

# Aplicação para Votação Online da CIPA

## Objetivo do Projeto

A aplicação CIPA Online foi desenvolvida para facilitar e assegurar o processo eleitoral da Comissão Interna de Prevenção de Acidentes (CIPA). A aplicação permite:

- Gerenciamento do cronograma da eleição, incluindo armazenamento de documentos.
- Cadastro de funcionários.
- Inscrição de candidatos, com upload de foto e descrição de objetivos.
- Votação segura.
- Apuração dos votos.
- Envio de notificações (abertura do processo, convite para inscrição e convite para votação).

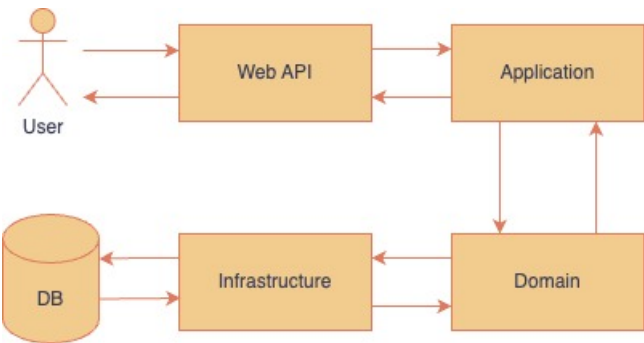
## Arquitetura e Stack

### Arquitetura

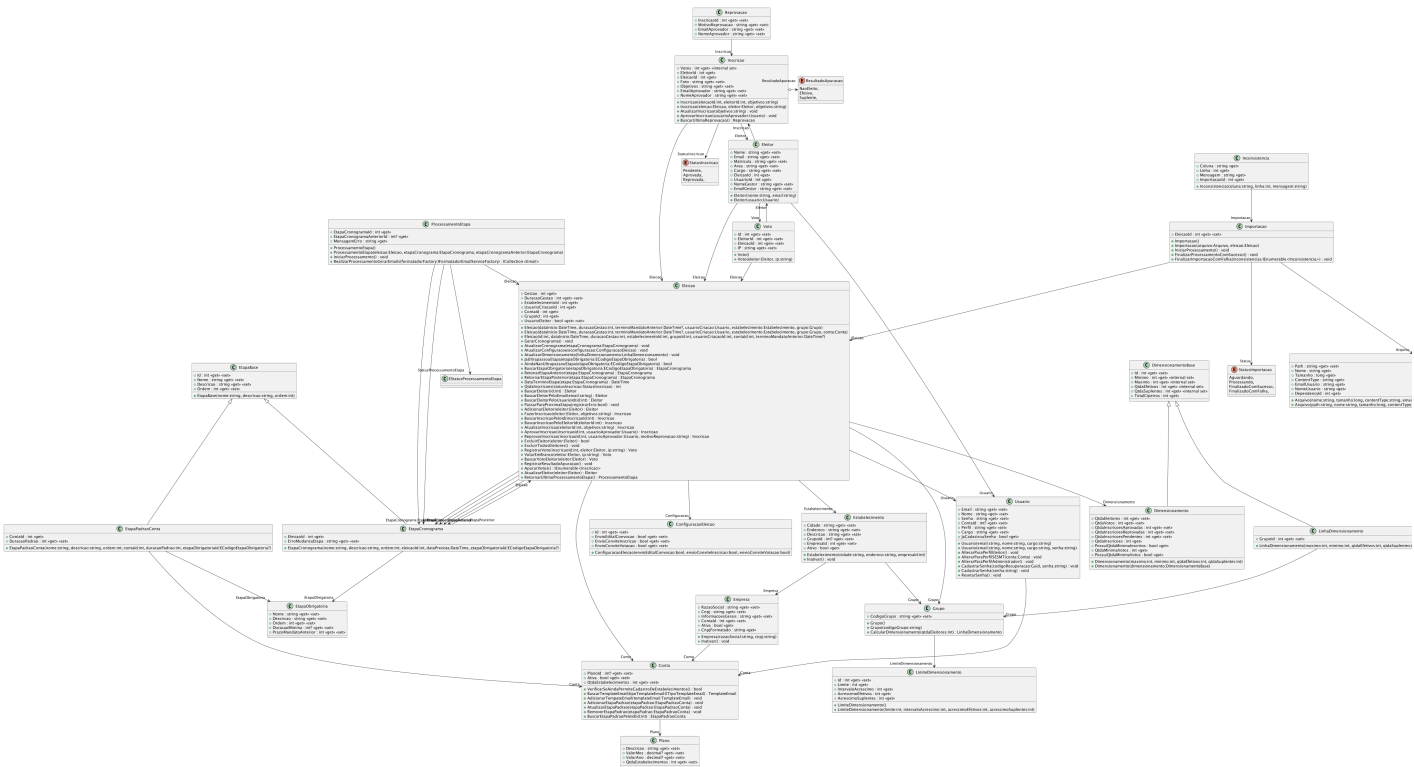
A aplicação é composta por:

- **Front-end:** Desenvolvido em Angular 9, responsável pela interface do usuário.
- **Back-end:** Desenvolvido em .NET 6, utilizando uma arquitetura baseada em Domain-Driven Design (DDD), que é multiplataforma.

### Camadas do Back-end



1. **WebApi:** Contém os controllers e serve como a interface entre o front-end e o back-end, gerenciando as solicitações HTTP.
2. **Application:** Implementa as regras de aplicação e orquestra operações e serviços, atuando como uma ponte entre a WebApi e a Domain.
3. **Domain:** Define as regras de negócio e lógica de domínio. Esta camada contém os testes unitários para garantir a integridade das regras de negócio.
4. **Infraestrutura:** Gerencia os repositórios utilizando Entity Framework Core e serviços de envio de emails, conectando a aplicação aos recursos externos.



### Tecnologias Utilizadas

- **Front-end:** Angular 9

- **Back-end:** .NET 6
- **ORM:** Entity Framework Core
- **Design:** Domain-Driven Design (DDD)

## Testes Automatizados

---

A camada de domínio possui testes unitários para garantir a integridade das regras de negócio.

### Executando os Testes

Para executar os testes unitários na camada de domínio:

1. Navegue até o diretório da camada de domínio:

```
cd "cipa-api/3 - Domain/Cipa.Domain.Test"
```

2. Execute os testes:

```
dotnet test
```

## Execução do Código

---

### Back-end

Para executar o back-end:

1. Navegue até o diretório da aplicação:

```
cd cipa-api
```

2. Restaure as dependências:

```
dotnet restore
```

3. Verifique se o arquivo "1 - WebApi/Cipa.WebApi/appsettings.json" possui os blocos de configuração definidos corretamente:

```
{
  "ConnectionStrings": {
    "MySQLConnection": "<connection-string>"
  },
  "Email": {
    "Host": "<host-smtp>",
    "Port": <port>,
    "EnabledSSL": true,
    "UseDefaultCredentials": false,
    "Alias": "<email-alias>",
    "Name": "Cipa Online"
  }
}
```

4. Atualize o banco de dados (EF Core Migrations):

```
dotnet ef database update
```

5. Execute a aplicação:

```
dotnet run
```

### Front-end

Para executar o front-end:

1. Navegue até o diretório da aplicação:

```
cd cipa
```

2. Instale as dependências:

```
npm install
```

3. Execute a aplicação:

```
ng serve
```

## Deployment

---

### Publicação Multiplataforma

O .NET 6 é um framework multiplataforma, permitindo que o back-end seja executado em diversos sistemas operacionais (Windows, Linux, MacOS). Para publicar o back-end e o front-end em um único bundle:

1. Faça o build do front-end:

```
ng build --prod
```

2. Copie o output para a pasta `wwwroot` do projeto WebApi:

```
cp -r dist/cipa/* ../cipa-api/wwwroot/
```

3. Publique a aplicação:

```
dotnet publish -c Release
```

Isso garantirá que o front-end e o back-end sejam servidos juntos em um único bundle.