

EDITAL DE PRÁTICA – AV2

COMPONENTE CURRICULAR:	Práticas Integradas: Camada de Serviço
ESPAÇO DE APRENDIZAGEM:	AVA + Sala do Eixo Integrativo
DISCENTE(S):	

Objetivos de Aprendizagem

- Codificação de Python para Banco de dados;
- Dispositivo de IoT acessando base de dados.

Conteúdos Formativos

- Uso do Git e GitHub para versionamento e entrega.
- Introdução à Banco de Dados para registro de leituras com timestamp.
- Uso de Python (Flask/FastAPI ou SQLite) como backend para IoT.

Equipamentos e Materiais

Descrição	Quantidade
Computador com acesso à internet	1
Conta no Wokwi (se não tiver, crie uma gratuitamente)	1
Conta no GitHub	
Arduino IDE	1
Callmebot	1

Equipamentos e Materiais

Descrição	Quantidade
ESP32	1
Sensor de Temperatura	1
Fotorresistor	1
Motor	2
Sensor de presença (PIR)	1
LED Verde	1
LED Vermelho	1

Introdução

Bem-vindo à atividade da Camada de Serviço. Neste guia, exploraremos a criação de um projeto de um robô explorador espacial.

- **O que é o projeto?**
 - Um robô explorador que busca **indícios de vida extraterrestre** em um planeta simulado.
 - O robô físico coleta dados de sensores e calcula a **probabilidade de vida**.
 - O controle remoto simulado envia comandos de movimentação.
- **Qual o papel de cada ESP32?**
 - **ESP32 no Wokwi (controle remoto):** recebe dados do **joystick** e **envia comandos** para o robô.
 - **ESP32 no laboratório (robô explorador):** **recebe comandos**, movimenta motores, lê sensores e calcula probabilidade de vida.
- **Por que isso é importante?**
 - Introduz versionamento com **GitHub**.
 - Permite armazenar dados em **banco de dados** para futuras análises.
 - Apresenta uso prático de **Python** em integração IoT.,
- **Nosso projeto:**
 - O robô calcula a probabilidade de vida com base nos sensores e algoritmo de aprendizado de máquina, sendo controlado remotamente.
 - A atividade terá 5 etapas principais:
 - Etapa 01: Controle remoto com ESP32 + Joystick no Wokwi.
 - Etapa 02: Robô físico em laboratório.
 - Etapa 03: Armazenamento dos Dados dos Sensores.
 - Etapa 04: Github.
 - Etapa 05: Apresentação do Projeto.

Descrições e Procedimentos

Etapa 01: Controle Remoto no Wokwi

Passo 1: Cadastrar conta no Wokwi

- Para esta atividade, será usado o Wokwi.
- Se já tiver conta, vá direto para o passo 2.
- Se não tiver:
 - Acesse <https://wokwi.com>
 - Clique em “Cadastre-se” no canto superior direito.
 - Preencha nome de usuário, e-mail e senha.
 - Clique em “Cadastre-se” para finalizar.
 - Confirme sua conta clicando no link enviado para o e-mail.
 - Volte ao site e faça login com seu usuário e senha.
 - Depois do login, você terá acesso ao painel do Wokwi, onde poderá criar e simular circuitos eletrônicos.

Passo 2: Montagem do Circuito Simulado

- Criem um projeto no Wokwi.
- Escolham a placa **ESP32**.
- Adicionem os seguintes componentes:
 - **Joystick Analógico**

- **LEDs Verde e Vermelho**
- **Interruptor/Botão** (para parada/controle do sistema)
- Façam as conexões de acordo com o esquemático e datasheet do ESP32 e dos componentes e pequenos programas testando as funcionalidades dos componentes

Passo 3: Programação do Circuito Simulado

- Criem o código do firmware do ESP32 para:
 - **Joystick (analógico):**
 - Movimentar o robô em 4 direções:
 - Frente → motores para frente.
 - Trás → motores para trás.
 - Esquerda → motor direito ativo.
 - Direita → motor esquerdo ativo.
 - O ESP32 deve ler os valores do joystick (eixo X e Y) e enviar os comandos ao robô.
 - **Botão remoto de desligar:**
 - Um botão no circuito simulado deve servir para **desligar remotamente** o robô físico.
 - Ao ser pressionado:
 - O controle remoto envia comando “DESLIGAR”.
 - O robô deve parar imediatamente os motores.
 - No monitor serial do controle remoto, deve aparecer: “Comando enviado: DESLIGAR”.
 - **LEDs no controle remoto (status de envio):**
 - **LED verde aceso:** controle remoto funcionando normalmente e conectado ao robô.
 - **LED vermelho aceso:** robô desligado ou desconectado.
 - Esses LEDs não simulam os do robô, mas indicam o estado da comunicação no controle remoto.
 - **Monitor Serial (controle remoto):**
 - Mostrar continuamente os comandos enviados (ex.: “Comando: Frente”, “Comando: Direita”).
 - Mostrar mensagem clara ao pressionar o botão de desligar.
 - Exibir se o robô está **ligado** (LED verde) ou **desligado** (LED vermelho).

Passo 4: Funcionamento do Circuito Simulado

- Simular o projeto no Wokwi.
 - Verificar se o projeto funciona corretamente.
 - Corrigir possíveis erros no código.

Passo 5 – GitHub

- Criar repositório no GitHub com nome sugerido: **iot-esp32-robo-espacial**.
- Adicionar:
 - Código do controle remoto (arquivo .ino).
 - Link do projeto no Wokwi.
 - Arquivo README.md com:
 - Nome do projeto.
 - Objetivo da etapa.
 - Lista dos componentes do circuito.
 - Como rodar no Wokwi.
- **Entregável:**

- Projeto funcionando no Wokwi (com joystick, botão e LEDs).
- Link do projeto no README do GitHub.
- Código do firmware versionado no GitHub.

Etapa 02: Robô físico em laboratório

Passo 5: Cadastrar conta no CallmeBot

- Acesse <https://www.callmebot.com>
- Se já tiver conta → vá para a Etapa 3 (criação do projeto).
- Se não tiver conta → siga os passos para vincular seu número de celular e obter a APIKEY:
 - Adicione o número +34 644 78 33 97 aos seus contatos do WhatsApp.
 - Envie a mensagem: "I allow callmebot to send me messages".
 - Aguarde a resposta: "API ativada para o seu número de telefone. Sua APIKEY é XXXXX".
 - Guarde a APIKEY enviada, pois ela será usada para enviar mensagens pela API.

Passo 6: Montagem do Circuito Físico em Laboratório

- Conecte todos os componentes ao ESP32 conforme o esquema definido:
 - **Motores (2):** responsáveis pela locomoção (um do lado esquerdo e outro do lado direito).
 - **Sensor de temperatura:** coleta dados do ambiente.
 - **Fotorresistor:** mede a luminosidade do ambiente.
 - **Sensor de presença (PIR):** detecta movimento em volta do robô.
 - **LED verde:** aceso quando o robô está ligado e operando normalmente.
 - **LED vermelho:** aceso quando o robô é desligado remotamente pelo controle ou entra em modo de alerta (probabilidade de vida > 75%).
- Certifique-se de:
 - Verificar a polaridade dos motores (direita/esquerda).
 - Testar as conexões com o multímetro ou checar no Monitor Serial.
 - Deixar os sensores bem-posicionados para captar sinais do ambiente

Passo 7: Programação do Circuito Físico em Laboratório

- O Crie o código do firmware do ESP32 para:
 - **Sensor de Temperatura e Umidade (DHT):**
 - O sensor deve medir temperatura em °C e umidade em %.
 - Mostrar leituras no Monitor Serial.
 - **Fotorresistor (luz):**
 - Medir a intensidade de luz do ambiente.
 - Mostrar valor no Monitor Serial.
 - **Sensor de Presença (PIR):**
 - Detectar movimento no ambiente.
 - Mostrar no Monitor Serial: "Presença detectada" ou "Sem presença".
 - **Monitor Serial**
 - A cada 2 segundos, o ESP32 deve enviar informações atualizadas:
 - Temperatura medida.
 - Umidade medida.
 - Intensidade da luz.
 - Estado do sensor de presença.
 - Estado do robô (ligado, desligado ou alerta).
 - Valor calculado da probabilidade de vida (%).
 - **Cálculo da Probabilidade de Vida**

- O ESP32 deve usar os valores dos sensores como entrada de um algoritmo simples de aprendizado de máquina. Exemplo de regras:
 - Temperatura entre 15 °C e 30 °C → +25%.
 - Umidade entre 40% e 70% → +25%.
 - Luz adequada (valor acima de um limite) → +20%.
 - Presença detectada → +30%.
- A soma desses fatores gera a probabilidade de vida (0% a 100%).
- Mostrar no Monitor Serial:
 - “Probabilidade de vida: XX%”.
- **Decisão com base na Probabilidade**
 - Se a probabilidade for $\leq 75\%$:
 - LED verde aceso.
 - Mostrar no Monitor Serial: “Exploração normal. Nenhum indício relevante detectado.”
 - Se a probabilidade for $> 75\%$:
 - LED vermelho aceso.
 - Mostrar no Monitor Serial: “ALERTA! Alta probabilidade de vida detectada!”.
 - Enviar mensagens via Callmebot (WhatsApp):
 - “Alerta! Alta probabilidade de vida detectada no planeta.”
- **Motores (DC)**
 - O robô deve ter 2 motores para locomoção:
 - Motor esquerdo.
 - Motor direito.
 - A direção é controlada pelos comandos recebidos do joystick.
- **LEDs**
 - LED verde aceso → quando o robô está funcionando normalmente.
 - LED vermelho aceso → quando o robô foi desligado remotamente ou quando a probabilidade de vida for superior a 75%.

Passo 11: Funcionamento do Circuito

- Testar o projeto completo no circuito físico.
 - Verificar se o projeto funciona corretamente.
 - Corrigir possíveis erros no código e montagem.
- **Entregável:**
 - **Robô físico funcionando (movimento, sensores e LEDs).**
 - **Cálculo da probabilidade de vida ativo.**
 - **Mensagem de alerta via Callmebot quando $> 75\%$.**

Etapa 03: Armazenamento dos Dados dos Sensores

Passo 8: Registro no Banco de Dados

- A cada 2 segundos, enviar dados via HTTP POST para o backend Python ou serviço REST, por exemplo:

```
{
  "timestamp": "2025-09-02T14:35:00Z",
  "temperatura_c": 24.3,
  "umidade_pct": 55,
```

```
"luminosidade": 723,  
"presenca": 1,  
"probabilidade_vida": 78.0  
}
```

- Conferir resposta do servidor.
- Caso falhe, armazenar em buffer e reenviar depois.

Passo 9: Integração com Python (Backend e Banco)

- Criar API em Python (Flask/FastAPI) com endpoint /leituras.
- Banco de dados recomendado: SQLite.
- Estrutura da tabela leituras:
 - id (chave primária).
 - timestamp.
 - temperatura_c.
 - umidade_pct.
 - luminosidade.
 - presenca.
 - probabilidade_vida.
- Adicionar rota GET /leituras para consultar últimas 100 leituras.
- **Entregável:**
 - **Leituras armazenadas no banco de dados.**

Etapa 04: Github

Passo 10 – GitHub

- Criar branch feat/robo-lab.
- Adicionar ao repositório:
 - Código do ESP32 do robô.
 - Scripts Python do backend.
 - Arquivo SQL de criação da tabela.
 - Atualizar README.md com instruções:
 - Como montar o robô.
 - Como rodar o backend Python.
 - Como consultar dados salvos.
- **Entregável:**
 - **Código e backend versionados no GitHub.**

Etapa 05: Apresentação do Projeto

Passo 9: Apresentação do Circuito Simulado e em Laboratório

- Preparar uma apresentação oral (5–10 minutos).
- Mostrar o circuito no **Wokwi** e no **laboratório real**.
- Explicar de forma simples e clara:
 - Como cada sensor e atuador foi usado.
 - Quais foram as dificuldades e como foram resolvidas.
- Repositório GitHub atualizado:
 - Código do controle remoto (Wokwi).
 - Código do robô físico (ESP32).

- Backend Python + banco de dados.
- README.md com instruções completas.
- **Entregável: Apresentação oral com demonstração prática e Repositório GitHub.**

Cronograma			
Nº	Etapas	Entregável	Prazo
01	01	Circuito Simulado	Final do encontro
02	02	Circuito em Laboratório e	Final do encontro
03	03	Banco de Dados	Final do encontro
04	04	Githb	Final do encontro
05	05	Apresentação oral	Final do encontro

Avaliação		
1	Etapa 01	1,0
2	Etapa 02	1,0
3	Etapa 03	4,0
4	Etapa 04	2,0
5	Etapa 05	2,0
Total		10 pontos

Referências

ARDUINO. *Arduino IDE*. Disponível em: <https://www.arduino.cc/en/software>. Acesso em: 31 ago. 2025.

ESPRESSIF SYSTEMS. *ESP32 technical reference manual*. Shanghai: Espressif, 2022.

RANDOM NERD TUTORIALS. *ESP32 tutorials and projects*. Disponível em: <https://randomnerdtutorials.com/>. Acesso em: 31 ago. 2025.

WOKWI. *Online ESP32 and Arduino simulator*. Disponível em: <https://wokwi.com/>. Acesso em: 31 ago. 2025