

# Mini-Projetos

## Ficha Mini-Projetos

### Assunto:

MQTT | Node-RED | ESP8266 | InfluxDB | Serviços REST | Mecanismos de Segurança

### Por favor, tome as seguintes notas em consideração:

- O projeto deve ser realizado mantendo os mesmos grupos das aulas práticas da disciplina.
- Avaliação: Cada grupo deverá explicar e demonstrar o funcionamento do projeto.
  - \* Critérios de avaliação: participação do aluno ( $c_1$ ), apresentação do projeto ( $c_2$ ), cumprimento dos objetivos ( $c_3$ ) e relatório ( $c_4$ ).
  - \*  $NotaProjeto = 0.25 \times (\sum_{n=1}^4 c_n)$
- Data de apresentação: Os projetos deverão ser apresentados durante a última aula prática de IdC na semana que compreende de 03/Jun/2024 a 07/Jun/2024.
- Recursos: Durante o desenvolvimento do mini-projeto, os alunos podem utilizar as seguintes ferramentas:
  - \* O *broker MQTT público* utilizado durante as aulas práticas da disciplina.
  - \* Nos projetos os alunos devem utilizar os conceitos e ferramentas que foram desenvolvidas durante as aulas práticas e teóricas de IdC: protocolo MQTT e broker publico, Node-RED, placa de desenvolvimento ESP8266, base de dados InfluxDB, serviços REST e mecanismos de segurança.
  - \* Adicionalmente, os alunos podem utilizar um simulador de ESP online disponível em <https://wokwi.com/>. Esta plataforma consegue simular um ESP, capaz de enviar e receber dados através de MQTT além de possuir diversos sensores e leds para conectar a placa de desenvolvimento.
  - \* Base de dados InfluxDB e criptografia de dados.

## Entregáveis

- Cada aluno deverá submeter os entregáveis (ficheiro PDF) em **Atividades** no IPB.Virtual. O limite de submissão dos entregáveis será antes da aula de apresentação do trabalho.
  - Relatório:
    - \* O nome do ficheiro deve ser PRel\_nome\_ID\_TurnoX.pdf (=PRel\_<nomeAluno>\_<IDAluno>\_<TurnoX>.pdf).
  - Apresentação (máximo de 10 slides):
    - \* O nome do ficheiro deve ser PAp\_nome\_ID\_TurnoX.pdf (=PAp\_<nomeAluno>\_<IDAluno>\_<TurnoX>.pdf).

## 1. Internet of Medical Things

- (a) **Descrição:** A pandemia de COVID-19 tem acelerado a necessidade da inclusão de tecnologias emergentes no setor de saúde, principalmente tecnologias relacionadas a Internet das Coisas (IdC), visando promover a conectividade e troca de informações em tempo real entre pacientes, profissionais da saúde e hospitais.
- (b) **Objetivo:** Desenvolver um sistema que monitore pacientes de forma remota utilizando tecnologias IdC.
- (c) **Contexto de aplicação:** Considere um cenário onde pacientes devem ser constantemente monitorados e para isso há um dispositivo que faz medições dos sinais vitais do paciente (ex: temperatura, pressão arterial, ECG, oximetria, etc.). O dispositivo envia os dados do paciente para duas aplicações distintas:
- Aplicação do paciente: mostra os dados do paciente em tempo real e notifica quando algum parâmetro estiver fora de uma escala aceitável.
  - Aplicação do hospital: permite visualizar os dados de sinais vitais dos pacientes e outros dados como o número de pacientes monitorados, sexo, faixa etária, tipo de doença, estado do paciente, etc. A aplicação também deve ser capaz de emitir alertas quando algum parâmetro de um dos pacientes não estiver em um nível aceitável e fazer o cadastro de novos pacientes.
- (d) **Considerações:**
- Pode-se utilizar um sensor de temperatura e um ESP8266, por exemplo, para simular a temperatura corporal de um paciente.
  - Para os demais sensores/dispositivos que monitoram os sinais vitais do paciente, o aluno deve simular essas medições utilizando o Node-RED.
  - A aplicação do hospital deve mostrar no mínimo 4 pacientes sendo monitorados em tempo real.
  - Armazene os dados dos pacientes no InfluxDB.
  - Inclua mecanismos de segurança (criptografia) sobre os dados dos pacientes.
- (e) **Recursos:**
- O aluno deve utilizar os conceitos e ferramentas que foram apresentadas durante as aulas práticas e teóricas de IdC: protocolo MQTT, ESP8266 e sensores, Node-RED, base de dados (InfluxDB), serviços REST e mecanismos de segurança.
  - Outros recursos/materiais ficam a critério do aluno.

## 2. IoT-based Smart Parking

- (a) **Descrição:** Problemas de estacionamento não são incomuns, especialmente nas grandes cidades. O aumento da demanda por vagas de estacionamento decorre predominantemente devido a crescente urbanização e o aumento do número de veículos em circulação. Nesse sentido, os smart parkings permitem que as vagas sejam monitoradas remotamente e automaticamente. Dentre as diversas metodologias para implementar um smart parking, aquelas que utilizam tecnologias IdC oferecem soluções que facilmente podem ser implementadas, oferecendo aos motoristas meios de rapidamente encontrar uma vaga livre. Além disso, esses sistemas também podem ser utilizados para aumentar a eficiência das instalações, determinar tendências e padrões em relação ao fluxo de motoristas.
- (b) **Objetivo:** Desenvolver um sistema para monitoramento de vagas de estacionamento utilizando tecnologias IdC.
- (c) **Contexto de aplicação:** Considere que um sistema de smart parking será implementado na cidade de Bragança em 2 setores/regiões. Cada setor/região deve ter um número 5 de vagas. Cada vaga é composta por um sensor de presença que verifica se a vaga está livre ou não. Existem duas aplicações:
- Aplicação do motorista: consulta as vagas que estão disponíveis (conforme o setor especificado) e retorna o(s) número(s) da(s) vaga(s) livre(s). Para entrar no estacionamento, o motorista deve pagar uma taxa em euros através da aplicação. Por exemplo, antes de entrar o motorista aperta um botão na aplicação que dispara um evento que faz o pagamento e envia os dados do motorista (nome, modelo do veículo, sexo, idade, valor pago, setor, vaga, etc.) para a aplicação do estacionamento.
  - Aplicação do estacionamento: armazena os dados dos utilizadores e permite visualizar o número de utilizadores por dia/semana/mês, setor mais frequentado, etc. Além disso, a aplicação faz recomendações (ex: o setor está sempre lotado, melhore a infraestrutura), mostra o perfil dos motoristas (idade, sexo, veículo, etc.) que utilizam o smart parking e o total do valor arrecadado.
- (d) **Considerações:**
- Para apenas 1 vaga utilize um sensor de presença e o ESP8266. Os demais sensores devem ser implementados virtualmente no Node-RED (utilize objetos *flow* para guardar o estado dos sensores no Node-RED. Ex: 1 - ocupado; 0 - vaga livre). Ao entrar na vaga, o estado do sensor (simulado) deve ser actualizado de 0 para 1. E ao sair, de 1 para 0.
  - Os dados dos motoristas devem ficar armazenados no InfluxDB.
  - Considere que os clientes tenham saldo inacabável (obs: note que isso é para facilitar o desenvolvimento, não sendo compatível com uma aplicação real).
  - Inclua mecanismos de segurança (criptografia) sobre os dados dos motoristas.
- (e) **Recursos:**
- O aluno deve utilizar os conceitos e ferramentas que foram apresentadas durante as aulas práticas e teóricas de IdC: protocolo MQTT, ESP8266 e sensores, Node-RED, base de dados (InfluxDB), serviços REST e mecanismos de segurança.
  - Outros recursos/materiais ficam a critério do aluno.

### 3. Farm2Fork Strategy

- (a) **Descrição:** Alinhada com a estratégia “*Farm to Fork*” no âmbito do acordo “*European Green Deal*” que visa criar um sistema alimentar mais robusto, seguro e sustentável, a inclusão de tecnologias IdC assume um papel crucial para promover o monitoramento e troca de informações em tempo real entre todo o processo que vai desde a produção primária até a comercialização e consumo dos alimentos.
- (b) **Objetivo:** Desenvolver um sistema que contribua para a estratégia “*Farm to Fork*” utilizando tecnologias IdC.
- (c) **Contexto de aplicação:** Considere um cenário que envolve desde a produção primária, transporte, comercialização/consumo do alimento.
- No nível da produção primária, produtores/fazendeiros necessitam de mecanismos (ex: sensores) para monitorar parâmetros como temperatura, humidade e pH do solo, assim como uma aplicação para armazenar, visualizar esses dados e se comunicar com atuadores, por exemplo, para irrigar o solo. Quando o alimento já estiver pronto para ser transportado, deve ser feito um cadastro e armazenamento das suas informações, indicando a data de validade de consumo, origem, nome e quantidade.
  - No nível do transporte dos alimentos, é necessário verificar se parâmetros como temperatura e humidade dentro do veículo de transporte estão em níveis aceitáveis para transportar os alimentos e emitir um aviso caso não esteja.
  - No nível de comercialização/consumo do alimento (ex: restaurante), é necessário ter uma aplicação que mostre o rastreio do transporte, garantindo que produto fez o percurso necessário da fazenda ao restaurante. Além disso, a aplicação deve ter acesso as informações dos alimentos para emitir alertas quando um alimento estiver próximo da data de validade.
- (d) **Considerações:**
- Considere a produção de um único tipo de alimento (ex: batatas).
  - No nível do transporte dos alimentos, pode-se utilizar um ESP8266 e sensor de temperatura e humidade.
  - Os demais sensores/atuadores devem ser implementados virtualmente utilizando o Node-RED.
  - Guardar os dados no InfluxDB.
  - Simular o rastreio do transporte através da aplicação móvel OwnTracks (disponível para iOS e Android), que publica tópicos MQTT sobre informações GPS do telemóvel.
  - Use o node “worldmap” (Node-RED) para monitorar o rastreio (<https://tinyurl.com/yf4p46c3>).
- (e) **Recursos:**
- O aluno deve utilizar os conceitos e ferramentas que foram apresentadas durante as aulas práticas e teóricas de IdC: protocolo MQTT, ESP8266 e sensores, Node-RED, base de dados (InfluxDB), serviços REST e mecanismos de segurança.
  - Outros recursos/materiais ficam a critério do aluno.

#### 4. Domótica

- (a) **Descrição:** A “Domótica” permite que as funções de equipamentos eléctricos e electrónicos em uma edificação se possam controlar e automatizar, local ou remotamente, através de um sistema integrado.
- (b) **Objetivo:** Desenvolver um controlo de iluminação local, da temperatura e da qualidade do ar do ambiente.
- (c) **Contexto de aplicação:** Considere uma edificação com vários dispositivos IdC que podem atuar (por ex.: na iluminação) e monitorizar dados atmosféricos como temperatura, humidade, CO<sub>2</sub>, etc..
- (d) **Considerações:**
- Considere a informação do nascer/pôr do sol;
  - Considere as informações atmosféricas de temperatura e de humidade, a informação da medida da iluminação e da concentração de CO<sub>2</sub> obtida dos sensores;
  - Considere um atuador simulado onde **não** houver sensor para o controle individual:
    1. da cortina nas posições: alta, média e baixa;
    2. da janela nas posições: semi-aberta, aberta e fechada;
    3. do condicionador de ar nos modos: ligado (aquecendo, arrefecendo, ventilando) e desligado;
  - Considere os serviços externos “gratuitos” listados na página web do:
    1. OpenWeather (<https://openweathermap.org/api>);
    2. Instituto Português do Mar e da Atmosfera – IPMA (<https://api.ipma.pt>).
  - Devem ser indicadas em uma *dashboard*:
    1. a duração do dia (histórico dos últimos 7 dias) e a hora do nascer/pôr do sol do dia corrente;
    2. a qualidade do ar interna (conforme a concentração de CO<sub>2</sub>);
    3. as temperaturas e humidades (interna e externa);
    4. as posições da cortina e da janela;
    5. o modo do condicionador de ar.
- (e) **Recursos:** Broker: broker.emqx.io ou broker.hivemq.com | Porta TCP: 1883
- Use um ESP8266 para conectar os sensores:
    1. LDR para a medida da iluminação;
    2. LED para as lâmpadas;
    3. DHT11 para a temperatura;
    4. servo motor para a janela.
  - Tópico (**IPB/IoT/Lab/AirQuality**) – sensores para a medida interna:
    1. da quantidade de luz (em lux);
    2. da concentração de CO<sub>2</sub> (em ppm), de temperatura (em °C) e da humidade do ar (em %).
  - Tópico (**IPB/IoT/Lab/ExternalNode**) – sensores para a medida externa:
    1. de temperatura (em °C) e da humidade do ar (em %).
  - Outros recursos/materiais ficam a critério do aluno e podem ser simulados.

## 5. Controlo Sanitário

- (a) **Descrição:** A pandemia de COVID-19 tem acelerado a necessidade da inclusão de tecnologias emergentes no setor de saúde, principalmente tecnologias relacionadas a Internet das Coisas (IdC).
- (b) **Objetivo:** Desenvolver o controlo da lotação e da qualidade do ar do auditório Eng<sup>o</sup> Alcínio Miguel.
- (c) **Contexto de aplicação:**
1. Considere uma edificação com vários dispositivos IdC que detectam o movimento e dados atmosféricos como temperatura, humidade, etc..
  2. Considere a informação de um par de sensores ultrassônicos de movimento (“detetores de presença” **A** e **B**) instalados no único acesso disponível (entrada/saída):
    - Obs.: Estes devem ser montados numa placa de ensaios, usando um ESP8266, e afastados entre si de tal forma que possam indicar o sentido do movimento (por ex.: de **A** → **B** ou **A** ← **B**).
- (d) **Considerações:**
- Considere as informações atmosféricas do tópico IPB/IoT/Lab/AirQuality;
  - Considere um atuador simulado onde não houver sensor para controle individual:
    1. da cortina nas posições: alta, média e baixa;
    2. da janela nas posições: semi-aberta, aberta e fechada;
    3. do condicionador de ar nos modos: ligado (aquecendo, arrefecendo, ventilando) e desligado;
  - Considere os serviços externos “gratuitos” listados na página web do:
    1. OpenWeather (<https://openweathermap.org/api>);
    2. Instituto Português do Mar e da Atmosfera – IPMA (<https://api.ipma.pt>).
  - Devem ser indicados em uma *dashboard*:
    1. a qualidade do ar interna (conforme a concentração de CO<sub>2</sub>) com histórico dos últimos 7 dias;
    2. as temperaturas e humidades (interna e externa);
    3. as posições da cortina e da janela;
    4. o modo do condicionador de ar;
    5. a lotação instantânea e a máxima (para facilitar, use um valor baixo).
      - Caso a lotação instantânea e a máxima sejam **iguais** deve ser emitido um alerta sonoro.
- (e) **Recursos:** Broker: broker.emqx.io ou broker.hivemq.com | Porta TCP: 1883
- Use um ESP8266 para conectar os sensores:
    1. LDR para a medida da iluminação;
    2. LED para as lâmpadas;
    3. DHT11 para a temperatura;

- 4. servo motor para a janela.
- Tópico (**IPB/IoT/Lab/AirQuality**) – sensores para a medida interna:
  - 1. da qualidade do ar – índice IAQ (em escala 0-500 (pior));
  - 2. da concentração de CO<sub>2</sub> (em ppm), qualidade do ar – IAQ (em escala 0-500 (pior)) de temperatura (em °C) e da humidade do ar (em %).
- Tópico (**IPB/IoT/Lab/ExternalNode**) – sensores para a medida externa:
  - 1. de temperatura (em °C) e da humidade do ar (em %).
- Outros recursos/materiais ficam a critério do aluno e podem ser simulados.

## 6. Controlo da Rega

- (a) **Descrição:** A necessidade de incremento da produção agrícola tem acelerado a inclusão de tecnologias emergentes para a monitorização da saúde das plantas, principalmente tecnologias relacionadas a Internet das Coisas (IdC).
- (b) **Objetivo:** Monitorar dados de temperatura, humidade do ar e humidade do solo de uma planta, e quando for necessário, realizar a rega automática desta planta.
- (c) **Contexto de aplicação:** Considere uma planta, onde será monitorado temperatura, humidade do ar e humidade do solo. Utilizando o Node-RED como dashboard, uma placa de desenvolvimento ESP8266 e sensores de temperatura, humidade e motobomba, deve ser desenvolvido um sistema de rega automática para uma planta.
- (d) **Considerações:** O Sistema desenvolvido precisa ter as seguintes especificações (os alunos podem adicionar outras características se julgar necessário):
  - Um ESP8266 com sensores de temperatura, humidade do ar, humidade do solo, e uma motobomba para regar a planta.
  - O dashboard precisa conter um gráfico com informações de temperatura, humidade do ar e humidade do solo das ultimas 24 horas referente a planta.
  - O dashboard deve conter a temperatura, humidade do ar e humidade do solo atual.
  - Uma garrafa de água, conectada a uma mangueira que estará conectada a uma motobomba. Este sistema também será ligado ou desligado pelo ESP8266 de forma automático. Portanto, a planta será regada automaticamente (ligar a motobomba quando for necessário, e desligar quando a planta estiver regada).
  - Através de análise de dados, identificar quando a garrafa de água estiver vazia e enviar um alerta sonoro para o usuário.
  - O dashboard deve conter um slide para ligar ou desligar de forma manual (adicionalmente ao automático) o sistema de rega.
  - Os dados devem ser armazenados numa base de dados.
  - Os dados devem ser criptografados.
- (e) **Recursos:**
  - Node-RED, InfluxDB, ESP8266, Sensor de temperatura e humidade, sensor de humidade do solo e motobomba.
  - Outros recursos/materiais ficam a critério do aluno.



## 7. Solicitação Automática de Transporte de Pallets por Empilhadeiras.

(a) **Descrição:** Desenvolver um sistema IoT, onde existem 4 clientes (Máquina 1, Máquina 2, Máquina 3 e Máquina 4) e 1 Empilhadeira.

(b) **Objetivo:** Cada cliente (operador de máquina) irá requisitar através de um dashboard o transporte de produtos por uma empilhadeira que será realizado da Máquina X para um armazém.

O motorista da empilhadeira terá uma dashboard indicando a fila FIFO de solicitações de máquinas (por ordem de pedido). Também terá um botão para remover o pedido da fila após finalizar a operação.

(c) **Considerações:**

Na interface dos operadores de máquina:

- Apresentar o tempo médio e desvio padrão (desde a solicitação até o fim do transporte) de transporte para cada máquina;
- Cada máquina deve possuir um bucket na base de dados e armazenar os dados relevantes da aplicação.

Na interface do motorista da empilhadeira:

- Apresentar o tempo médio de todos os transportes;
- Apresentar o menor tempo de transporte e para qual máquina foi realizado este menor tempo de transporte;
- Apresentar o maior tempo de transporte e para qual máquina foi realizado este maior tempo de transporte;
- Uma mensagem de áudio informando que um novo pedido de transporte da máquina x chegou.
- Armazenar os dados no InfluxDB em um bucket.

Obs: Toda a troca de informações entre o operador da empilhadeira e das máquinas deve ser criptografada.

(d) **Recursos:**

- O aluno deve utilizar os conceitos e ferramentas necessários que foram apresentadas durante as aulas práticas e teóricas de IdC: protocolo MQTT e broker público, Node-RED, base de dados InfluxDB e mecanismos de segurança.
- Outros recursos/materiais ficam a critério do aluno.

## 8. Audio Chat

- (a) **Descrição:** Enviar mensagens de áudio de no máximo 10 segundos de um cliente A para outro cliente B e do Cliente B para o Cliente A (voice chat);
- (b) **Objetivo:** Criar um voice chat entre dois clientes com o dashboard do Node-RED. Na interface de cada cliente, deverá ser possível escutar 10 últimos áudios trocados entre os clientes; Todas as mensagens devem ser enviadas depois de serem criptografadas. Adicionalmente, as mensagens enviadas pelos clientes, devem ser convertidas para texto e enviadas por telegram, ou seja, cada cliente terá o seu telegram para registrar todas as mensagens.
- (c) **Considerações:**
  - Instalar a biblioteca “node-red-ui-microphone”;
  - Verificar permissões de acesso no navegador para conseguir gravar o audio do microfone.
  - Deve ser possível acessar os 10 ultimos audios. Por exemplo, se for enviado 11 o primeiro audio enviado pode ser descartado.
  - Para realizar a comunicação entre node-red e telegram utilizar a biblioteca “node-red-contrib-telegrambot”, a qual os alunos deverão estudar a documentação para utilizar.
- (d) **Recursos:**
  - O aluno deve utilizar os conceitos e ferramentas necessários que foram apresentadas durante as aulas práticas e teóricas de IdC: protocolo MQTT e broker, Node-RED, mecanismos de segurança.
  - Outros recursos/materiais ficam a critério do aluno.

## 9. Álbum Fotográfico Digital

(a) **Descrição:** Enviar imagens obtidas da webcam de um cliente A para outro cliente B;

(b) **Objetivo:** Criar um repositório ou álbum de foto digital.

(c) **Considerações:**

O sistema conta com 2 clientes. O cliente A deverá ter a opção de tirar fotos utilizando a webcam, se ele gostar da foto, deve inserir um texto descritivo e enviar a foto para um cliente B. Caso não goste da foto, deve descartá-la.

Na interface do Cliente B, deverá ser listado em ordem cronológica as fotos e suas descrições. O cliente B fornece um aviso de modo automático ao cliente A quando o limite de 10 fotos estiver esgotando para ele selecionar o ID de qual foto ele pretende remover e então remover esta foto.

Quando uma nova foto for enviada, o usuário deve ser informado através do telegram que uma nova foto foi recebida e qual é o texto descritivo.

(d) **Recursos:**

- Para acessar a webcam através do Node-RED deverá utilizar o pacote: "node-red-node-ui-webcam"
- O aluno deve utilizar os conceitos e ferramentas necessários que foram apresentadas durante as aulas práticas e teóricas de IdC: protocolo MQTT e broker, Node-RED, InfluxDB e mecanismos de segurança.
- Outros recursos/materiais ficam a critério do aluno.

## 10. Sistema de Ponto Digital

(a) **Descrição:** Monitorar a assiduidade de um trabalhador (sistema ponto) de forma automática utilizando dados do telemóvel do utilizador. Portanto, deverá ser verificada a hora de chegada e a hora de saída do local de trabalho e com isso explorar informações sobre o tempo de trabalho. As informações do GPS podem ser obtidas através do telemóvel do utilizador instalando a aplicação Owntracks.

(b) **Objetivo:** Através de uma aplicação utilizando o GPS, conseguir identificar o padrão de trabalho de um funcionário.

(c) **Considerações:** Primeiramente, explorar a documentação do Owntracks para dominar as funcionalidades da aplicação para o desenvolvimento do projeto.

Então, com a aplicação Owntracks a enviar dados por MQTT, deve subscrever por Node-RED e verificar o horário de chegada e saída do local de trabalho de um funcionário ou estudante. Os horários devem ser armazenados numa base de dados para análise de dados. Com isto, deve ser verificado o tempo de trabalho dos últimos sete dias e apresentar de forma amigável no dashboard. Calcular a média de tempo de trabalho dos últimos 7 dias. Enviar e-mails de alerta conforme a rotina de trabalho ("Trabalhou o tempo esperado", "Trabalhou menos que o tempo esperado", "Trabalhou mais que o tempo esperado").

Utilizar o "node-red-contrib-web-worldmap" para disponibilizar os dados do GPS no mapa.

(d) **Recursos:**

- Utilizar a aplicação móvel owntracks, que publica informações do GPS do telemóvel;
- Na aplicação Owntrack, configurar o modo para MQTT o host para o broker publico ou para um broker criado pelos estudantes, considerando que a segunda opção é a mais segura.
- Mais informações verificar a documentação em "<https://owntracks.org/booklet/>"
- O aluno deve utilizar os conceitos e ferramentas necessários que foram apresentadas durante as aulas práticas e teóricas de IdC: protocolo MQTT e broker, Node-RED, base de dados InfluxDB e mecanismos de segurança.
- Outros recursos/materiais ficam a critério do aluno.

## 11. Sistema de Detecção de Incêndios com Alerta em Tempo Real

- (a) **Descrição:** Sistema de alarme de incêndio que detecta fumaça, temperatura elevada e envia alertas imediatos. Dashboard para monitoramento e histórico de eventos.
- (b) **Objetivo:** Desenvolver um sistema de detecção de incêndios baseado em dados coletados de temperatura, humidade e gás (fumaça) e uma dashboard para monitorar esses parâmetros coletados, indicando o local de detecção do incêndio (através do GPS).
- (c) **Contexto de aplicação:** Considere um parque natural com vários dispositivos IdC instalados em diferentes pontos, que podem monitorar dados atmosféricos como temperatura, humidade, CO<sub>2</sub>, etc..
- (d) **Considerações:**
  - Considere um dispositivo: ESP32 + sensor de fumaça (MQ-2/MQ-135) + sensor de temperatura/humidade + GPS;
  - Considere a aplicação Owntracks no caso de não haver sensor GPS;
  - Considere os seguintes atuadores:
    1. Buzzer e LED vermelho para simular sirene e luzes de evacuação;
  - Devem ser indicadas em uma *dashboard*:
    1. O registo de alarmes e tempos de resposta (início e fim, intervalo, etc.);
    2. As coordenadas do incêndio (latitude e longitude);
    3. as temperaturas e humidades (em diferentes pontos/dispositivos);
- (e) **Recursos:** Broker: broker.emqx.io ou broker.hivemq.com | Porta TCP: 1883
  - Use um ESP32 para conectar os sensores:
    1. LED para as lâmpadas;
    2. DHT11/DTH22 para a temperatura e humidade;
    3. MQ-2 para detecção de fumaça (gases).
  - Aplicação Owntracks para dados de latitude e longitude.
  - Outros recursos/materiais ficam a critério do aluno e podem ser simulados.

## 12. Monitoramento de Plantas Domésticas com Feedback em Tempo Real

- (a) **Descrição:** Sistema doméstico que monitora a saúde de plantas com sensores de solo, luminosidade e temperatura. O ESP32 envia os dados, e um LED RGB indica o “estado de saúde” da planta.
- (b) **Objetivo:** Desenvolver um sistema doméstico para monitorar a saúde de plantas que seja capaz de coletar informações ambientais e de acordo com as informações coletadas, indicar o estado de saúde da planta (saudável, alerta) e ser capaz de atuar no ambiente (irrigação).
- (c) **Contexto de aplicação:** Considere uma planta doméstica em que será colocado um dispositivo IdC para verificar sua saúde.
- (d) **Considerações:**
  - Considere um dispositivo: ESP32 + sensor de humidade do solo + LDR (luminosidade) + temperatura;
  - Considere os seguintes atuadores:
    1. Válvula para acionar irrigação;
  - Considere uma integração com o Telegram para enviar lembretes para realizar a rega da planta;
  - Devem ser indicadas em uma *dashboard*:
    1. O registo de cuidados da planta (watering logs), indicando início, fim, intervalo/duração, etc.);
    2. As condições ambientais (temperatura, humidade, etc.);
    3. A condição de saúde da planta (saudável, em alerta..).
- (e) **Recursos:** Broker: broker.emqx.io ou broker.hivemq.com | Porta TCP: 1883
  - Use um ESP32 para conectar os sensores:
    1. LED para as lâmpadas;
    2. DHT11/DTH22 para a temperatura e humidade;
    3. LDR para luminosidade.
  - Outros recursos/materiais ficam a critério do aluno e podem ser simulados.

### 13. Sistema de Alerta de Enchente

- (a) **Descrição:** Estação de monitoramento de enchentes em rios. Mede o nível da água e envia alertas em tempo real.
- (b) **Objetivo:** Desenvolver um sistema de monitoramento de enchentes em rios/valas capaz de prever enchentes e ligar o sistema de evacuação do local.
- (c) **Considerações:**
  - Considere um dispositivo: ESP32 + sensor de chuva + ultrassônico (nível da água);
  - Considere a API do IPMA para obter probabilidade de precipitação dos próximos dias;
  - Considere os seguintes atuadores:
    1. Buzzer e LED vermelho para simular sirene e luzes de evacuação;
  - Considere uma integração com o Telegram para enviar alertas de enchente;
  - Devem ser indicadas em uma *dashboard*:
    1. O registo de enchentes (indicando início, fim, intervalo/duração, etc.);
    2. As condições em tempo real (nível da água, probabilidade de precipitação, chuva, etc.);
    3. A condição do rio (abaixo do nível, normal, em alerta..).
- (d) **Recursos:** Broker: broker.emqx.io ou broker.hivemq.com | Porta TCP: 1883
  - Use um ESP32 para conectar os sensores:
    1. LED para as lâmpadas;
    2. Sensor de chuva;
    3. Sensor ultrassônico para nível da água.
  - Outros recursos/materiais ficam a critério do aluno e podem ser simulados.