo en sleep mode <10uA
(Entre mayor sea el consumo de corriente la distancia y velocidad será mayor)

Conclusiones

Tener un control de asistencia o control acceso registrado en una base de datos con acceso a internet nos permite tener un historial de las horas y fechas en que el usuario accedió al sistema. Y poder consultarlas posteriormente en cualquier momento independientemente si la aplicación está funcionando o no. Las aplicaciones IoT se frecuentan cada vez más y tener un control de la información en cualquier parte del mundo aporta muchas ventajas para administrar el personal; Esta es una aplicación ideal para personas que tienen a más personas bajo su cargo y desean saber quién llega temprano, tarde o simplemente no se presentó de una manera automatizada. Además las tarjetas tienen una memoria interna donde se puede almacenar información personalizada y así tener llaves para abrir cerraduras electrónicas para personal exclusivo en ciertas áreas

Referencias

<https://www.ubuntu.com/>

<https://www.arduino.cc/>

RFID con arduino <https://www.youtube.com/watch?v=LvRfxGTUEpE>

<https://hetpro-store.com/TUTORIALES/rfid-mfrc522-arduino-uno/>

<https://hetpro-store.com/TUTORIALES/esp8266-arduino-crear-servidor-web/>

<https://hetpro-store.com/TUTORIALES/configurar-el-esp8266-con-arduino-leccion-1/>

Instalar: NGINX, PHP 7, MYSQL

<https://devanswers.co/installing-nginx-mysql-php-lemp-stack-ubuntu-18-04/>

Instalar phpMyAdmin

<https://linuxize.com/post/how-to-install-phpmyadmin-with-nginx-on-ubuntu-18-04/>

Tabla en JavaScript para ver asistencia

<https://developers.google.com/chart/interactive/docs/gallery/table>

Hoja de datos ESP8266

https://hetprostore.com/images/companies/1/DATSH/ESP8266_Specifications_English.pdf?1459783298618

<https://www.epochconverter.com/>

Proyecto Final

Modificar el archivo capturarAsistencia.php

```
<?php
//Tiempo de la zona Ciudad de Mexico
date_default_timezone_set('America/Mexico_City');
$postdata = file_get_contents("php://input");
$fecha = new DateTime();
$fechaString = $fecha->format('Y-m-d H:i:s');
$fechaNumero = $fecha->getTimestamp();
//Variable obj para depositar el mensaje JSON decodificado
$obj = json_decode($postdata);
//Comprobar que la clave del mensaje es correcta
$pass = $obj->{'key'};
if ($pass == 'x9xyLx56v9G7b3WC'){
//Depositar en la variable código la variable var de arduino
$codigo = $obj->{'var'};
//Variables para almacenar el host el usuario y contraseña de la base de datos
$servername = "localhost";
$username = "muthe";
$password = "51";
try{
//Realizar conexión con mysql y obtener la base de datos Practica4
$conn = new PDO("mysql:host=$servername;dbname=Practica4", $username, $password);
$conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
//Selecciona el código de la tabla I7262
$sql = "SELECT * FROM I7262 WHERE codigo =" . $codigo;
$q = $conn->query($sql);
$fila = $q->fetch();
if ($fila == NULL)
else{
$id = $fila['numeroLista'];
//Variable para determinar si se permite el registro de asistencia
$registrar = false;
//Verificar que no esté registrado
$sql = "SELECT * FROM Asistencia WHERE código =" . $código . " ORDER BY fecha DESC";
$q = $conn->query($sql);
$fila = $q->fetch();
if ($fila == NULL){
$registrar = true;
}else{
$fechaAsistencia = $fila['fecha'];
$día = date('d');
$mes = date('m');
$año = date('Y');
//Convertir a época principal
$fechaEpochPrincipal = mktime(0, 0, 0, $mes, $día, $año);
$fechaEpochPrincipal . "\n";
//Evaluar si se permite realizar el registro de asistencia solo se permite una vez cada dos días
if ($fechaAsistencia >= $fechaEpochPrincipal && $fechaAsistencia < ($fechaEpochPrincipal + 172800))
{
//No registrar si ya se tiene asistencia anteriormente
$registrar = false;
}else
$registrar = true;
}
}
//Evaluar si se puede hacer el registro
if ($registrar)
{try{
$conn = new PDO("mysql:host=$servername;dbname=Practica4", $username, $password);
$conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
//Registrar asistencia en la tabla Asistencia
$sql = "INSERT INTO Asistencia (fecha, codigo) VALUES ('$fechaNumero', '$codigo')";
$conn->exec($sql);
?>
```