🧠 Arquitetura Geral

| Camada | Dispositivo | Função Principal |

|--------------------|-------------------|-------------------------------------------------------|

| Sensores e atuadores | Arduino Uno | Leitura de sensores e controle físico do reator |

| Processamento e interface | Raspberry Pi 4B | Interface web, armazenamento, controle remoto |

| Controle remoto | Smartphone | Acesso via navegador ou app à interface do Raspberry Pi|

---

## 🔧 Componentes do MVP

### Sensores conectados ao Arduino:

- DHT22 – Temperatura e umidade do ar

- Sensor capacitivo de umidade do solo

- DS18B20 – Temperatura interna do composto

- Sensor de pH analógico

- MQ135 – Gases (CO₂, NH₃)

- BMP280 – Pressão interna

### Atuadores:

- Ventilador 5V ou 12V

- Bomba de água

- Resistência de aquecimento (via relé)

- Módulo Peltier (opcional)

---

## 🔌 Comunicação Arduino ↔ Raspberry Pi

- Via \*Serial USB\* usando a biblioteca PySerial

- Arduino envia leituras dos sensores

- Raspberry Pi envia comandos de controle (ex: ligar bomba)

---

## 📱 Interface Web para Smartphone

### Desenvolvida com Flask + Bootstrap

- Painel com:

- Leituras em tempo real

- Botões para controle manual dos atuadores

- Gráficos de histórico (Plotly ou Chart.js)

- Alertas visuais (ex: temperatura fora do ideal)

### Acesso remoto:

- Raspberry Pi conectado ao Wi-Fi

- Interface acessível via IP local (ex: http://192.168.0.100)

- Para acesso externo:

- Configurar \*port forwarding\* no roteador

- Ou usar \*ngrok\* para criar um túnel seguro

---

## 🧠 Lógica de Controle

### Arduino (C++):

- Lê sensores e aplica regras locais

- Envia dados via Serial

- Recebe comandos do Raspberry Pi (ex: “ligar ventilador”)

### Raspberry Pi (Python):

- Recebe dados via Serial

- Atualiza interface web

- Salva dados em CSV ou SQLite

- Envia comandos via Serial conforme interação do usuário

---

## 🧪 Exemplo de Fluxo

1. Usuário acessa a interface pelo celular

2. Vê que a umidade está abaixo de 40%

3. Clica em “Ativar bomba”

4. Raspberry Pi envia comando via Serial

5. Arduino aciona a bomba por 5 segundos

6. Interface atualiza com novo status

---

## 🚀 Recursos Extras

- Notificações por Telegram (via bot)

- Modo automático/manual

- Histórico exportável

- Timelapse com câmera USB (opcional)

---

## 🧰 Requisitos de Software

- Arduino IDE (para programar o Uno)

- Python 3 no Raspberry Pi

- Bibliotecas: pyserial, flask, plotly, sqlite3, gpiozero (se quiser usar GPIO direto)