**Observações de Desenvolvimento**

* A aplicação foi construída com Springboot no Backend. No frontend foi utilizada uma aplicação WEB com Thymeleaf, embarcado na própria aplicação.
* Nas instruções era solicitado que o banco de dados não se perdesse com o restart da aplicação. Entretanto, foi utilizado o banco H2 para facilitar o desenvolvimento.
* Também nas instruções, era solicitado que a aplicação rodasse em algum serviço de nuvem. Há pouco tempo eu utilizava o Heroku para fazer o deploy de aplicações, entretanto, com o fim dos serviços gratuitos, optei por disponibilizar o projeto na minha conta do Github

<https://github.com/wdiegosfreire/election>.

* Deixarei também os links do github de um projeto pessoal em desenvolvimento, caso queiram analisa-lo, o qual está bem mais elaborado.
  + <https://github.com/wdiegosfreire/sisfin-maintenance>
  + <https://github.com/wdiegosfreire/sisfin-transaction>
  + <https://github.com/wdiegosfreire/sisfin-frontend>
  + Trata-se de uma aplicação de controle financeiro com Springboot(back) e VueJs (front)
  + O back está dividido em dois projetos:
    - Maintenance: trata da camada de autenticação e manutenção de usuários
    - Transaction: trata da camada de controle financeiro em si. Tem comunicação com o projeto maintenance via FeingClient.

**Instruções para Rodar a Aplicação**

1. Rodar a aplicação localmente
2. Acessar o console do banco de dados H2 e rodar o script SQL abaixo para dar uma carga inicial de pautas e associados

* insert into sub\_subject (sub\_identity, sub\_title, sub\_description, sub\_end\_date) values (1, 'Ainda assim, existem dúvidas a respeito de como a valorização de fatores subjetivos afeta positivamente a correta previsão dos índices pretendidos.', null, null);
* insert into sub\_subject (sub\_identity, sub\_title, sub\_description, sub\_end\_date) values (2, 'É claro que a adoção de políticas descentralizadoras ainda não demonstrou convincentemente que vai participar na mudança das formas de ação.', null, null);
* insert into sub\_subject (sub\_identity, sub\_title, sub\_description, sub\_end\_date) values (3, 'Assim mesmo, a execução dos pontos do programa exige a precisão e a definição da gestão inovadora da qual fazemos parte.', null, null);
* insert into mem\_member(mem\_cpf, mem\_name) values('11111111111', 'Bruce Dickinson');
* insert into mem\_member(mem\_cpf, mem\_name) values('65849532154', 'Freddie Mercury');
* insert into mem\_member(mem\_cpf, mem\_name) values('77448596325', 'James Hetfield');
* insert into mem\_member(mem\_cpf, mem\_name) values('85241236541', 'Rob Halford');
* insert into mem\_member(mem\_cpf, mem\_name) values('96385274112', 'Ozzy Osbourne');

1. Acessar a home da aplicação no endereço. Uma lista de opções será apresentada. <http://localhost:8080/index>

Texto, Carta

Descrição gerada automaticamente

1. Módulo “Pautas”

Clicar na opção “Pautas” leva à home do módulo de pautas. A tela à seguir será apresentada com a lista de pautas cadastradas e as opções básicas de um CRUD. Os demais módulos seguem o mesmo padrão:

Interface gráfica do usuário, Texto, Aplicativo

Descrição gerada automaticamente

1. Módulo “Sessões de Votação”

Clicar na opção “Sessões de Votação” leva à home do respectivo módulo.

Interface gráfica do usuário, Aplicativo

Descrição gerada automaticamente com confiança média

1. Formulário “Cadastrar Nova Sessão de Votação”

Interface gráfica do usuário, Aplicativo

Descrição gerada automaticamente

1. De volta à home do módulo “Sessões de Votação”

Após cadastrada, a sessão é apresentada na home.

Tabela

Descrição gerada automaticamente

1. Formulário “Votar”

Interface gráfica do usuário, Texto, Aplicativo, Email

Descrição gerada automaticamente

1. De volta ao Módulo “Sessões de Votação”

Os votos cadastrados são exibidos na coluna “Votos”

Tabela, Linha do tempo

Descrição gerada automaticamente