

[← VOLTAR À LISTA](#)

## ATIVIDADE EM GRUPO

### Cap 1 - CardiolA Conectada: IoT e Visualização de Dados para a Saúde Digital

DE 09/10/2025 A 24/10/2025

ENTREGA PENDENTE

ENTREGAR ATIVIDADE

Atividades entregues **até 3 dias após o prazo** receberão até **70% da nota**.  
O cálculo é feito automaticamente pelo sistema, o professor não tem controle sobre o percentual da nota atribuída.

## INTRODUÇÃO

### ATIVIDADE - FASE 3: Monitoramento Contínuo - IoT na Saúde

#### **Atenção Atividade Avaliativa:**

- Verifique se o **arquivo do upload está correto**, não é possível enviar um outro arquivo após fechamento da entrega na plataforma ou correção do professor.
- Não deixe para realizar a entrega da atividade **nos últimos minutos do prazo**, você pode ter algum problema e perder a entrega. As entregas são realizadas apenas pela plataforma.
- **Não disponibilize a resposta** da sua atividade em grupos de WhatsApp, Discord, Microsoft Teams, pois pode gerar plágio e zerar a atividade para todos.
- Você tem um período máximo de **15 dias** após a publicação da nota para solicitar a revisão da correção.

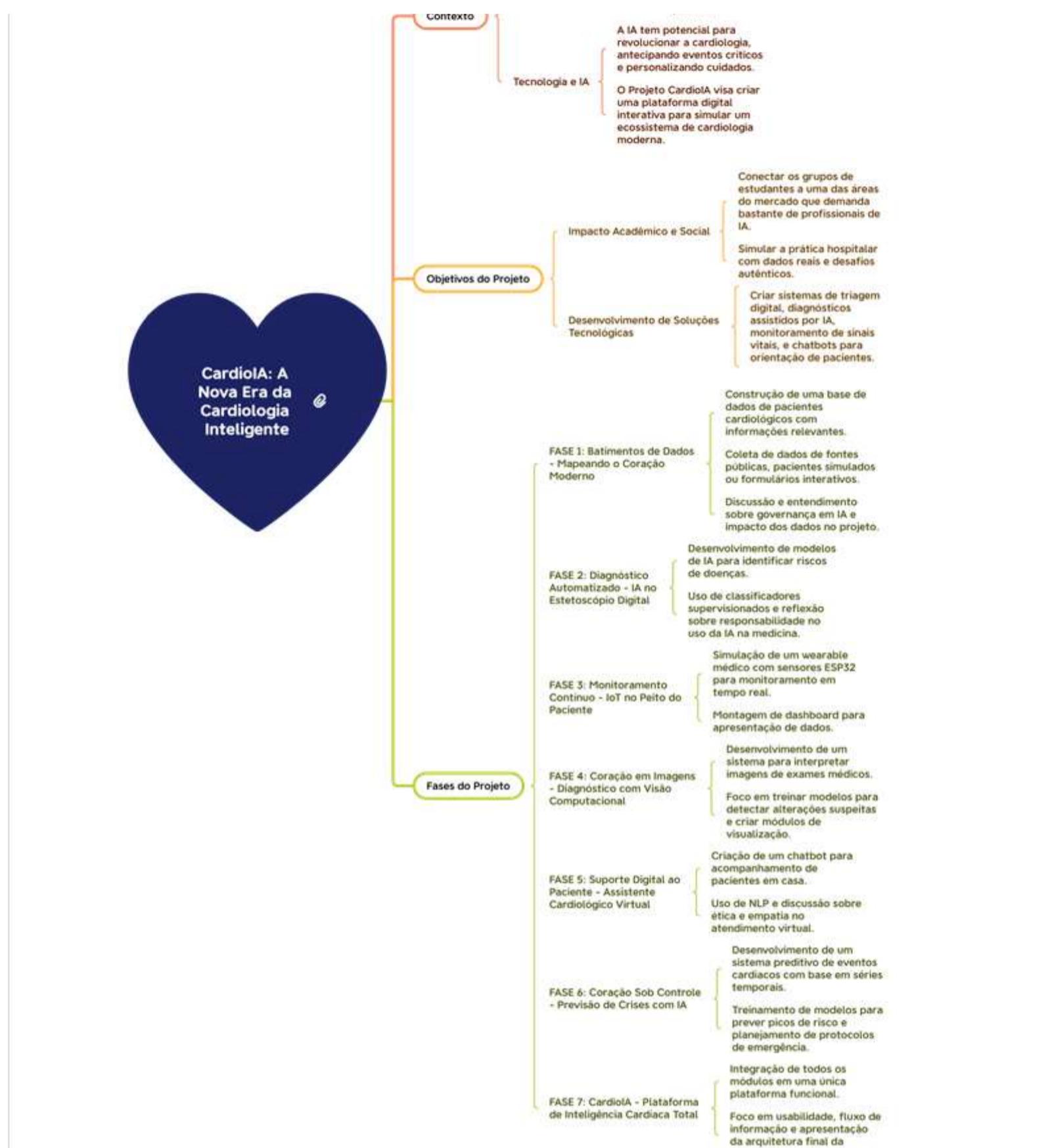
#### Enunciado da atividade:

Nesta fase, você e sua equipe irão avançar no projeto CardiolA, integrando conceitos de **IoT**, **computação em nuvem**, **Edge/Fog Computing** e **visualização de dados**. O desafio é simular um sistema de monitoramento contínuo que capture sinais vitais de pacientes cardiológicos, armazene e processe informações localmente, transmita os dados para a nuvem e exiba resultados em dashboards interativos.

Você irá trabalhar com **ESP32 no Wokwi**, **protocolos de comunicação (MQTT)**, **armazenamento local (SPIFFS)** e **dashboards (Node-RED/Grafana)**. Essa combinação permitirá compreender o fluxo completo de dados em IoT aplicado à saúde: captura → processamento → transmissão → visualização → alerta.

Para situar melhor a jornada, observe no mapa mental em que ponto o projeto CardiolA se encontra nesta fase. É importante destacar que o material teórico que dará suporte a este projeto está concentrado nas fases 2 e 3, consideradas mais adequadas para você neste momento. A visualização

[!\[\]\(99f58673407353e96a019fbca558fd72\_img.jpg\) VOLTAR À LISTA](#)



Para visualizar o mapa mental completo, acesso o link: [mapaMental - CardioIA\\_A Nova Era da Cardiologia Inteligente.svg](#)

#### Objetivo geral dessa atividade:

O objetivo é desenvolver um protótipo funcional que simule um sistema vestível de monitoramento cardíaco, capaz de:

- Capturar sinais vitais simulados (exemplo: batimentos cardíacos, temperatura, umidade e movimento).
- Armazenar localmente e garantir resiliência offline (Edge Computing).
- Transmitir dados para a nuvem via MQTT.
- Visualizar informações em dashboards com alertas automáticos.
- Refletir sobre eficiência, segurança e boas práticas em IoT médico.

**Observação:** o aluno não é obrigado a comprar nenhum hardware ou sensores para realizar esta atividade. O uso de dispositivos físicos é opcional, no entanto recomendamos que você tenha todo o material necessário por perto para uma melhor formação profissional. Caso deseje utilizar equipamentos

[← VOLTAR À LISTA](#)

Segurança e responsabilidade no uso dos dados. Essa solução é 100% aplicável em projetos reais no mercado.

#### Atividade detalhada:

##### **PARTE 1 – Armazenamento e processamento local (Edge Computing):**

Desenvolver no **Wokwi** uma aplicação com ESP32, onde:

- Seu projeto tenha no mínimo 2 sensores distintos.
  - Um sensor obrigatório de temperatura, responsável por ler periodicamente dados do sensor DHT22 (como o DHT22, que mede tanto temperatura quanto umidade — por isso, vamos considerar o DHT22 como sendo 1 sensor). O segundo sensor é de livre escolha do grupo para desenvolver a sua aplicação.
  - Armazene cada leitura localmente no sistema de arquivos SPIFFS.
  - Simule conectividade Wi-Fi (variável booleana). Quando “conectado”, envie os dados armazenados para a nuvem via Serial.println e apague o arquivo local.
  - Garanta resiliência offline: mesmo sem conexão, o sistema continua coletando dados e quando voltar a ficar online, os dados não enviados sejam sincronizados. Escolha uma estratégia de armazenamento limitado nesse processo de resiliência alinhado com o modelo de negócio do seu projeto, isto é, quantas milhares de amostra você pretende registrar no seu ESP32 até ficar online novamente.
  - Essa etapa demonstra o papel do Edge Computing em aplicações críticas de saúde.

#### Entregáveis:

- Link do projeto no Wokwi.
- Código em C++ comentado.
- Relatório simples (mínimo de uma página) descrevendo o fluxo de funcionamento e a lógica de resiliência.

##### **PARTE 2 – Transmissão para nuvem e visualização (Fog/Cloud Computing):**

Criar um sistema completo de monitoramento que:

- Envie dados simulados do ESP32 para a nuvem via protocolo MQTT (exemplo de broker: HiveMQ Cloud).
- Monte uma dashboard no Node-RED para exibir em tempo real:
  - Gráfico de um sinal vital escolhido pelo grupo (temperatura + outro sensor simule batimentos cardíacos, como um botão de pressão, que você aperta x vezes por minuto).
  - Medidor (gauge) de outro parâmetro relevante (exemplo: temperatura ou sensor alternativo).
  - Indicador visual de alerta (LED virtual ou texto) quando valores ultrapassarem limites definidos pelo grupo (exemplo: > 120 bpm, temperatura > 38 °C ou outro parâmetro relevante).
- Opcionalmente, integrar com Grafana Cloud para visualização avançada.

[← VOLTAR À LISTA](#)

-----

- Código do ESP32 (Arduino IDE/Wokwi).
- Prints evidenciando a aplicação ou export do dashboard no Node-RED (e Grafana, se utilizado).
- Relatório (mínimo de duas páginas) sobre o fluxo de comunicação MQTT e configuração do dashboard.

#### Critérios de avaliação (10 pontos totais):

Critério	Pontos
Leitura de sensores e armazenamento no SPIFFS	2
Implementação de resiliência offline (Edge Computing)	2
Envio via MQTT e integração com broker	2
Dashboard funcional e alertas automáticos	2
Documentação clara (código + relatórios)	2

### IR ALÉM 1 – Comunicação automatizada com REST e e-mail

#### Objetivo:

Simular um sistema de monitoramento que consuma e envie dados de sinais vitais via uma API REST em Python, incorporando lógica de detecção de riscos (exemplo: taquicardia, febre, ausência de movimento) e acionando uma automação de envio de e-mail em caso de alerta. O exercício conecta protocolos HTTP, bibliotecas Python e conceitos de RPA aplicados à saúde digital.

#### Entregáveis:

- Código em Python implementando cliente REST para envio e recebimento de dados.
- Implementação da lógica de verificação de risco.
- Simulação de disparo de e-mail automatizado em caso de alerta.
- Relatório curto (de uma a duas páginas) descrevendo o fluxo implementado.

#### Critérios de avaliação:

- Implementação correta do consumo de API REST.
- Lógica de verificação de riscos bem definida.
- Automação do envio de e-mail funcional.
- Clareza do código e documentação.
- Organização da entrega (manter o código limpo, documentação clara e estrutura de arquivos bem definida).

### IR ALÉM 2 – Inteligência Artificial em séries temporais de saúde

[← VOLTAR À LISTA](#)**ENTREGAVEL:**

- Notebook Python comentado.
- Repositório GitHub público com código e README.
- Relatório comparativo (duas páginas) sobre o desempenho dos modelos.
- Vídeo de até 4 minutos apresentando resultados, postado no YouTube como "não listado" e linkado no README.

**Critérios de avaliação:**

- Implementação funcional dos modelos.
- Comparaçao clara entre métodos.
- Clareza e organização do notebook.
- Qualidade do relatório comparativo.
- Demonstração em vídeo com análise crítica.

**Mensagem final:**

Integrar **Edge, Fog e Cloud Computing** em aplicações médicas é um dos maiores desafios atuais da IoT. Nesta fase, você terá vivenciado o ciclo completo de captura, processamento, transmissão e visualização de dados em saúde digital. Ao avançar para os desafios “Ir Além”, você expande ainda mais seu repertório, conectando IoT a Front-End e Inteligência Artificial aplicada.

**GRUPO**

Mínimo: 1 | Máximo: 5

Participante (s)

<input checked="" type="checkbox"/>	Fernando Miranda Segregio segregio@gmail.com	<input type="button" value="X"/>
<input checked="" type="checkbox"/>	Wellington Nascimento de Brito well334@hotmail.com	<input type="button" value="X"/>
<input checked="" type="checkbox"/>	Mário Roberto Silva de Almeida marioalmeida1980@gmail.com	<input type="button" value="X"/>
<input checked="" type="checkbox"/>	Caio Rodrigues Castro caiorcastro@gmail.com	<input type="button" value="X"/>
<input checked="" type="checkbox"/>	Felipe Soares Nascimento consultor.casteliano@gmail.com	<input type="button" value="X"/>

ENTREGA PENDENTE

 Avaliação

AVALIAÇÃO PENDENTE

**PRAZO DE ENTREGA**



[VOLTAR À LISTA](#)

---

#### ÚLTIMA MODIFICAÇÃO

---

#### GRUPO

Grupo 1

---

#### INTEGRANTES DO GRUPO 1

Fernando Miranda Segregio  
segregio@gmail.com

Wellington Nascimento de Brito  
well334@hotmail.com

Mário Roberto Silva de Almeida  
marioalmeida1980@gmail.com

Caio Rodrigues Castro  
caiorcastro@gmail.com

Felipe Soares Nascimento  
consultor.casteliano@gmail.com