

## Tutoriais:

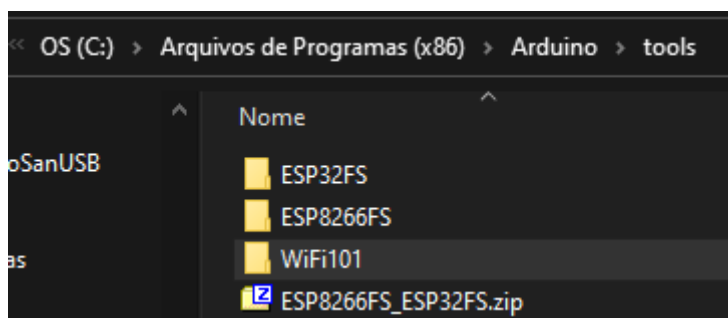
1. Exemplo **1Form\_LED.ino** (Comutar pino do ESP32 e ESP8266 através de página html e Javascript na memória SPIFFS): <https://youtu.be/5d9Z8eptu8A>
2. Exemplo **2Form\_LED\_PWM.ino** (Comutar a intensidade luminosa do pino por PWM no ESP32 e ESP8266 através de página html e Javascript na memória SPIFFS).
3. Exemplo **3Forms\_Reactjs.ino** (Servindo um exemplo de App React.js com ESP8266 e ESP32 na memória SPIFFS) faz parte de mais uma aplicação Async com código disponível nos exemplos do repositório <https://github.com/SanUSB/AsyncIOT> onde todas as bibliotecas Async descritas no README do repositório devem ser previamente instaladas.

*Este exemplo tem o objetivo de mostrar os passos para servir um app do React.js no ESP32 e no ESP8266 e não de explicar o desenvolvimento do código, pois será utilizado um exemplo de aplicativo padrão do React já existente, que pode ser visto nesse link.*

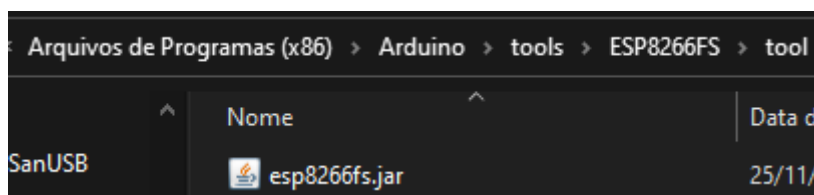
Depois de baixar e extrair o arquivo .zip do exemplo do aplicativo React, são necessários somente o arquivos *demo.js* e *index.html*, que devem ser colocados na pasta **data** para serem gravados na memória SPIFFS de acordo com os tutoriais <https://youtu.be/XiBGUGb7Fdo> e <https://youtu.be/5d9Z8eptu8A>, sendo possível excluir todos os outros arquivos do exemplo .zip.

## Memória SPIFFS:

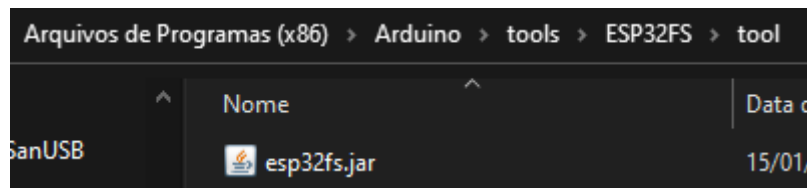
Baixe e extraia as pastas ESP8266FS e ESP32FS disponíveis em [http://sanusb.org/tools/ESP8266FS\\_ESP32FS.zip](http://sanusb.org/tools/ESP8266FS_ESP32FS.zip) dentro de C:\Program Files (x86)\Arduino\tools.



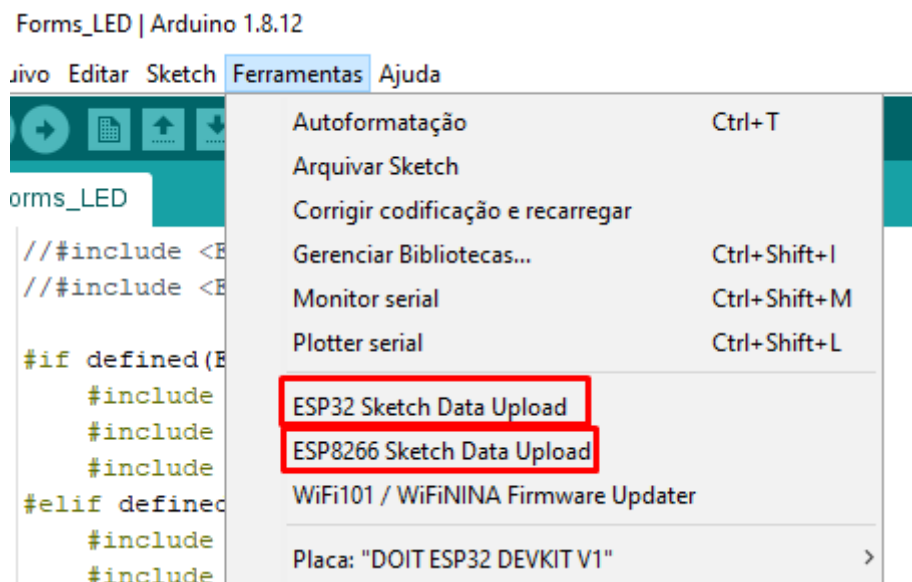
Verifique se o arquivo *esp8266fs.jar* está contido dentro da pasta tools, no endereço abaixo C:\Program Files (x86)\Arduino\tools\ESP8266FS\tool.



Verifique se o arquivo *esp326fs.jar* está contido dentro da pasta tools, no endereço abaixo  
 C:\Program Files (x86)\Arduino\tools\ESP32FS\tool.



Depois disso, reinicie a IDE do Arduino e transfira pela serial os arquivos contidos na pasta Data pela Arduino IDE clicando em **Sketch Data Upload**.



Clicando em *ESP8266 Sketch Data Upload*, os arquivos contidos na pasta DATA sobem para a memória SPIFFS do ESP.