



INTEGRANTES

GONZALEZ DIEGO

ORTIZ ARNALDO

SINCHIGUANO BRYAN

TAPIA WILIAN

PROYECTO INTEGRADOR

GRUPO 2

NETWORKING:

ING. VIVIANA DE LOS ANGELES TIXILIMA CISNEROS

SISTEMAS EMBEBIDOS

ING. RAFAEL JAYA

PROGRAMACIÓN HIPERMEDIAL

RODRIGO TUFÍÑO

Contenido

1. Tema	3
2. Descripción general del proyecto	3
3. Justificación.....	6
3.1. Sistemas Embebidos.....	6
3.2. Networking.....	6
Programación Hipermedial	6
3.3. Implementaciones y relación con el diario vivir	7
4. Objetivos.....	7
4.1. Objetivo general	7
4.2. Objetivos Específicos.....	8
5. Desarrollo.	9
5.1. Topología Para Usar en el desarrollo del proyecto y detalles de las configuraciones.	9
Switch 1(SERVER).....	9
Sensores Utilizados.	15
1. Sensor Infrarrojo	15
2. Sensor Ultrasónico	16
3. Motor.....	17
Circuito en Fritzing.	17
Control de Node MCU y programación de los embebidos.	18
Control de Node MCU.	19
MVC(Modelo-Vista-Controlador).	22
Diagramas de base de datos y componentes de la aplicación Web	22
Enlace a GitHub.....	29
6. Conclusiones.....	30
Recomendaciones	31
Bibliografía y Referencias	31

1. Tema

Proyecto integrador de las materias: Networking , sistemas embebidos, programación Hipertextual, para la creación e implementación de una banda transportadora automatizada por medio de sensores de lectura y ciertas herramientas de escritura para almacenar estos datos dentro de un base de datos alojada en un servidor.

2. Descripción general del proyecto

El proyecto integrador para este período consiste en generar una aplicación web sobre una infraestructura de red previamente definida, que permita monitorear el estado de un proceso y/o sistema embebido. Para el sistema embebido a monitorear, se ha establecido dos variantes que serán asignados a cada grupo según el criterio del profesor responsable de la asignatura. Cabe mencionar que se debe construir una maqueta que represente al sistema embebido a controlar. A continuación, se especifica los detalles de cada uno de los componentes del sistema.

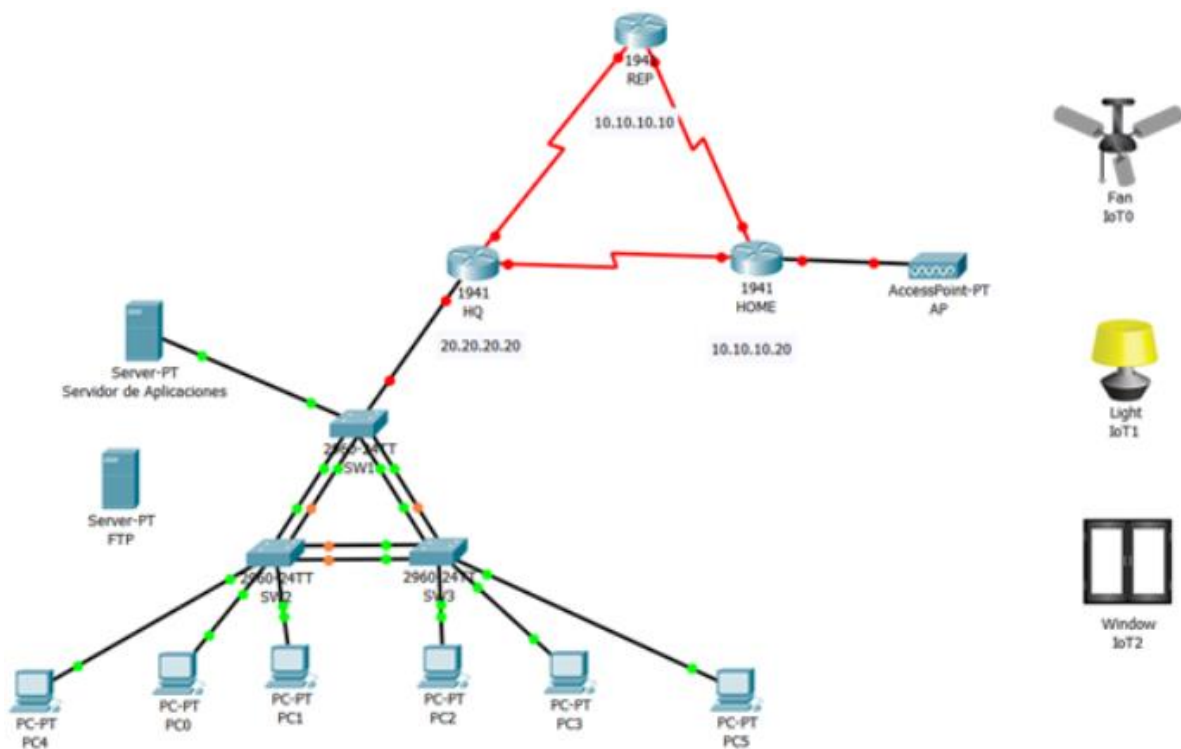


Tabla de direccionamiento

DISPOSITIVO	INTERFAZ	IP	MASCARA	GATEWAY
HQ	S0/0/0	200.165.209.65	255.255.255.252	
	G0/1	172.16.42.253	255.255.255.224	
HOME	S0/0/0	180.165.100.33	255.255.255.252	
	G0/1	192.168.1.1	255.255.255.224	
REP	S0/0/0	200.165.209.66	255.255.255.252	
	S0/1/0	180.165.100.34	255.255.255.252	
SERVIDOR DE APLICACIONES	Fa0	172.16.42.226	255.255.255.224	172.16.42.1
SERVIDOR BDD	Fa1	172.16.42.230	255.255.255.224	172.16.42.1
VLAN 10		172.16.42.1	255.255.255.224	
VLAN 20		172.16.42.33	255.255.255.224	
VLAN 30		172.16.42.65	255.255.255.224	
PC0	Fa0	172.16.42.1	255.255.255.224	
PC1	Fa0	172.16.42.35	255.255.255.224	
PC2	Fa0	172.16.42.50	255.255.255.224	
PC3	Fa0	172.16.42.10	255.255.255.224	
PC4	Fa0	172.16.42.55	255.255.255.224	
PC5	Fa0	172.16.42.20	255.255.255.224	
VLAN'S	RANGOS	NAME		
VLAN 10	F0/1-9	ADMINISTRADOR		
VLAN 20	F0/10-15	WEBDESIGNER		
VLAN 30	F0/16-20	CLIENTES		

Networking

La configuración Inter VLAN debe ser llevada a cabo en el SW1

- Configurar DHCP para las VLAN'S en el router HQ
- Configurar todas las interfaces utilizando la tabla de direccionamiento proporcionada.
- Realizar la configuración básica en cada uno de los dispositivos, utilizando password "proyecto" para vty como para console, y "sexto" para enable.
- Configurar VTP server y el VTP client con el dominio "NETWORKING" y el password "12345NET"
- Crear las VLANS con sus correspondientes interfaces. •Configurar para que puedan comunicarse entre VLAN'S
- Levantar una ACL estándar para permitir solo a la VLAN 10 que pueda acceder al SERVIDOR BDD

- Levantar una ACL extendida para permitir a las VLAN's 20 y 30 para que pueda acceder al puerto 80 y 443 del SERVIDOR DE APLICACIONES
- Levantar una ACL extendida para permitir solo a la VLAN 20 pueda acceder por el puerto 22 al SERVIDOR DE APLICACIONES
- Implementar EtherChannel en los switch's correspondientes
- Aplicar enrutamiento entre los 3 routers utilizando un protocolo de enrutamiento

Programación Hipermedial

Crear una aplicación Web que permita monitorear el estado de los sensores y controlar el estado de los actuadores del sistema embebido. La aplicación debe mostrar un esquema simplificado del sistema a través de un Dashboard con la siguiente información:

- Estado en tiempo real de los sensores (retraso máximo de 1 segundo)
- Controlar el estado de actuadores (encender / apagar)
- Registro de los cambios de estado de los sensores (tabla con fecha y hora)
- Registro de los cambios de estado de los actuadores (tabla con fecha y hora)
- Exportación de reportes a hoja de cálculo y PDF.La aplicación deberá implementarse utilizando los siguientes requerimientos no funcionales:
- Base de datos MySQL.
- Lenguaje de programación PHP 7.0 o superior y el framework CodeIgniter.
- Utilizar AJAX y HighCharts en la presentación de datos y gráficas.
- Utilizar Bootstrap para la interfaz gráfica.
- Servidor Web NGINX.

El sistema deberá estar alojado en un servidor de aplicaciones GNU/Linux (172.16.42.226) sin interfaz gráfica. La base de datos se instalará sobre un sistema Microsoft® Windows® (172.16.42.230).

Sistemas Embebidos

Diseñar el sistema embebido para adquisición y control de variables analógicas y digitales que constara de 2 sensores (Temperatura/Presión y Movimiento (PIR)/Fotodiodo) y 2 Actuadores (relés para Apagado y Encendido de Luces/Calefactores, Control de Aire acondicionado/Motores). Todos estos dispositivos irán conectados a un módulo Arduino y este a su vez se conectará con un módulo ESP8266/ESP32 (Modulo WiFi con uC de 32 bits),los mismos que serán los encargados de enviar o recibir los datos a la red o Base de

datos (DB) a través de un punto de acceso. Adicional para observar si los datos se están transfiriendo o se está realizando alguna acción, este se visualizará en un display LC

3. Justificación

3.1. Sistemas Embebidos.

El presente proyecto integrador que implica las materias de: Networking , Sistemas Embebidos, programación Hipermedial, pretende fomentar la investigación por parte de los estudiantes por lo cual ,el punto de partida de este proyecto es implementar una comunicación mediante el puerto serial entre la placa Arduino a usar y el nodeMCU para poder transferir datos y ser almacenados en la base de datos correspondiente, datos que deben ser tomados por medios de diferentes sensores y de acuerdo a las medidas o condiciones que cada uno de estos sensores tenga activar un determinado actuador que permita clasificar los diferentes objetos que pasan por medio de la banda transportadora, la cual a su vez tiene un motor alimentado externamente que ayude a su funcionamiento adecuado , dado que si se trata de tomar energía de la placa Arduino no permite el correcto funcionamiento de los actuadores y sensores haciendo que los datos sean imprecisos y por lo tanto la base de datos no contenga información precisa sobre los valores tomados por medio de los diferentes sensores provocando un error en el funcionamiento de un determinado actuador.

3.2. Networking

La topología propuesta una vez implementada ha sido tomada en cuenta para que funcione de una manera adecuada conjuntamente con un punto de acceso inalámbrico al cuál estará conectado a un nodeMCU para la transmisión de archivos, con sus respectivas VLAN'S y ACL'S correspondientes a ser configuradas dentro de cada dispositivo facilitado por la universidad, además de implementar el protocolo DHCP dado que este permitirá un mejor funcionamiento con los puntos de acceso que se conecten a la topología permitiendo así asignar una IP automática a los dispositivos conectados a la red respectiva, permitiéndonos así optimizar tiempo y recursos dentro de una red.

Programación Hipermedial

Es importante implementar el uso de codeigniter dentro del siguiente proyecto dado que al poder implementar las diferentes sentencias de programación dentro de un modelo una vista y un controlador podemos manipular de una manera mucho más eficiente los datos que se van a mostrar gracias a las librerías de highcharts desarrolladas en JavaScript y así posteriormente graficar los datos respectivos de acuerdo al tiempo y datos almacenados por cada uno de los

sensores que han sido ubicados en la banda transportadora, además que se ha decidido usar AJAX cuyo significado es JavaScript asíncrono y XML que nos permite principalmente hacer que el procesamiento de solicitudes a un servidor dentro de la red se ejecute en segundo plano evitando así la recarga completa de la página a cada momento.

3.3. Implementaciones y relación con el diario vivir

Con ayuda de las conexiones físicas de los equipos proporcionados se procede a las diferentes configuraciones de los equipos físicos tanto para la parte de Networking como para la parte de los sistemas embebidos, dado que siempre como primer punto se debe verificar conectividad entre los dispositivos y posteriormente validar cada una de las tecnologías implementadas dentro del presente proyecto para poder así juntar todas y hacer las respectivas configuraciones para que se pueda tener comunicación entre las diferentes tecnologías y transmitir los datos tomados por los diferentes sensores y activar un determinado actuador para una acción específica, guardando esos datos recolectados en una base de datos por medio de un nodeMCU wifi para que los datos almacenados dentro de esta base de datos puedan ser tomados para mostrar una gráfica de los datos recolectados en un determinado tiempo por los diferentes sensores para mostrarlos en un sitio web con ayuda de las diferentes tecnologías mencionadas anteriormente. Esta es una de las aplicaciones que nos muestran un gran paso para poder implementar el internet de las cosas dentro de tareas automatizadas que pueden involucrar varias tecnologías ya sean de programación o de conexiones inalámbricas a una determinada red, pero además se puede tomar datos que ayuden en la generación de reportes en el caso de ser una tarea automatizada a gran escala esos datos pueden ayudar a un determinado sector de la empresa dado que tienen datos históricos y pueden ser tomados para un determinado análisis que necesite la empresa, esta puede ser una de las muchas aplicaciones del presente proyecto.

4. Objetivos

4.1. Objetivo general

Implementar una red con diferentes puntos de acceso ya sean por medio de cables o a su vez inalámbricos por donde se transportará datos recolectados por medio de sensores que permitirán activar un determinado actuador, a su vez guardar los valores en una base de datos alojada dentro de un servidor, mediante la conexión de diferentes equipos físicos que ayuden a poder pasar estos datos por medio de una red recolectando datos por medio de un sistema embebido para poder graficar una determinada variabilidad de los datos de acuerdo al

tiempo transcurrido con la ayuda de diferentes componentes web que permitan una mejor implementación web del servidor que nos permitirá visualizar estos datos.

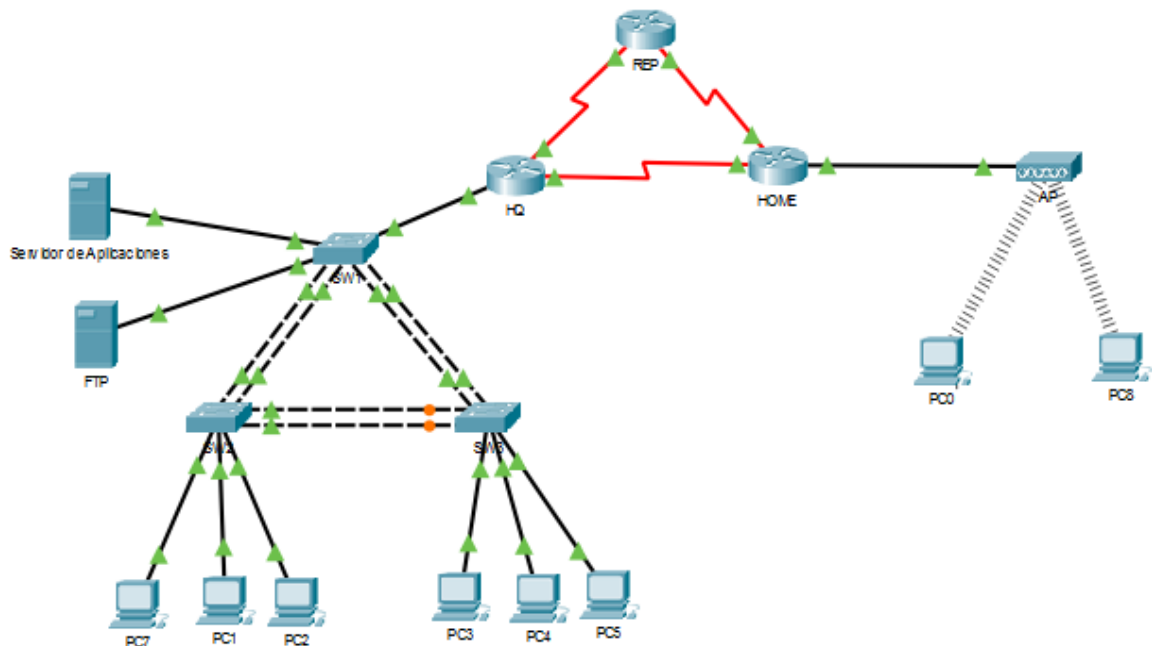
4.2. Objetivos Específicos

- Adquirir los materiales necesarios para poder ser conectados dentro de la banda transportadora y así poder añadir el funcionamiento de diferentes sensores y actuadores.
- Verificar el estado de los equipos proporcionados para construir la topología de la red por la cual viajarán los datos dado que si un equipo no está en óptimas condiciones de funcionamiento podría producir errores que afecten a los demás equipos.
- Implementar las configuraciones que se han mencionado en los requisitos dentro de la topología presentada para así poder enviar los datos a través de la red a un determinado servidor que alojara los datos recolectados.
- Crear un sistema embebido el cual contenga 1 sensor infrarrojo y 1 sensor de ultrasonido para la optimización de un sistema para una banda transportadora.
- Instalar los diferentes complementos necesarios dentro del software de Arduino para poder así controlar la placa y transmisión de los datos entre los diferentes componentes.
- Instalar el sistema operativo que se ha determinado en los requerimientos para instalar posteriormente el servidor que permitirá alojar la aplicación.
- Recolectar los datos necesarios de cada uno de los sensores que están dentro de la placa Arduino para poder así controlar los actuadores y enviar esos datos al nodeMCU.
- Conectar la placa de Arduino con el nodeMCU para poder transmitir los datos entre estos dos dispositivos y posteriormente almacenarlos en una base de datos.
- Verificar la conectividad del nodeMCU a un punto de acceso dentro de la red por la cual se enviarán los datos y una vez que se ha podido verificar su correcto funcionamiento y transmisión se crea la base de datos donde se almacenarán los datos recolectados con anterioridad .
- Verificar el correcto funcionamiento del Framework de Codeneighter así como verificar el correcto funcionamiento de las librerías para highcharts y así poder hacer las gráficas necesarias de acuerdo al sensor seleccionado .
- Implementar las sentencias respectivas dentro del framework mediante Codeneighter para poder visualizar reportes de los diferentes datos y a su vez gráficas de los datos

recolectados por cada uno de los sensores y así verificar el correcto funcionamiento de la conexión entre las diferentes tecnologías.

5. Desarrollo.

5.1. Topología Para Usar en el desarrollo del proyecto y detalles de las configuraciones.



Switch 1(SERVER)

Crear de vlan en el switch que realizara de servidor.

```
Switch(config)#vlan 10
Switch(config-vlan)#name ADMINISTRADOR
Switch(config-vlan)#vlan 20
Switch(config-vlan)#name WEBDESIGNER
Switch(config-vlan)#vlan 30
Switch(config-vlan)#name CLIENTES
Switch(config-vlan)#exit
```

Realizar modo servidor para distribuir las vlans en los demás switch.

```
Switch(config)#
Switch(config)#vtp mode server
Device mode already VTP SERVER.
Switch(config)#vtp domain NETWORKING
Changing VTP domain name from NULL to NETWORKING
Switch(config)#vtp password 12345NET
Setting device VLAN database password to 12345NET
```

Mode trunk :g0/1, f0/21-24 para traficar la comunicación entre paquetes que se enviaran en la red.

```

Switch(config)#int r g0/1-2
Switch(config-if-range)#switchport mode trunk

Switch(config-if-range)#
%LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet0/2,
changed state to down

%LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet0/2,
changed state to up

Switch(config-if-range)#exit
Switch(config)#int f0/24
Switch(config-if)#switchport mode trunk

Switch(config-if)#
%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/24,
changed state to down

%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/24,
changed state to up

```

Etherchannel

En cada Switch se configuran dos canales para el tráfico de paquetes.

Crear el primer protocolo en las interfaces 21-22 modo etherchannel

```

Switch(config)#int r f0/21-22
Switch(config-if-range)#channel-group 1 mode active
Switch(config-if-range)#
Creating a port-channel interface Port-channel 1

%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/21,
changed state to down

%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/21,
changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/22,
changed state to down

%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/22,
changed state to up

Switch(config-if-range)#exit
Switch(config)#interface port-channel 1
Switch(config-if)#switchport mode trunk
Switch(config-if)#switchport trunk allowed vlan 10,20,30
Switch(config-if)#exit
Switch(config)#

```

Creamos el primer protocolo en las interfaces 23-24 modo etherchannel

```

Switch(config)#int r f0/23-24
Switch(config-if-range)#sh

Switch(config-if-range)#
%LINK-5-CHANGED: Interface FastEthernet0/23, changed state to
administratively down

%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/23,
changed state to down

%LINK-5-CHANGED: Interface FastEthernet0/24, changed state to
administratively down

%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/24,
changed state to down

Switch(config-if-range)#channel-
Switch(config-if-range)#channel-g
Switch(config-if-range)#channel-group 2 mode active
Switch(config-if-range)#
Creating a port-channel interface Port-channel 2

Switch(config-if-range)#no sh

Switch(config-if-range)#
%LINK-5-CHANGED: Interface FastEthernet0/23, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/23,
changed state to up

%LINK-5-CHANGED: Interface FastEthernet0/24, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/24,
changed state to up

Switch(config-if-range)#exit
Switch(config)#interface port-channel 2
Switch(config-if)#switchport mode trunk
Switch(config-if)#switchport trunk allowed vlan 10,20,30
Switch(config-if)#exit
Switch(config)#exit

```

SWITCH 2 y SWITCH 3(CLIENT)

Digitar los siguientes comandos para configurar vtp en los switch clientes.

```

Switch(config)#vtp mode client
Setting device to VTP CLIENT mode.
Switch(config)#vtp domain NETWORKING
Domain name already set to NETWORKING.
Switch(config)#vtp password 12345NET
Setting device VLAN database password to 12345NET
Switch(config)#

```

Asignar interfaces a nuestras vlans 10,20,30

```

Switch(config)#int r f0/1-9
Switch(config-if-range)#switchport mode access
Switch(config-if-range)#switchport access vlan10

% Invalid input detected at '^' marker.

Switch(config-if-range)#switchport access vlan 10
Switch(config-if-range)#int r f0/10-15
Switch(config-if-range)#switchport mode access
Switch(config-if-range)#switchport access vlan 20
Switch(config-if-range)#int r f0/16-20
Switch(config-if-range)#switchport mode access
Switch(config-if-range)#switchport access vlan 30
Switch(config-if-range)#exit

```

Etherchannel

Creamos el primer protocolo en las interfaces 21-22 modo etherchannel que estarán conectados al SW1.

```

Switch(config)#int r f0/21-22
Switch(config-if-range)#sh

Switch(config-if-range)#
%LINK-5-CHANGED: Interface FastEthernet0/21, changed state to
administratively down

%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/21,
changed state to down

%LINK-5-CHANGED: Interface FastEthernet0/22, changed state to
administratively down

%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/22,
changed state to down

Switch(config-if-range)#channel-group 1 mode active
Switch(config-if-range)#
Creating a port-channel interface Port-channel 1

Switch(config-if-range)#sh
Switch(config-if-range)#exit
Switch(config)#interface port-channel 1
Switch(config-if)#switchport mode trunk
Switch(config-if)#switchport trunk allowed vlan 10,20,30
Switch(config-if)#exit
Switch(config)#exit

```

Creamos el primer protocolo en las interfaces g0/1-2 modo etherchannel que estarán conectados con el SW3.

```

Switch(config)#int r g0/1-2
Switch(config-if-range)#sh

Switch(config-if-range)#
%LINK-5-CHANGED: Interface GigabitEthernet0/1, changed state to
administratively down

%LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet0/1,
changed state to down

%LINK-5-CHANGED: Interface GigabitEthernet0/2, changed state to
administratively down

%LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet0/2,
changed state to down

Switch(config-if-range)#channel-group 3 mode active
Switch(config-if-range)#
Creating a port-channel interface Port-channel 3

Switch(config-if-range)#no sh

Switch(config-if-range)#
%LINK-5-CHANGED: Interface GigabitEthernet0/1, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet0/1,
changed state to up

%LINK-5-CHANGED: Interface GigabitEthernet0/2, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet0/2,
changed state to up
ex
%LINK-5-CHANGED: Interface Port-channel3, changed state to up

Switch(config-if)#switchport mode trunk
Switch(config-if)#switchport trunk allowed vlan 10,20,30
Switch(config-if)#exit

```

Switch 3

Creamos el primer protocolo en las interfaces 23-24 modo etherchannel, estarán conectadas con el switch 2

```

Switch(config)#int r f0/23-24
Switch(config-if-range)#sh

Switch(config-if-range)#
%LINK-5-CHANGED: Interface FastEthernet0/23, changed state to
administratively down

%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/23,
changed state to down

%LINK-5-CHANGED: Interface FastEthernet0/24, changed state to
administratively down

%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/24,
changed state to down

Switch(config-if-range)#channel-group 2 mode active
Switch(config-if-range)#
Creating a port-channel interface Port-channel 2

Switch(config-if-range)#sh
Switch(config-if-range)#no sh

Switch(config-if-range)#
%LINK-5-CHANGED: Interface FastEthernet0/23, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/23,
changed state to up

%LINK-5-CHANGED: Interface FastEthernet0/24, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/24,
changed state to up

%LINK-5-CHANGED: Interface Port-channel2, changed state to up
Switch(config-if-range)#exit
Switch(config)#interface port-channel 2
Switch(config-if)#switchport mode trunk
Switch(config-if)#switchport trunk allowed vlan 10,20,30
Switch(config-if)#exit
Switch(config)#exit

```

Creamos el primer protocolo en las interfaces g0/1-2 modo etherchannel

```

Switch(config)#int r g0/1-2
Switch(config-if-range)#sh

Switch(config-if-range)#
%LINK-5-CHANGED: Interface GigabitEthernet0/1, changed state to
administratively down

%LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet0/1,
changed state to down

%LINK-5-CHANGED: Interface GigabitEthernet0/2, changed state to
administratively down

%LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet0/2,
changed state to down

Switch(config-if-range)#channel-group 3 mode active
Switch(config-if-range)#
Creating a port-channel interface Port-channel 3

Switch(config-if-range)#no sh

Switch(config-if-range)#
%LINK-5-CHANGED: Interface GigabitEthernet0/1, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet0/1,
changed state to up

%LINK-5-CHANGED: Interface GigabitEthernet0/2, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet0/2,
changed state to up

Switch(config-if-range)#exit
Switch(config)#interface port-channel 3

Switch(config-if-range)#exit
Switch(config)#interface port-channel 3
Switch(config-if)#switchport mode trunk
Switch(config-if)#switchport trunk allowed vlan 10,20,30
Switch(config-if)#exit

```

A continuación se muestra las configuraciones del router más importante de nuestra topología ya que abarca la mayoría de configuraciones en el router REP y HOME se ingresa la ip a los seriales.

ROUTER HQ

Crear subinterfaces para el encapsulamiento entre vlan g0/1.10; g0/1.20; g0/1.30

```

Router(config)#int g0/1.10
Router(config-subif)#
%LINK-5-CHANGED: Interface GigabitEthernet0/1.10, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface
GigabitEthernet0/1.10, changed state to up

Router(config-subif)#encapsulation dot1Q 10
Router(config-subif)#ip add 172.16.42.1 255.255.255.224
Router(config-subif)#exit
Router(config)#int g0/1.20
Router(config-subif)#
%LINK-5-CHANGED: Interface GigabitEthernet0/1.20, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface
GigabitEthernet0/1.20, changed state to up

Router(config-subif)#encapsulation dot1Q 20
Router(config-subif)#ip add 172.16.42.33 255.255.255.224
Router(config-subif)#exit
Router(config)#int g0/1.30
Router(config-subif)#
%LINK-5-CHANGED: Interface GigabitEthernet0/1.30, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface
GigabitEthernet0/1.30, changed state to up

Router(config-subif)#encapsulation dot1Q 30
Router(config-subif)#ip add 172.16.42.65 255.255.255.224
Router(config-subif)#exit

```

Configurar dhcp para direcciones ip de forma automática asignar a vlans

```

-----
Router(config)#ip dhcp pool ADMINISTRADOR
Router(dhcp-config)#network 172.16.42.0 255.255.255.224
Router(dhcp-config)#default-router 172.16.42.1
Router(dhcp-config)#exit
Router(config)#ip dhcp pool WEBDESIGNER
Router(dhcp-config)#network 172.16.42.32 255.255.255.224
Router(dhcp-config)#default-router 172.16.42.33
Router(dhcp-config)#exit
Router(config)#ip dhcp pool CLIENTES
Router(dhcp-config)#network 172.16.42.64 255.255.255.224
Router(dhcp-config)#default-router 172.16.42.65
Router(dhcp-config)#exit

```

Ingresar ip en g0/1

```

Router(config)#int g0/1
Router(config-if)#ip add 200.165.209.65
% Incomplete command.
Router(config-if)#ip add 172.16.42.253 255.255.255.224
Router(config-if)#exit

```

Enrutamiento OSPF

Router HQ

```

router ospf 1
router-id 20.20.20.20
log-adjacency-changes
network 172.16.42.224 0.0.0.31 area 0
network 200.165.209.64 0.0.0.3 area 0
network 172.16.42.0 0.0.0.31 area 0
network 172.16.42.32 0.0.0.31 area 0
network 172.16.42.64 0.0.0.31 area 0
network 200.165.209.60 0.0.0.3 area 0

```

Router REP

```

router ospf 1
router-id 10.10.10.10
log-adjacency-changes
network 200.165.209.64 0.0.0.3 area 0
network 180.165.100.32 0.0.0.3 area 0

```

Router HOME

```

:
router ospf 1
router-id 10.10.10.20
log-adjacency-changes
network 180.165.100.32 0.0.0.3 area 0
network 192.168.1.0 0.0.0.31 area 0
network 200.165.209.60 0.0.0.3 area 0
!

```

ACL

En este apartado realizamos acl para permitir el acceso de vlan 20,30 al puerto 80, 443

```

access-list 100 permit tcp 172.16.42.64 0.0.0.31 host 172.16.42.226
eq www
access-list 100 permit tcp 172.16.42.32 0.0.0.31 host 172.16.42.226
eq www
access-list 100 permit tcp 172.16.42.64 0.0.0.31 host 172.16.42.226
eq 443
access-list 100 permit tcp 172.16.42.32 0.0.0.31 host 172.16.42.226
eq 443
access-list 100 permit tcp 172.16.42.32 0.0.0.31 host 172.16.42.226
eq 22
access-list 100 permit icmp any any

```

Asignar a la interfaz para realizar el tráfico de las ACL.

```

:
interface GigabitEthernet0/1.10
 encapsulation dot1Q 10
 ip address 172.16.42.1 255.255.255.224
 ip access-group 100 in
!
interface GigabitEthernet0/1.20
 encapsulation dot1Q 20
 ip address 172.16.42.33 255.255.255.224
 ip access-group 100 in
!
interface GigabitEthernet0/1.30
 encapsulation dot1Q 30
 ip address 172.16.42.65 255.255.255.224
 ip access-group 100 in
.

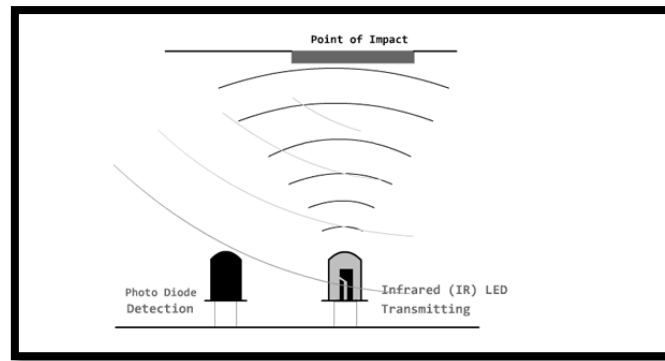
```

Sensores Utilizados.

1. Sensor Infrarrojo

Un detector de obstáculos infrarrojo es un dispositivo que detecta la presencia de un objeto mediante la reflexión que produce en la luz. El uso de luz infrarroja (IR) es simplemente para que esta no sea visible para los humanos.

Constitutivamente son sensores sencillos. Se dispone de un LED emisor de luz infrarroja y de un fotodiodo (tipo BPV10NF o similar) que recibe la luz reflejada por un posible obstáculo.

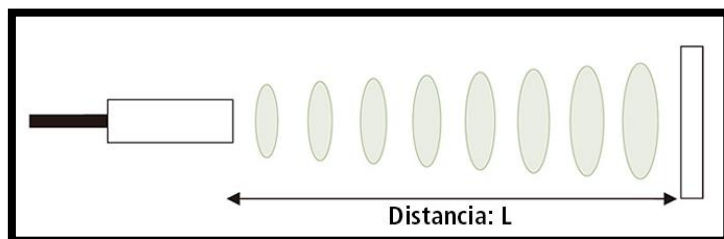


Los detectores de obstáculo suelen proporcionarse con una placa de medición estándar con el comparador LM393, que permite obtener la lectura como un valor digital cuando se supera un cierto umbral, que se regula a través de un potenciómetro ubicado en la placa.

Este tipo de sensores actúan a distancias cortas, típicamente de 5 a 20mm. Además, la cantidad de luz infrarroja recibida depende del color, material, forma y posición del obstáculo, por lo que no disponen de una precisión suficiente para proporcionar una estimación de la distancia al obstáculo.

2. Sensor Ultrasónico

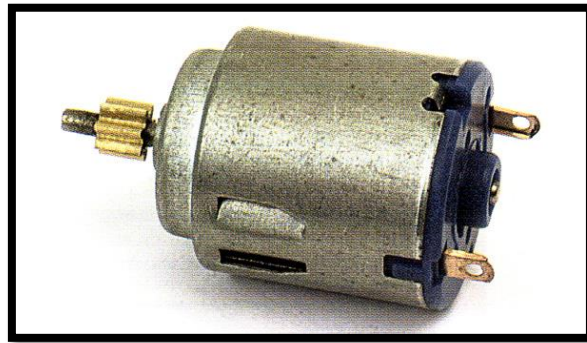
Como su nombre lo indica, los sensores ultrasónicos miden la distancia mediante el uso de ondas ultrasónicas. El cabezal emite una onda ultrasónica y recibe la onda reflejada que retorna desde el objeto. Los sensores ultrasónicos miden la distancia al objeto contando el tiempo entre la emisión y la recepción.



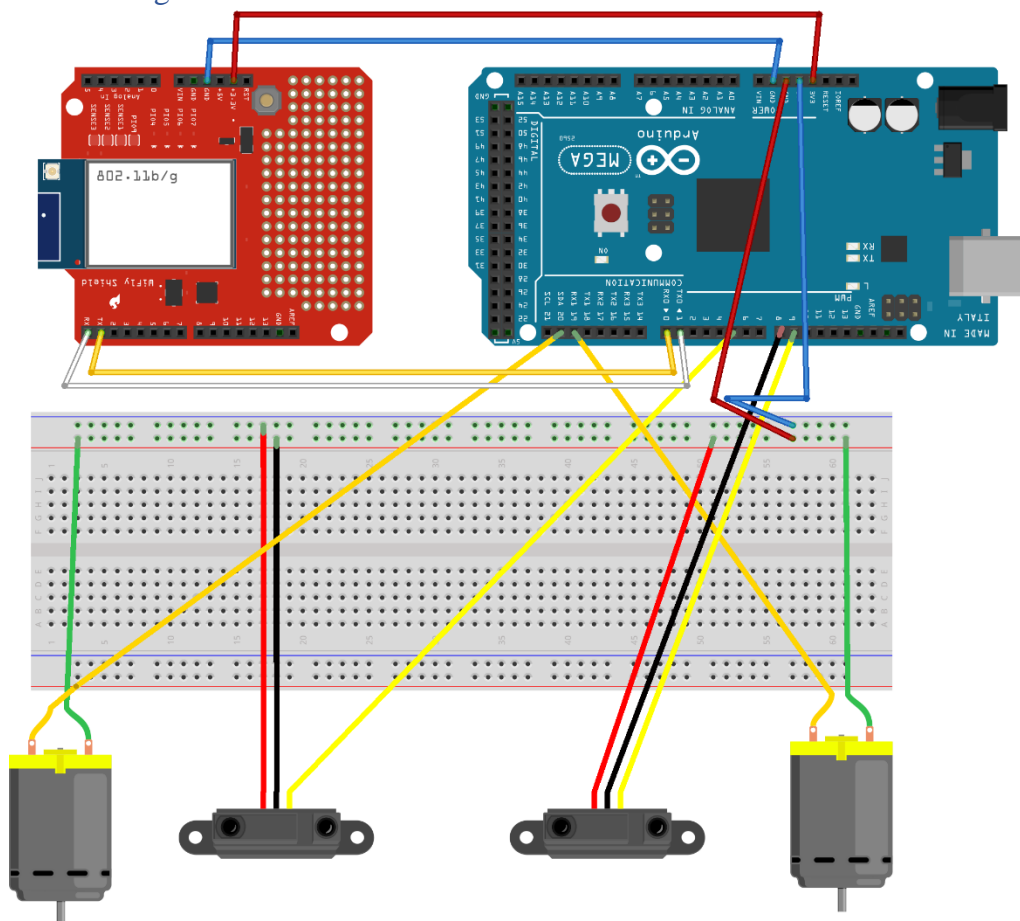
Como su nombre lo indica, los sensores ultrasónicos miden la distancia mediante el uso de ondas ultrasónicas. El cabezal emite una onda ultrasónica y recibe la onda reflejada que retorna desde el objeto. Los sensores ultrasónicos miden la distancia al objeto contando el tiempo entre la emisión y la recepción.

3. Motor

Los motores eléctricos son máquinas eléctricas rotatorias. Transforman una energía eléctrica en energía mecánica de rotación en un eje. Tienen múltiples ventajas, entre las que cabe citar su economía, limpieza, comodidad y seguridad de funcionamiento, el motor eléctrico a reemplazado en gran parte a otras fuentes de energía, tanto en la industria como en el transporte, las minas, el comercio, o el hogar.

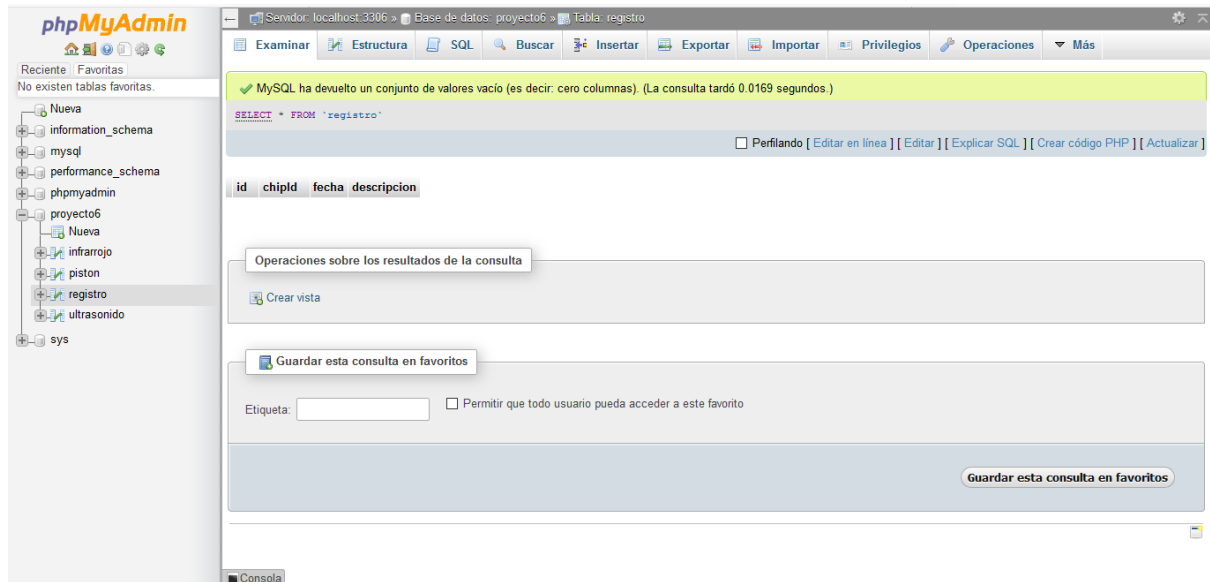


Circuito en Fritzing.

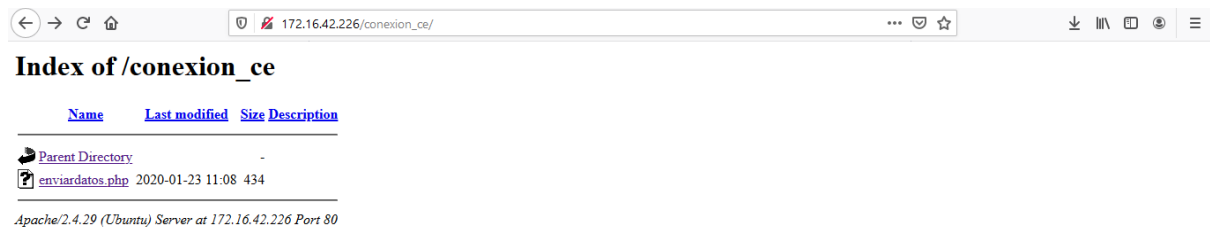


Control de Node MCU y programación de los embebidos.


Verificación de la tabla desde una máquina de acceso en modo cliente desde el punto de acceso inalámbrico.



Verificación de acceso a la IP del servidor para poder ver el archivo que permite envío de datos a la base de datos desde el módulo MCU.



Control de Node MCU.

 mcu Arduino 1.9.0-beta

Archivo Editar Programa Herramientas Ayuda



```
mcu
1 #include <SoftwareSerial.h>
2 #include <ESP8266WiFi.h>
3
4
5 int contconexion = 0;
6
7 const char *ssid = "GRUPO 2";
8 const char *password = "";
9
10 unsigned long previousMillis = 0;
11
12 char host[48];
13 String strhost = "172.16.42.226"; //
14 String strurl = "/conexion_ce/enviardatos.php";
15 String chipid = "";
16 int chipidl = 0;
17 SoftwareSerial NodeMCU(D2,D3);
18
19 //-----Función para Enviar Datos a la Base de Datos SQL-----
20
21 String enviardatos(String datos) {
22     String linea = "error";
23     WiFiClient client;
24     strhost.toCharArray(host, 49);
25
26     if (client.connect(host, 80)) {
27         Serial.println("exitosa ....");
28         return linea;
29     }else if (!client.connect(host, 80)) {
30         Serial.println("Fallo de conexión...");
31     }
```



```

28     return linea;
29 }else if (!client.connect(host, 80)) {
30     Serial.println("Fallo de conexión...");
31     return linea;
32 }
33
34 client.print(String("POST ") + strurl + " HTTP/1.1" + "\r\n" +
35             "Host: " + strhost + "\r\n" +
36             "Accept: */*" + "\r\n" +
37             "Content-Length: " + datos.length() + "\r\n" +
38             "Content-Type: application/x-www-form-urlencoded" + "\r\n" +
39             "\r\n" + datos);
40 delay(10);
41
42 Serial.print("Enviando datos a SQL...");
43
44 unsigned long timeout = millis();
45 while (client.available() == 0) {
46     if (millis() - timeout > 5000) {
47         Serial.println("Cliente fuera de tiempo!");
48         client.stop();
49         return linea;
50     }
51 }
52 // Lee todas las líneas que recibe del servidor y las imprime por el terminal serial
53 while(client.available()){
54     linea = client.readStringUntil('\r');
55 }
56 Serial.println(linea);
57 return linea;
58

```



```

64 Serial.begin(115200);
65 NodeMCU.begin(115200);
66 Serial.println("");
67 pinMode(D2, INPUT); //rx ----14ardunio
68 pinMode(D3, OUTPUT); //tx ----13arduino
69
70 Serial.print("chipId: ");
71 chipid = String(ESP.getChipId());
72 Serial.println(chipid);
73
74 // Conexión WIFI
75 WiFi.begin(ssid, password);
76 while (WiFi.status() != WL_CONNECTED and contconexion <50) {
77     ++contconexion;
78     delay(500);
79     Serial.print(".");
80 }
81 if (contconexion <50) {
82     //para usar con ip fija
83     IPAddress ip(192,168,1,2);
84     IPAddress gateway(192,168,1,1);
85     IPAddress subnet(255,255,255,0);
86     WiFi.config(ip, gateway, subnet);
87
88     Serial.println("");
89     Serial.println("Conexion establecida a: ");
90     Serial.println(WiFi.localIP());
91 }
92 else {
93     Serial.println("");
94 }
95

```

mcu Arduino 1.9.0-beta

Archivo Editar Programa Herramientas Ayuda

```
mcu $
77 while (WiFi.status() != WL_CONNECTED) and !conconexion
78 ++conconexion;
79 delay(500);
80 Serial.print(".");
81 }
82 if (conconexion < 50) {
83 //para usar con ip fija
84 IPAddress ip(192,168,1,2);
85 IPAddress gateway(192,168,1,1);
86 IPAddress subnet(255,255,255,0);
87 WiFi.config(ip, gateway, subnet);
88
89 Serial.println("");
90 Serial.println("Conexion establecida a: ");
91 Serial.println(WiFi.localIP());
92 }
93 else {
94 Serial.println("");
95 Serial.println("Error de conexion");
96 }
97 }
98 void loop() {
99
100 if(NodeMCU.available()){
101 String c = NodeMCU.readStringUntil('\n');
102 enviardatos("chipid=" + chipid + "&dato=" + c );
103 Serial.println(c);
104 }}
```

```
datosstring $
28 | // ENVIAR PULSO DE DISPARO EN EL PIN "TRIGGER"
29 digitalWrite(pintrigger, LOW);
30 delayMicroseconds(100000);
31 digitalWrite(pintrigger, HIGH);
32 // EL PULSO DURA AL MENOS 10 uS EN ESTADO ALTO
33 delayMicroseconds(100000);
34 digitalWrite(pintrigger, LOW);
35 tiempo = pulseIn(pinecho, HIGH); // MEDIR EL TIEMPO EN ESTADO ALTO DEL PIN "ECHO" EL PULSO ES PROPORCIONAL A LA DISTANCIA MEDIDA
36 // LA VELOCIDAD DEL SONIDO ES DE 340 M/S O 29 MICROSEGUNDOS POR CENTIMETRO
37 // DIVIDIMOS EL TIEMPO DEL PULSO ENTRE 58, TIEMPO QUE TARDA RECORRER IDA Y VUELTA UN CENTIMETRO LA ONDA SONORA
38 distancia = tiempo / 58;
39 // ENVIAR EL RESULTADO AL MONITOR SERIAL
40 Serial.print(distancia); //imprimir distancia
41 Serial.println(" cm");
42
43
44 if(digitalRead(3)) s_high = 1;
45 if(!digitalRead(3) && s_high)
46 {
47 s_high = 0;
48 counter += 1;
49 Serial.println("este es el valor del contador ");
50 Serial.println(counter); //dato a enviar numero de paquetes
51 // Serial3.write(counter);
52
53 }
54 if (distancia < 5) {
```

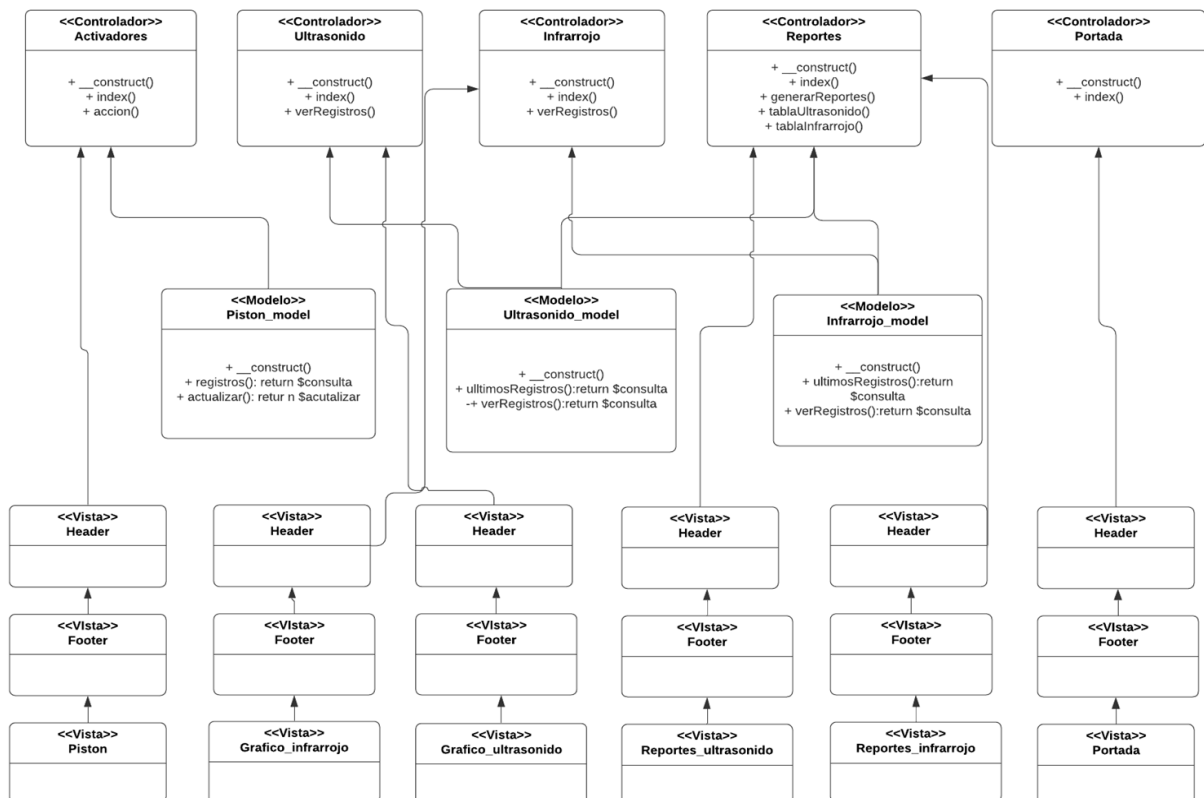
```

datosstring$
50 Serial.println(counter);
51 // Serial3.write(ccounter);
52
53 }
54 if (distancia <5 ) {
55   digitalWrite(13, HIGH);
56   delay(3000);
57
58
59   estado = 1;
60
61   Serial.print(distancia); //imprimir cuantos no pasan
62   envio= String('a')+String(counter)+ String('b')+String(distancia) ;
63   Serial.println(envio);
64   Serial3.println(envio);
65
66   }else {
67     digitalWrite(13, LOW);
68     estado =0;
69
70   }
71   envio= String('a')+String(counter)+ String('b')+String(distancia) ;
72   Serial.println(envio);
73   Serial3.println(envio);
74
75
76 }

```

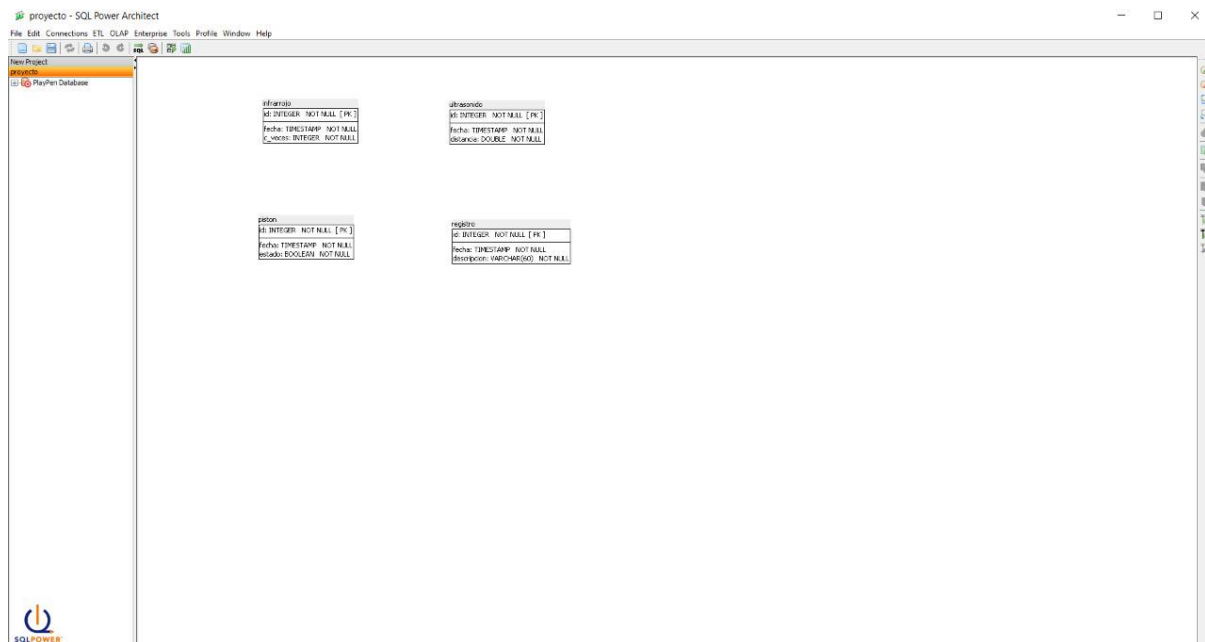
MVC(Modelo-Vista-Controlador).

La arquitectura MVC empleada para el desarrollo de este proyecto tiene la siguiente estructura para la comunicación y manejo de datos:



Diagramas de base de datos y componentes de la aplicación Web.

Diagrama de la base de datos



Los controladores empleados son los siguientes:

Activadores.php

```
<?php
defined('BASEPATH') OR exit('No direct script access allowed');

class Activadores extends CI_Controller{

    function __construct(){
        parent::__construct();
        $this->load->helper('url');
    }

    function index(){
        $this -> load -> view('encabezados/header.php');
        $this -> load -> view('activadores/piston.php');
        $this -> load -> view('encabezados/footer.php');
    }

    function grafico(){
        $this -> load -> view('encabezados/header.php');
        $this -> load -> view('activadores/piston.php');
        $this -> load -> view('encabezados/footer.php');
    }

}
```

Infrarrojo.php

```

Infrarrojo.php ×
application > controllers > Infrarrojo.php > Infrarrojo > __construct
1  <?php
2  defined('BASEPATH') OR exit('No direct script access allowed');
3
4  class Infrarrojo extends CI_Controller{
5
6      function __construct(){
7          parent::__construct();
8          $this->load->model('Infrarrojo_model');
9          $this->load->database();
10         $this->load->helper('url');
11     }
12
13     function index(){
14         $this->load->view('encabezados/header.php');
15         $datos["inf"] = $this->Infrarrojo_model->verRegistros();
16         $this->load->view('Infrarrojo/Grafico_infrarrojo.php');
17         $this->load->view('encabezados/footer.php');
18     }
19 }
20
21 }
22
23 ?>

```

Portada.php

```

Portada.php ×
application > controllers > Portada.php > ...
1  <?php
2  defined('BASEPATH') OR exit('No direct script access allowed');
3
4  class Portada extends CI_Controller{
5
6      function __construct(){
7          parent::__construct();
8          $this->load->helper('url');
9      }
10
11     function index(){
12         $this->load->view('encabezados/header.php');
13         $this->load->view('portada/Portada.php');
14         $this->load->view('encabezados/footer.php');
15     }
16
17     function grafico(){
18         $this->load->view('encabezados/header.php');
19         $this->load->view('activadores/piston.php');
20         $this->load->view('encabezados/footer.php');
21     }
22 }
23
24
25 ?>

```

Reporte.php


```

Reportes.php
application > controllers > Reportes.php > Reportes
1  <?php
2  defined('BASEPATH') OR exit('No direct script access allowed');
3
4  class Reportes extends CI_Controller{
5
6      function __construct(){
7          parent::__construct();
8          $this->load->helper('url');
9          $this->load->model('Ultrasonido_model');
10         $this->load->model('Infrarrojo_model');
11     }
12
13     function index(){
14         //$this->load->library('Pdf');
15         $this->load->view('encabezados/header.php');
16         $this->load->view('Reportes/Reportes.php');
17         $this->load->view('encabezados/footer.php');
18     }
19
20
21     public function generarReportes(){
22         $data = [];
23         $html = $this->input->post('txtPDF');
24         $pdfFilePath = "output_pdf_name.pdf";
25         $this->ob_clean();
26         $this->load->library('M_pdf');
27         $this->m_pdf->pdf->WriteHTML($html);
28         //$this->ob_end_clean();
29         $this->error_reporting(E_ALL);
30         $this->m_pdf->pdf->Output($pdfFilePath, "D");
31     }
32
33     function tablaUltrasonido(){
34         $this->load->view('encabezados/header.php');
35         $datos["ult"] = $this->Ultrasonido_model->verRegistros();
36         $this->load->view('Reportes/Reportes_ultrasonido.php',$datos);
37         $this->load->view('encabezados/footer.php');
38     }
39
40     function tablaInfrarrojo(){
41
42         $this->load->view('encabezados/header.php');
43         $datos["inf"] = $this->Infrarrojo_model->verRegistros();
44         $this->load->view('Reportes/Reportes_infrarrojo.php',$datos);
45         $this->load->view('encabezados/footer.php');
46     }

```

Ultrasonido.php

```

Ultrasonido.php
application > controllers > Ultrasonido.php > ...
1  <?php
2  defined('BASEPATH') OR exit('No direct script access allowed');
3
4  class Ultrasonido extends CI_Controller{
5
6      function __construct(){
7          parent::__construct();
8          $this->load->model('Ultrasonido_model');
9          $this->load->database();
10         $this->load->helper('url');
11     }
12
13     function index(){
14         $this->load->view('encabezados/header.php');
15         $datos["reg"] = $this->Ultrasonido_model->verRegistros();
16         $this->load->view('Ultrasonido/Grafico_ultrasonido.php',$datos);
17         $this->load->view('encabezados/footer.php');
18     }
19
20
21
22     function grafico(){
23         $this->load->view('encabezados/header.php');
24         $datos["dat"] = json_encode($this->Ultrasonido_model->obtenerJSON());
25         $this->load->view('Ultrasonido/Grafico_ultrasonido.php',$datos);
26         $this->load->view('encabezados/footer.php');
27     }
28
29 }
30
31

```

Modelos.

Infrarrojo_model

```
Infrarrojo_model.php X
application > models > Infrarrojo_model.php > ...
1  <?php
2
3  defined('BASEPATH') OR exit('No direct script access allowed');
4
5  class Infrarrojo_model extends CI_Model{
6
7      function __construct(){
8
9          parent::__construct();
10
11      }
12
13      function ultimosRegistros(){
14
15          $consulta = $this->db->query('SELECT * FROM `infrarrojo` ORDER BY `id` DESC LIMIT 20');
16          return $consulta;
17      }
18
19      function verRegistros(){
20
21          $consulta = $this->db->query('SELECT * FROM infrarrojo');
22          return $consulta;
23      }
24
25  }
26
27  }
28
29  >>
```

Ultrasonido_model.php

```
Ultrasonido_model.php X
application > models > Ultrasonido_model.php > ...
1  <?php
2
3  defined('BASEPATH') OR exit('No direct script access allowed');
4
5  class Ultrasonido_model extends CI_Model{
6
7      function __construct(){
8
9          parent::__construct();
10
11      }
12
13      function ultimosRegistros(){
14
15          $consulta = $this->db->query('SELECT * FROM `ultrasonido` ORDER BY `id` DESC LIMIT 20');
16          return $consulta;
17      }
18
19      function verRegistros(){
20
21          $this->load->database();
22          $consulta = $this->db->query('SELECT * FROM ultrasonido');
23          return $consulta;
24      }
25
26      function grafica(){
27
28          $this->load->database();
29          $consulta = $this->db->query('SELECT * FROM ultrasonido ORDER BY fecha DESC LIMIT 1');
30          foreach($consulta->result_array() as $dist){
31              echo $dist['distancia'];
32          }
33      }
34
35      function graficaReal(){
36
37          $this->load->database();
38          $consulta = $this->db->query('SELECT * FROM ultrasonido ORDER BY fecha DESC LIMIT 1');
39          return $consulta;
40      }
41
42  }
43
44  }
45
46  }
```

Vistas.

Footer.php

```
footer.php X
application > views > encabezados > footer.php
1 </div>
2 <!-- /.container-fluid -->
3
4 </div>
5 <!-- /#page-wrapper -->
6
7 </div>
8 <!-- /#wrapper -->
9
10
11
12 <!-- jQuery -->
13 <script src="<?php echo base_url(); ?>js/jquery.js"></script>
14
15 <!-- Bootstrap Core JavaScript -->
16 <script src="<?php echo base_url(); ?>js/bootstrap.min.js"></script>
17
18 <!-- Morris Charts JavaScript -->
19 <script src="<?php echo base_url(); ?>js/plugins/morris/raphael.min.js"></script>
20 <script src="<?php echo base_url(); ?>js/plugins/morris/morris.min.js"></script>
21 <script src="<?php echo base_url(); ?>js/plugins/morris/morris-data.js"></script>
22
23 </body>
24
25 </html>
```

Header.php

```
header.php X
application > views > encabezados > header.php
1 <!DOCTYPE html>
2 <html lang="en">
3
4 <head>
5
6 <meta charset="utf-8">
7 <meta http-equiv="X-UA-Compatible" content="IE=edge">
8 <meta name="viewport" content="width=device-width, initial-scale=1">
9 <meta name="description" content="">
10 <meta name="author" content="">
11
12 <title>Proyecto Integrador</title>
13
14 <!-- Bootstrap Core CSS -->
15 <link href="<?php echo base_url(); ?>css/bootstrap.min.css" rel="stylesheet">
16
17 <!-- Custom CSS -->
18 <link href="<?php echo base_url(); ?>css/sb-admin.css" rel="stylesheet">
19
20 <!-- Morris Charts CSS -->
21 <link href="<?php echo base_url(); ?>css/plugins/morris.css" rel="stylesheet">
22
23 <!-- Custom Fonts -->
24 <link href="<?php echo base_url(); ?>font-awesome/css/font-awesome.min.css" rel="stylesheet" type="text/css">
25
26
27
28 <!-- HTML5 Shim and Respond.js IE8 support of HTML5 elements and media queries -->
29 <!-- WARNING: Respond.js doesn't work if you view the page via file:// -->
30 <!--[if lt IE 9]>
31 <script src="https://oss.maxcdn.com/libs/html5shiv/3.7.0/html5shiv.js"></script>
32 <script src="https://oss.maxcdn.com/libs/respond.js/1.4.2/respond.min.js"></script>
33 <![endif]>
34
35 </head>
36
37 <body>
38
39 <div id="wrapper">
40
41 <!-- Navigation -->
42 <nav class="navbar navbar-inverse navbar-fixed-top" role="navigation">
43 <!-- Brand and toggle get grouped for better mobile display -->
44 <div class="navbar-header">
45 <button type="button" class="navbar-toggle" data-toggle="collapse" data-target=".navbar-ex1-collapse">
46 <span class="sr-only">Toggle navigation</span>
47 </div>
```

Grafico_ultrasonido.php

```
Grafico_ultrasonido.php X
application > views > Ultrasonido > Grafico_ultrasonido.php
33 }
34 .highcharts-data-table caption {
35     padding: 1em 0;
36     font-size: 1.2em;
37     color: #555;
38 }
39 .highcharts-data-table th {
40     font-weight: 600;
41     padding: 0.5em;
42 }
43 .highcharts-data-table td, .highcharts-data-table th, .highcharts-data-table caption {
44     padding: 0.5em;
45 }
46 .highcharts-data-table thead tr, .highcharts-data-table tr:nth-child(even) {
47     background: #f8f8f8;
48 }
49 .highcharts-data-table tr:hover {
50     background: #f1f7ff;
51 }
52
53 </style>
54
55 <body>
56
57 <figure class="highcharts-figure">
58     <div id="container"></div>
59     <p class="highcharts-description">
60         Gráfico de temperatura que muestra la actualización de datos cada segundo, con la eliminación de datos antiguos.
61     </p>
62 </figure>
63
64
65 <script type="text/javascript">
66     var intervaloTiempo;
67
68     Highcharts.chart('container', [{
69         chart: {
70             type: 'spline',
71             animation: Highcharts.svg, // don't animate in old IE
72             marginRight: 10,
73             events: {
74                 load: function () {
75
76                     // set up the updating of the chart each second
77                     var series = this.series[0];
```

Tabla_Ultrasonido

```
Reportes_infrarrojo.php X
application > views > Reportes > Reportes_infrarrojo.php
1 <!DOCTYPE html>
2 <html>
3
4 <head>
5 <title></title>
6 </head>
7
8 <body>
9
10
11 <!-- /.row -->
12 <div class="row">
13     <div class="col-lg-6">
14         <h2>Datos Infrarrojo</h2>
15         <div class="table-responsive">
16             <table class="table table-bordered table-hover">
17                 <thead>
18                     <tr>
19                         <th>ID</th>
20                         <th>Fecha</th>
21                         <th>Nº Veces</th>
22                     </tr>
23                 </thead>
24                 <tbody>
25                     <?php
26
27                     foreach($inf -> result_array() as $i){
28                         echo "<tr>";
29                         echo "<td>". $i['id']. "</td>";
30                         echo "<td>". $i['fecha']. "</td>";
31                         echo "<td>". $i['c_veces']. "</td>";
32                         echo "</tr>";
33                     }
34                 </tbody>
35             </table>
36         </div>
37     </div>
38 </div>
39
40 </body>
41
42 </html>
```

Tabla_Infrarrojo

```
Reportes_infrarrojo.php
application > views > Reportes > Reportes_infrarrojo.php
1 <!DOCTYPE html>
2 <html>
3
4 <head>
5 <title></title>
6 </head>
7
8 <body>
9
10
11 <!-- /.row -->
12 <div class="row">
13 <div class="col-lg-6">
14 <h2>Datos Infrarrojo</h2>
15 <div class="table-responsive">
16 <table class="table table-bordered table-hover">
17 <thead>
18 <tr>
19 <th>ID</th>
20 <th>Fecha</th>
21 <th>Nº Veces</th>
22 </tr>
23 </thead>
24 <tbody>
25 <?php
26
27 foreach($inf -> result_array() as $i){
28 echo "<tr>";
29 echo "<td>". $i['id']. "</td>";
30 echo "<td>". $i['fecha']. "</td>";
31 echo "<td>". $i['c_veces']. "</td>";
32 echo "</tr>";
33 }
34 </tbody>
35 </table>
36 </div>
37 </div>
38
39
40 </body>
41
42 </html>
```

Portada.php

```
Portada.php
application > views > portada > Portada.php
1 <html>
2
3 <head>
4
5 </head>
6
7 <body>
8
9 <center>
10
11 <h1>Proyecto Integrador</h1>
12
13 <h1>Networking-Programación Hipermedial-Sistemas Embebidos</h1>
14
15 <h1>Integrantes</h1>
16
17 <h1>Diego González, Arnaldo Ortiz, Bryan Sinchiguano, Wilian Tapia</h1>
18
19 </center>
20
21 </body>
22
23 </html>
```

Enlace a GitHub

<https://github.com/Grotso/Proyecto-Sexto>

6. Conclusiones

- Las placas de Arduino pueden llegar a ser un beneficio o un perjuicio para un determinado proyecto dado que no todas las placas usan los mismos drivers o métodos de comunicación para en envío y recepción de datos.
- Los equipos proporcionados, los cuales provienen de la empresa CISCO permiten una implementación de diferentes comandos que ayudan a que las configuraciones sean mucho mejor dentro de cada dispositivo evitando fallos.
- Cada uno de los sensores de lectura de datos puede tener librerías que ayuden a una manera de implementación más rápida de una determinada funcionalidad.
- Los puntos de acceso inalámbrico muchas veces pueden fallar dado que no todos tienen una certificación avalada que verifique si funcionamiento correcto por lo que se debe tomar en cuenta eso al momento de usar un punto de acceso.
- Existen plataformas de hardware libre, como lo es el Arduino, una placa con un controlador y un entorno de desarrollo, su sencillez y bajo costo nos permiten hacer múltiples diseños y múltiples tipos de uso como fue este proyecto.
- La comunicación entre la placa de Arduino y el módulo MCU no siempre puede ser llevada a cabo por medio del comando `Serial.write(variable)` dado que si se necesita enviar más e un dato se de usar una cabecera que permita este transporte de información.
- En conclusión, para poder compartir datos entre el módulo MCU y la base de datos es necesario enviar siempre los datos al monitor serial para poder visualizar previamente los datos que se van a almacenar.
- Se puede deducir por medio de las fallas dadas que un framework puede ser una ventaja o una desventaja al momento de ser implementado dentro de un proyecto , dado que no siempre ofrece una manera fácil de transmitir los datos.
- El uso de AJAX dentro de una aplicación web ayuda mucho dado que solo se ejecutan en segundo plano ciertas funciones que nosotros determinemos dejando de lado la recarga de la página completamente.
- EL internet de las cosas puede ser complicado de manipular, pero es una herramienta que está en constante crecimiento y puede ayudar a implementar cosas más grandes dentro de varios campos laborales.

Recomendaciones

- ✓ Verificar las conexiones , con sus respectivos pines asignados dentro de la placa de Arduino y nodeMCU respectivamente.
- ✓ Verificar que las interfaces a ser configuradas sean las correctas dado que si no lo hacemos puede dar un error o la configuración puede estar asignada a otra interfaz.
- ✓ Conectarse a la red de manera inalámbrica desde otros dispositivos antes de probar conexiones de envío desde el nodeMCU.
- ✓ Ir guardando cada una de las configuraciones que se hace ya sea en el nodeMCU o en las interfaces de cada uno de los equipos físicos dado que si se llega a cortar la energía las configuraciones se borrarán.
- ✓ Antes de hacer cualquier acción sobre alguno de los equipos físicos se debe verificar el correcto funcionamiento de estos, dado que esto puede afectar posteriormente.
- ✓ Probar cada uno de los sensores por separado antes de unificar en una sola unidad de código dado que así será más fácil verificar errores.
- ✓ Mediante el monitor serial verificar los datos a ser enviados por medio de la comunicación para no mandar valores erróneos a la base de datos.
- ✓ Investigar previamente el uso de una herramienta que no se conozca o que no se haya manejado antes para minimizar los errores y maximizar la eficiencia del código dentro de lo que respecta al trabajo con Frameworks.
- ✓ Brindar una guía mucho más desarrollada que ayude a demostrar un aprendizaje significativo en lo que respecta a las nuevas tecnologías y comunicación entre ellas.

Bibliografía y Referencias

- ❖ <https://www.arduino.cc/>
- ❖ <https://codeigniter.com/>
- ❖ <https://getbootstrap.com/docs/4.4/examples/>
- ❖ <https://www.highcharts.com/demo>
- ❖ https://www.nodemcu.com/index_en.html#fr_5475f7667976d8501100000f
- ❖ https://www.cisco.com/c/es_ec/index.html?CCID=cc000727&OID=0&DTID=pseggl000015&POSITION=SEM&COUNTRY_SITE=ec&CAMPAIGN=nbt-00&CREATIVE=EC_SEM_GEN_Pure-Brand_EM_B_RLSA_GGL_0_All-Visitors_Targeting_MLT-Cisco&REFERRING_SITE=Google&KEYWORD=cisco&ds_rl=1261909&ds_rl=1261909&gclid=Cj0KCQiAsbrxBRDpARIsAAnnz_NFu9diRtLssHTBmyptjNhNnYemIli7dCwMojxFGHTWa48KfILbqq8aAt1yEALw_wcB

- ❖ https://www.cisco.com/c/es_mx/support/docs/ip/access-lists/26448-ACLsamples.html
- ❖ <https://echaleunvistazo.wordpress.com/2012/06/11/configurar-dhcp-en-router-cisco/>