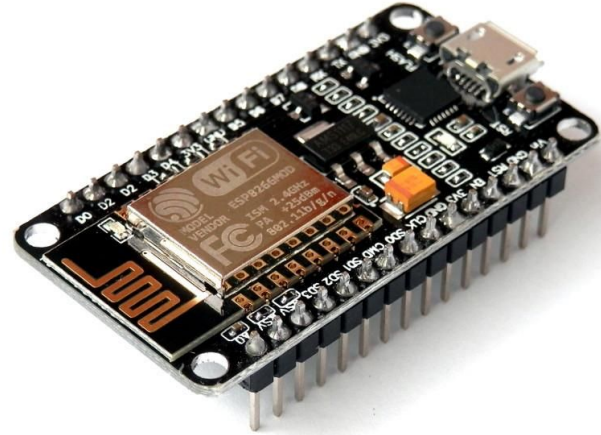


# **Minicurso**

## **IoT**

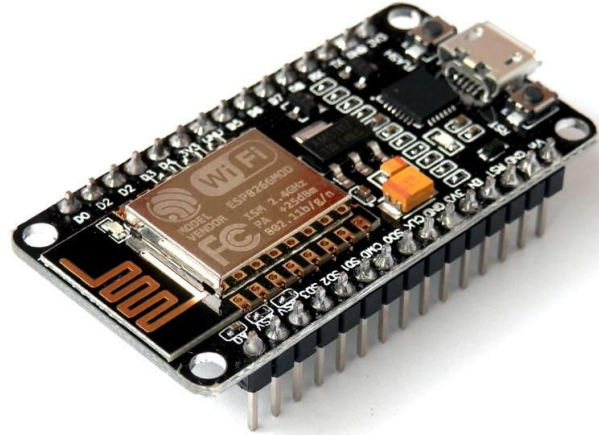
### **ESP8266**

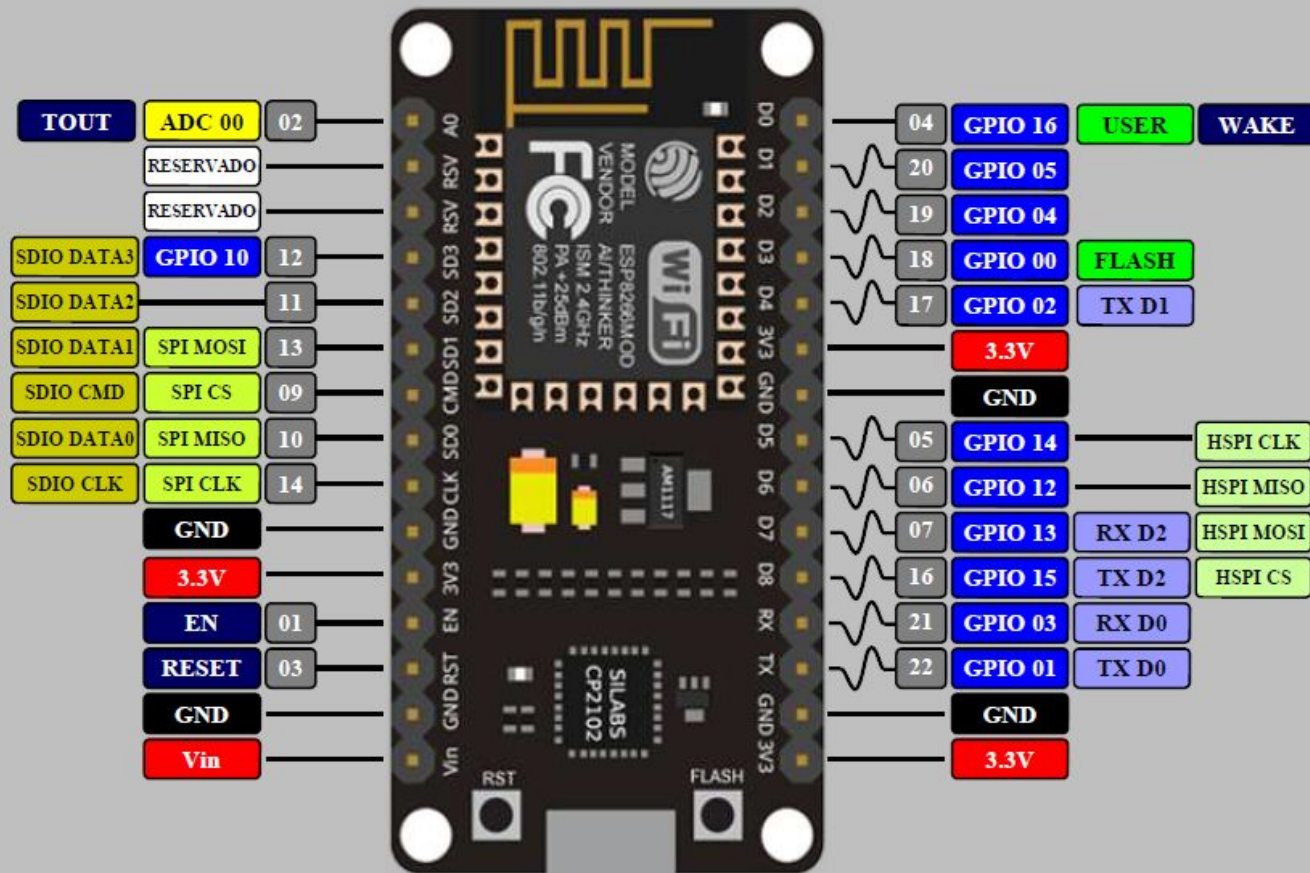
#### **Aula 2**



# Jimmy PETECO

<https://github.com/jimmytsoares>  
[jimmy@alunos.utfpr.edu.br](mailto:jimmy@alunos.utfpr.edu.br)





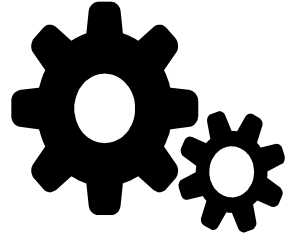
# Blynk

Um app que permite utilizar o NodeMCU (e várias outras placas) de maneira fácil e rápida.



# 1.

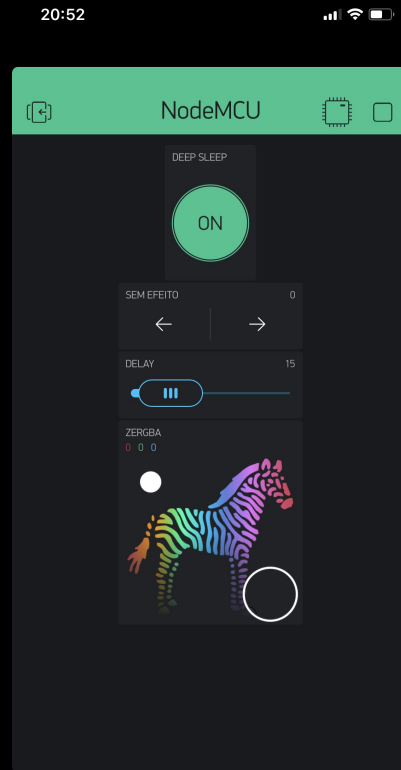
**O Blynk é uma combinação de app + servidor que facilita o uso do nodeMCU totalmente conectado na internet, podendo acessá-lo de qualquer lugar.**



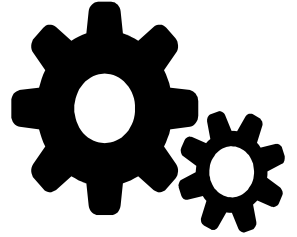
**Ao baixar o app e fazer uma conta, será possível criar projetos. Cada projeto tem um token de autenticação único. Esse token deve ser informado a placa, normalmente por hardcode, para haver a comunicação. Dentro do projeto você pode colocar botões, displays, sliders, existem diversos widgets a disposição para escolher.**

# Blynk

Exemplo de projeto



**Resumidamente, o que ocorre é que com esse token tanto o app quanto a placa utilizam a arquitetura REST para se comunicarem através do servidor.**



**Para abstrair mais um pouco, imagine que seu projeto tem uma página na web. Tanto a placa quanto o app ficam monitorando essa página. Normalmente o app realiza mudanças na página e a placa age de acordo, mas o contrário pode acontecer também. Nessa página se encontram informações de todos os seus pinos.**

1. **Baixar o Blynk no celular, fazer conta/login com o facebook o que preferir.**
2. **Ir até <https://www.blynk.cc/getting-started/>**
3. **Download Blynk library**
4. **Extraí o conteúdo e encontre a sua pasta do sketchbook Arduino, em File -> Preferences ou no Mac Arduino -> Preferences**
5. **Dentro da pasta do sketchbook, se já houver pasta libraries, coloque o que você baixou que estava dentro de libraries dentro dela e a mesma coisa para tools**
6. **Qualquer dúvida o próprio site do Blynk deixa claro**



- 1. Abra IDE do Arduino**
- 2. File -> Examples -> Examples from custom libraries  
-> Blynk -> Boards\_WiFi -> NodeMCU**
- 3. Abra o app, crie um novo projeto, selecionando a placa certa e crie um botão nele e coloque no D4 ou D0 depende da placa**
- 4. Encontre o token nas configurações do seu projeto e coloque no código na IDE do Arduino**
- 5. Altere o SSID e senha**
- 6. V ->, se o node conseguir conectar na internet, ao clicar Play no app, será possível ver se o Node está online e o botão irá fazer o LED da placa ligar/desl.!**

**Por enquanto só sabemos acessar diretamente as portas digitais, mas nem sempre queremos esse acesso direto, queremos processar uma informação antes. Ex: Cor de uma fita de LED.**



**O Blynk criou então pinos virtuais, que basicamente servem para você passar informações de widgets do app e acessar pelo código na placa e vice-versa.**

**No app, no lugar de um pino físico, você coloca um virtual. Ex: V1, V2.... V20...**



**No código para ter acesso a esse valor você utiliza (normalmente widgets de controle):**

```
BLYNK_WRITE(PINO) {  
int variavel = param.asInt(); } (como uma função)
```

**Para widgets de display, que a informação vai da placa pro app:**

```
BLYNK_READ(PINO){  
Blynk.virtualWrite(PINO, valor);} para mandar em  
intervalos definidos no app ou  
Blynk.virtualWrite(PINO, valor); quando quiser
```



```
setup(){
```

```
Inicializa o Blynk como no exemplo
```

```
}
```

```
loop(){
```

```
Normalmente apenas o Blynk.run() deve estar
```

```
}
```

```
BLYNK_WRITE(V1){
```

```
int var = param.asInt(); Pega a variável do widget  
de controle e coloca em var quando você mudar  
pelo app
```

```
}
```

**continua**

**->**

```
BLYNK_READ(V2){
```

```
  Blynk.virtualWrite(V2, valor);
```

**Nos widgets de display existe um parâmetro chamado ‘Refresh Interval’ que define de quanto em quanto tempo aquele widget deve ser atualizado com as informações do servidor. Essa função é acionada conforme o intervalo definido lá.**

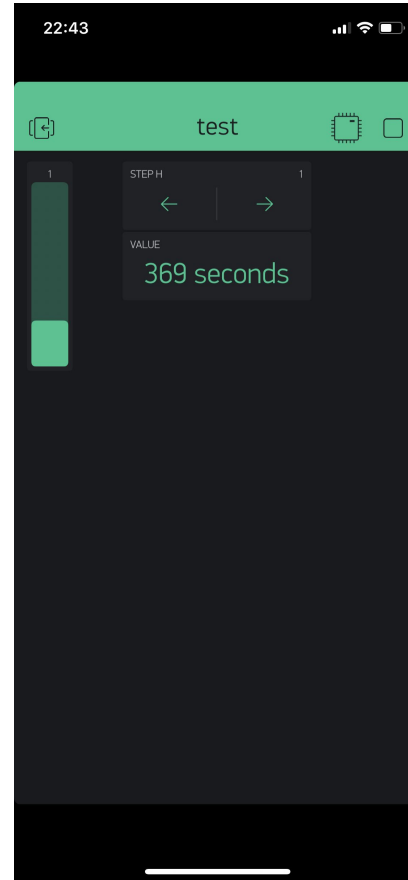
```
}
```

**Caso não queira atualizar em intervalo definido é possível chamar Blynk.virtualWrite(PINO, valor); em qualquer lugar.**



## Exercício

- Criar um projeto com 1 labeled display, 1 step H e 1 level V (nomes no app)
- O labeled display deve mostrar o tempo ativo. Configurar refresh interval em 1s. Dica: `millis()/1000`
- O step H e o level V devem ir de 0-4 e ao mudar o step H o level deve mudar de acordo



Em 5 linhas de código da para fazer!

# 2.

## **WiFiManager**

Biblioteca para conectar em redes ou adicionar informações (como o token do Blynk) sem precisar compilar sempre

- 1. Abra IDE do Arduino**
- 2. Sketch -> Include Library -> Manage Libraries -> WiFiManager by tzapu**
- 3. Pegar o código no github BlynkWiFiManager\_Simple**
- 4. Encontrar a linha**  
**if (!wifiManager.autoConnect("MUDARAQUI",**  
**"12345678")) e alterar o nome para algo único**
- 5. V ->, logo irá aparecer uma rede com o nome que você colocou, assim que conectar nela deverá ter um popup de configuração criado pelo WiFiManager**
- 6. Colocar as informações, o Node irá resetar e se estiver tudo certo tudo estará funcionando**