

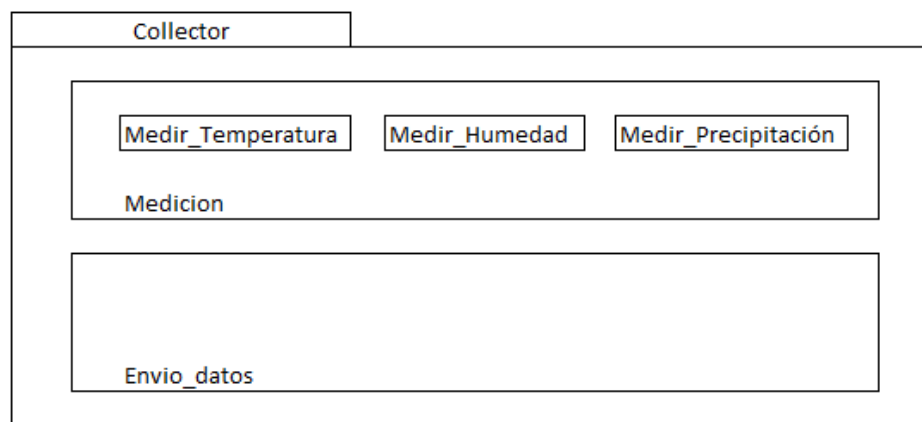


Estación meteorológica

1. Descripción de los módulos del sistema:

Se contarán con tres modulos, un modulo para el recolectar de los datos **Collector**, un modulo para el servidor intermedio **Brocker**, y un modulo para los servidores **Server**.

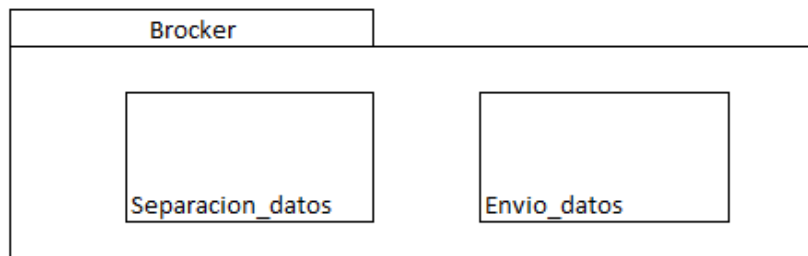
1.1. Collector: Se encarga de controlar un pequeño computador, que es utilizado para la recolección de información climática. Toma cada cierto tiempo mediciones de temperatura, humedad y nivel de lluvias (precipitación). Envía esta información al servidor intermedio.



1.1.1. Medicion: Se encarga de medir las variables planteadas: Temperatura, Humedad, Precipitación. Como se trata de un ejercicio académico, los medidores a implementar no van a leer datos reales, en su lugar van a producir datos aleatorios para cada medida. Estos datos deben ser realistas.

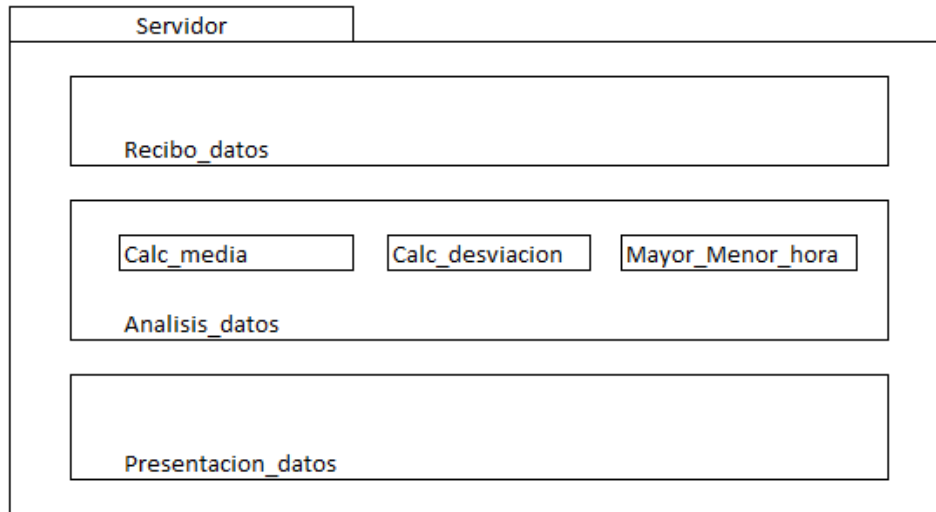
1.1.2. Envio_datos: Se encarga de estructurar los datos medidos en un JSON, serializará y enviará el mensaje al Servidor intermedio.

1.2. Brocker: Se encarga de que al llegar los datos sean separados para su análisis. Ya con los datos separados, envía a cada servidor específico el dato correspondiente, es decir las mediciones de temperatura van al servidor de temperatura, las de humedad al servidor de humedad, las de precipitación al servidor de precipitación.





1.3. Server: Este módulo es general, puede ser usado para las tres medidas planteadas. Se encarga de recibir los datos, interpretarlos y producir informes estadísticos de los datos. Se calculan las siguientes medidas: Media, Desviación estándar y hora del día donde se producen la menor y la mayor de las medidas. Estos resultados son guardados en un archivo



2. Interacción por paso de mensajes entre cada uno de los módulos

2.1. Collector, después de recoger toda la información en su módulo de Medición, genera un JSON con la información:

```
data = {
    "Computer": "Comp15A",
    "hour" : "Sat Jan 8 20:07:41 2017",
    "Temperature": 32,
    "Humidity": 70,
    "Precipitation": 500
}
```

Información que serializa y envía en un mensaje al servidor intermedio **Broker**

2.2. Broker recibe el mensaje de **Collector**, el cual deserializa y divide en tres partes:

```
dataT = {
    "hour": "Sat Jan 8 20:07:41 2017",
    "Temperature": 32
}

dataH = {
```



Universidad
Tecnológica
de Pereira

"hour": "Sat Jan 8 20:07:41 2017",

"Humidity": 32

}

dataP = {

"hour": "Sat Jan 8 20:07:41 2017",

"Precipitation": 32

}

Esta información es serializada y enviada al **Server** respectivo.

2.3. El **Server** recibe el mensaje y lo deserializa, y realiza los cálculos que le corresponde.

3. El estado de cada módulo y la manera en que será almacenado.

3.1. Collector: Este módulo no guarda ninguna información en memoria física, la información que maneja la trabaja directamente en la memoria RAM.

3.2. Brocker: No guarda información en memoria física, trabaja directamente en la memoria RAM

3.3. Server: Después de realizar los cálculos estadísticos planteados, guarda la información en dos archivos. Un archivo **.json** que contiene todos los datos medidos y otro archivo **.json** que contiene el análisis estadístico de los mismos.