



Faculdade de Informática e Administração Paulista

Giovanna Revito Roz - RM558981

Kaian Gustavo de Oliveira Nascimento - RM558986

Lucas Kenji Kikuchi - RM554424

DISRUPTIVE ARCHITECTURES IOT, IOB and GENERATIVE

Global Solution – Swift Safety

INTEGRANTES

RM (SOMENTE NÚMEROS)	NOME COMPLEMENTO (SEM ABREVIAR)
558981	Giovanna Revito Roz
558986	Kaian Gustavo de Oliveira Nascimento
554424	Lucas Kenji Kikuchi



SUMÁRIO

1.Descritivo.....	6
2.Hardware Utilizado	7
3. Bibliotecas	7
4.Configurações	8
5.Funcionalidades	9
6.Nós do Node-RED	14
7. Conclusão.....	20

1 – Descritivo

Em um mundo cada vez mais vulnerável às forças imprevisíveis da natureza, os alertas climáticos se tornaram uma das nossas principais ferramentas de defesa. Fenômenos repentinos, como tempestades violentas, enchentes, secas prolongadas e incêndios florestais, têm se tornado mais frequentes e intensos. Diante dessa realidade, surge uma pergunta crucial: como podemos nos preparar para o imprevisto? Como garantir que os alertas cheguem a tempo de salvar vidas e reduzir os danos? A resposta, muitas vezes, reside em algo invisível, mas presente em nosso cotidiano: a tecnologia."

O projeto Swift Safety é um sistema inovador de monitoramento ambiental, projetado para detectar condições climáticas extremas e riscos ambientais em tempo real. Utilizando uma rede de sensores de temperatura, umidade, precipitação, gases, vento e movimento (aceleração), conectados a um microcontrolador ESP32, o sistema coleta e analisa dados de maneira contínua. Essas informações são então exibidas em um display LCD e, em caso de eventos críticos, são enviadas via MQTT para um broker público. Alertas visuais e sonoros são acionados para notificar imediatamente os usuários sobre a situação."

Além disso, o projeto transforma a forma como lidamos com desastres naturais, funcionando como uma rede social de alertas climáticos. Através do aplicativo, usuários e órgãos públicos podem compartilhar informações, imagens e comentários em tempo real, criando uma rede colaborativa de monitoramento. Sensores IoT inteligentes detectam automaticamente situações de risco, como mudanças abruptas no clima ou eventos extremos, e emitem alertas instantâneos. Esses alertas são disseminados não só dentro do sistema, mas também por meio de notificações push, diretamente nos celulares dos usuários, garantindo que todos estejam informados e preparados para agir.

2 – Hardware Utilizado

Componente	Descrição	Pino/Endereço
ESP32	Microcontrolador principal	
Sensor DHT22	Temperatura e Umidade	GPIO 12
Sensor de Chuva	Sensor analógico para intensidade de chuva	GPIO 34
Sensor de Gás	Sensor analógico para concentração de gás	GPIO 35
Sensor de Vento	Sensor analógico para velocidade do vento	GPIO 32
MPU6050	Acelerômetro e giroscópio	I2C (SDA/SCL via Wire)
LCD I2C	Display LCD 16x2 com interface I2C	Endereço I2C 0x27
LEDs de Alerta	LED vermelho e amarelo para sinais visuais	GPIO 13 (vermelho), GPIO 14 (amarelo)
Buzzer	Alerta sonoro	GPIO 15

3 – Bibliotecas Utilizadas

- **WiFi.h:** Gerenciamento da conexão Wi-Fi.
- **LiquidCrystal_I2C.h:** Controle do display LCD via protocolo I2C.
- **Adafruit_Sensor.h** e **Adafruit_MPU6050.h:** Leitura dos dados do acelerômetro MPU6050.
- **WiFiClient.h** e **PubSubClient.h:** Cliente MQTT para comunicação com broker.
- **DHT.h:** Comunicação com o sensor DHT22.
- **ArduinoJson.h:** Construção e serialização do JSON para envio MQTT.
- **time.h:** Obtenção da hora via NTP para timestamp das leituras.

4 – Configurações

Rede Wi-Fi

- SSID: "Wokwi-GUEST"
- Senha: (null)

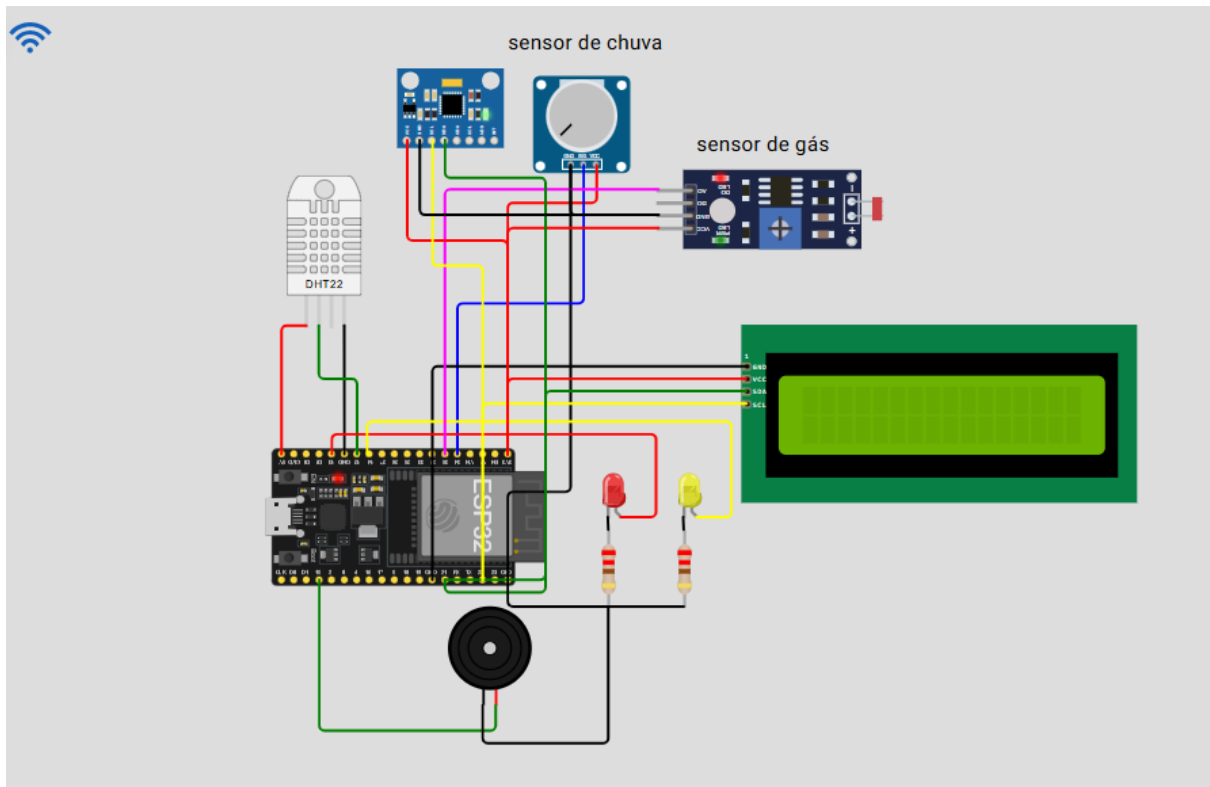
MQTT

- Broker: broker.hivemq.com
- Porta: 1883
- Tópicos:
 - fiap/eventos-extremos/sensor1
 - fiap/eventos-extremos/sensor2
 - fiap/eventos-extremos/sensor3
 - fiap/eventos-extremos/alertas
- Client ID: "esp32-sensor-extremo-001"

Limites de Alertas

Temperatura Alta	40.0 °C
Temperatura Baixa	0.0 °C
Umidade Alta	80 %
Umidade Baixa	20 %
Chuva	3000 (valor analógico)
Gás	700 (valor analógico)
Vento	10.0 m/s
Movimento (aceleração)	1.5 m/s ²

5– Funcionalidades



1. Inicialização (setup())

- Configura comunicação serial.
- Inicializa pinos dos LEDs e buzzer.
- Inicializa LCD e sensores DHT22 e MPU6050.
- Estabelece conexão Wi-Fi.
- Configura o horário via servidor NTP.
- Configura cliente MQTT.

2. Conexão Wi-Fi (setup_wifi())

- Tenta conectar à rede Wi-Fi.
- Aguarda até conexão estabelecida.
- Imprime IP local no terminal.

```
// Wi-Fi
const char* ssid = "Wokwi-GUEST";
const char* password = "";
```

```
Conectando a Wokwi-GUEST
....
WiFi conectado
10.10.0.2
Conectando ao MQTT...Conectado.
```

3. Reconexão MQTT (reconnect())

- Tenta conectar ao broker MQTT.
- Caso sucesso, inscreve ao tópico de alertas.
- Em caso de falha, tenta reconectar após 5 segundos.

4. Obtenção da Hora Atual (horaAtual())

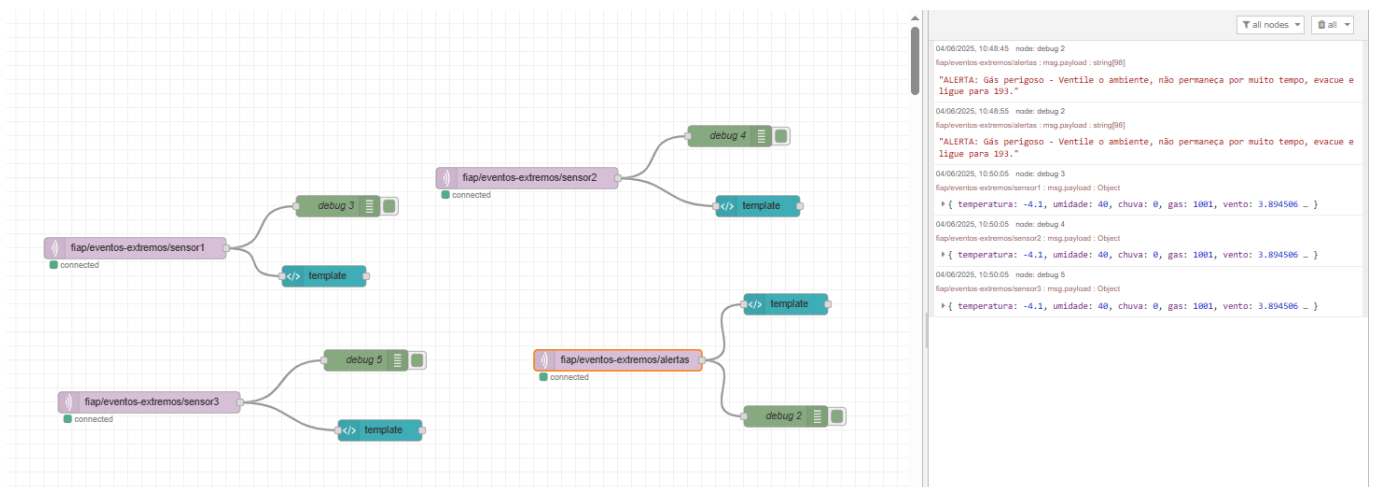
- Utiliza o sistema NTP para obter horário local formatado (HH:MM:SS).
- Retorna string com hora atual.

5. Envio de Dados via MQTT (enviarDadosMQTT())

- Gera documento JSON contendo:
 - Temperatura
 - Umidade
 - Chuva
 - Gás
 - Vento
 - Movimento
 - Alerta formatado (se houver)

- Trouxe os dados de São Paulo – Zona Leste de alguns locais específicos
- Horário da leitura com atualização a cada 10 segundos
- Publica dados nos três tópicos de sensores.
- Publica alertas no tópico de alertas (se houver).
- Imprime JSON enviado no terminal.

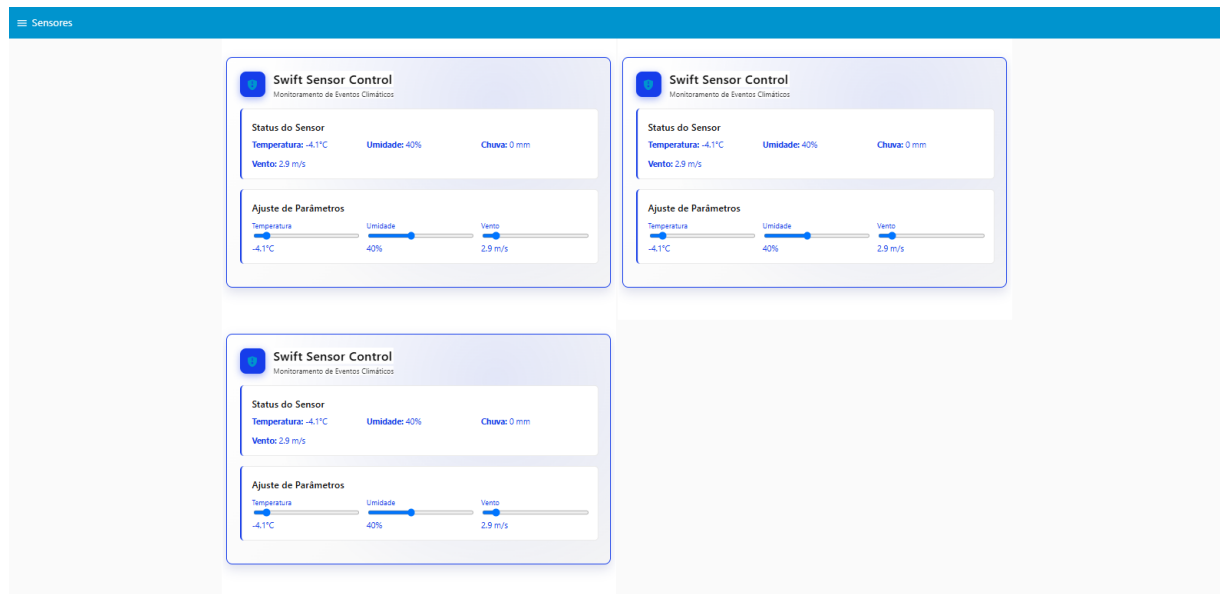
```
....
WiFi conectado
10.10.0.2
Conectando ao MQTT...Conectado.
Gás perigoso
JSON MQTT enviado:
{
  "temperatura": 24,
  "umidade": 40,
  "chuva": 0,
  "gas": 1001,
  "vento": 5.641026,
  "movimento": 9.80665,
  "alerta": "ALERTA: Gás perigoso - Ventile o ambiente, não permaneça por muito tempo, evacue e ligue para 193.",
  "localizacao": "Zona Leste de São Paulo [Tatuapé, Sapopemba, Mooca]",
  "horario": "10:48:44"
}
```



6. Verificação de Eventos Extremos (verificarEventosExtremos())

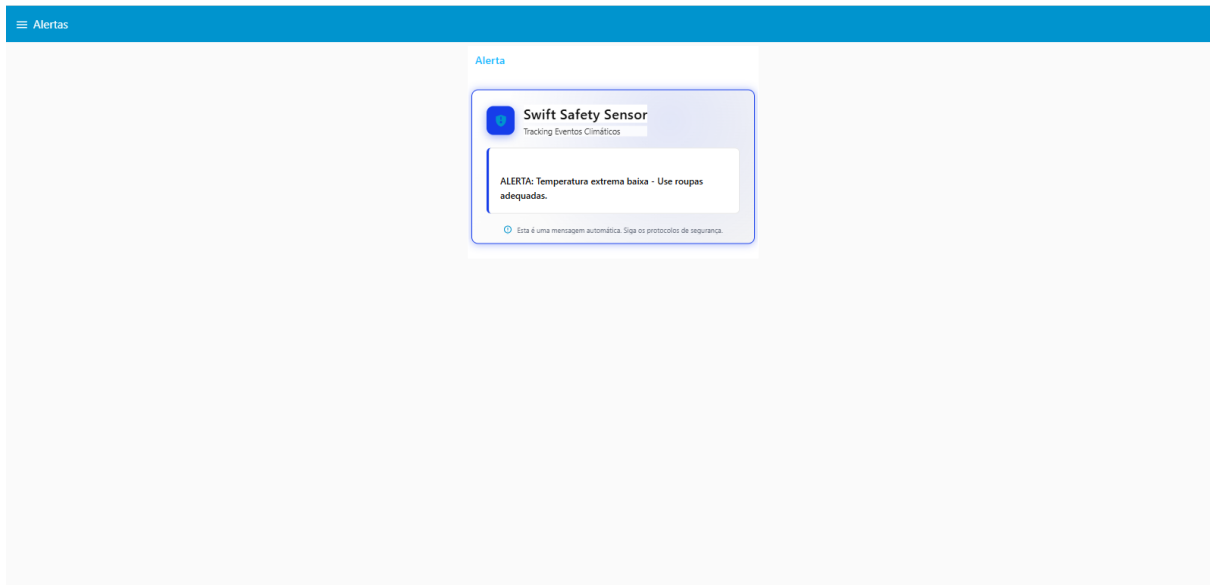
- Avalia valores dos sensores contra limites para determinar alertas.
- Detecta eventos como risco de enchente, temperatura extrema, gás perigoso, tempestade severa, furacão, tornado, seca, granizo, deslizamento, entre outros.
- Aciona LEDs e define estado de alerta ativo.

- Retorna descrição do alerta.



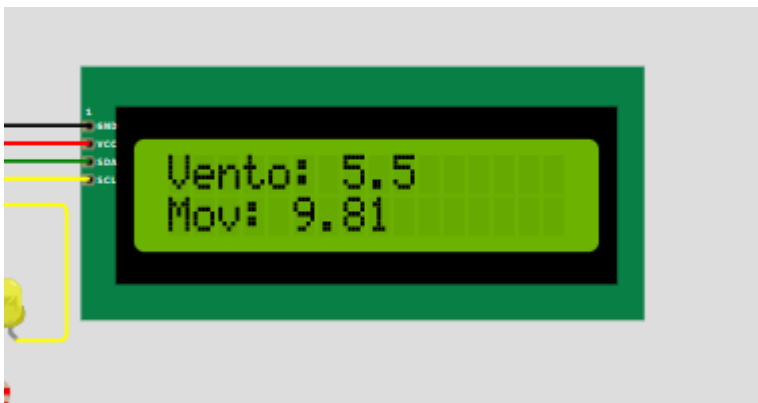
7. Mensagem de Instrução por Tipo de Alerta (mensagemInstrucao ())

- Retorna mensagem de orientação para o usuário conforme o tipo do alerta detectado.
- Exemplos:
 - Enchente: "Evite áreas baixas. Use rotas seguras..."
 - Gás: "Ventile o ambiente, evacue e ligue para 193."
 - Tornado: "Busque abrigo subterrâneo ou lugar seguro."



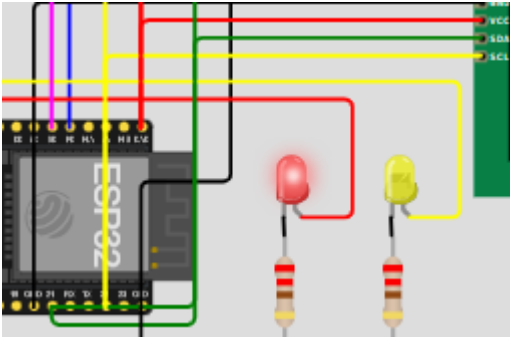
8. Atualização do Display LCD (atualizarLCD ())

- Atualiza o display em ciclos a cada 3 segundos.
- Mostra:
 - Temperatura e Umidade
 - Chuva e Gás
 - Vento e Movimento



9. Controle dos LEDs de Alerta (piscaLEDsAlerta ())

- Pisca o LED vermelho para alertas graves.
- Pisca o LED amarelo para alertas moderados.
- Mantém LED desligado se não há alerta.



10. Alerta Sonoro (`emitirAlertaSonoro()`)

- Aciona buzzer com tom constante se alerta ativo.
- Desliga buzzer se não há alerta.

11. Loop Principal (`loop()`)

- Garante conexão MQTT.
- Lê sensores de temperatura, umidade, chuva, gás, vento e movimento.
- Verifica e classifica possíveis eventos extremos.
- Atualiza display LCD.
- Envia dados via MQTT.
- Controla LEDs e buzzer para alertas.
- Aguarda 10 segundos antes da próxima leitura.

6– Nós do Node-RED

Visão Geral do Fluxo

O fluxo consiste em:

- 3 subfluxos de sensores climáticos (`sensor1`, `sensor2`, `sensor3`)
- 1 fluxo de alertas
- Interface de usuário organizada em abas ("Sensores" e "Alertas")

Nós Detalhados

1. Nó Tab (Flow Container)

json

Copy

Download

```
{
  "id": "2dd2e60046001308",
  "type": "tab",
  "label": "Flow 1",
  "disabled": false,
  "info": "",
  "env": []
}
```

- **Função:** Contêiner principal que agrupa todos os outros nós deste fluxo.
- **Configuração:** Nomeado como "Flow 1", ativo e sem variáveis de ambiente especiais.

2. Nós MQTT Input

Existem 4 nós MQTT de entrada que se conectam ao broker HiveMQ:

a) Nó de Alertas

json

Copy

Download

```
{
  "id": "54b2f6e5bd892460",
  "type": "mqtt in",
  "name": "",
  "topic": "fiap/eventos-extremos/alertas",
  "qos": "2",
}
```

```
"broker": "adb114682f965e27"
}
```

- **Função:** Recebe mensagens de alerta no tópico `fiap/eventos-extremos/alertas` com QoS 2 (entrega exatamente uma vez).
- **Saídas:** Encaminha para um nó de debug e um template UI.

b) Nós de Sensores (3 no total)

json

Copy

Download

```
{
  "id": "ef0ba926ed29f95a",
  "type": "mqtt in",
  "topic": "fiap/eventos-extremos/sensor1",
  "qos": "2",
  "broker": "adb114682f965e27"
}
```

- **Função:** Cada um recebe dados de um sensor específico (sensor1, sensor2, sensor3).
- **Saídas:** Encaminham para nós de debug e templates UI correspondentes.

3. Nós Debug (5 no total)

```
{
  "id": "236381b94009bc28",
  "type": "debug",
  "name": "debug 2",
  "active": true,
  "complete": "payload"
}
```

- **Função:** Exibe o conteúdo das mensagens recebidas na sidebar do Node-RED.
- **Configuração:** Cada um está configurado para mostrar o payload completo das mensagens.

4. Nós UI Template (4 no total)

Templates HTML estilizados para exibição na interface do usuário:

a) Template de Alerta

```
{
  "id": "5ba536943e13b2cd",
  "type": "ui_template",
  "group": "54dd2e902608cee2",
  "format": "...
}
```

- **Função:** Exibe mensagens de alerta em um cartão estilizado com animação.
- **Características:**
 - Design moderno com gradiente e sombras
 - Ícone de segurança
 - Animação de pulsação azul
 - Exibe o conteúdo de `msg.payload`

b) Templates de Sensores (3 no total)

```
{
  "id": "59775960773ff8f7",
  "type": "ui_template",
  "group": "29181e19c8f3dd0b",
  "format": "...
}
```

- **Função:** Exibe dados dos sensores e controles simulados.
- **Características:**
 - Mostra temperatura, umidade, chuva e vento
 - Controles deslizantes para ajustar parâmetros
 - Mesmo estilo visual do template de alerta
 - Função JavaScript `updateSensorData()` para atualizar valores

5. Nó Spacer

```
{  
  "id": "29692773c0e3c292",  
  "type": "ui_spacer",  
  "group": "9209ebba6fcab075"  
}
```

- **Função:** Espaçador na interface do usuário para organização visual.

6. Configuração do Broker MQTT

```
{  
  "id": "adb114682f965e27",  
  "type": "mqtt-broker",  
  "broker": "broker.hivemq.com",  
  "port": "1883"  
}
```

- **Função:** Configuração da conexão com o broker público HiveMQ na porta 1883 sem TLS.

7. Grupos UI (4 no total)

Organizam os elementos na interface:

a) Grupo de Alertas

```
{  
  "id": "54dd2e902608cee2",  
  "type": "ui_group",  
  "name": "Alerta",  
  "tab": "e02975db086f16db",  
  "width": "6"  
}
```

- **Associado à aba "Alertas"**
- Largura de 6 colunas

b) Grupos de Sensores (2 no total)

```
{  
  "id": "29181e19c8f3dd0b",  
  "type": "ui_group",  
  "tab": "9b0d322ede19aa9c",  
  "width": 9  
}
```

- **Associados à aba "Sensores"**
- Largura de 9 colunas cada

8. Abas UI (2 no total)

Organizam a interface em seções:

a) Aba "Alertas"

```
{  
  "id": "e02975db086f16db",  
  "type": "ui_tab",  
  "name": "Alertas",  
  "icon": "dashboard"  
}
```

b) Aba "Sensores"

```
{  
  "id": "9b0d322ede19aa9c",  
  "type": "ui_tab",  
  "name": "Sensores",  
  "icon": "dashboard"  
}
```

Fluxo de Dados

1. Os nós MQTT recebem dados dos tópicos:
 - `fiap/eventos-extremos/sensor[1-3]` para dados de sensores
 - `fiap/eventos-extremos/alertas` para alertas

2. Os dados são:

- Exibidos no debug para monitoramento
- Encaminhados para os templates UI correspondentes

3. A interface do usuário é organizada em duas abas principais:

- "Sensores": Mostra os painéis de controle dos três sensores
- "Alertas": Exibe mensagens de alerta formatadas

Considerações de Design

- **Segurança:** Utiliza QoS 2 para garantir entrega de mensagens
- **Experiência do Usuário:**
 - Interface moderna com animações e feedback visual
 - Organização clara em abas e grupos
 - Design responsivo que se adapta a diferentes tamanhos de tela
- **Monitoramento:** Múltiplos nós debug para verificação do fluxo de dados

7-Conclusão

À medida que o planeta enfrenta o aumento da frequência e da intensidade dos eventos climáticos extremos, a necessidade de resposta rápida e eficiente nunca foi tão urgente. Mas como podemos nos proteger de algo que, muitas vezes, surge sem aviso? Em um mundo onde os fenômenos da natureza se tornam cada vez mais imprevisíveis, a tecnologia surge como nossa aliada. Sistemas de alerta, como o **Swift Safety**, não são apenas ferramentas de monitoramento; são sentinelas que nos ajudam a enxergar o invisível, a antecipar o desconhecido.

Com a ajuda de sensores inteligentes, que detectam desde mudanças na temperatura até a intensidade do vento, e com a força da conectividade digital, a informação sobre perigos iminentes circula de forma rápida e eficiente, criando uma rede de segurança colaborativa. Os alertas não são mais apenas números ou gráficos em uma tela distante; eles se transformam em mensagens vitais, capazes de salvar vidas e minimizar danos.

Em um mundo cada vez mais vulnerável, onde o clima não faz mais promessas, iniciativas como essa nos lembram da importância de estarmos preparados. O **Swift Safety** é mais do que um sistema; é a prova de que, com a tecnologia certa e a colaboração de todos, podemos enfrentar o imprevisível, salvar vidas e, quem sabe, até mesmo proteger o nosso futuro.