

Environments Monitoring & Management Application

EMMA es una aplicación web desarrollada por Antonio Díaz Pozuelo (adpozuelo@gmail.com) cuyo objetivo es monitorizar y gestionar Instalaciones Científico Técnicas (ICT).

Diseño y arquitectura

EMMA se ha diseñado e implementado utilizando dos componentes principales:

1. Sensores (Python): “scripts” implementados a medida, que se encargan de recoger las métricas del sensor y guardarlas en la base de datos. Estos programas se instalan sobre una Raspberry Pi, la cual tiene conectado el sensor correspondiente, o sobre un equipo de computación o nodo a monitorizar. Actualmente se encuentran implementados sensores para DHT11/DHT22 (temperatura y humedad), KY038 (sonido), equipo o nodo de cálculo (con y sin GPU).
2. Servidor LAMP (Linux+Apache2+MariaDB+PHP):
 1. Aplicación Web (CodeIgniter+Bootstrap+ChartJS): nos permite monitorizar y gestionar la ICT mediante una interfaz web que se adapta a todos los navegadores y dispositivos electrónicos (*responsive web*).
 2. Base de datos (MariaDB): la cual almacena las métricas recogidas por los sensores y todos los datos necesarios para el correcto funcionamiento de la aplicación web.
 3. Gestión de alertas e informes (Python+Telegram): “scripts” que envían alertas a los teléfonos móviles (mediante Telegram) ante fallos en los sensores (caída de un equipo o sensor) y/o en las métricas (alta temperatura o humedad, sonido procedente de una alarma, etc.). Además, cada cierto intervalo, el servidor envía un informe estadístico de las métricas deseadas a los teléfonos móviles que pertenezcan al grupo de Telegram indicado.

Instalación

Por un lado, la instalación del servidor se realiza de la siguiente manera (Ubuntu 20.04):

1. Instalación de Python >= 3.6:
\$ apt-get install python3
2. Instalación de paquetes requeridos:
\$ pip install psutil gputil py-cpuinfo peewee PyMySQL
3. Instalación y configuración de la base de datos:
\$ apt-get install mariadb-server
\$ mysql_secure_installation
\$ mysql -u root -p
> CREATE DATABASE emma;
> CREATE USER 'emma'@'%' IDENTIFIED BY 'yourpassword';
> CREATE USER 'emma'@'localhost' IDENTIFIED BY 'yourpassword';
> GRANT ALL PRIVILEGES ON emma.* to 'emma'@'%';
> GRANT ALL PRIVILEGES ON emma.* to 'emma'@'localhost';
> SELECT user,host FROM mysql.user;
> SHOW GRANTS for emma;
4. Instalación y configuración de Apache2:
\$ apt-get install apache2 php php-mysql libapache2-mod-php \
php-json php-mbstring php-mysqlnd php-xml php-intl php-curl
\$ a2enmod rewrite
\$ nano /etc/apache/sites-enabled/000-default

- ```

+ <Directory "/var/www/html/public">
+ Options Indexes FollowSymLinks
+ AllowOverride All
+ Require all granted
+ </Directory>

```
5. Descomprimir CodeIgniter 4 (<https://codeigniter.com/>) en el directorio web de Apache2:
 

```

$ unzip codeigniter4-framework-v4.3.1-0-gb1e5c64.zip
$ mv codeigniter4-framework-b1e5c64/* /var/www/html/.

```
  6. Descomprimir la aplicación EMMA dentro del framework CodeIgniter:
 

```

$ cd /var/www/html
$ tar zxvf ~/EMMA.tgz
$ chown -R www-data:www-data *
$ systemctl restart apache2.service

```
  7. Configurar EMMA:
 

```

$ nano app/Database/Seeds/Init.php ← (Define tus credenciales de administrador).
$ nano env ← (Define tu configuración de la base de datos).
$ cp env .env

```
  8. Copiar los siguientes “scripts” en /usr/local/bin y a añadir las siguientes líneas en CRON:
 

```

$ cp sensors/check_alarms.py /usr/local/bin/.
$ nano /usr/local/bin/check_alarms.py ← (Cambia GROUP por tu grupo de Telegram).
$ cp sensors/check_cluster.py /usr/local/bin/.
$ cp sensors/sensors_report.py /usr/local/bin/.
$ nano /usr/local/bin/sensors_report.py ← (Cambia GROUP por tu grupo de Telegram).
$ cp sensors/emma_model.py /usr/local/bin/.
$ nano /usr/local/bin/emma_model.py ← (Define tu configuración de la base de datos).
$ export EDITOR=nano && crontab -e
+ SHELL=/bin/bash
+ */1 * * * * /usr/local/bin/check_alarms.py&>>/var/log/emma.log
+ */10 * * * * /usr/local/bin/check_cluster.py&>>/var/log/emma.log
+ 0 20 * * * /usr/local/bin/sensors_report.py&>>/var/log/emma.log

```
  9. Rotación de logs:
 

```

$ nano /etc/logrotate.d/emma
+ /var/log/emma.log {
+ rotate 4
+ weekly
+ compress
+ missingok
+ notifempty
+ }

```
  10. Inicializa EMMA para cargar las credenciales de administrador:
 

```

$ cd /var/www/html
$ php spark db:seed Init

```
  11. Puede modificar el HOME de la web en el siguiente fichero app/Views/home.php
  12. EMMA está instalado y configurado para acceder mediante <http://localhost>

Por otro lado, la instalación de los clientes (sensores o equipos) se realiza de la siguiente manera (Ubuntu 20.04 o Raspberry Pi OS):

1. Instalación de Python >= 3.6:
 

```

$ apt-get install python3-dev python3-pip

```
2. Instalación de paquetes requeridos:
 

```

$ python3 -m pip install --upgrade pip setuptools wheel
$ pip3 install adafruit-circuitpython-dht psutil peewee PyMySQL ← (Raspberry Pi OS)

```

- \$ pip3 install psutil gputil py-cpuinfo peewee PyMySQL ← (Ubuntu 20.04)
3. En Raspberry es recomendable desactivar el Wifi/Bluetooth (si no se va a utilizar) y eliminar Avahi-Daemon:
 

```
$ apt-get remove avahi-daemon
$ nano /etc/sysctl.d/99-sysctl.conf
+ net.ipv6.conf.all.disable_ipv6 = 1
+ net.ipv6.conf.default.disable_ipv6 = 1
+ net.ipv6.conf.lo.disable_ipv6 = 1
$ sysctl -p
```
  4. Copiar el sensor deseado desde el directorio “sensors” al directorio \$HOME (Raspberry Pi OS):
 

```
$ cp sensors/emma_model.py ~/.
$ nano ~/emma_model.py ← (Define tu configuración de la base de datos).
$ cp sensors/dht11_sensor.py ~/. ← Sensor DHT11.
$ cp sensors/dht22_sensor.py ~/. ← Sensor DHT22.
$ cp sensors/KY038_sound_sensor.py ~/. ← Sensor KY038.
$ cp sensors/node_sensor.py ~/. ← Sensor equipo o nodo.
```
  5. Configurar CRON (ejemplo con sensor DHT11 para que registre lecturas cada 5 minutos):
 

```
$ export EDITOR=nano && crontab -e
+ SHELL=/bin/bash
+ */5 * * * * ~/dht11_sensor.py&>>/home/pi/dht11_sensor.log
```
  6. Rotación de logs:
 

```
$ nano /etc/logrotate.d/dht11_sensor
+ /home/pi/dht11_sensor.log {
+ rotate 4
+ weekly
+ compress
+ missingok
+ notifempty
+ }
```

Para cualquier duda o sugerencia, por favor, enviar correo electrónico a [adpozuelo@gmail.com](mailto:adpozuelo@gmail.com).

Simbología:

- \$ → Comando a ejecutar desde el interprete de comandos del sistema operativo.  
 > → Comando a ejecutar desde el interprete de comandos de MariaDB.  
 + → Línea a añadir en el fichero editado.