

# Documentação do Projeto: Interface de Utilizador com Autenticação de Duas Fases (A2F)

## 1. Visão Geral do Projeto

- **Nome do Projeto:** Client Authentication
- **Descrição:** Este projeto consiste em uma interface de utilizador desenvolvida em Angular para autenticação de duas fases. O sistema valida clientes com base em um nome e código fornecido, e, após a validação, gera um código temporário que o utilizador deve inserir para completar a autenticação.

## 2. Arquitetura do Sistema

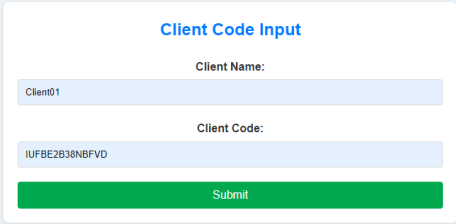
- **Frontend:** Desenvolvido em Angular usando componentes standalone.
- **Backend:** Não é necessário para este projeto específico, pois os dados dos clientes são armazenados em um arquivo JSON local.
- **Serviços:**
  - **ClientService:** Serviço responsável por buscar e validar os dados do cliente a partir de um arquivo JSON.
  - **TempCodeService:** Serviço responsável por gerar códigos temporários criptograficamente seguros.

## 3. Componentes e Serviços

- **AppComponent:** Componente raiz que gerencia a exibição do *ClientCodeInputComponent*.
- **ClientCodeInputComponent:** Componente que lida com a entrada do nome e código do cliente, valida essas informações, e gerencia a exibição e verificação do código temporário.
- **ClientService:** Serviço que interage com o arquivo JSON contendo os dados dos clientes, responsável pela validação do nome e código fornecido.
- **TempCodeService:** Serviço que gera códigos temporários de forma segura, utilizados na segunda fase da autenticação.

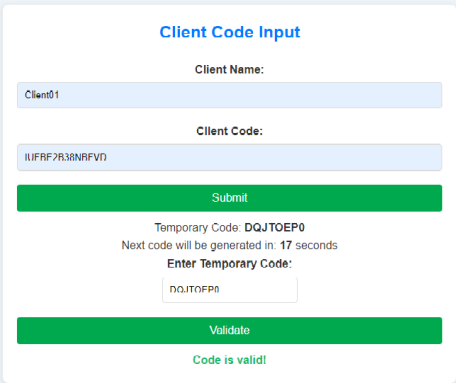
## 4. Funcionamento

O usuário insere o nome e o código do cliente na interface, após isso o ClientService valida o nome e o código comparando com os dados no arquivo JSON.



The image shows a web form titled "Two-Factor Authentication" with a subtitle "Client Code Input". It contains two input fields: "Client Name:" with the value "Client01" and "Client Code:" with the value "IUFBE2B38NBFVD". Below the fields is a green "Submit" button.

Se o cliente for validado com sucesso, o TempCodeService gera um código temporário que é exibido para o utilizador, este que deve inserir o código temporário dentro do tempo estipulado para completar a autenticação. Caso o utilizador não insira o código temporário correto será mostrado uma mensagem de erro, e caso exceda o tempo estimado, é gerado um novo código temporário e o temporalizador é reiniciado.



The image shows the same "Two-Factor Authentication" form, but now it displays a "Temporary Code: DQJTOEP0" and a message "Next code will be generated in: 17 seconds". Below this, there is an "Enter Temporary Code:" field with the value "DQJTOEP0" and a green "Validate" button. At the bottom, a green message "Code is valid!" is displayed.

Se o cliente não for validado com sucesso, é mostrado uma mensagem de erro.

## 5. Possíveis Melhorias Futuras

- **Backend Integration:** Implementar uma API backend para substituir o arquivo JSON, permitindo uma validação mais dinâmica e segura com uma base de dados real.
- **Melhoria na Segurança:** Implementar técnicas adicionais de segurança, como encriptação dos códigos no frontend e proteções contra ataques de força bruta.
- **Testes Automatizados:** Desenvolver testes unitários e end-to-end (E2E) para garantir a robustez e confiabilidade do sistema.
- **Expansão para Múltiplos Clientes:** Permitir que o sistema suporte múltiplos clientes com diferentes níveis de acesso e autenticação.

## 6. Requisitos de Configuração

- **Node.js;**
- **Angular CLI:** Usado para iniciar, desenvolver e construir o projeto Angular.
- **HTTP Server.**

## 7. Considerações Finais

- Este projeto foi desenvolvido como uma demonstração básica de autenticação de duas fases utilizando Angular. Ele é um bom ponto de partida, mas pode ser expandido e melhorado para atender a casos de uso mais complexos e robustos.