



# **Relatório da Arquitetura**

## **Project Factory**

**Docente: Fábio Guilherme**

03/06/2024

Francisco Santos nº20211206  
Mariana Gonçalves nº20210539

# Índice

<b>Funções implementadas</b>	2
Recolha de dados	2
Menu	2
Autenticação entre esp32 e a API	2
Controlo de led	3
Botão emergência	3
Autenticação entre utilizador e API	3
Aplicação Móvel	4
<b>Funções a implementar</b>	4
Mesh	4
Outras funcionalidades	4
<b>Diagramas</b>	5

# Funções implementadas

## Recolha de dados

O esp32 obtém o *Epoch Time* através de um servidor NTP, o microcontrolador depois vai calibrando o relógio interno a cada 30 minutos. Isto é necessário, pois não pretendemos estar a enviar os dados dos sensores a cada segundo para a API, assim guardamos os dados e o respetivo epoch time num cartão SD, depois mandamos tudo de uma vez para API que guarda na base de dados.

## Menu

O nosso menu terá três opções: visualizar os dados dos sensores, o próximo concerto que irá decorrer no festival, e os ESPs em proximidade.

## Autenticação entre esp32 e a API

Para evitar confusões, a “apitoken” é o token que a API cria para o esp32 comunicar, e existe o “esptoken”, que server, para a API comunicar com o esp32, criando uma comunicação bidirecional.

Quando o esp32 é ligado, é mandado para a API, a “apikey”, o MACaddress do esp, o “esptoken” que é gerado cada vez que o esp32 inicia e a “apitoken”, se tiver. A API verifica, se a “apikey” é a correta e se tem o MACaddress registado, se não tiver, regista, se tiver, verifica se o “apitoken” é o correto, se este não for correto ou se o esp32 não tiver registado, a API devolve um token gerado, e um “userToken” para depois existir maneira de um utilizador controlar o esp32.

A API de seguida, também guarda o “esptoken” e o ip do esp32 na base de dados, para poder mandar lhe mandar comandos se necessário.

O esp32 depois, se necessário, guarda o token na EEPROM e utiliza-o para comunicar com a API.

## Controlo de led

Esta função permite controlar ou para controlar o RGB Led,

Faz se um pedido á API central, com o macaddress desejado, a API vai á base de dados procurar pelo “esptoken” para comunicar com o esp32 e o seu IP, a API de seguida manda um pedido para a API do esp32, com as instruções, o esp32 verifica se o token é correto, e se for executa as funções.

## Botão emergência

Quando o utilizador carrega um botão na tenda, a tenda envia um pedido para a API central para indicar que esta tenda está em modo de emergência, e a tenda em si, enquanto está neste modo, para de gravar informação no cartão SD e n aceita mais pedidos de controlo do led, pois o led irá estar a piscar vermelho e um buzzer fica a tocar.

## Autenticação entre utilizador e API

Quando o esp liga, autenticasse com a API da maneira implementada e explicada em cima, só que com a autenticação poderá receber um “UserToken” (só o receberá se for necessário autenticar-se um utilizador novo). Se receber o esp manda uma mensagem ao Arduíno que está conectado através dos pins TX/RX no seguinte formato:”\0NFC\0\*usrToken\*\0”, os ‘\0’ servem para o Arduíno distinguir entre lixo de debugging, e de comandos específicos.

Quando o Arduíno deteta que o comando é NFC, este liga passa um led vermelho para verde, indicando que está á espera de um cartão NFC “mifare”, para guardar neste o UserToken. O utilizador depois deverá encostar o cartão até o led passar de verde para vermelho, indicando que não tem nada para guardar. Agora o utilizador tem um cartão NFC com um token para aceder á tenda guardado, este Token pode ser reiniciado por um admin, basta mandar para a API central no caminho “/api/node/reset”, o seguinte corpo:

```
{
  "macaddress": "macaddress"
}
```

Isto faz com que o userToken seja alterado, e é necessário a autenticação novamente com um cartão NFC.

## Aplicação Móvel

Quando o utilizador abre a APP, se for a primeira vez que a está a abrir, ou se o seu UserToken tenha sido alterado, terá que encostar o cartão NFC atrás autenticado, no sensor NFC do telemóvel.

Após ter o token e ter acesso á tenda, poderá ver os valores dos sensores da tenda, localizar a tenda e ligar a luz interior.

## Funções a implementar

### Mesh

Para fazer a mesh cada nó guarda as conexões que consegue fazer com outros nós na proximidade. Depois na API guardamos essa informação, e no Unity utilizamos “Force-directed graph drawing” para desembaralhar a mesh. Com essa visualização, o utilizador depois poderá escolher uma data e cada nó na mesh, irá mudar a altura e a cor, dependendo do valor do sensor. Com essa visualização, se tudo correr bem também irá se conseguir programar os light shows precisamente.

### Outras funcionalidades

Possivelmente no futuro, utilizaremos a biblioteca “painless mesh”, para tornar mais fácil o envio de dados para a api e o controlo de cada dispositivo.

# Diagramas

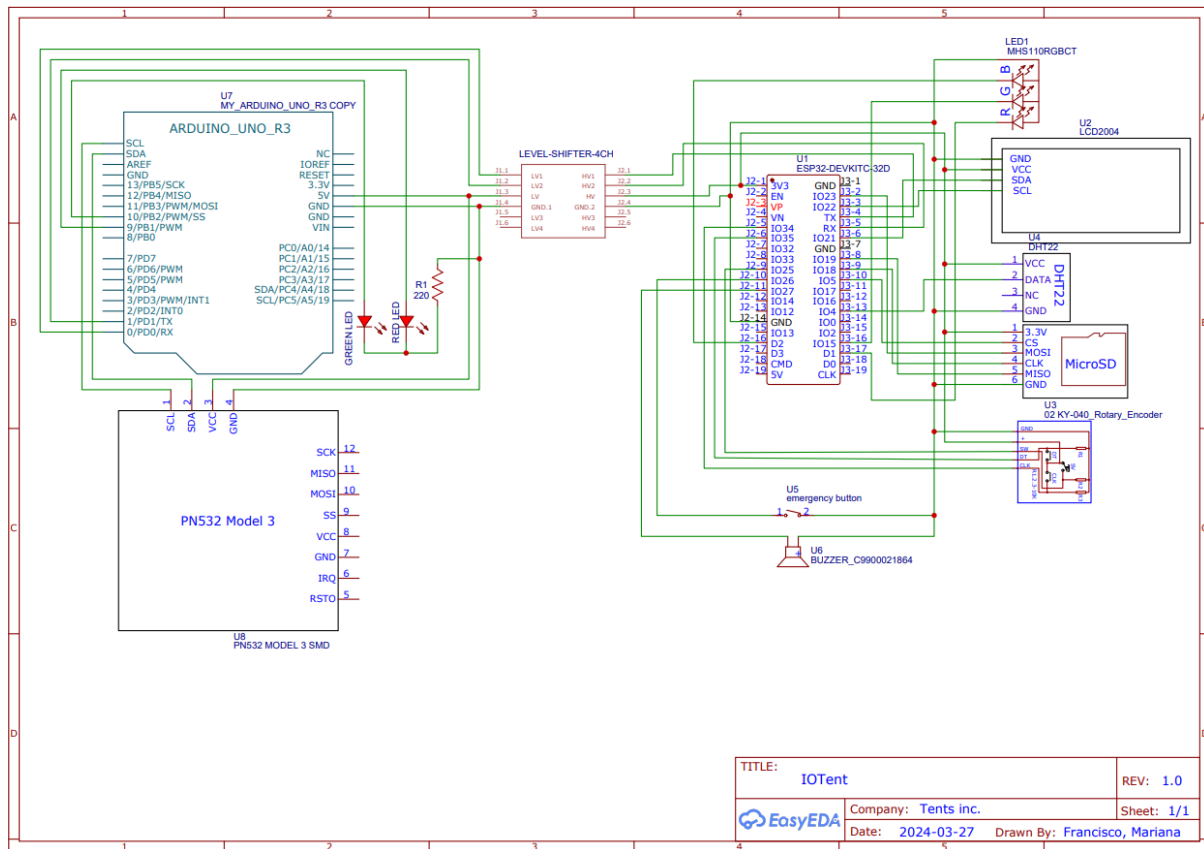


Diagrama do circuito realizado no EasyEDA