

# Programando ESP32 com Atom Text Editor e PlatformIO IDE

<https://randomnerdtutorials.com/programming-esp32-with-atom-text-editor-and-platformio-ide/>

Em nossos [projetos e tutoriais ESP32](#), recomendamos o uso do Arduino IDE para programar a placa de desenvolvimento ESP32. No entanto, em alguns computadores Windows, é um pouco complicado instalar o complemento ESP32 usando a GUI do Git devido a erros de permissão, à versão do Arduino IDE ou a várias instalações do Arduino IDE.



Há também outro método popular para programar as placas de desenvolvimento ESP32 usando o [editor de texto Atom](#) combinado com o [PlatformIO IDE](#). Com esse método, você ainda pode usar a mesma linguagem de programação usada no Arduino IDE.

Este próximo tutorial foi testado em um PC com Windows 10 e em um computador com Mac OS X.

## 1. Instalando o Atom Text Editor

O primeiro passo é ir ao site do [Atom.io](https://atom.io/) e baixar o editor de texto gratuito. <https://atom.io/>

Depois disso, abra o arquivo de instalação baixado e execute-o. A instalação é bastante simples. Complete as instruções na tela para concluir a instalação do Atom.

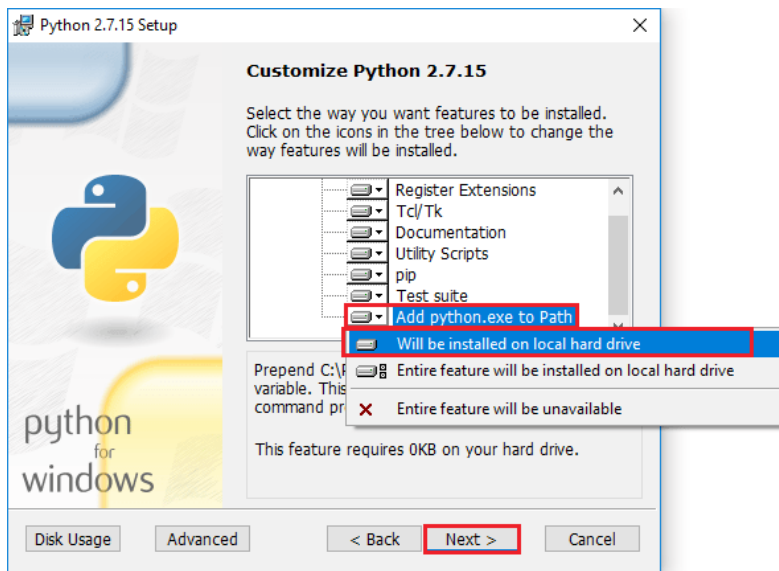
## 2a. Instalando o Python 2.7.X

Para usar o PlatformIO IDE e programar suas placas ESP32, você deve ter o Python 2.7.X instalado no seu computador. Vá para a página de [downloads do Python](#) e baixe a versão mais recente do Python 2.7.X para o seu sistema operacional (sistema operacional). <https://www.python.org/downloads/>

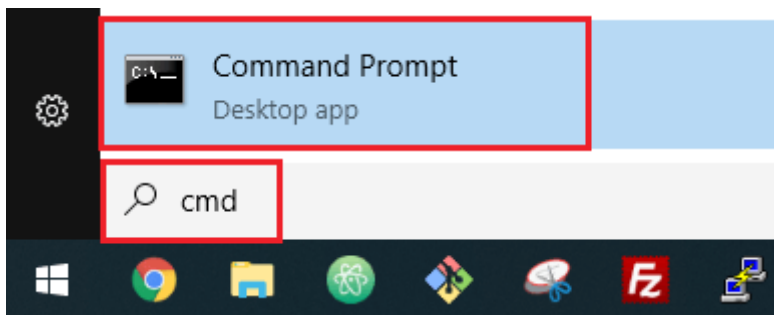
Release version	Release date	Click for more	
<a href="#">Python 3.7.0</a>	2018-06-27	<a href="#">Download</a>	<a href="#">Release Notes</a>
<a href="#">Python 3.6.6</a>	2018-06-27	<a href="#">Download</a>	<a href="#">Release Notes</a>
<a href="#">Python 2.7.15</a>	2018-05-01	<a href="#">Download</a>	<a href="#">Release Notes</a>
<a href="#">Python 3.6.5</a>	2018-03-28	<a href="#">Download</a>	<a href="#">Release Notes</a>
<a href="#">Python 3.4.8</a>	2018-02-05	<a href="#">Download</a>	<a href="#">Release Notes</a>
<a href="#">Python 3.5.5</a>	2018-02-05	<a href="#">Download</a>	<a href="#">Release Notes</a>
<a href="#">Python 3.6.4</a>	2017-12-19	<a href="#">Download</a>	<a href="#">Release Notes</a>

Nota: para esta unidade, usamos o Python 2.7.15. Qualquer outra versão do Python 2.7.X também deve funcionar. Abra o arquivo baixado para iniciar o assistente de instalação do Python. Durante o passo 2, siga estas próximas instruções:

- Role para baixo pela janela " Personalizar Python 2.7.15 ";
- Abra o " Adicionar python.exe ao caminho ";
- E selecione a opção " Será instalado no disco rígido local ".



Depois disso, pressione o botão " **Next** " para concluir a instalação. Depois de instalar o Python 2.7.X, você precisa abrir o " **Prompt de Comando** ":



Execute a próxima sequência de comandos para verificar a versão do Python e do pip instalada:

```
python --version  
Python 2.7.15  
pip --version  
pip 9.0.3
```

Ambos os comandos devem retornar uma saída similar (a versão pode ser mais diferente no seu caso). Depois disso, verifique se você possui o **virtualenv** instalado:

```
virtualenv --version
```

Se já estiver instalado, você pode ir para a próxima seção. Caso contrário, você precisa instalá-lo com este comando:

```
pip install virtualenv
```

Depois disso, execute este comando novamente para verificar se **virtualenv** foi instalado corretamente:

```
virtualenv --version  
16.0.0
```

### 3. Instalando o Clang para conclusão de código

O PlatformIO IDE usa o Clang para a conclusão do código inteligente. Para verificar se o Clang está disponível em seu sistema, abra o Terminal / Command Prompt e execute:

```
clang --version
```

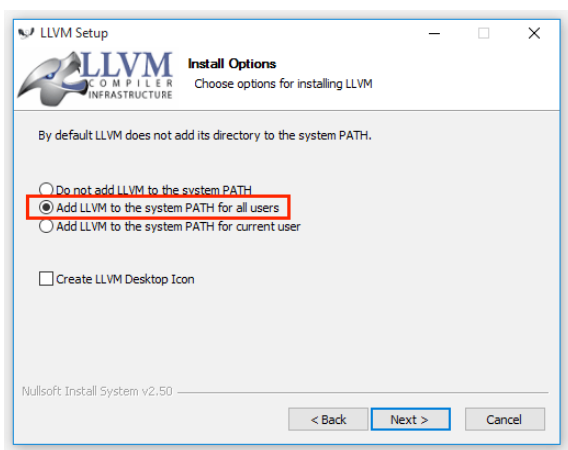
Se o clang não estiver instalado, instale-o seguindo as instruções do seu sistema operacional:

**Linux: using package managers: apt-get install clang or yum install clang.**

•Windows: baixe o [Clang 3.9.1 para Windows](#) . Selecione a opção “ Adicionar LLVM ao sistema PATH ” durante a etapa de instalação mostrada na imagem abaixo.

•[Clang 3.9.1 para Windows \(32 bits\)](#) <http://releases.llvm.org/3.9.1/LLVM-3.9.1-win32.exe>

•[Clang 3.9.1 para Windows \(64 bits\)](#) <http://releases.llvm.org/3.9.1/LLVM-3.9.1-win64.exe>

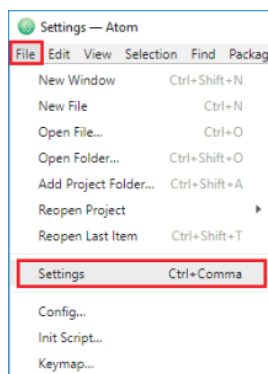


**Aviso: NÃO INSTALE O CLANG 4.0, APENAS O CLANG 3.9 ESTÁ APOIADO NO MOMENTO.**

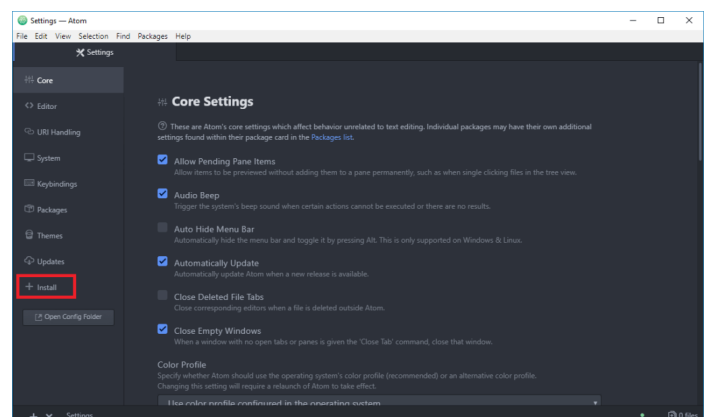
### 4. Instalando o PlatformIO IDE no Atom

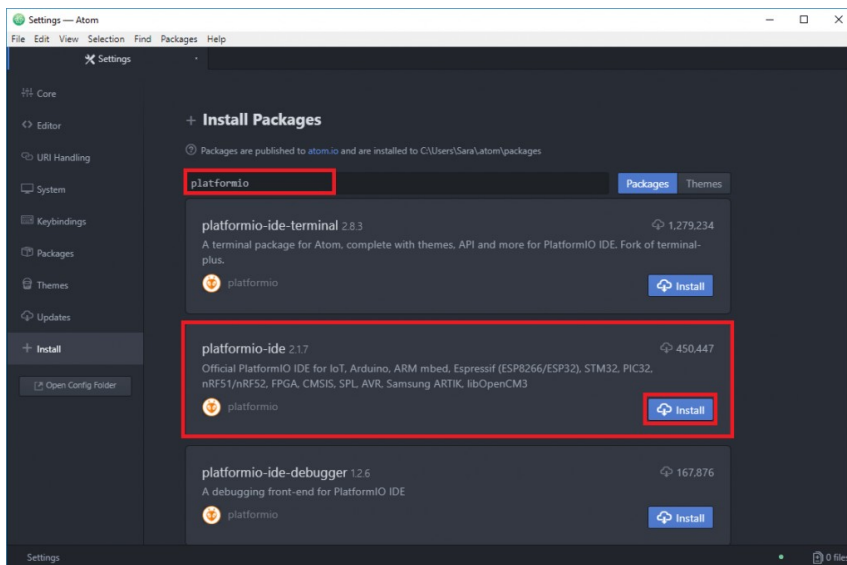
pip install -U platformio (linux)

Windows. Depois de instalar todas as dependências do PlatformIO IDE, abra o editor de texto Atom e vá para **Arquivo > Configurações** :



No menu à esquerda, abra a guia " **Instalar** ":



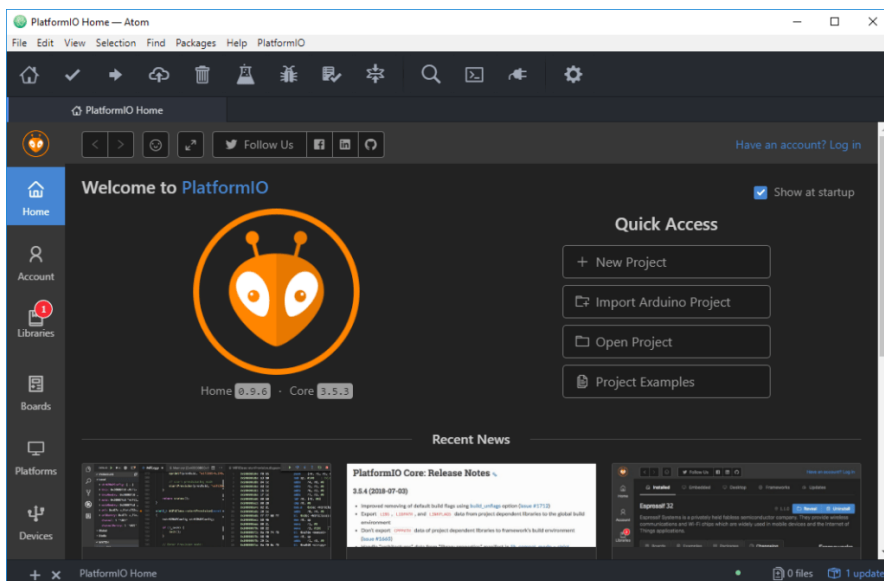


Procure por “ **platformio** ” e pressione a tecla Enter / Return:

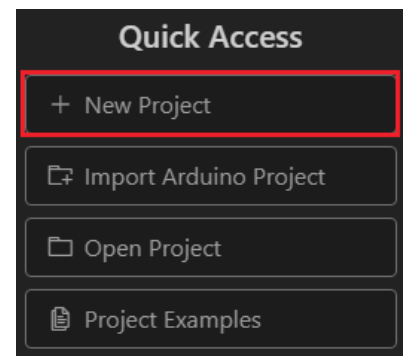
Instale a opção “ **platformio-ide** ” destacada na imagem anterior. Após a conclusão da instalação, reinicie o editor de texto Atom para que as alterações entrem em vigor.

## 5. Visão Geral do PlatformIO IDE

Agora, quando você abrir o editor de texto Atom, uma nova janela deverá ser carregada com a tela “Welcome to PlatformIO”:



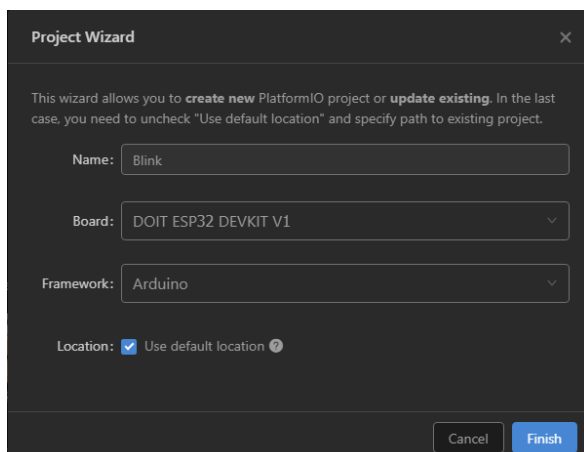
Pressione o botão “ **Novo projeto** ” no menu de acesso rápido:



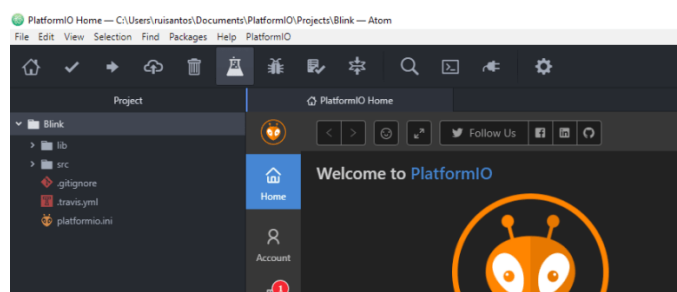
Uma nova janela carrega que permite criar um novo projeto para sua placa, siga estes próximos passos:

- Nomeie seu projeto (exemplo: Blink);

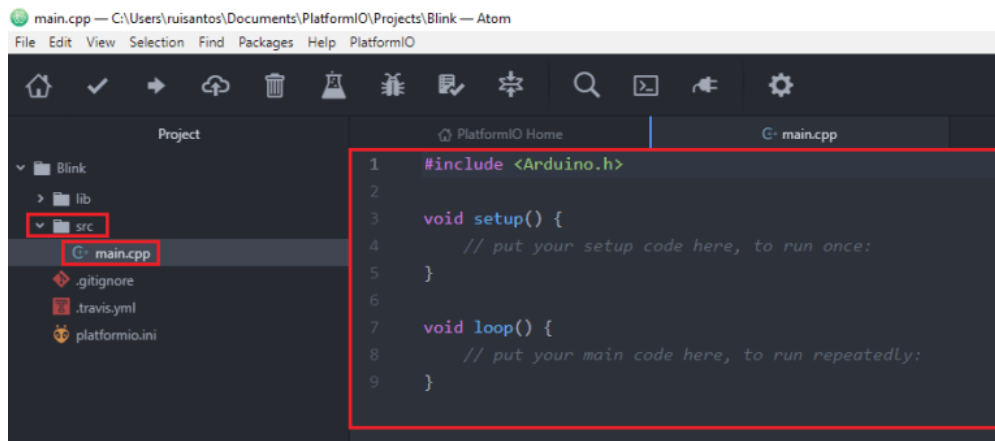
- Procure por “ ESP32 ” e selecione sua placa ESP32 (exemplo: DOIT ESP32 DEVKIT V1);
- Selecione o framework Arduino ;
- Pressione o botão “ Finish ”.



Depois que o novo projeto for criado, você verá a pasta do projeto no menu à esquerda que pode ser usada para navegar pelos arquivos e pastas.



Abra a pasta **src** e clique duas vezes no arquivo **main.cpp** para abri-lo. Uma nova janela é aberta no Atom com esse arquivo, para que você possa editá-lo:



O arquivo *main.cpp* é como o seu arquivo *Blink.ino* usado no Arduino IDE. Você pode escrever seu código do Arduino, mas precisa começar com o arquivo incluindo a estrutura do Arduino. Portanto, basicamente todos os skets do Arduino funcionam com o PlatformIO IDE, se você iniciar o sket com esta linha, a seguinte linha:

```
#include <Arduino.h>
```

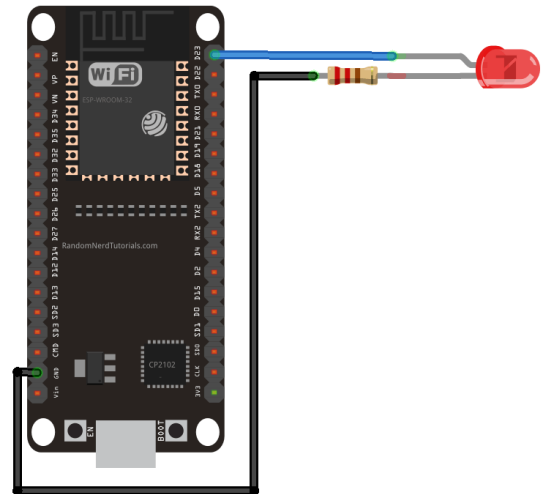
## 6. PlataformIO IDE de teste

Vamos tentar um exemplo para testar o PlataformIO IDE. Vamos piscar um LED conectado ao **GPIO23**. Aqui está a lista de peças que você precisa para seguir este exemplo:

- Placa de desenvolvimento ESP32
- LED de 5mm
- Resistor de 330 Ohm
- Breadboard
- Fios de ligação

Aqui está um sket para fins de teste que pisca o LED:

```
#include <Arduino.h>
// ledPin refers to ESP32 GPIO 23
const int ledPin = 23;
// the setup function runs once when you press reset or power the board
void setup() {
    // initialize digital pin ledPin as an output.
    pinMode(ledPin, OUTPUT);
}
// the loop function runs over and over again forever
void loop() {
    digitalWrite(ledPin, HIGH); // turn the LED on (HIGH is the voltage level)
    delay(1000);                // wait for a second
    digitalWrite(ledPin, LOW);  // turn the LED off by making the voltage LOW
    delay(1000);                // wait for a second
}
```

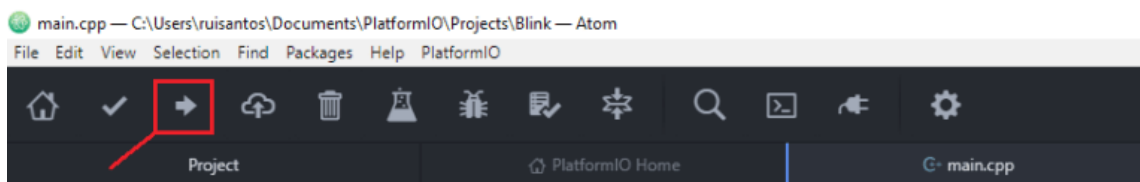


disonivel em:

[https://github.com/RuiSantosdotme/ESP32-Course/raw/master/code/Blink\\_LED/Blink\\_LED\\_PlatformIO.ino](https://github.com/RuiSantosdotme/ESP32-Course/raw/master/code/Blink_LED/Blink_LED_PlatformIO.ino)

Copie o código para o editor de texto Atom e siga os próximos passos para fazer o upload do código para o seu painel ESP32:

1. Conecte sua placa ESP32 ao seu computador;
2. Salve o sket recém-criado ( Arquivo > Salvar );
3. Pressione o botão “ Upload ” (destacado na próxima imagem).



```
platformio run --target upload

Scanning dependencies...
No dependencies
Configuring upload protocol...
Looking for upload port...
Auto-detected: COM4
Uploading .pioenvs\esp32doit-devkit-v1\firmware.bin
esptool.py v2.1-beta1
```

Após o upload do sketch, o seu ESP32 deve estar piscando o LED conectado ao **GPIO 23** a cada 1 segundo.  
É isso aí! O PlatformIO foi instalado com sucesso e você pode usá-lo para programar sua placa ESP32.

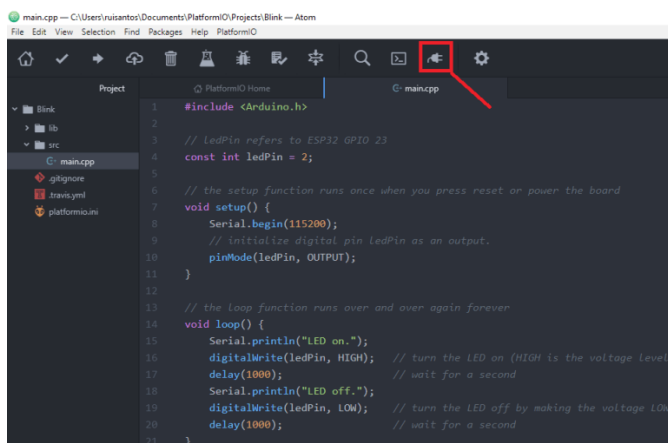
## 7. Dicas Adicionais do PlatformIO IDE

Acabamos de arrancar a superfície do que o PlatformIO IDE pode fazer. Aqui está o que cada botão no PlatformIO IDE faz / significa:

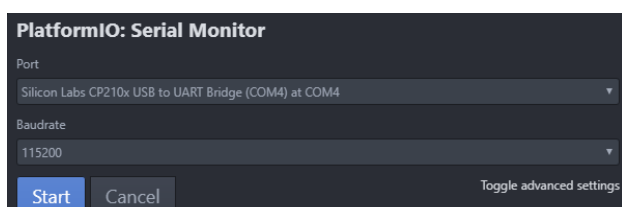


- 1.Home do PlatformIO
- 2.Build PlatformIO
- 3.Upload do PlatformIO
- 4.Carregar para dispositivo remoto
- 5.PlatformIO Clean
- 6.Teste PlatformIO
- 7.Depuração do PlatformIO
- 8.Executar outro destino
- 9.Alternar painel de compilação
- 10.Encontrar no projeto
- 11.terminal
- 12.Monitor Serial (é como o Monitor Serial do Arduino IDE)
- 13.Atom Settings

Nós modificamos o código Blink usado anteriormente para incluir alguns comandos **Serial.println ()** para demonstrar como o Monitor Serial se parece. Você pode abrir o Monitor Serial clicando no ícone:



O software PlatformIO deve completar automaticamente suas configurações. Caso contrário, selecione sua porta COM do ESP32 e sua taxa de transmissão. Em seguida, pressione o botão " **Iniciar** ":



Assim como o Arduino IDE Serial Monitor, você tem uma janela que gera todos os comandos *Serial.println ()* usados no seu código:

Como você pode ver, está imprimindo as mensagens: "LED aceso" e "LED apagado".