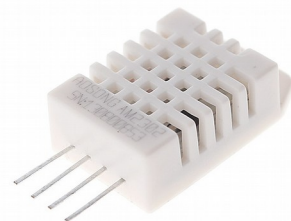
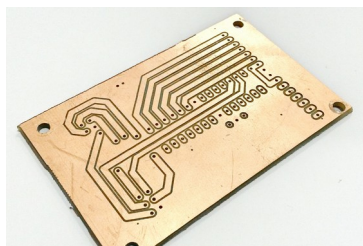
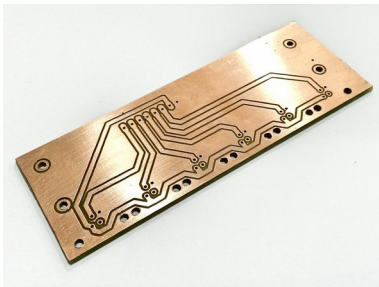
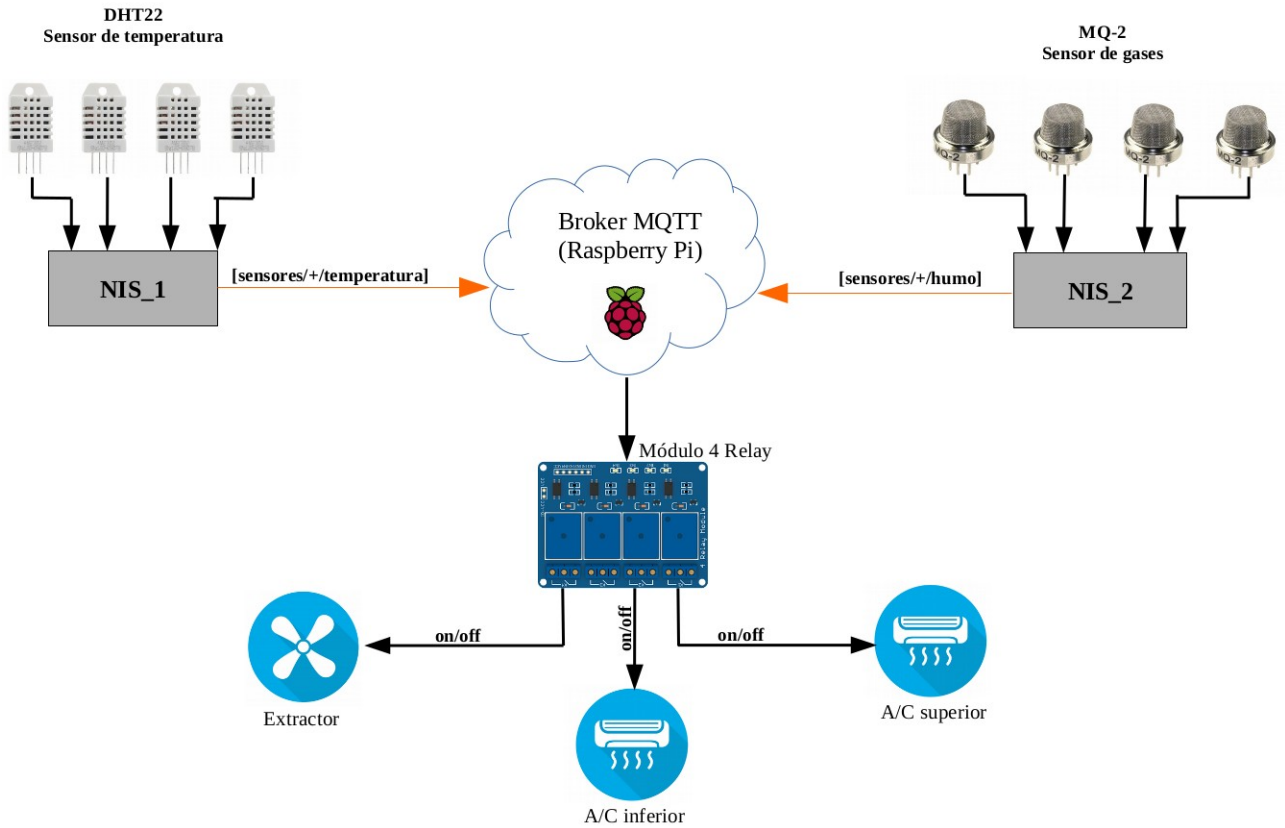


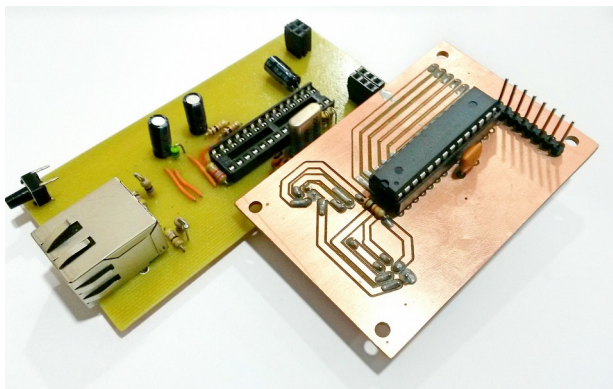
Manual de Instalación

Montaje de dispositivos

Tomando como base el esquema siguiente y guiado por las fotografías adjuntas mostrar los sensores en el Raspberry

DIAGRAMA DE COMUNICACIÓN





Para ver las fotografías a mayor resolución consulte la carpeta “fotografías” adjunto con el código fuente.

Manual de instalación del servicio para obtener información del estado de los sensores ambientales (Backend)

Nota.- Para el correcto funcionamiento del programa, es necesario que los dispositivos estén montados y conectados. Mediante modo remoto instalar el backend en el Raspberry.

Instalación de programas basicos y librerias

```
sudo apt-get update
sudo apt-get install -y curl
sudo apt-get install -y git

sudo apt-get install -y libzmq3-dev
sudo apt-get install -y libkrb5-dev
```

Instalación de Node

```
curl -sL https://deb.nodesource.com/setup_4.x | sudo -E bash -
sudo apt-get install -y nodejs
sudo apt-get install -y build-essential
```

Compilar el programa

Ingresar a la carpeta del código fuente en especifico a la carpeta **mqtt**

```
cd sensores-ambientales-datacenter
cd mqtt
```

Instalar las dependencias necesarias

```
cd librerias
npm install
```

Instalacion de MongoDB

Instalamos el gestor de base e iniciamos el servicio

Actualizamos repositorios

Primero actualizamos los repositorios para poder instalar mongo

```
sudo apt-key adv --keyserver hkp://keyserver.ubuntu.com:80 --recv
0C49F3730359A14518585931BC711F9BA15703C6
```

```
echo "deb http://repo.mongodb.org/apt/debian jessie/mongodb-org/3.4 main" | sudo
tee /etc/apt/sources.list.d/mongodb-org-3.4.list
```

```
sudo apt-get update
```

Instalar mongodb

```
sudo apt-get install -y mongodb-org
```

```
sudo service mongod start
```

Creamos la base de datos

Ejecutar los siguientes scripts

```
cd ..
cd ..
cd mongo
cd sensores
mongo principal.js
cd ejemplo
mongo principal.js
```

```
cd ..
cd ..
cd usuarios
mongo principal.js
cd ejemplo
mongo principal.js
```

Probar los scripts

Probar la ejecución de cada script para verificar que funcionen correctamente.

Nota.- Para que estos scripts funcionen correctamente:

- Todos estos scripts deben estar instalados en el dispositivo.
- Todos los conectores deben estar montados según el manual de montaje de dispositivos.

```
cd ..
cd ..
```

```

cd ..
cd mqtt

node suscriptores/subMosca.js

node suscriptores/subSensorBD.js

node suscriptores/subActuadorBD.js

node suscriptores/subDisparadorTemperatura.js

node suscriptores/subDisparadorHumedad.js

node suscriptores/subDisparadorHumo.js

node suscriptores/subActuador.js

node suscriptores/subGpio.js

```

Instalar el programa como servicio

Si los anteriores scripts funcionan correctamente levantarlos todos como servicio utilizando *forever*

Configurar forever

Configurar las aplicaciones segun la carpeta en donde se encuentre instalado el programa

```
pwd
```

Prestar atención en la ruta que se muestra, sera algo parecido a: */home/jaimito/sensores-ambientales-datacenter* este sera utilizado para el archivo de configuración del **forever** Editar el archivo *sensores-ambientales-datacenter.json*

```
nano forever/sensores-ambientales-datacenter.json
```

Cambiar el **sourceDir** de todos los parametros segun corresponda

```

[
  {
    "uid": "ServidorMosca",
    "append": true,
    "watch": true,
    "script": "subMosca.js",
    "sourceDir": "/home/jaimito/sensores-ambientales-
datacenter/mqtt/suscriptores"
  },
  {
    "uid": "SensorBD",
    "append": true,
    "watch": true,
    "script": "subSensorBD.js",
    "sourceDir": "/home/jaimito/sensores-ambientales-
datacenter/mqtt/suscriptores"
  },
  {
    "uid": "ActuadorBD",
    "append": true,

```

```

    "watch": true,
    "script": "subActuadorBD.js",
    "sourceDir": "/home/jaimito/sensores-ambientales-
datacenter/mqtt/suscriptores"
  },
  {
    "uid": "DisparadorTemperatura",
    "append": true,
    "watch": true,
    "script": "subDisparadorTemperatura.js",
    "sourceDir": "/home/jaimito/sensores-ambientales-
datacenter/mqtt/suscriptores"
  },
  {
    "uid": "DisparadorHumedad",
    "append": true,
    "watch": true,
    "script": "subDisparadorHumedad.js",
    "sourceDir": "/home/jaimito/sensores-ambientales-
datacenter/mqtt/suscriptores"
  },
  {
    "uid": "DisparadorHumo",
    "append": true,
    "watch": true,
    "script": "subDisparadorHumo.js",
    "sourceDir": "/home/jaimito/sensores-ambientales-
datacenter/mqtt/suscriptores"
  },
  {
    "uid": "Actuador",
    "append": true,
    "watch": true,
    "script": "subActuador.js",
    "sourceDir": "/home/jaimito/sensores-ambientales-
datacenter/mqtt/suscriptores"
  },
  {
    "uid": "GPIO",
    "append": true,
    "watch": true,
    "script": "subGpio.js",
    "sourceDir": "/home/jaimito/sensores-ambientales-
datacenter/mqtt/suscriptores"
  }
]

```

Ejecutar todos los scripts con forever

```

sudo npm install forever -g
forever start forever/sensores-ambientales-datacenter.json

```

Si todo salio correctamente los servicios ya fueron levantados y pueden ser probados con el frontend

Nota.- Por defecto el usuario es *demo* y la contraseña *arcon412*

Manual de instalación del servicio para obtener información del estado de los sensores ambientales (Frontend)

Nota.- Para continuar con la instalación es estrictamente necesario que los sensores estén encendidos y el servicio del backend esté levantado correctamente caso contrario el sistema no funcionara y mostrara errores.

Instalacion de programas basicos

```
sudo apt-get update
sudo apt-get install -y curl
sudo apt-get install -y git
```

Instalación de Node

```
curl -sL https://deb.nodesource.com/setup_4.x | sudo -E bash -
sudo apt-get install -y nodejs
sudo apt-get install -y build-essential
```

Compilar el programa

Ingresar a la carpeta del codigo fuente en especifico a la carpeta frontend

```
cd sensores-ambientales-datacenter
cd frontend
```

Configurar la IP donde esta instalado el servidor

```
nano app/configuracion.js
```

Editar la primera linea segun corresponda a la ip del Raspberry (backend)

```
var ip_backend='127.0.0.1';

exports.bd = {
  usuarios: 'mongodb://' + ip_backend + ':27017/usuarios',
  sensores:
'mongodb://' + ip_backend + ':27017/sensores_control_ambiental_datacenter'
}

exports.clave = {
.
.
.

nano publico/js/app.js
```

Editar la primera linea segun corresponda a su ip

```
var serv_backend='http://127.0.0.1:3000';
```

```
var app = angular.module('app', ['ngRoute', 'angular-jwt']);

app.config(function($routeProvider, $locationProvider, $httpProvider,
jwtInterceptorProvider) {
  .
  .
  .
})
```

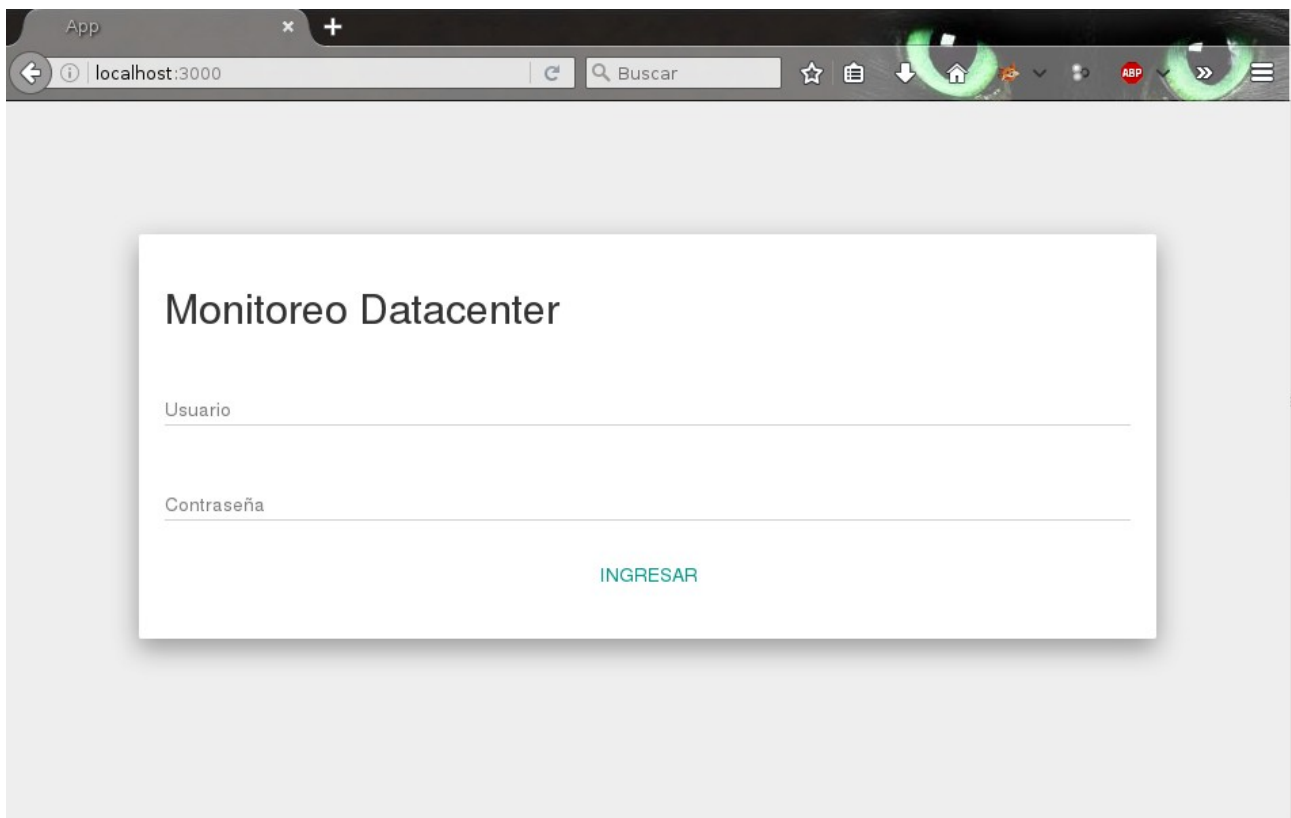
Instalar las dependencias necesarias

```
npm install
```

Ejecutar el programa:

```
node app.js
```

Para comprobar que el frontend este ejecutandose ingresar en el navegador a la ruta
<http://localhost:3000>



Instalar el programa como servicio

Instalar Supervisor para que el servidor pueda ejecutar la aplicación como servicio

```
sudo apt-get install supervisor
sudo /etc/init.d/supervisor restart
```

Crear el archivo de configuración

```
sudo nano /etc/supervisor/conf.d/frontend-sensores.conf
```

Ejemplo de contenido del archivo *frontend-sensores.conf*. Se Debe cambiar los nombres de la ruta con las que correspondan a tu aplicación.

```
[program:run-frontend-sensores]
directory=/home/jaimito/desarrollo/sensores-ambientales-datacenter/frontend
command=node app.js
autostart=true
autorestart=true
stderr_logfile=/var/log/frontend-sensores.err.log
```

Reiniciar supervisor, para que los cambios tomen efecto.

```
sudo /etc/init.d/supervisor restart
```