



**BOSCH**

Invented for life

## Aula 6 - Wi-Fi + Firebase

### Docupedia Export

Author: Goncalves Donathan (CtP/ETS)  
Date: 03-Mar-2023 12:47

# Table of Contents

Com o avanço do uso de dispositivos móveis e os benefícios que eles proporcionam, a comunicação sem fio é um fator fundamental. Em companhias, ela pode agregar maior capacidade de monitoramento para a gestão, mobilidade e agilidade operacional, bem como redução de custos e riscos.

Neste treinamento estaremos utilizando o Wi-Fi, ele possui alta taxa de transmissão de dados, porém alcance limitado de sinal e alto consumo de energia, existem diversas outras opções que podem ser bons substitutos para o Wi-Fi. LoRaWAN, por exemplo, funciona para grandes distâncias, tendo alcance de alguns km.



O Wi-Fi funciona através de ondas de rádio, assim como as TVs, aparelhos de rádio e celulares. A antena do roteador é a responsável por captar e emitir os sinais, bem como decodificá-los. E é assim que os aparelhos conseguem trocar informações.

Essa troca de informações pode ocorrer em duas frequências diferentes, 2.4GHz ou 5 GHz. E quanto mais alta a frequência, maior é a capacidade de transferência de dados.

Também existe um protocolo que os aparelhos precisam seguir para se entenderem, que é o padrão 802.11. Tal padrão é uma série de regulamentações que especifica regras de transmissão e codificação para a troca de informações.

O roteador, cuja função é realizar a distribuição dos sinais da rede, além de "escolher" o melhor caminho para o envio de um conjunto de dados, é quem recebe o sinal e o decodifica. É ele quem envia as informações para a internet usando uma conexão (com fios), a Ethernet, responsável pela interconexão de redes locais.

Nesta aula estaremos usando a ESP como um Client da rede Wi-Fi, em outras palavras, a ESP estará mandando dados para o roteador, que decodificará e enviará esses dados via cabo para a Internet.

Vamos utilizar a biblioteca Wi-Fi.h:

```
1 #include <Wi-Fi.h>
```

Precisamos definir o SSID e a senha da rede que estamos utilizando:

```
1 #define WIFI_SSID "<NOME DA SUA REDE>"
2 #define WIFI_PASSWORD "<SENHA DA SUA REDE>"
3 Wi-Fi.begin(WIFI_SSID, WIFI_PASSWORD);
```

Podemos checar o status da conexão, vamos criar um while para só seguir o programa quando a ESP estiver conectada à rede:

```
1 while (WiFi.status() != WL_CONNECTED)
2 {
3   Serial.print(".");
4   delay(300);
5 }
```

Por fim, podemos ver o IP da rede em que estamos conectados.

```
1 Serial.print("Connected with IP: ");
2 Serial.println(WiFi.localIP());
```

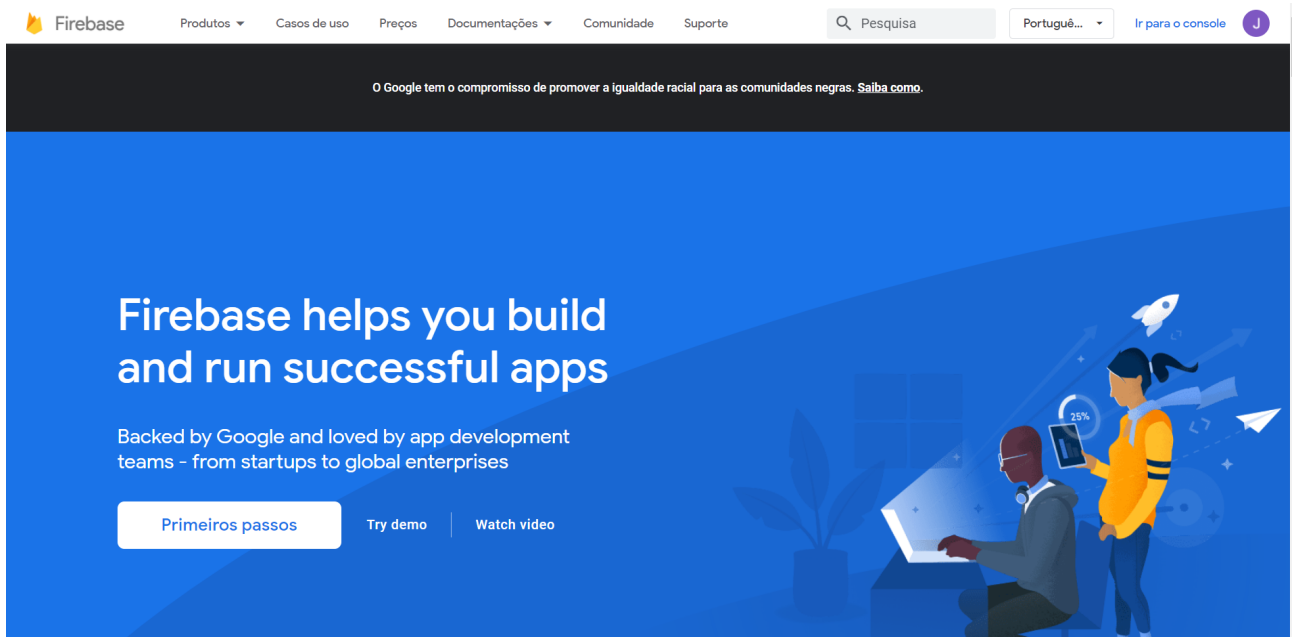
Se as configurações foram feitas corretamente, neste ao carregar a ESP com esse script, a conexão com a rede já está ativa.

```
11:48:42.707 -> Connecting to Wi-Fi.....
11:48:45.111 -> Connected with IP: 192.168.43.108
11:48:45.111 ->
11:48:45.111 -> -----
11:48:45.111 -> Connected...
```

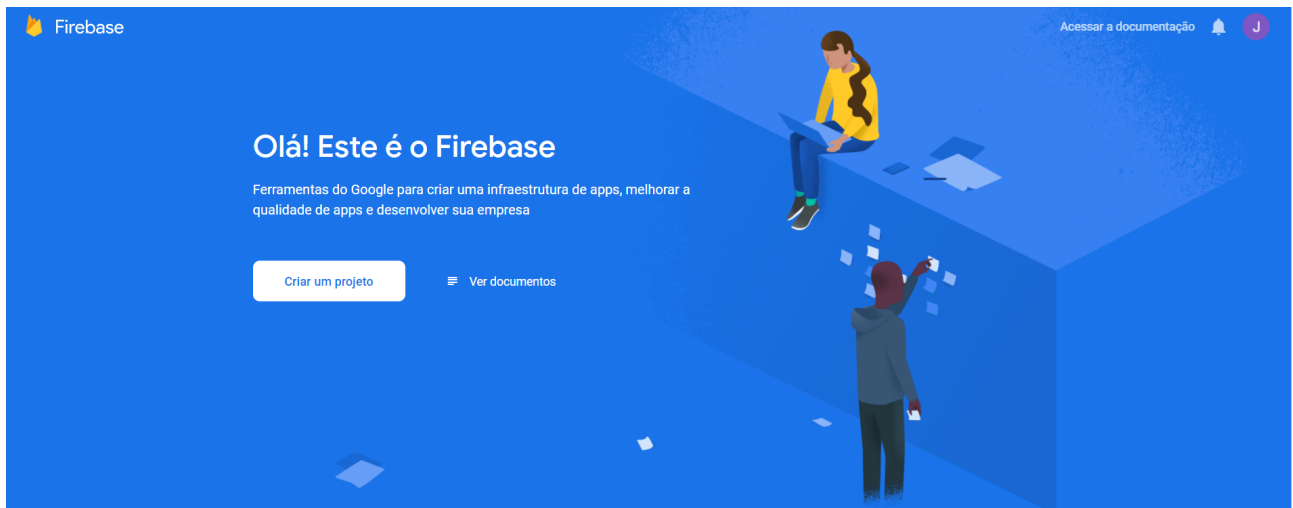


- O [Firebase Realtime Database](#) é um banco de dados hospedado na nuvem. Os dados são armazenados como JSON e sincronizados em tempo real com todos os clientes conectados.
- Com ele poderemos enviar os dados da ESP, então eles ficarão disponíveis para visualização até que o JSON seja atualizado.

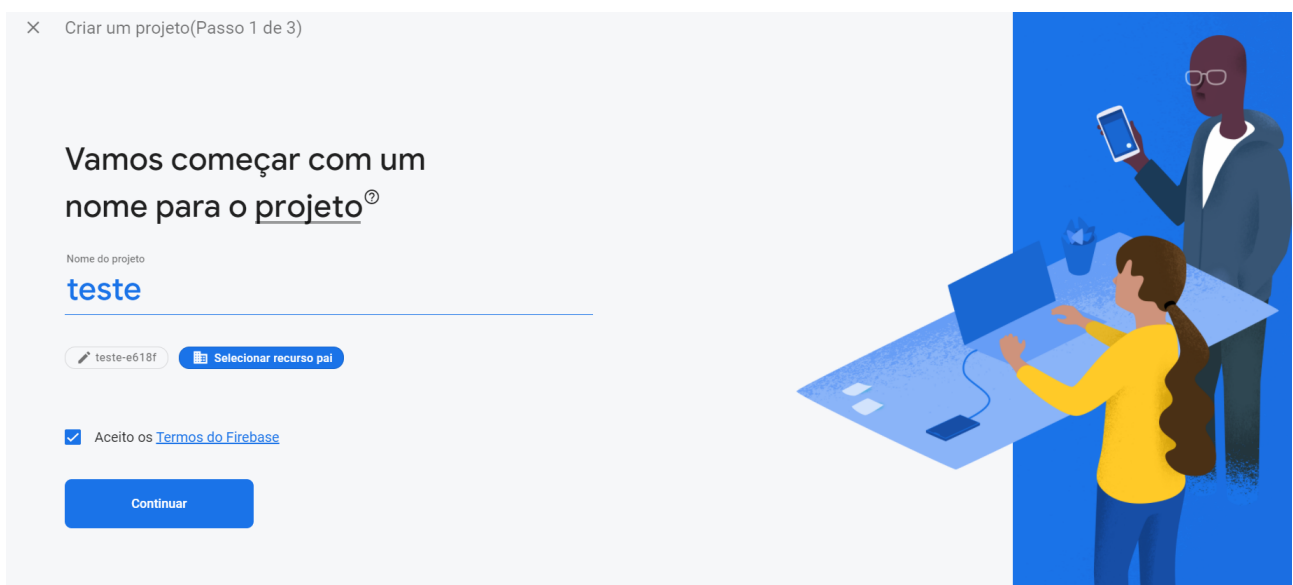
O primeiro passo é criar um projeto:



Clique em "Ir para o console".



Clique em "Criar um projeto".



Escolha um nome para o projeto, aceite os termos e clique em Continuar. Na próxima página clique em Continuar novamente.

✕ Criar um projeto(Passo 3 de 3)

Localização do Analytics ⑦

Estados Unidos

Configurações de compartilhamento de dados e termos do Google Analytics


☒ Usar as configurações padrão para o compartilhamento de dados do Google Analytics. [Learn more](#)

- ✕ Compartilhe seus dados do Analytics com o Google para melhorar os produtos e serviços da empresa
- ✓ Compartilhe seus dados do Analytics com o Google para ativar o Comparativo de mercado
- ✓ Compartilhe seus dados do Analytics com o Google para ativar o suporte técnico
- ✓ Compartilhe seus dados do Analytics com os especialistas em contas do Google

☒ Eu aceito os [Termos do Google Analytics](#)

Após a criação do projeto, uma nova propriedade do Google Analytics será criada e vinculada ao seu projeto do Firebase. Esse processo permitirá o fluxo de dados entre os produtos. Os dados da propriedade do Google Analytics exportados para o Firebase ficam sujeitos aos Termos de Serviço do Firebase, e os dados do Firebase importados para o Google Analytics ficam sujeitos aos Termos de Serviço do Google Analytics. [Saiba mais](#)

[Anterior](#) [Criar projeto](#)



Aceite os novos termos e clique em Criar projeto.

🔥 Firebase teste

Receber atualizações por e-mail sobre novos recursos, pesquisas e eventos do Firebase [Inscreva-se](#) ✕

[Plano Spark](#)

Comece adicionando o Firebase ao seu aplicativo

iOS Android Web Flutter

Adicione um app para começar

Armazene e sincronize dados de app em milissegundos ✕

Spark  
Plano gratuito US\$ 0/mês [Fazer upgrade](#)



Esta é a tela principal do projeto. Na árvore ao lado esquerdo, clique em Realtime Database, e depois em Criar banco de dados.

### Configurar banco de dados

1 Opções de banco de dados

2 Regras de segurança

Sua configuração de local é onde os dados do Realtime Database serão armazenados.

Local do Realtime Database

Estados Unidos (us-central1)

CancelarAvançar

Selecione Estados Unidos e clique em Avançar.

### Configurar banco de dados

1 Opções de banco de dados

2 Regras de segurança

Após definir a estrutura de dados, será necessário criar regras para proteger seus dados.

[Saiba mais](#)

☐ Iniciar no modo bloqueado

Seus dados são privados por padrão. O acesso de leitura/gravação do cliente será concedido apenas se especificado por suas regras de segurança.

☒ Iniciar no modo de teste

Por padrão, seus dados estão definidos para permitir uma configuração rápida. Porém, você precisa atualizar suas regras de segurança em até 30 dias para permitir em longo prazo o acesso de leitura/gravação do cliente.

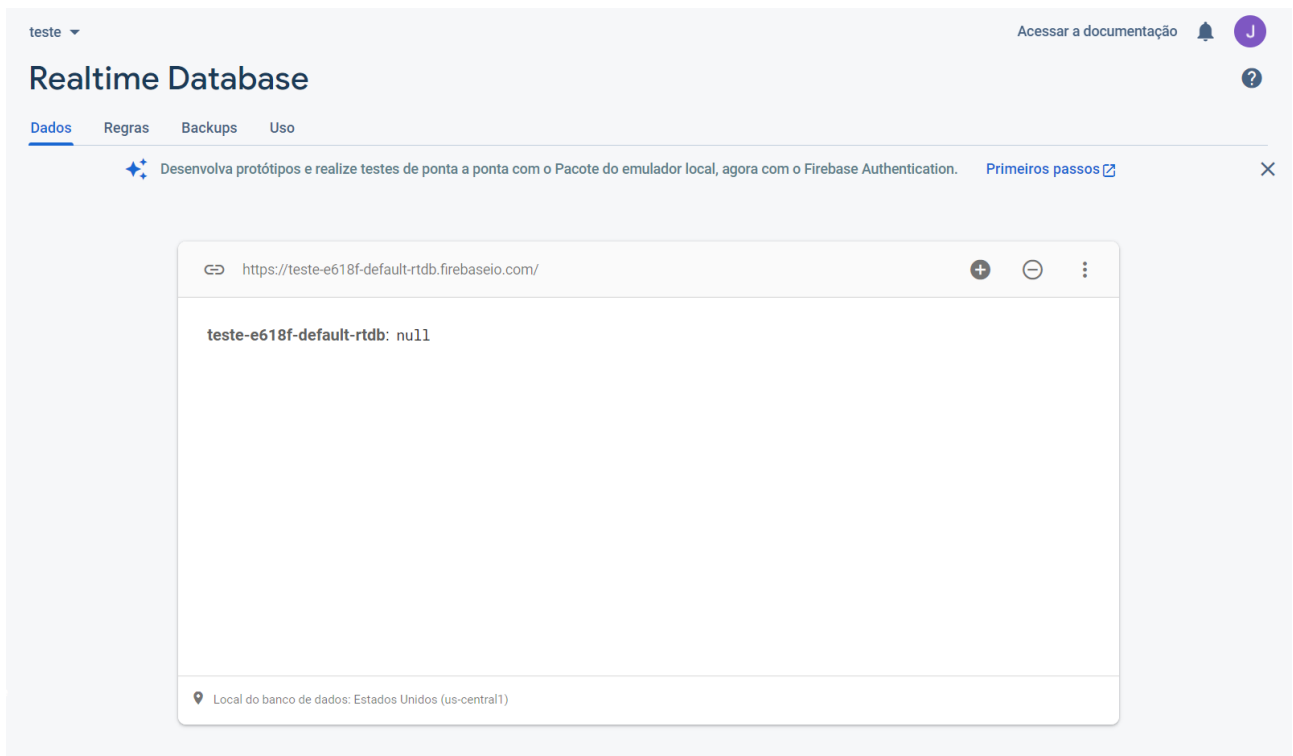
```
{
  "rules": {
    ".read": "now < 1623466800000", // 2021-6-12
    ".write": "now < 1623466800000", // 2021-6-12
  }
}
```

!

As regras de segurança padrão para o modo de teste permitem que qualquer pessoa com a referência do seu banco de dados acesse, edite e exclua todos os dados nele por 30 dias

CancelarAtivar

Selecione "modo de teste" e clique em Ativar.



Agora nosso Realtime Database está pronto para receber nossos dados. Sua aparência deve ser como o da imagem acima.

Vamos utilizar a biblioteca [FirebaseESP32.h](#).

```
1 #include <FirebaseESP32.h>
```

É preciso definir duas variáveis para conectar ao Firebase:

```
1 #define FIREBASE_HOST <SEU_HOST_LINK> // URL acima do console de dados
2 #define FIREBASE_AUTH <SUA_AUTH_KEY> // Configurações do projeto > Contas
  de serviço > Chaves secretas do banco de dados > Copiar chave secreta
```

Precisamos criar os objetos do Firebase:

```
1 FirebaseData firebaseData;
2 FirebaseJson json;
```

E executamos as seguintes funções no `setup()`:

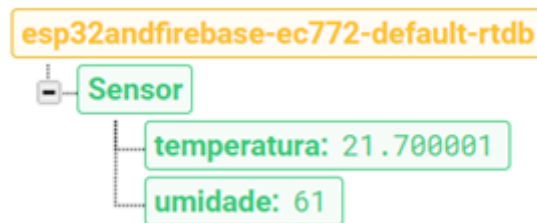
```
1 Firebase.begin(FIREBASE_HOST, FIREBASE_AUTH);
2 Firebase.reconnectWiFi(true);
3 Firebase.setReadTimeout(firebaseData, 1000 * 60);
4 Firebase.setWriteSizeLimit(firebaseData, "tiny");
```

Para enviar os dados precisamos montar o JSON, a biblioteca do Firebase oferece uma opção que faz isso de uma forma bem simples:



```
1 json.set("/temperatura", t);  
2 json.set("/umidade", h);  
3 Firebase.updateNode(firebaseData, "<Seu_Nome>/Sensor", json);
```

Utilizamos o objeto **json** criado anteriormente, e basta setar o nome das chaves, seguidos de seus valores, que serão os dados lidos dos sensores. Depois disso é só atualizar o Node, que demos o nome de **Sensor**, o resultado deve ser esse:



Agora nós temos nossos dados disponíveis em um endereço na Web, sendo atualizados com uma certa frequência. Na próxima aula vamos ver como recuperar esses dados utilizando requests em python.