



# **TRABALHO PRÁTICO 1**

## **Raspberry Pi Pico**

**Infraestruturas de Tecnologia de Informação e Segurança**  
**Mestrado em Engenharia Informática – Internet das Coisas**

**Trabalho realizado por:**

**Romão Filipe Nº20923**

**Frederico Carneiro Nº24391**

**Jonatas Ribeiro Nº26099**

## ÍNDICE

Introdução .....	3
Objetivo .....	3
Thonny e Python .....	4
End-Point: .....	4
Web server: .....	5
HiveMQ - MQTT Público .....	5
Funcionamento Básico .....	5
Conclusão .....	7

## Introdução

O avanço contínuo da Internet das Coisas (IoT) tem transformado a maneira como interagimos com dispositivos eletrónicos no nosso quotidiano. Neste contexto, o presente projeto destaca-se ao explorar as capacidades do Raspberry Pi Pico W, um microcontrolador poderoso e versátil, para criar um sistema web interativo.

O Raspberry Pi Pico W desempenha o papel central como um servidor web, proporcionando uma experiência de utilizador única. Utilizando MQTT (Message Queuing Telemetry Transport), estabelece-se uma comunicação eficiente e bidirecional, permitindo aos utilizadores controlar o estado do LED interno do Pico W e monitorizar a temperatura interna remotamente.

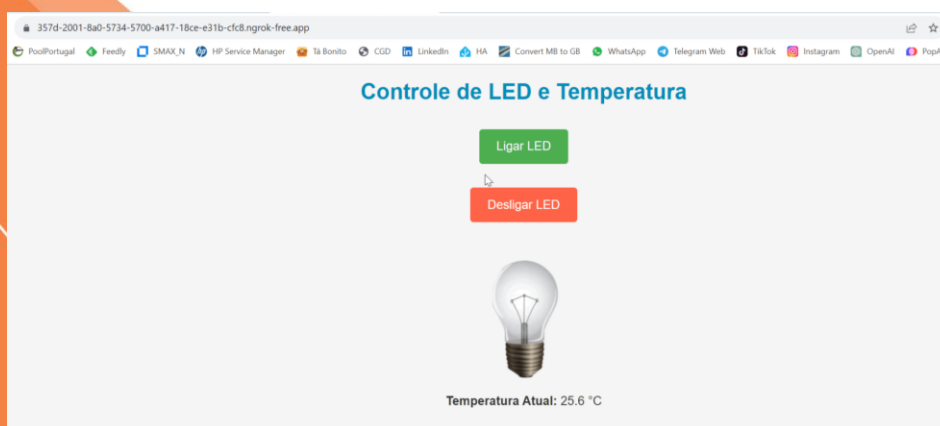
Ao longo deste trabalho, iremos explorar os fundamentos da implementação, desde a configuração do servidor web até à integração de MQTT para uma comunicação eficaz. Adicionalmente, serão discutidas as implicações práticas para tornar o web server acessível globalmente, sem a necessidade de configurações complexas de rede.

Este projeto não apenas demonstra a capacidade de criação de um sistema de IoT funcional, mas também realça a importância da acessibilidade global e da comunicação eficiente na conceção de soluções inovadoras. Vamos explorar em detalhes as etapas e os desafios encontrados durante este processo, contribuindo assim para uma compreensão mais profunda da aplicação prática dos conceitos de IoT no mundo real.

## Objetivo

O objetivo central deste projeto é destacar a integração harmoniosa de várias tecnologias para criar um sistema de controlo remoto utilizando o Raspberry Pi Pico W. Pretende-se explorar não apenas a implementação técnica, mas também proporcionar uma compreensão clara dos benefícios e desafios associados à utilização destas ferramentas no contexto específico do Raspberry Pi Pico W.

Com foco na simplicidade e eficiência, foram utilizadas ferramentas específicas que permitem uma implementação acessível mesmo para iniciantes.



*Figura 1- Página Web para controlar Raspberry Pico W*

## Thonny e Python

A escolha do ambiente de desenvolvimento Thonny, associado à linguagem Python, foi motivada pela facilidade de utilização e pela vasta comunidade de apoio. Esta combinação proporcionou um desenvolvimento rápido e eficaz, permitindo uma abordagem pragmática na implementação das funcionalidades desejadas.

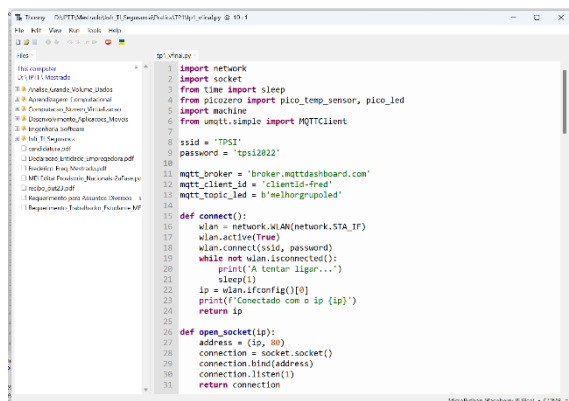


Figura 2 - Ambiente de desenvolvimento Thonny

O desenvolvimento do web server e controlo do end-point foi realizado através da linguagem de programação Python e da IDE Thonny, proporcionando uma abordagem eficiente e acessível. A construção da página web foi executada exclusivamente em HTML.

Para alcançar os resultados almejados, implementamos diversos pontos fundamentais, detalhados a seguir:

### End-Point:

Um end-point é um dispositivo que se conecta a uma rede de computadores, servindo como ponto de comunicação entre um dispositivo remoto e a rede à qual está vinculado. Pode ser físico, como computadores desktop, servidores, ou dispositivos móveis, ou lógico, como pontos de conexão em software, como API end-points. A segurança dos end-points é crucial, pois podem representar pontos vulneráveis à invasão. No nosso projeto, o end-point é o Raspberry Pi Pico, equipado com sensor de temperatura e um indicador LED de estado.

### Descrição do Raspberry Pi Pico:

O Raspberry Pi Pico é uma placa microcontrolador de alto desempenho, incorporando o chip RP2040.

**Processador:** Dispõe de dois núcleos ARM Cortex-M0+ operando a 133MHz.

**Memória:** Apresenta 264kB de SRAM interna distribuída em seis bancos independentes.

**Armazenamento:** Suporta até 16MB de memória Flash externa através de uma interface QSPI dedicada.

**Conectividade:** Oferece interfaces digitais versáteis para conexão com sensores, atuadores e outros microcontroladores.

**Documentação:** A Raspberry Pi Foundation fornece documentação abrangente, incluindo guias de uso, exemplos de projetos e informações técnicas detalhadas.

**Compatibilidade:** Suporta várias linguagens de programação, como MicroPython e C/C++.

**Aplicações:** Pode ser empregado em diversos projetos, incluindo automação residencial, robótica, Internet das Coisas (IoT), dispositivos vestíveis, entre outros.

Este projeto destaca não apenas a implementação técnica, mas também a escolha criteriosa de ferramentas e tecnologias para atingir os objetivos de forma eficaz e funcional.

## Web server:

Um servidor web tem o propósito de hospedar e fornecer páginas da web para os utilizadores. Ele recebe solicitações de clientes (navegadores) e envia as páginas correspondentes como resposta. O servidor web também pode processar formulários, executar scripts e fornecer recursos adicionais, como bases de dados e serviços de autenticação. Ele desempenha um papel fundamental na disponibilização de conteúdo na internet.

```
ipt_vfnal.py
53 def webpage(temperature, state):
54     html = """
55     <!DOCTYPE html>
56     <html lang="en">
57
58     <head>
59     <meta charset="UTF-8">
60     <meta name="viewport" content="width=device-width, initial-scale=1.0">
61     <title>Controle de LED e Temperaturas</title>
62     <style>
63     body {
64         font-family: 'Arial', sans-serif;
65         background-color: #f5f5f5;
66         text-align: center;
67         margin: 20px;
68         color: #333;
69     }
70
71     h1 {
72         color: #008CBA;
73     }
74
75     form {
76         margin-top: 20px;
77         display: flex;
78         flex-direction: column;
79         align-items: center;
80     }
81
82     input[type="submit"] {
83         padding: 15px 25px;
84         font-size: 18px;
85         border: none;
86         border-radius: 5px;
87         cursor: pointer;
88         margin-top: 10px;
89     }
90     """
91     return html
```

Figura 3 - Código HTML para criação da página

Neste projeto, embarcámos o código do web server dentro do próprio end-point, pois a finalidade é somente académica. Obviamente, este dispositivo (Raspberry pi pico) possui grande versatilidade e recursos, porém, é limitado. Num projeto IoT de larga escala, será necessária a implementação de um server com grande capacidade de resposta as requisições web, feitas por diversos dispositivos ao mesmo tempo.

## HiveMQ - MQTT Público

A integração do protocolo MQTT foi essencial para estabelecer uma comunicação eficaz entre o Raspberry Pi Pico W e a interface web remota. A opção por utilizar o servidor público HiveMQ, com a sua fiabilidade e robustez reconhecidas, permitiu uma implementação fluida da subscrição de tópicos, possibilitando assim o controlo remoto do dispositivo.

Um broker MQTT é uma peça-chave na arquitetura MQTT, agindo como um intermediário entre quem envia mensagens (clientes) e quem as recebe (assinantes). A sua função é garantir uma comunicação eficiente e confiável entre os dispositivos conectados.

### Funcionamento Básico:

**Conexão:** Os clientes MQTT se conectam ao broker por meio de uma conexão TCP/IP. O broker fica "ouvindo" em uma porta específica para aceitar as conexões dos clientes.

**Publicação:** Um cliente pode enviar mensagens para um tópico específico no broker. Essas mensagens podem conter diferentes tipos de informações, como dados de sensores, comandos ou atualizações de estado.

Assinatura: Assinantes MQTT podem se inscrever em tópicos específicos no broker, indicando seus interesses. O broker então direciona as mensagens publicadas nesses tópicos para os assinantes correspondentes.

Roteamento: O broker é responsável por encaminhar as mensagens certas para os assinantes corretos. Ele mantém um registo dos tópicos aos quais cada cliente está inscrito, garantindo que as mensagens sejam entregues apenas aos interessados.

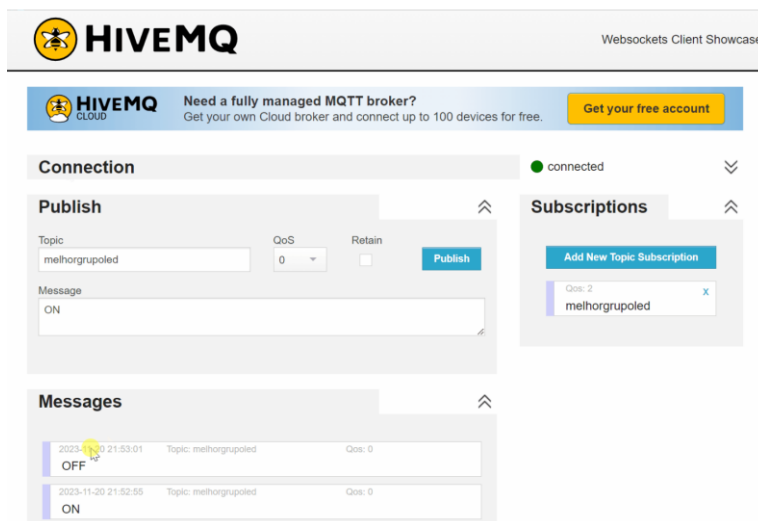


Figura 4- Hive - Servidor público MQTT

## NGROK para Exposição Pública

O recurso ao NGROK foi fundamental para tornar o servidor web acessível publicamente, eliminando as barreiras associadas ao encaminhamento de portas. A utilização de uma conta gratuita no NGROK revelou-se suficiente para os objetivos deste projeto, destacando assim a acessibilidade e a simplicidade da solução.



Figura 5 - Aplicação NGROK

Ngrok é um aplicativo multiplataforma que permite que os desenvolvedores exponham um servidor de desenvolvimento local para a Internet com o mínimo de esforço. É usado principalmente para testar e depurar aplicativos da web durante o desenvolvimento. Com o Ngrok, é possível criar um túnel seguro para o servidor local e aceder remotamente através de um URL público. Isso é útil quando é necessário partilhar temporariamente um aplicativo com outras pessoas ou integrar serviços externos que exigem um URL acessível publicamente.



O Ngrok é independente do ambiente, o que significa que pode entregar tráfego para serviços em execução em qualquer lugar, sem alterações na rede. No contexto do nosso projeto, o ngrok serve como um mecanismo para expor o endereço de IP local de nosso web server para a web (externo), e assim, possibilitar a gestão das requisições web fora da rede local.

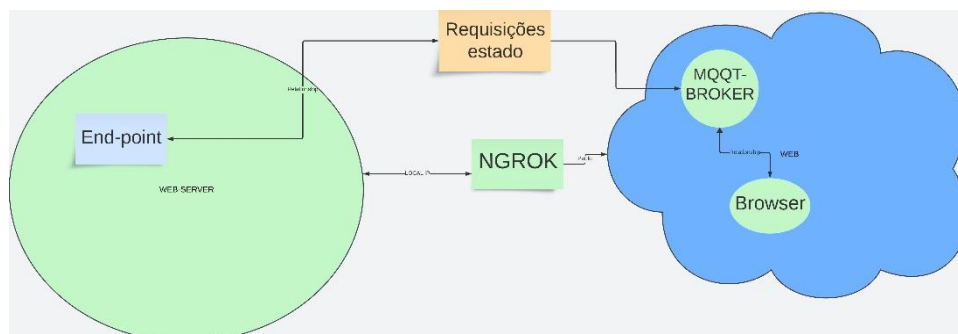


Figura 6 - Diagrama

## Conclusão

A realização deste trabalho académico proporcionou uma imersão profunda no mundo da automação e conectividade de dispositivos, com foco especial em uma aplicação prática envolvendo o Raspberry Pi Pico como web server controlado por MQTT.

Durante todo o processo, desde a programação do web server até a interação com o Hive MQTT e a utilização do NGROK para disponibilizar o servidor de forma pública, os desafios foram superados com uma abordagem rigorosa e inovadora.

A escolha do Thonny como ambiente de desenvolvimento Python revelou-se acertada, proporcionando uma experiência fluida e eficiente na criação do código necessário para o funcionamento integrado do sistema. O Hive MQTT, enquanto broker de mensagens, desempenhou um papel fundamental, permitindo uma comunicação estável e eficaz entre o web server e o dispositivo remoto.

A utilização do Raspberry Pi Pico como end-point, com seu sensor de temperatura e LED de estado, adicionou uma camada de complexidade que enriqueceu significativamente o projeto. A descrição detalhada deste componente, desde suas características técnicas até suas aplicações práticas, contribuiu para uma compreensão aprofundada da infraestrutura subjacente.

Este trabalho proporcionou não apenas o desenvolvimento de habilidades técnicas, mas também uma apreciação mais profunda das complexidades e considerações práticas envolvidas na criação de sistemas conectados. A compreensão adquirida sobre a importância de elementos como endpoints, brokers MQTT e escolha cuidadosa de hardware é valiosa para a futura aplicação em projetos mais amplos de automação.

Em suma, a conclusão bem-sucedida deste projeto valida não apenas a capacidade de aplicar conceitos teóricos, mas também a habilidade de enfrentar desafios práticos de maneira sistemática. Este trabalho serve como uma sólida base para explorações futuras no domínio da automação industrial e conectividade de dispositivos.