

# Infraestructura Tecnológica

The TICK stack is a collection of products from the developers of the time-series database InfluxDB. It is made up of the following<sup>1)</sup>

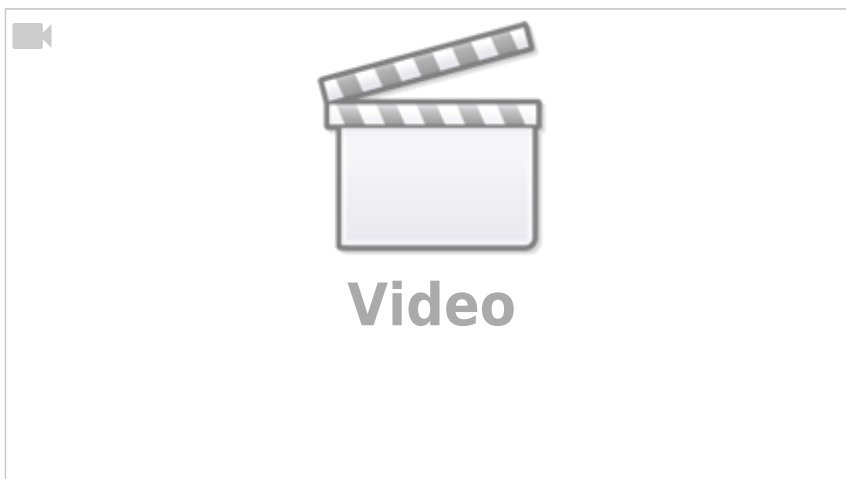
components:

- **Telegraf** collects time-series data from a variety of sources.
- **InfluxDB** stores time-series data.
- **Chronograf** visualizes and graphs the time-series data.
- **Kapacitor** provides alerting and detects anomalies in time-series data.

You can use each of these components separately, but if you use them together, you'll have a scalable, integrated open-source system for processing time-series data.

In this tutorial you'll set up and use this platform as an open-source monitoring system. You'll generate a bit of CPU usage and receive an email alert when the usage gets too high.

<https://www.influxdata.com/blog/building-iot-time-series-demo/>



[https://www.youtube.com/watch?v=woepS38F\\_90](https://www.youtube.com/watch?v=woepS38F_90)

<https://www.influxdata.com/time-series-platform/>

<https://forums.docker.com/t/docker-and-iptables-configuration-startup/904>

<https://github.com/CWempe/rest2influxdb/blob/master/rest2influxdb.sh>

<http://blog.sequenceiq.com/blog/2014/08/12/docker-networking/>

<https://www.digitalocean.com/community/tutorials/how-to-configure-the-linux-firewall-for-docker-swarm-on-ubuntu-16-04>

## MQTT

<http://platformio.org/lib/show/89/PubSubClient/examples> <https://pubsubclient.knolleary.net/>

<http://platformio.org/lib/show/346/AsyncMqttClient> <https://github.com/marvinroger/async-mqtt-client>

<http://marvinroger.viewdocs.io/async-mqtt-client/>

**influx data format** → [https://docs.influxdata.com/influxdb/v1.3/write\\_protocols/line\\_protocol\\_tutorial/](https://docs.influxdata.com/influxdb/v1.3/write_protocols/line_protocol_tutorial/)

## Convertir CSV en Flash a influx data format para enviar por MQTT

### Función strtok:

#### EXAMPLE

The program below uses nested loops that employ strtok\_r() to break a string into a two-level hierarchy of tokens. The first command-line argument specifies the

string to be parsed. The second argument specifies the delimiter byte(s) to be used to separate that string into "major" tokens. The third argument specifies the

delimiter byte(s) to be used to separate the "major" tokens into subtokens.

An example of the output produced by this program is the following:

```
$ ./a.out 'a/bbb///cc;xxx:yyy:' ':' '/'
1: a/bbb///cc
    --> a
    --> bbb
    --> cc
2: xxx
    --> xxx
3: yyy
    --> yyy
```

#### Program source

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

int
main(int argc, char *argv[])
{
    char *str1, *str2, *token, *subtoken;
    char *saveptr1, *saveptr2;
    int j;

    if (argc != 4) {
        fprintf(stderr, "Usage: %s string delim subdelim\n",
                argv[0]);
        exit(EXIT_FAILURE);
    }
```

```
    for (j = 1, str1 = argv[1]; ; j++, str1 = NULL) {
        token = strtok_r(str1, argv[2], &saveptr1);
        if (token == NULL)
            break;
        printf("%d: %s\n", j, token);

        for (str2 = token; ; str2 = NULL) {
            subtoken = strtok_r(str2, argv[3], &saveptr2);
            if (subtoken == NULL)
                break;
            printf(" --> %s\n", subtoken);
        }
    }

    exit(EXIT_SUCCESS);
}
```

1)

<https://www.digitalocean.com/community/tutorials/how-to-monitor-system-metrics-with-the-tick-stack-on-ubuntu-16-04>

From:  
<https://wiki.unloquer.org/> -

Permanent link:  
[https://wiki.unloquer.org/personas/brolin/proyectos/agentes\\_calidad\\_aire/infraestructura\\_iot?rev=1506560678](https://wiki.unloquer.org/personas/brolin/proyectos/agentes_calidad_aire/infraestructura_iot?rev=1506560678)

Last update: **2017/09/28 01:04**

