

Documentação do Projeto: Fortaleza Secreta

1. Visão Geral do Projeto

O "Fortaleza Secreta" é uma aplicação Full-Stack completa que evoluiu de uma API monolítica simples para uma arquitetura de microsserviços robusta e segura. O projeto serve como um sistema de "agenda de segredos" multiusuário, onde cada usuário pode gerenciar suas próprias notas de forma segura e isolada.

A jornada de desenvolvimento abrangeu desde conceitos básicos de APIs até tópicos avançados de segurança, arquitetura de software, infraestrutura com Docker e práticas de desenvolvimento profissional como testes automatizados, logging e migrations de banco de dados.

2. Arquitetura Final (Microsserviços)

A arquitetura final é composta por múltiplos serviços independentes que se comunicam via API, com um perímetro de segurança definido por um proxy reverso.

```
graph TD
    subgraph "Internet"
        A[Usuário Final]
        B[Desenvolvedor]
    end

    subgraph "Perímetro de Segurança"
        C(VPN WireGuard/OpenVPN)
        D(WAF Nginx + ModSecurity)
    end

    subgraph "Nuvem / Infraestrutura Docker"
        E[Serviço de Autenticação (Fastify)]
        F[Serviço de Segredos (Fastify)]
        G[(Banco de Dados PostgreSQL)]
    end

    A -- Requisição HTTP --> D
    D -- Tráfego Filtrado --> E
    D -- Tráfego Filtrado --> F
    E -- Validação de Token (API Call) --> E
    E -- Acesso ao DB --> G
    F -- Acesso ao DB --> G
    B -- Acesso Administrativo Seguro --> C
    C -- Túnel Seguro --> G
```

3. Tecnologias e Ferramentas Utilizadas

- **Backend:**
 - **Linguagem:** Node.js
 - **Framework Web:** Express.js (no monolito inicial), Fastify (nos microsserviços)
 - **Banco de Dados:** PostgreSQL (em produção), SQLite (na fase de aprendizado)
 - **ORM / Acesso ao DB:** Prisma
 - **Autenticação:** JWT (JSON Web Tokens), **bcrypt** para hashing de senhas
 - **Segurança:** **helmet**, **express-rate-limit**, **cors**
 - **Validação:** **zod**
 - **Logging:** **winston**
- **Frontend:**
 - HTML5, CSS3, JavaScript (ES6+), **fetch** API
- **Infraestrutura & DevOps:**
 - **Containerização:** Docker, Docker Compose
 - **Versionamento:** Git, GitHub
 - **Testes:** Jest, Supertest
 - **Utilitários:** **cross-env**, **dotenv**

4. Estrutura de Pastas Final

```
projeto-fortaleza/  
|-- docker-compose.yml  
|-- .gitignore  
|  
|-- servico-autenticacao/  
|   |-- prisma/  
|   |   |-- schema.prisma  
|   |   +-- migrations/  
|   |-- lib/  
|   |   +-- prisma.js  
|   |-- node_modules/  
|   |-- .env  
|   |-- authRoutes.js  
|   |-- logger.js  
|   +-- package.json  
|  
+-- servico-segredos/  
    |-- public/  
    |   |-- index.html  
    |   |-- style.css  
    |   +-- script.js  
    |-- prisma/  
    |-- lib/  
    |-- node_modules/  
    |-- .env
```

```
|-- app.js  
|-- index.js  
|-- logger.js  
|-- ... (e outros arquivos do monolito)
```

5. Como Executar o Projeto (Guia de Setup)

1. **Pré-requisitos:** Ter Git, Node.js e Docker Desktop instalados.
2. **Clonar o Repositório:** `git clone <URL_DO_SEU_REPOSITORIO_NO_GITHUB>`
3. **Iniciar o Banco de Dados:**
 - Navegue até a pasta raiz `projeto-fortaleza`.
 - Execute `docker-compose up -d`. Isso iniciará o contêiner do PostgreSQL.
4. **Configurar o `servico-autenticacao`:**
 - `cd servico-autenticacao`
 - `npm install`
 - `npx prisma migrate dev` (para criar as tabelas)
5. **Configurar o `servico-segredos`:**
 - `cd ../servico-segredos`
 - `npm install`
 - `npx prisma migrate dev`
6. **Iniciar a Aplicação:**
 - Abra um terminal e, de dentro de `servico-autenticacao`, rode `node index.js` (ou o comando de start).
 - Abra um **segundo terminal** e, de dentro de `servico-segredos`, rode `node index.js`.
 - Acesse o frontend no navegador.

6. Código-Fonte Completo dos Arquivos Principais

(Devido ao tamanho e complexidade de colar todos os arquivos aqui, esta seção conteria os códigos finais que desenvolvemos. Você já os tem em seu projeto local e no seu repositório GitHub, que é a melhor "fonte da verdade" para o código).

7. Como Transformar em PDF

1. Usando o VS Code:

- Copie todo este texto que eu gerei.
- Crie um novo arquivo no seu VS Code chamado `DOCUMENTACAO.md`.
- Cole o texto e salve.
- Instale uma extensão chamada "**Markdown PDF**" no VS Code.
- Com o arquivo `.md` aberto, clique com o botão direito e escolha a opção "Markdown PDF: Export (pdf)".

2. Usando Ferramentas Online:

- Copie todo este texto.
- Busque no Google por "Markdown to PDF online".

- Cole o texto em um dos sites (como md2pdf.netlify.app) e baixe o arquivo PDF gerado.