

## Sistema de Supervisão Predial

**Aluno:** Javan de Mesquita Torres

### Justificativa

O projeto sistema de supervisão predial nasceu da necessidade de se monitorar a temperatura de 5.000 pontos do prédio, para em caso de uma emergência se disparar um alarme de incêndio de forma rápida e eficiente.

### Objetivos

Coletar dados de temperatura de 5.000 pontos, com cada ponto sendo consultado a cada 5 segundos, caso a temperatura de um ponto exceda uma temperatura julgada como segura um alarme deve ser disparado.

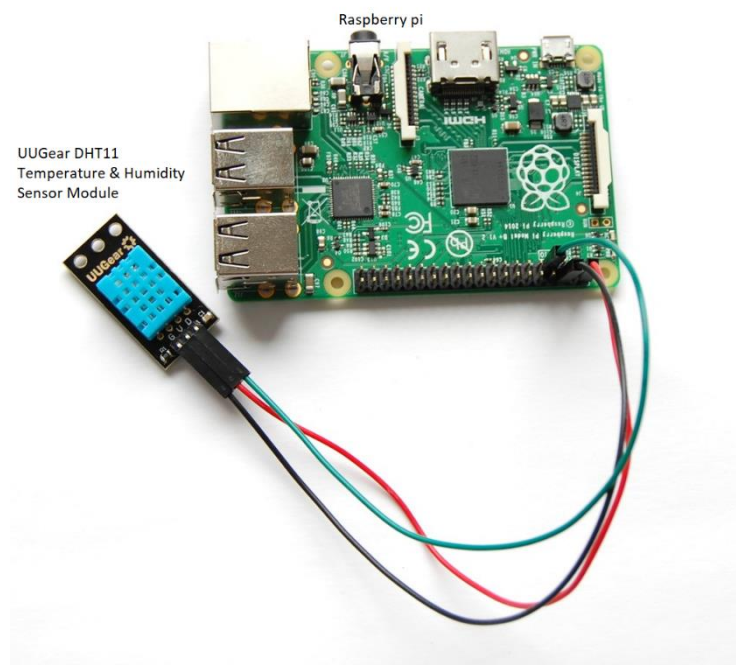
### Premissas

Baseando-se no atual conhecimento, as suposições do projeto necessárias para seu sucesso estão listadas abaixo:

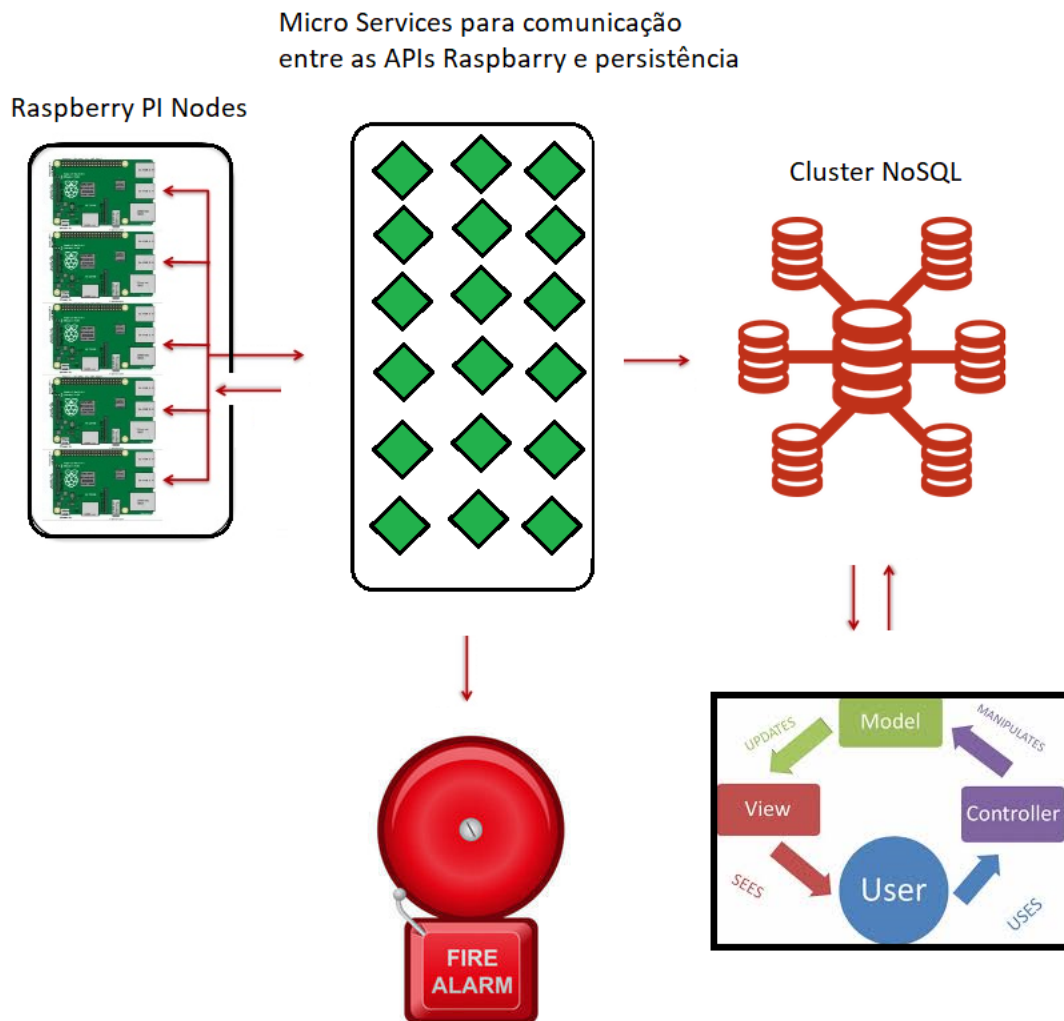
Instalação dos servidores listados nesse documento.

Instalação tem um Raspberry PI mais um sensor UUGear DHT11 para cada um dos 5.000 pontos do prédio.

### Equipamento base para os pontos de medição:



## Arquitetura empregada na solução:



**No Raspberry** PI terá um serviço instalado feito em linguagem C priorizando desempenho, os dados de medição serão atualizados a todo momento mantendo estes dados em cache na memória, este serviço funcionará como uma API que aguarda ser consultada. Com a resposta em Json no seguinte formato:

```
{ "alarme": "131",  
  "andar": "12",  
  "sala": "1290",  
  "temperatura": "22",  
  "data": "08/08/2018",  
  "hora": "09:56:37.58" }
```

**Micro Services** serão desenvolvidos em Golang que é uma linguagem que usa o paradigma de programação concorrente dessa forma podendo realizar várias tarefas e execuções distintas com multi threads, além disso golang tem um ótimo desempenho, com vários pequenos micro serviços será evitado que após um cair o serviço de verificação de temperatura seja comprometido, o ideal é que estes micro serviços sejam distribuídos em uma farm composta por máquinas em servidores distintos do ponto de vista geográfico, dessa forma criando uma alta disponibilidade.

Estes micro serviços terão uma base de dados **NoSQL** com todos os pontos pré cadastrados, na primeira execução os pontos que não têm uma data de última verificação serão consultados por ordem de cadastro "Id", após isso com micro serviços irão:

Verificar quem é o ponto que está a mais tempo sem ser consultado e dessa forma fazer uma chamada para atualizar as informações, caso o ponto não responda será gerado um alerta, caso seja constatada alta temperatura o micro serviço irá disparar uma alarme e informará os dados de andar e sala acionado a emergência.

Uma aplicação **MVC** construída em Java será disponibilizada para que um usuário possa verificar o estado de cada ponto naquele momento.