

# Passo a Passo Completo para Configurar Firebase, Arduino e HTML do Projeto Controle de Porta

## 1. Criar Projeto no Firebase

1. Acesse <https://console.firebase.google.com/>
2. Clique em **"Adicionar projeto"**.
3. Dê um nome ao seu projeto (ex: controle-porta).
4. Aceite os termos e clique em **"Continuar"**.
5. Desative Google Analytics se preferir e finalize a criação.

## 2. Configurar Realtime Database

1. No painel do projeto Firebase, clique em **"Realtime Database"** no menu lateral.
2. Clique em **"Criar banco de dados"**.
3. Escolha a localização do banco e clique em **"Ativar modo bloqueado"** (recomendado para segurança).
4. Depois, configure as regras conforme o arquivo **rules** que está no seu GitHub:
  - Copie o conteúdo do arquivo **rules**.
  - Na aba **Regras** do Realtime Database, substitua o conteúdo atual pelo do arquivo.
  - **Substitua o trecho <UID\_DO\_USUARIO> pelas letras e números do UID real do usuário criado no Firebase Authentication.**  
**O UID é exibido ao clicar no nome do usuário dentro do painel Firebase > Authentication > Usuários.**

### 3. Criar Usuário no Firebase Authentication (Email e Senha)

1. No painel Firebase, vá em **Authentication > Método de login**.
2. Habilite o método **Email/Senha**.
3. Vá em **Usuários** e clique em **Adicionar usuário**.
4. Crie um usuário com o e-mail e senha que deseja para testar (ex: usuario@exemplo.com e 123456).
5. Anote o **UID** desse usuário (clcando nele, na lista de usuários), pois será usado nas regras e importação do banco.

### 4. Importar Dados JSON no Realtime Database

1. No Realtime Database, clique nos três pontinhos do painel (ao lado de "Dados").
2. Clique em **Importar JSON**.
3. Escolha o arquivo JSON que você já subiu no GitHub (baixe-o primeiro).
4. Antes de importar, **abra o arquivo JSON** e substitua o UID placeholder pelo UID do usuário criado.
5. Importe o arquivo para o banco.

### 5. Configurar o Arduino

- Abra o código do Arduino que controla o servo e conecta ao WiFi/Firebase.
- No início do código, procure e edite as variáveis:

```
const char* ssid = "SEU_SSID";      // Substitua pelo nome da sua rede WiFi
const char* password = "SUA_SENHA"; // Substitua pela senha da sua rede WiFi
```

```
const char* firebaseHost = "SEU_FIREBASE_HOST";
// Exemplo: "seu_projeto-rtadb.firebaseio.com"
```

```
const char* firebaseAuth = "SEU_FIREBASE_AUTH";
```

Você encontra isso no console do Firebase em Configurações do projeto > Contas de serviço > Chave privada

**Observação:** Algumas bibliotecas do Arduino exigem um token de autenticação.

Para isso, vá em *Configurações do Projeto* > *Contas de Serviço* > clique em **"Gerar nova chave privada"**.

Isso irá baixar um arquivo .json que contém o campo `private_key`, `client_email`, etc. Use conforme a biblioteca que estiver utilizando.

**Atenção:** em alguns casos, o token de autenticação pode não ser necessário se você já configurou corretamente as regras do banco para aceitar o UID.

## 6. Como encontrar o `firebaseHost`?

1. No console Firebase, vá em **Realtime Database**.
2. Na parte superior, verá a URL do seu banco, algo como:  
`blablabla-default-rtdb.firebaseio.com`
3. Use essa parte (`blablabla-default-rtdb.firebaseio.com`) no Arduino.

## 7. Como pegar o `firebaseAuth`?

1. Para projetos simples, você pode usar o token gerado por [Firebase Console > Configurações do projeto > Contas de serviço > Gerar nova chave privada].
2. Em projetos simples, a autenticação pode ser feita via regras do banco configuradas para aceitar seu UID, então o Arduino não precisa desse token diretamente, dependendo da biblioteca usada.

## 8. Ajustar o HTML do Projeto

- No seu arquivo index.html, localize a variável firebaseConfig dentro do <script type="module"> (ou no script onde está):

```
const firebaseConfig = {  
  apiKey: "SUA_API_KEY",  
  authDomain: "SEU_PROJETO.firebaseio.com",  
  databaseURL: "https://SEU_PROJETO.firebaseio.com",  
  projectId: "SEU_PROJECT_ID",  
  storageBucket: "SEU_BUCKET.appspot.com",  
  messagingSenderId: "SEU_SENDER_ID",  
  appId: "SEU_APP_ID"  
};
```

## 9. Para pegar esses dados:

1. No console do Firebase, clique no ícone de engrenagem (Configurações do projeto).
2. Em **Configurações gerais**, na seção **Seus apps**, selecione seu app web.
3. Copie as configurações para substituir no seu HTML.

### Observações:

- No seu arquivo `index.html`, localize a variável `firebaseConfig` dentro da tag `

## 10. **Certifique-se que:**

Os imports do Firebase estejam descomentados e corretos, por exemplo:

```
import { initializeApp } from "https://www.gstatic.com/firebasejs/9.6.1/firebase-app.js";
```

```
import { getAuth, signInWithEmailAndPassword, signOut } from  
"https://www.gstatic.com/firebasejs/9.6.1/firebase-auth.js";
```

```
import { getDatabase, ref, set } from  
"https://www.gstatic.com/firebasejs/9.6.1/firebase-database.js";
```

Depois disso, o login, controle da porta e comunicação com o banco devem funcionar.

## **Fim! Agora é só testar:**

- Publique o HTML no GitHub Pages.
- Ligue o Arduino na sua rede.
- Faça login com o usuário criado.
- Controle a porta com os botões.