



Internet das Coisas em um Mundo Conectado

UNIDADE 07

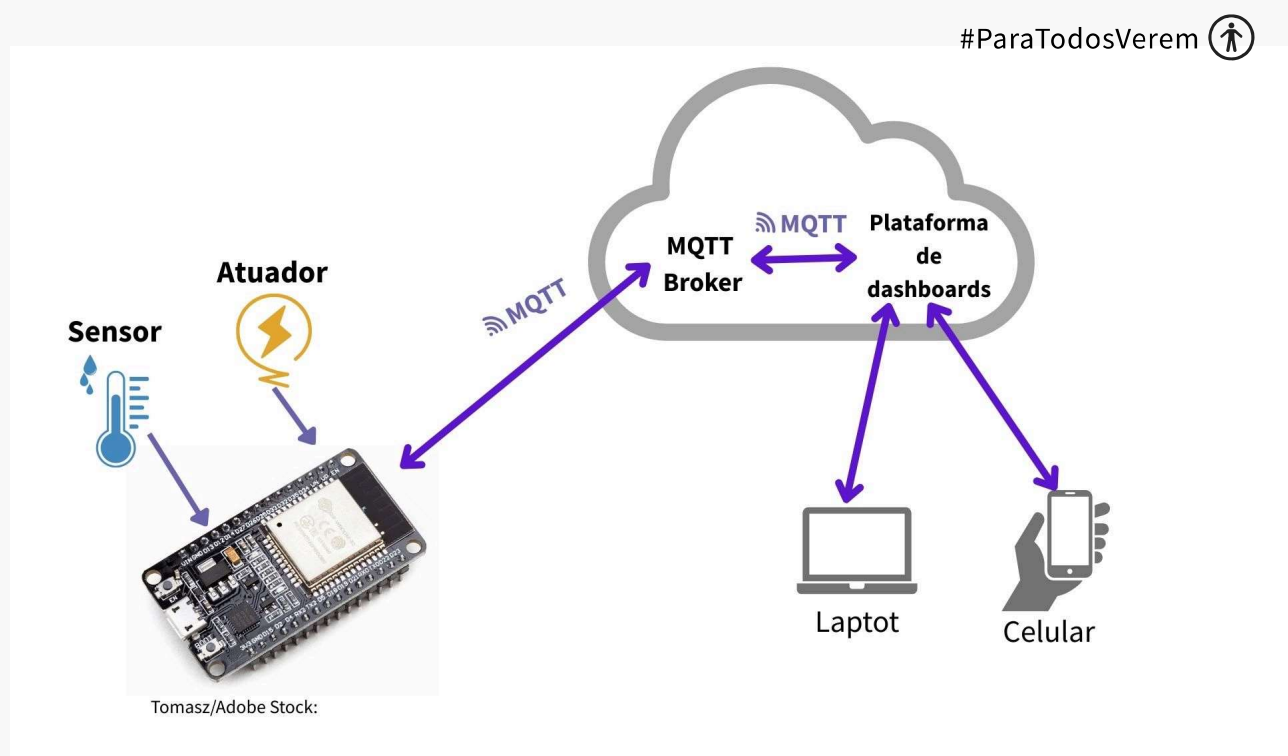
Serviços em nuvem

Nesta unidade, o foco será na construção de uma interface visual que permita ao usuário monitorar ou controlar o circuito simulado remotamente. Para isso, será utilizado o aplicativo Blynk, que se conecta ao ESP32 via internet e proporciona uma experiência visual amigável.

Ao construir esse painel, você aplicará os conceitos de interatividade e controle em tempo real, essenciais para soluções modernas de IoT voltadas ao usuário final. O painel criado no Blynk funciona como uma ponte intuitiva entre o mundo digital e o físico, permitindo desde a leitura de sensores até o acionamento remoto de dispositivos, como lâmpadas ou relés simulados.

Você poderá criar elementos interativos como botões, medidores (gauge), LEDs virtuais e outros widgets para representar o funcionamento do seu projeto. Essa etapa torna o sistema mais acessível para o usuário final e simula situações reais de monitoramento remoto, como em sistemas de automação residencial ou agrícola. Além de facilitar o uso, esse tipo de interface aumenta a confiabilidade do sistema e permite seu uso em domínios variados, como residências inteligentes, ambientes agrícolas, controle industrial e monitoramento ambiental, reforçando a aplicabilidade prática da atividade.

A ideia é estabelecer a conexão do dispositivo que criamos na nossa última aula com o Blynk, utilizando o editor de fluxo para conectar e visualizar os dados através da interface web da ferramenta. Finalmente moveremos o código de nosso dispositivo para um ESP32 simulado, via Wokwi. A arquitetura da solução iniciada anteriormente e representada na figura a seguir.



O autor

Nesta arquitetura iremos estabelecer a conexão de um dispositivo através do Simulador WowKi, integrando com o MQTT Broker e plataforma Blynk, onde visualizaremos os dados através da interface web da ferramenta. Finalmente implementaremos uma solução integrada de IoT com o protocolo MQTT.

Videoaula: Integração com Blynk

Neste vídeo, você aprenderá a utilizar o Blynk para criar um painel interativo em nuvem, integrando-o ao projeto iniciado na anteriormente com Wokwi e MQTTX.



MUNDO DO TRABALHO

Lembra do nosso case do setor agrícola? Vamos pensar em uma evolução para ele.

Após o sucesso do protótipo inicial, a startup avançou para a **segunda fase do projeto**. Na etapa anterior, o sistema monitorava temperatura e umidade em tempo real, acionando automaticamente um ventilador ou a irrigação. Agora, o desafio é tornar a solução **mais interativa e escalável**, permitindo que os agricultores acompanhem os dados **à distância**, em seus celulares ou laptops, e também controlem manualmente os dispositivos.

Para isso, a equipe decidiu:

- Manter a simulação no Wokwi como ambiente de testes, mas preparar o sistema para futura implementação em hardware real.

- Programar o ESP32 em MicroPython, permitindo a integração com serviços em nuvem.
- Configurar comunicação via MQTT para garantir que as informações cheguem rapidamente ao servidor e possam ser distribuídas a múltiplos clientes.
- Criar um dashboard no Blynk, onde o agricultor pode:
 1. Visualizar gráficos históricos de temperatura e umidade.
 2. Receber notificações quando valores saírem da faixa adequada.
 3. Acionar manualmente ventilador e irrigação.
 4. Alterar os parâmetros de acionamento automático (por exemplo, definir se a irrigação deve ser acionada a 35% ou 40% de umidade).

Você foi chamado(a) novamente pela **empresa** para contribuir nesta fase. Sua tarefa é:

1. **Configurar o Blynk** com widgets que permitam visualizar dados, controlar atuadores e ajustar parâmetros de forma interativa.
2. **Programar o ESP32 em MicroPython** para que ele não só leia sensores e acione atuadores, mas também receba comandos externos do dashboard.
3. **Testar no MQTTX** se os tópicos de publicação e subscrição estão funcionando corretamente.
4. **Implementar lógica de notificação**, em que o agricultor receba um alerta quando a temperatura ou a umidade atingirem níveis críticos.

Agora, propomos que você faça uma pausa e anote como poderia resolver a estes desafios. Depois, consulte as respostas.



Respostas do desafio



1. Como configuração do Blynk

Neste processo é necessário criar um **dashboard** com os seguintes widgets:

- **Gauge** ou **Display Value** → mostrar temperatura e umidade em tempo real.
- **Chart** → exibir o histórico dos dados coletados (ex: últimas 24h).
- **Button Switch** → ligar/desligar ventilador e irrigação manualmente.
- **Slider** → definir parâmetros de automação, como limite de umidade (ex: 35–50%).
- **Notification** → enviar alerta em caso de valores críticos.

Por fim, associar cada widget a um **Virtual Pin (Vx)** no Blynk, para que o ESP32 consiga publicar e receber dados.

1. Programação do ESP32 em MicroPython

Para este desafio, vamos seguir os passos:

- Conectar o ESP32 à rede Wi-Fi e ao broker MQTT.
- Ler os sensores de temperatura e umidade (ex: DHT22).
- Publicar periodicamente (ex: a cada 5 segundos) os dados nos tópicos MQTT.
- Assinar tópicos relacionados ao controle remoto dos atuadores (ex: estufa/ventilador/cmd).
- Implementar lógica para:
 - Acionar automaticamente atuadores quando sensores ultrapassarem limites.
 - Receber comandos do Blynk (via MQTT) e sobrescrever o modo automático.
 - Atualizar no Blynk os estados atuais (atuadores ligados/desligados).

Exemplo de estrutura em pseudocódigo:

```
</>
1  import dht
2  import machine
3  import time
4
5  sensor = dht.DHT22(machine.Pin(15))
6
7  while True:
8      temp = ler_sensor_temperatura()
9      umid = ler_sensor_umidade()
10
11      mqtt.publish("estufa/temperatura", temp)
12      mqtt.publish("estufa/umidade", umid)
13
14      if umid < limite_umidade or comando_irrigacao == "ON":
15          ligar_irrigacao()
16      else:
17          desligar_irrigacao()
18
19      if temp > limite_temperatura or comando_ventilador == "ON":
```

O próximo passo é no MQTXX. Publique ON ou OFF no tópico estufa/ventilador e o atuador deve responder de acordo.



REFLEXÃO

Depois de analisar o case e verificar as respostas, responda as questões.

- Quais vantagens e limitações você enxerga no uso de simuladores em comparação com dispositivos físicos?
- Explique como o protocolo MQTT facilita a comunicação entre o ESP32 e o MQTTX nesse projeto.
- Se esse protótipo fosse implantado em uma estufa real, quais outros sensores ou funcionalidades poderiam ser adicionados para tornar o sistema mais eficiente?
- Como o uso desse tipo de tecnologia pode impactar a gestão agrícola e a sustentabilidade no agronegócio?

Após fazer suas anotações, confira as sugestões de respostas que preparamos para você.



Sugestões de respostas



Como o uso do Wokwi pode auxiliar no aprendizado prático sem a necessidade imediata de hardware físico?

O Wokwi permite simular o ESP32, sensores e atuadores em um ambiente virtual, oferecendo uma experiência próxima ao mundo real. Isso elimina a barreira de custos e disponibilidade de componentes, facilitando que os estudantes testem ideias, errem e

ajustem sem medo de danificar equipamentos. Assim, o foco fica no entendimento da lógica, da programação e da integração entre dispositivos, criando uma base sólida antes do contato com o hardware físico.

Quais vantagens o protocolo MQTT oferece na comunicação entre dispositivos IoT nesse cenário?

O MQTT é um protocolo leve, eficiente e projetado para dispositivos com recursos limitados, como o ESP32. No contexto da estufa inteligente, ele permite:

- Comunicação **em tempo real** entre sensores e usuários.
- Organização dos dados em **tópicos**, o que facilita a escalabilidade do sistema.
- **Baixo consumo de rede**, ideal para aplicações de IoT.
- Facilidade para integrar diferentes aplicações e plataformas, como o MQTTX, dashboards ou serviços em nuvem.

De que forma esse protótipo pode ser expandido para aplicações reais em estufas inteligentes?

O protótipo pode ser ampliado em várias direções:

- **Adicionar novos sensores**, como de luminosidade, umidade do solo e CO₂, para enriquecer o monitoramento.
- **Automatizar outros atuadores**, como sistemas de irrigação ou cortinas de sombreamento.
- **Integrar com serviços em nuvem**, para armazenar dados históricos e gerar relatórios.
- **Implementar inteligência artificial**, permitindo prever necessidades da planta e otimizar os recursos de forma autônoma.
- **Acesso remoto via aplicativo**, para que o agricultor gerencie sua estufa de qualquer lugar.

Quais desafios de segurança da informação devem ser considerados em uma aplicação real como essa?

Em aplicações reais, a segurança é fundamental. Alguns desafios incluem:

- **Proteção do broker MQTT**, evitando acesso não autorizado aos dados e comandos.

- **Criptografia na transmissão (TLS/SSL)** para impedir interceptação de informações.
 - **Controle de autenticação e permissões**, garantindo que apenas usuários autorizados enviem comandos.
 - **Resiliência contra ataques de negação de serviço (DoS)**, que poderiam paralisar o sistema.
 - **Atualização e manutenção dos dispositivos**, para corrigir falhas e vulnerabilidades que possam surgir.
-



CURIOSIDADE

Você Sabia?

Que o **Wokwi** não é apenas um simulador visual: ele executa o código em um **interpretador real de MicroPython ou Arduino C++**, permitindo depuração e testes de bibliotecas de forma quase idêntica ao hardware físico?

Que o **ESP32 simulado no Wokwi** suporta **GPIOs, comunicação serial, I²C e SPI**, possibilitando a integração virtual de sensores como DHT22, displays OLED e até motores de passo?

Que diferente de outros simuladores, o Wokwi permite **integração direta com brokers MQTT**, tornando viável testar arquiteturas IoT antes da implementação em dispositivos reais?

Que o protocolo **MQTT (Message Queuing Telemetry Transport)** é leve, baseado em **publicação/assinatura (publish/subscribe)**, e roda sobre TCP/IP? Isso garante baixo consumo de banda e torna-o ideal para IoT em redes instáveis.

Que o **MQTTX** atua como cliente gráfico, mas também suporta **QoS (Quality of Service)**, **retenção de mensagens** e **persistência**, permitindo que o estudante experimente diferentes cenários de entrega de dados?

Que ao conectar o **Wokwi** com o **MQTTX**, você pode simular **fluxos completos de IoT**, como um sensor publicando dados para um broker em nuvem (ex: Mosquitto), enquanto múltiplos clientes (MQTTX, dashboards, outros ESP32) recebem e processam as informações?

| Ligando conceitos

Nesta unidade, nosso foco foi em compreender como funciona a arquitetura básica de um sistema IoT embarcado. Aprendemos o passo a passo, a simular circuitos utilizando o ESP32 no ambiente online Wokwi, que nos permite trabalhar com sensores e atuadores sem precisar de um protótipo físico. Também vimos como configurar a comunicação MQTT para que o ESP32 envie e receba dados, utilizando o cliente gráfico MQTTX. Colocamos tudo em prática, usando o Wokwi como simulador, o MQTTX para gerenciar as mensagens trocadas pelo protocolo MQTT e o MicroPython como linguagem de programação do ESP32 no ambiente simulado.

