

Projeto IoT com Raspberry Pi e ESP32/ESP8266

- **Disciplina:** Eletiva 2 - Comunicação e Redes
 - **Professor:** Jymmy Barreto
 - **Data de Entrega:** 04/12
 - **Duração da Apresentação:** 15 a 20 minutos
-

1. Objetivo do Projeto

Desenvolver um sistema IoT que integre:

- **Raspberry Pi 3 B ou B+ (ou superior)** como Broker MQTT. Pode ser outro dispositivo (**ESP32, PC ou outra placa as mesmas características de uma Raspberry Pi**), caso não tenha disponibilidade de Raspberry Pi 3 B ou B+, no entanto isso deve ser acertado com o professor.
- **ESP32 ou ESP8266** como dispositivos sensores/atuadores.
- **Comunicação sem fio** (Wi-Fi) e protocolo **MQTT** para troca de dados.
- Preferencialmente, usar o **protocolo IPv6 como pilha**.
- **Dashboard** para visualização em tempo real dos dados coletados.

Os estudantes terão liberdade para propor a aplicação (ex.: monitoramento ambiental, automação residencial, segurança, etc.), desde que cumpram os requisitos técnicos.

2. Requisitos Técnicos

Hardware Obrigatório

- 1x Raspberry Pi 3 B ou B+ (ou modelo superior: Pi 4, Pi 5) como Broker e Servidor.
 - Pode ser trocado por **AWS Learner Lab** para executar toda a parte de infra (**Broker, Back-end, Front-end**) na AWS
 - Caso o grupo necessite do **AWS IoT Core**, tratar comigo para que eu libere um **Learner Lab na AWS Academy**
- 1x ou 2x módulos ESP32 ou ESP8266 (NodeMCU).
- Sensores básicos (ex.: DHT11/DHT22, LM35, LDR, PIR, etc.).
- Componentes eletrônicos (resistor, LED, jumper, etc.).

Software/Ferramentas

- **Raspberry Pi:**
 - **Sistema Operacional:** Preferencialmente o **Raspberry Pi OS (Linux) ou Raspbian**.
 - Broker MQTT.
 - Protocolo IPv6.
 - Aplicação Web e/ou Dashboards (Node-RED, Grafana, Python + Flask/Django, NodeJs + React ou outras formas similares previamente acordada com o professor).
- **ESP32/ESP8266:**

- Firmware (FreeRTOS, MicroPython, Lua ou C/C++).
- Biblioteca MQTT.
- Biblioteca IPv6
- **Versionamento:** GitHub (repositório público).

3. Critérios de Avaliação

Item	Pontuação	Detalhes
Check Point	5 pts	Demonstração e pequenas entregas do projeto.
Funcionamento do Sistema	25 pts	Projeto operando conforme requisitos (comunicação MQTT, Protocolo IPv6 coleta de dados, dashboard).
Aplicação Web / Dashboard	20 pts	Interface visual funcional, exibindo dados em tempo real.
Relatório Técnico (ABNT2)	20 pts	Estrutura completa: introdução, metodologia, resultados, códigos em apêndice.
Apresentação (04/12)	15 pts	Clareza, demonstração ao vivo, divisão de tarefas no grupo.
GitHub (Organização)	15 pts	Repositório bem documentado, commits colaborativos, README detalhado.

Nota: Projetos com funcionalidades extras (ex.: **Alertas por e-mail ou Apps de mensagens, Integração com Zabbix, Infraestrutura de comunicação em IPv6**) ganharão bônus de até **10 pts**.

4. Cronograma e Acompanhamento

- **06/11 a 02/12:** Checkpoints semanais (presenciais/remotos).
 - **06/11:** Definição da ideia.
 - **18/11:** Testes preliminares (Prototipação de hardware, esboço de código e outros artefatos de encaminhamento do projeto).
 - **02/12:** Validação do dashboard e ajustes finais.
- **04/12:** Apresentação final (**15 min por grupo**).
- **09/12:** Entrega final de todos os artefatos do Projeto.

5. Estrutura do Relatório (ABNT2)

1. **Capa** (Nome da Faculdade, título, autores, data).
2. **Introdução** (Contexto, motivação, objetivos).
3. **Metodologia**
 - Diagrama do sistema.
 - Lista de hardware/software.
 - Fluxo de comunicação (MQTT + Wi-Fi).

4. Resultados

- Prints do dashboard.
- Dados coletados (gráficos/tabelas).
- Evidências de funcionamento (fotos/vídeos).

5. Conclusão (Desafios, aprendizados, melhorias futuras).

6. Apêndice

- Códigos completos (Raspberry Pi + ESPs).
- Configurações do broker MQTT.

6. Template do Repositório GitHub

```
|— README.md          # Descrição do projeto, requisitos, instruções.
|— /docs              # Relatório em PDF (ABNT2) + imagens.
|— /raspberrypi       # Códigos do broker MQTT + dashboard.
|— /esp32-esp8266     # Firmware dos módulos (FreeRTOS/MicroPython/C/C++/IDE
Arduíno).
|— /schematics        # Diagramas eletrônicos (Fritzing ou KiCad).
```

Dica: Use o [GitHub Projects](#) para gerenciar tarefas em equipe.

7. Ideias Sugeridas e Links de Apoio para o Projeto

Aqui estão algumas ideias para inspirar seu projeto IoT, além de links úteis com tutoriais e exemplos práticos:

Ideias Sugeridas

- **Estação Meteorológica:** ESPs enviam temperatura/umidade para o Pi, que exibe em um gráfico.
- **Alarme com Sensor PIR:** ESP detecta movimento e envia alerta MQTT para o Pi acionar um aviso no dashboard.
- **Irrigação Automática:** Sensor de umidade do solo (ESP) controla uma válvula solenoide via Pi.
- **Monitoramento de Energia:** Medição de consumo elétrico com sensor de corrente (ACS712) + exibição no dashboard.

Links de Apoio

1. Projetos com Raspberry Pi:

- [Random Nerd Tutorials – Projetos Raspberry Pi](#)
- Exemplos: servidor MQTT, controle de GPIO, integração com sensores.

2. Projetos IoT Prontos (ESP e Pi):

- [IoT Design Pro – Ideias de Projetos](#)
- Tutoriais passo a passo com código e esquemas.

3. Projetos com ESP8266/ESP32:

- [IoT Circuit Hub – ESP8266 Projects](#)
- Projetos focados em MQTT, Wi-Fi e sensores.

4. Extras:

- [Mosquitto MQTT Broker Setup](#) (para configurar o broker no Pi).
- [Node-RED Dashboard Tutorial](#) (para criar interfaces visuais).

Dica: Combine ideias dos links com os requisitos do projeto (ex.: usar um tutorial de **ESP8266 + MQTT** e adaptar ao dashboard no Pi).

Se precisar de ajuda para integrar componentes, consulte os tutoriais ou peça orientação durante os checkpoints! 🚀

8. Formação dos Grupos

Total de participantes por grupos:

- **Mínimo 5**
- **Máximo 6**

- Todo grupo deverá eleger 1 integrante para representar o projeto;
- O professor irá sortear + 1 integrante;
- O representante deverá indicar+ 1 integrante.

9. Entrega Final

- **Formato:**
 - Repositório GitHub (link do projeto).
 - Relatório em PDF (nome no padrão: **Projeto_GrupoX.pdf**).
- **Prazo:** 04/12 (até 23h59).

As entregas serão acompanhadas via Google Classroom.

Observação: **Projetos não funcionais na apresentação perderão 50% da pontuação de "Funcionamento".**
