

# Checkpoint 1

## Grupo 3:

Mateus Zanetti Camargo Penteado - 11219202

Érika Hortência Pereira Cardoso - 10696830

## O projeto

O projeto se trata de um sistema que utiliza um sensor para detectar movimento e enviar para o usuário todas as ocorrências, juntamente com um registro em imagem. Cada uma dessas ocorrências ficará registrada no banco de dados e poderá ser consultada posteriormente. A seguir detalhamos cada uma das ferramentas escolhidas.

## Aplicação

A aplicação é responsável por realizar a comunicação com a camada de micro-serviço e é a camada que o usuário final irá ver e utilizar de fato, portanto é necessário que tenha uma interface de fácil utilização.

A definição de quais tecnologias utilizar depende de para qual plataforma desejamos desenvolver, as principais plataformas relacionadas com suas principais tecnologias podem ser observadas abaixo:

Plataforma	Principais Tecnologias
IOS	Swift, Flutter, React Native
Android	Java, Kotlin, Flutter, React Native
Desktop	Java, C#
Interface WEB	HTML, CSS, JS

Dado o contexto de nossa aplicação e pela facilidade de implementação e acesso, optamos por desenvolver uma interface WEB para ser nossa aplicação, deste modo utilizaremos HTML, CSS e Javascript, para desenvolver tal tipo de sistema normalmente são utilizadas frameworks, como React, Angular, Vue, entre outras. Para nossa aplicação iremos utilizar o framework React, o qual oferece velocidade, simplicidade e escalabilidade à produção de aplicações.

Algumas vantagens e desvantagens em desenvolver uma interface WEB e de utilizar o React como framework estão listadas abaixo:

### Interface WEB:

- **Vantagens:**
  - Fácil desenvolvimento
  - Acesso ao sistema de qualquer lugar com internet
  - Facilidade de integrações com os outros sistemas
  - Flexibilidade
  - Não há a necessidade de download de nada por parte do usuário
- **Desvantagens:**
  - Necessidade de conexão à internet para utilização do sistema
  - Preocupação com a responsividade do sistema a diferentes telas
  - Necessário subir o sistema em algum servidor

### React:

- **Vantagens:**
  - Flexível, padrão definido pelo desenvolvedor
  - Permite utilizar Javascript, Typescript, ES6, entre outros
  - Permite escrever arquivos HTML e CSS dentro de um arquivo Javascript
  - Fácil migração entre versões
  - Muito popular
  - Reaproveitamento de código facilitado
- **Desvantagens:**
  - Documentação oficial enxuta
  - Flexibilidade de não possuir um padrão pode ocasionar códigos sem padrão algum ou com padrão muito bagunçado

## Broker

O broker é o elemento responsável por gerir as publicações e as subscrições do protocolo MQTT - Message Queuing Telemetry Transport, protocolo de mensagens entre máquinas. Ele é um mediador entre as máquinas, atua como um servidor intermediário da informação (recebe os dados enviados pelos sensores, faz seu tratamento envia para os clientes subscritos àquele tópico).

Atualmente, existem quatro variações de brokers, relacionadas abaixo:

Tipo de Broker MQTT	Exemplo
Open source	HiveMQ, Mosquitto
Comercial	HiveMQ Professional, EMQ, VerneMQ
Cloud	HiveMQ Cloud, CloudMQTT, AWS IoT Core, Azure IoT Hub
Broker de propósito geral com suporte MQTT	Solace PubSub+, IBM MQ, RabbitMQ, ActiveMQ

Ao se escolher um broker é preciso analisar os seguintes requisitos:

- Escalabilidade
- Segurança
- Resiliência
- Agilidade

- Observabilidade
- Disponibilidade

Para o projeto em questão, optou-se por escolher uma das opções de brokers open-source existentes. Abaixo relacionamos os mais utilizados:

Broker	Resumo
Mosquitto	Um dos mais populares, leve e escalável. Desenvolvido em C, suporta MQTT3.x e MQTT5
EMQ X	Altamente escalável, alta disponibilidade
Cassadana	Solução Java que surgiu como fork do Moquette
Ejaberd	Broker capaz de lidar facilmente com milhões de conexões simultâneas
HiveMQ	Broker desenvolvido em Java, funciona com MQTT 3.x e MQTT 5

Considerando o escopo do projeto, foi feita a opção por utilizar a alternativa mais popular, devido à sua facilidade de utilização, bem como disponibilidade de material complementar que torna mais acessível a implementação. Dessa forma, o broker escolhido foi o Mosquitto. O Mosquitto apresenta as seguintes vantagens e desvantagens:

- Vantagens:
  - Código aberto, leve e simples, que pode ser utilizado por todo tipo de máquina, até as mais antigas
  - Extremamente seguro e protegido de falhas
  - Fácil de ser instalado
- Desvantagens:
  - Não tem suporte para clustering
  - Não tem suporte para uso de CPU multi-thread

## Armazenamento

A camada de armazenamento tem a função de salvar os dados tratados pelo micro-serviço, sendo um fator fundamental na implementação da aplicação. O sistema de gerenciamento de banco de dados deve ser capaz de superar os seguintes desafios:

- Volume e variedade de dados: sistemas IoT coletam muitos tipos de dados que precisam ser gerenciados
- Escalabilidade: pode haver milhares de dispositivos em um determinado ambiente, cada um executando uma instância do banco de dados
- Alto desempenho: aplicativos IoT geralmente possuem um elemento de tempo real, ou quase-real que requerem um software de banco de dados de alto desempenho
- Dados de séries temporais: os dados de sensores são captados em locais e intervalos de tempo específicos. Funcionalidades de tempo e espaço são necessárias para armazenar, manipular e acessar esses dados de forma rápida e econômica

Para o projeto em questão foram considerados os seguintes bancos de dados:

SGBD	Tipo	Licença	Escalabilidade	Curva de Aprendizado
MySQL	SQL	GNU	Vertical	Média
PostgreSQL	objeto relacional	Open-source	Vertical	Alta
MongoDB	NoSQL	SSPL	Horizontal	Média

Para o projeto em questão, optou-se por um SGBD de uso livre, com curva de aprendizagem mais amena, dessa forma o escolhido foi o MySQL. O mesmo apresenta as seguintes vantagens e desvantagens:

- Vantagens:
  - Instalação grátis
  - Sintaxe simples e complexidade baixa
  - Compatibilidade com nuvem
- Desvantagens:
  - Problemas de escalabilidade
  - Suporte open-source limitado

## Micro-Serviço

A camada de micro-serviços é responsável por receber mensagens do Broker, realizar o tratamento adequado, solicitar a gravação de entradas no banco de dados e receber requisições da aplicação e retornar os dados necessários.

Há diversas linguagens e frameworks para o desenvolvimento do sistema em questão, algumas das mais populares são:

- **Linguagens:**
  - Python
  - C#
  - Javascript
  - PHP
  - Java
  - Ruby
- **Frameworks:**
  - Django
  - Flask
  - Ruby On Rails
  - Spring
  - ASP.NET Core
  - CakePHP

Dentre as opções listadas optou-se pelo desenvolvimento em C# utilizando o framework ASP.NET Core, pois já temos experiência com a utilização do framework em questão, além de ele ser robusto e atende todas as necessidades que precisamos para o projeto.

Algumas das vantagens e desvantagens em utilizar o framework escolhido estão listadas abaixo:

- Vantagens:
  - Código aberto
  - Multi-Plataforma
  - Sistema de rotas limpas
  - Fácil de aplicar testes
  - Fácil manutenção
  - Maioria das bibliotecas é desenvolvida pela própria Microsoft (maior consistência)
- Desvantagens:
  - Não há disponível muitos CMS como no PHP
  - ASP.NET tem um histórico de não ter muitos recursos básicos de segurança