

# Documento técnico - Avaliação Prática

**Nome do projeto:**

**Equipe:**

---

## 1. Arquitetura e Diagrama do Sistema

- Visão Geral

O sistema é dividido em **4 subsistemas**, conforme exigido:

### 1. Nó de Controle de Acesso (NCA)

- Sensor HC-SR04 detecta entrada/saída.
- Publica eventos de fluxo via MQTT.

### 2. Unidade de Monitoramento Ambiental e Feedback (UMAF)

- Coleta temperatura e umidade com DHT11.
- Recebe do Orquestrador o estado consolidado para atualizar OLED e LED RGB.

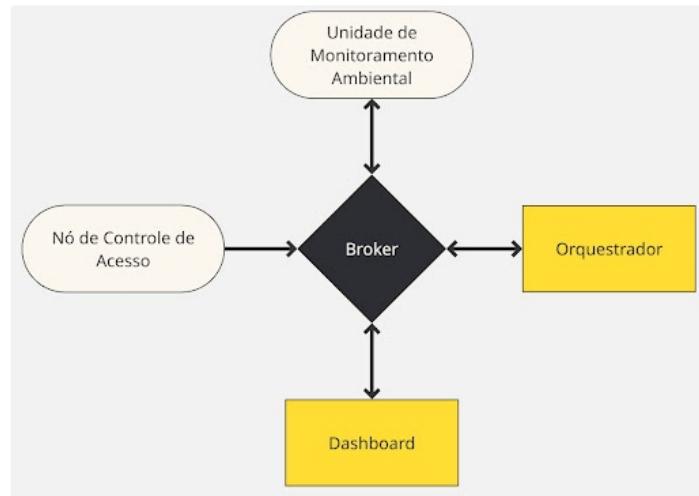
### 3. Orquestrador (Node-RED – Lógica)

- Calcula ocupação atual.
- Aplica regras de status (Livre, Atenção, Alerta).
- Publica estado consolidado.

### 4. Dashboard (Node-RED – Supervisão Remota)

- Exibe temperatura, umidade, ocupação e status.
- Permite configurar limite de ocupação.

#### - Fluxo Resumido de Comunicações (MQTT)



## 2. Descrição dos Componentes Utilizados

### - Componentes de Hardware

Componente	Função	Justificativa Técnica
<b>ESP32-C6</b>	Microcontrolador principal responsável pela leitura dos sensores e comunicação MQTT via Wi-Fi.	Alta capacidade de processamento, suporte nativo a Wi-Fi, baixo consumo de energia e integração facilitada com bibliotecas IoT.
<b>SR-HC04</b>	Sensor ultrassônico utilizado para detectar entrada e saída de pessoas.	Sensor de baixo custo, fácil integração com o ESP32 e boa precisão para detecção de movimento e distância.
<b>DHT11</b>	Sensor de temperatura e umidade do ambiente.	Fornece dados ambientais básicos com simplicidade, boa estabilidade e fácil implementação.
<b>NFP1315-61AY</b>	Display alfanumérico de 4 dígitos utilizado	Alta visibilidade mesmo em ambientes

	para exibir informações locais (ocupação, temperatura ou mensagens).	iluminados, baixo consumo e interface simples com o microcontrolador.
<b>LED RGB (ESP32-C6)</b>	Sinalização visual do status de lotação da sala (Livre, Enchendo, Meio Cheio, Quase Cheio e Lotado).	Já integrado ao ESP32-C6, dispensando componentes externos; operação via biblioteca NeoPixel, baixo consumo e alta confiabilidade.

---

#### - Software

Software	Função
Arduino IDE	Programação dos ESP32
Node-RED	Orquestração, tratamento de dados e dashboard
Mosquitto MQTT Broker	Comunicação MQTT entre dispositivos
Biblioteca PubSubClient	Comunicação MQTT no ESP32
Biblioteca Adafruit SSD1306	Controle do display OLED
Biblioteca DHT	Leitura do sensor DHT11

---

### 3. Tópicos MQTT e Payloads

NCA - Orquestrador

Tópico	Função	Exemplo de Payload
placas/on/off/ vini	Envia status <b>Online</b> quando a placa é	{ "status": "Online" }

		iniciada.
vini/topic/placa1/envia	Envia eventos de <b>entrada/saída</b> de pessoas, junto com o horário.	{ "evento": "Entrando", "timestamp": "18/11/2025" }

#### UMAF → Orquestrador

Tópico	Função	Exemplo de Payload
placas/on/off/bruno	Envia status <b>Online</b> quando a placa é iniciada.	{ "status": "Online" }
bruno/topic/placa2/envia	Envia <b>temperatura</b> e <b>umidade</b> medidas pelo DHT11.	{ "temp": 26, "humi": 50 }

#### Dashboard → Orquestrador

Tópico	Função	Exemplo de Payload
definir_ocupacao	Envia a ocupação máxima definida pelo usuário no painel.	{ "max": 50 }

#### Orquestrador → UMAF

Tópico	Função	Exemplo de Payload
bruno/topic/placa2/recebe/status	Envia o <b>status da sala</b> para a placa.	{ "status": "Livre" }

<code>bruno/topic/placa2/recebe/ocupacao</code>	Envia os dados de <b>ocupação atual/total.</b>	<code>{ "ocupacao": "10/50" }</code>
---	--	--------------------------------------

Orquestrador → Dashboard

Tópico	Função	Exemplo de Payload
<code>sensor/hora</code>	Envia o status do sensor (entrada/saída) e horário.	<code>{ "timestamp": "18/11/2025" }</code>
<code>grafico/temp/umi</code>	Envia temperatura e umidade para gráficos.	<code>{ "temp": 26, "humid": 50 }</code>
<code>ocupacao/grafico</code>	Envia ocupação atual para o gráfico.	<code>{ "atual": 10 }</code>
<code>ocupacao</code>	Envia ocupação atual e total.	<code>{ "ocupacao": "10/50" }</code>
<code>statusagora</code>	Envia o status geral da sala.	<code>{ "status": "Livre" }</code>
<code>status_placa1</code>	Envia status do subsistema 1 (placa 1).	<code>{ "status": "Online" }</code>
<code>status_placa2</code>	Envia status do subsistema 2 (placa 2).	<code>{ "status": "Online" }</code>

#### 4. Evidências de Funcionamento

- Dashboard

- Gráficos de temperatura, umidade e ocupação
- Indicador de status da sala (Livre, Meio Cheio, Quase Cheio e Lotado)
- Controle de limite de ocupação

- Indicador de status das placas (Online/Offline)

