

Projetos de Sistemas Ubíquos

Ambiente de Desenvolvimento CoAP

Patrick Davila Kochan

Araranguá,

08 de julho de 2019

**SUMÁRIO**

[**FERRAMENTAS**](#_wq22qljk5bdx) **3**

[Ferramentas obrigatórias](#_zg1vu78p9xze) 3

[Ferramentas opcionais](#_e5bf2wn8awe5) 3

[**INSTALAÇÃO DO AMBIENTE**](#_dje9s4hw0fu9) **4**

[**RECOMENDAÇÕES**](#_f7e2ybddka5h) **7**

[**LINKS PARA DOWNLOAD**](#_ipnn51spmpbw) **8**

# FERRAMENTAS

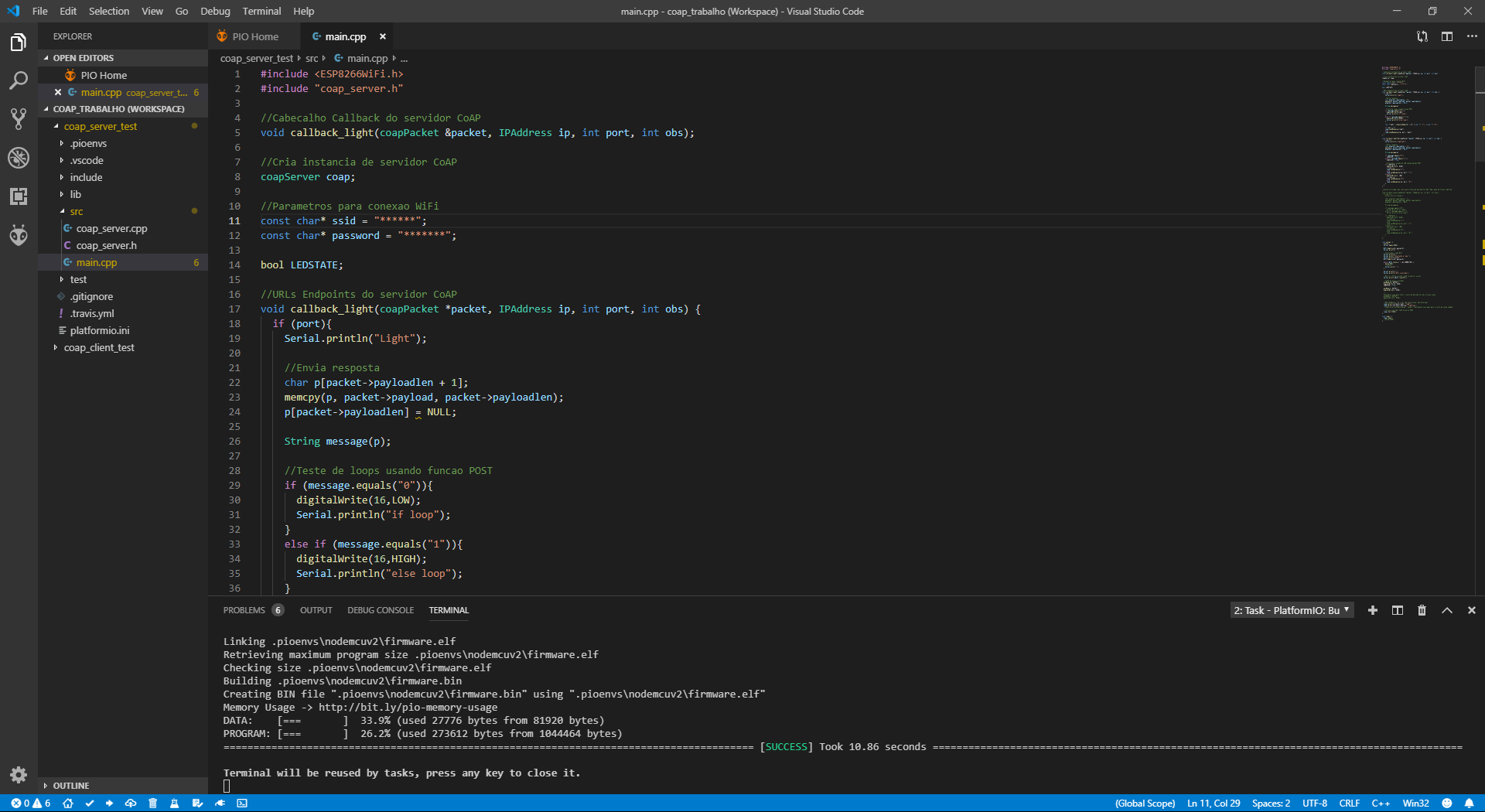
## Ferramentas obrigatórias

Todas as ferramentas listadas possuem seus *links* de *downloads* disponibilizados no final desse breve manual.

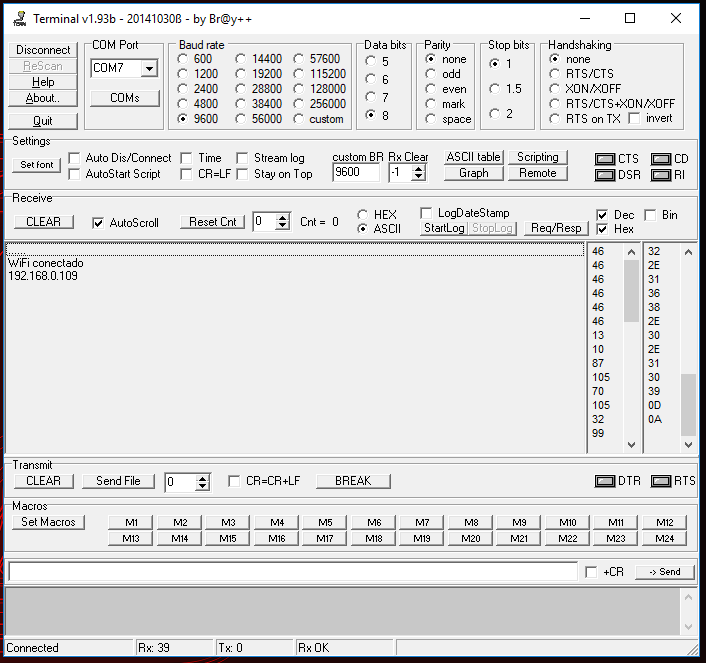
* Visual Studio Code: editor de código fonte desenvolvido pela Microsoft;
* PlatformIO: ecossistema para desenvolvimento de aplicações de internet das coisas (IoT). É uma extensão para Visual Studio Code;

## Ferramentas opcionais

* Terminal: emulador de monitor de porta serial. Esse programa apresenta maior flexibilidade do que os monitores de porta serial oferecidos pelo PlatformIO;
* Copper (Cu): uma extensão para Google Chrome que funciona como cliente CoAP. Muito didática e excelente para o bom entendimento do protocolo.



*Figura 01:* Visual Studio Code com integração ao PlatformIO*.*

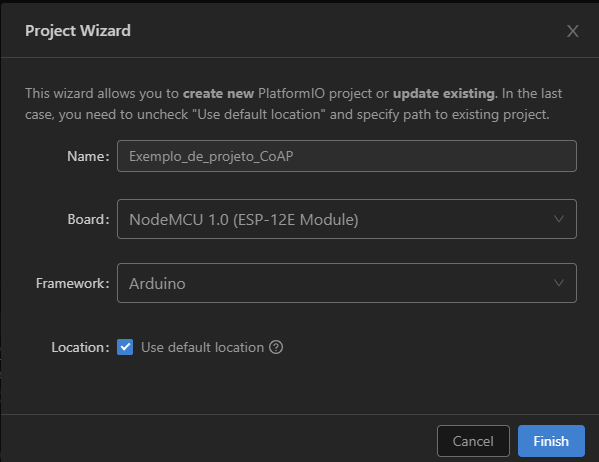


*Figura 02:* Terminal exibindo o endereço IP recebido através da porta serial (COM7)*.*

# INSTALAÇÃO DO AMBIENTE

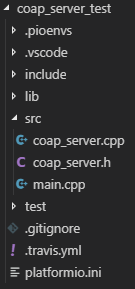
Este projeto foi desenvolvido utilizando o Windows como Sistema Operacional e C++ como linguagem de programação, visando facilidades e coerência com as atribuições adquiridas ao longo do curso de Engenharia de Computação. As instalações das ferramentas são simples e muito intuitivas, no entanto, alguns pontos de observação são importantes e levantados aqui.

Instale o Visual Studio Code e então o PlatformIO. Na criação de um novo projeto no PlatformIO, as seguintes configurações devem ser utilizadas para a correta execução do projeto:



*Figura 03:* Configurações utilizadas na criação do projeto no PlatformIO*.*

Após a criação do projeto, acesse os códigos fontes com alterações (disponíveis na pasta “[Materiais CoAP](https://drive.google.com/drive/folders/1Loa90dNF6ixwq7npghHjjK9Kub5feU3E?usp=sharing)” do Google Drive) ou o Projeto Base de Código Fonte no *link* disponibilizado ao final deste documento. Dentro da pasta “src” do projeto, crie os três arquivos mostrados na Figura 04. Os código dos arquivos “coap\_server.cpp” e “coap\_server.h” não foram alterados. O código do arquivo “main.cpp” possui alterações, mas corresponde ao arquivo “coapserver.ino” do Projeto Base. Basta copiar os códigos desejados e inseri-los nos seus respectivos arquivos.



*Figura 04:* Estrutura de pastas e arquivos do projeto do servidor CoAP no Visual Studio Code*.*

Com o projeto já pronto, pode-se observar uma barra azul no canto inferior da tela, no Visual Studio Code, representada pela Figura 05. Essa barra é da extensão PlatformIO e oferece diversas opções. Dentre elas, as duas mais importantes são: *Build* (compila o código) e *Upload* (envia o código para o dispositivo, no nosso caso, o ESP 8266). Conecte o ESP ao seu computador, use a opção *Build* e em seguida *Upload* para enviar seu código ao dispositivo.



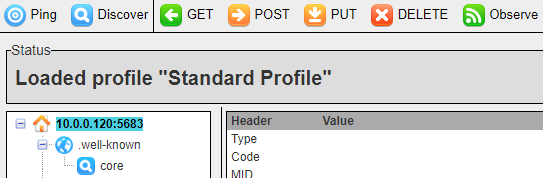
*Figura 05:* Opção *Build,* contornada na cor verde e opção *Upload*, em vermelho*.*

Dentre as ferramentas opcionais o único ponto que deve ser destacado é o Copper (Cu). Realize o *download* e instalação da extensão para Google Chrome seguindo as orientações dispostas no [projeto do GitHub](https://github.com/mkovatsc/Copper4Cr). Quando a extensão for aberta, solicitará uma URL. Adicione neste campo o endereço IP do seu servidor.



*Figura 06:* Janela desolicitação realizada pelo Copper (Cu).

Após adicionar o IP ao campo de URL, a extensão oferece as funções de descoberta, GET, POST etc. Utilize primeiramente a ferramenta de descoberta para obter a lista de funções disponíveis em seu servidor. Após isso, basta selecionar a função desejada e utilizar os métodos almejados.



*Figura 07:* Métodos oferecidos pela extensão Copper (Cu) para Google Chrome*.*

# RECOMENDAÇÕES

Existem muitas informações sobre o CoAP, no entanto esse grande volume não vem de fontes confiáveis e podem gerar muitas dúvidas sobre o protocolo, aumentando em muito a curva de aprendizado.

Ao decorrer do desenvolvimento deste trabalho, excelentes materiais foram encontrados e todos eles estão disponíveis neste link:

[**https://drive.google.com/drive/folders/1Loa90dNF6ixwq7npghHjjK9Kub5feU3E?usp=sharing**](https://drive.google.com/drive/folders/1Loa90dNF6ixwq7npghHjjK9Kub5feU3E?usp=sharing)

Poderosíssimas fontes de informação sobre o Constrained Application Protocol podem ser encontradas nas referências do artigo em desenvolvimento da matéria de Projetos de Sistemas Ubíquos que tratam de protocolos de IoT, inclusive o CoAP.

# LINKS PARA DOWNLOAD

Projeto Base do Código Fonte:

* <https://github.com/automote/ESP-CoAP>

Ferramentas:

* <https://code.visualstudio.com/download>
* <https://platformio.org/install/ide?install=vscode>
* <https://github.com/mkovatsc/Copper4Cr>
* <https://hw-server.com/terminal-terminal-emulation-program-rs-232>