

Democracia2.0

Dada a alta taxa de abstenção nas últimas eleições (44% em 2015, 51% em 2019 e 49% em 2022 ¹) e devido à insatisfação da população com a sua representação política, a Comissão Nacional de Eleições está a estudar um modelo alternativo de representatividade. A sua equipa foi escolhida para fazer o desenvolvimento da plataforma teste.

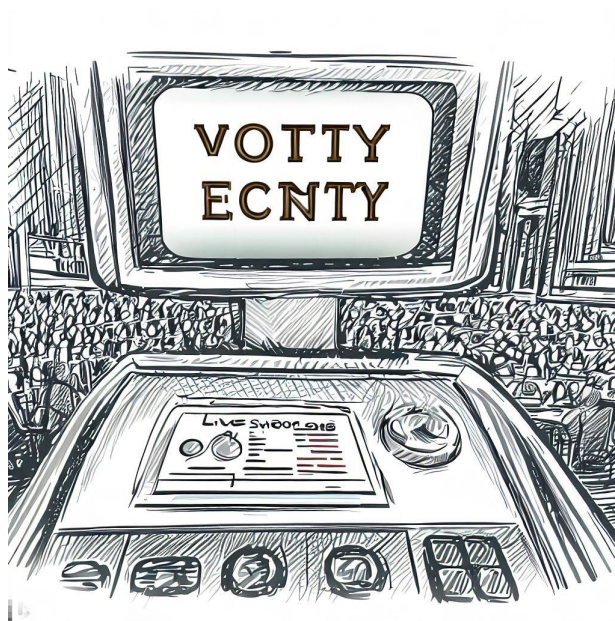


Figure 1: Ilustração da Assembleia Digital, obtida com o Bing Image Creator (<https://www.bing.com/images/create/>)

O modelo proposto: Democracia directa com representação opcional

No modelo proposto, pretende-se que os eleitores tenham a hipótese de votar directamente em todas as propostas de lei que vão a votação actualmente no parlamento. Assim, cada cidadão poderá fazer-se ouvir sobre cada assunto que lhe diz respeito. Este modelo é superior ao actual em virtude de que ao escolher um representante, o cidadão está a atribuir todos os seus votos à(s) mesma(s) pessoa(s), independentemente do tema. Com o novo modelo, o cidadão pode escolher uma sequência de votos que não corresponderia a nenhum deputado, até mesmo os que não o representam.

¹Fonte: Pordata, <https://www.pordata.pt/Portugal/Taxa+de+abstenç~ao+nas+eleiç~oes+para+a+Assembleia+da+República+total++residentes+em+Portugal++residentes+no+estrangeiro-2208>

No entanto, ao perder o conceito de deputados, também se perde a vantagem dos cidadãos não terem de se preocupar com todos os aspectos da governação e legislação. Muito do trabalho dos deputados é de desenho de propostas, estudo para compreensão das mesmas e estudo do impacto que têm na vida dos portugueses. De forma a manter a hipótese de delegar votos, é possível uma pessoa voluntariar-se como delegado (semelhante ao papel do actual deputado, mas sem eleições). Como delegado, os seus votos explícitos (i.e., que não foram copiados de outros delegados) passam a ser públicos. No entanto passa a existir a possibilidade de qualquer cidadão delegar os seus votos a um delegado, seja na totalidade, ou por tema (saúde, educação, obras públicas, imigração, etc...).

O projecto: Implementação do Software que faça a gestão dos eleitores, delegados, propostas e votos.

A sua equipa foi contratada para desenvolver este software, que se pretenda que tenha várias propriedades como segurança, privacidade, escalabilidade de performance (temos 10 milhões de habitantes com várias propostas por dia!), integração com outros sistemas do governo (como a autenticação por cartão de cidadão), etc...

1 Casos de Uso

Numa versão inicial pretendemos implementar alguns casos de uso, de forma a avaliar a proposta, deixando outros para segunda fase. No entanto, apresenta-se uma lista mais completa de casos de uso, de forma a que a equipa tenha uma noção do desenvolvimento futuro.

- A. Autenticação com o Cartão de Cidadão (<https://github.com/amagovpt/doc-AUTENTICACAO>).
- B. Apresentar-se como delegado. Um cidadão que seja delegado terá votos públicos em vez de votos secretos. Passa também a poder ser escolhido como delegado.
- C. Listar as votações passadas.
- D. (F_{2W} , F_{2R} , F_{2D}) **Listar as votações em curso.** Este caso de uso permite obter uma listagem das propostas de lei em votação neste momento.
- E. (F_{2W}) **Apresentar um projecto de lei.** Neste caso de uso, um delegado poderá propor um projecto de lei. Um projecto de lei é constituído por um título, um texto descritivo, uma data e hora de validade (máximo de um ano), um tema associado e o delegado proponente.
- F. (C) **Fechar projectos de lei expirados.** Todos os projectos de lei cuja data limite já tenha decorrido deverão ser fechados, sendo que não podem receber mais nenhuma assinatura.
- G. (F_{2W} , F_{2R} , F_{2D}) **Consultar projectos de lei.** Deve ser possível listar e consultar os projectos de lei não expirados.
- H. (F_{2W} , F_{2R} , F_{2D}) **Apoiar projectos de lei.** Cada projecto de lei pode ser apoiado por cidadãos (no limite de 1 apoio por cidadão). Quando um projecto de lei tiver pelo menos 10.000 apoios, é criada uma votação para esse projecto de lei imediatamente, com uma data de fecho igual à data de expiração do projecto de lei, com um limite mínimo de 15 dias e um limite máximo de 2 meses. Ao abrir a votação, é lançado automaticamente o voto do delegado proponente, como favorável.
- I. (F_{2W}) **Escolher delegado.** Um cidadão pode escolher vários delegados, mas apenas um para cada tema. Ou seja, pode ter um delegado para o tema de Saúde, um para Educação e um para outros temas. Quando fecha uma votação onde o cidadão não votou, é feito um voto automático com base no delegado do tema mais específico da proposta de lei.

- J. (F_{2W} , F_{2R} , F_{2D}) **Votar numa proposta.** Um cidadão deve pedir uma listagem das votações e escolher a que lhe interessa. Deverá ver qual o voto por omissão caso não o faça explicitamente (isto é, o voto do seu delegado para o tópico), caso este esteja disponível. Caso não concorde, o cidadão poderá lançar o seu voto (favorável ou desfavorável, não existem votos em branco nem nulos) numa proposta concreta. Deve ser verificado se o cidadão já votou nesta proposta em concreto, mas não deve ser registado em que opção foi o voto (dica: poderá ter de dividir o voto em duas partes, a do cidadão e a do conteúdo). Se o cidadão for também um delegado, então deverá ser registado um voto público, que qualquer um poderá confirmar.
- K. (C) **Fechar uma votação.** Assim que termina o prazo de uma votação, são atribuídos os votos dos delegados para cada cidadão que não tenha votado explicitamente. Depois são contados os votos, e se mais de metade dos votos forem favoráveis, então a proposta é fechada como aprovada. Caso contrário é fechada como rejeitada.

2 Requisitos Não Funcionais

O projecto tem ainda os seguintes requisitos:

- A autenticação deverá ser feita com cartão de cidadão, usando o protocolo OAUTH. Assim sendo, cada cidadão terá apenas o nome, o número do cartão de cidadão, e um token de autenticação. Esta funcionalidade será implementada apenas numa fase futura.
- Os votos dos não delegados deverão ser secretos. A base de dados não deverá guardar qual foi o voto de cada cidadão, apenas que votou, e o total dos votos favoráveis e desfavoráveis.
- Toda a informação deverá ser armazenada numa base de dados relacional.
- A plataforma deverá ser implementada em Spring Boot, de forma a ter um custo de desenvolvimento baixo, e a usar a linguagem Java, que é a linguagem para a qual é mais fácil de contratar engenheiros.
- A camada de dados deverá usar JPA e Spring Data.
- A camada de negócios deverá usar o Domain Model, com meta-dados suficientes para que o JPA faça o mapeamento para a base de dados.
- Deverá ser exposta uma API REST que os clientes (web ou mobile) poderão usar para interagir com a aplicação.
- O repositório deverá aceitar apenas código com o nível de qualidade aceite pela equipa (usando o pre-commit).
- O projecto deverá correr dentro de um docker, que será a forma como vai ser deployed no servidor de produção.
- O controlo da participação de cada membro da equipa será feita através da actividade no repositório git.

3 Tarefas

Nesta segunda fase do projecto queremos trabalhar sobre a versão anterior e trabalhar na camada de apresentação.

A equipa deverá:

- Desenvolver uma aplicação WEB com rendering Server-Side para os casos de uso identificados como F_{2W} .

- Desenvolver uma API REST para os casos de uso identificados como F_{2R} .
- Desenhar testes JUnit para cada um dos casos de uso anteriores.
- Desenvolver uma aplicação desktop usando JavaFX ou outra framework nativa, que permita executar os casos de uso F_{2D} . Deverá fazer mocks das APIs restantes.
- Implementar os casos de uso C à custa de Scheduled Tasks no Spring (<https://www.baeldung.com/spring-scheduled-tasks>).
- Escrever um relatório onde explica a arquitectura interna de cada um destes componentes.
- Fornecer um script `test.sh` que corra estes testes dentro de um container docker.
- Entregar um script `client.sh` que corra a interface gráfica (assuma ubuntu 22.04 ou um container docker).

4 Como entregar

Para entregarem o trabalho, basta identificarem a tag da fase1, e colocá-la no servidor. O repositório deve ter o código fonte do projecto, os ficheiros necessários para o correr no docker, bem como um único documento PDF na pasta docs, com todos os documentos necessários.

```
git tag fase2
git push origin fase2
```

5 Critérios de Avaliação

Para perceberem onde se devem focar, segue uma lista de critérios em que serão avaliados:

5.1 Processo de Desenvolvimento

- Granularidade dos commits
- Qualidade das mensagens de commit
- Qualidade do código presente no documento

5.2 Requisitos

- Em que medida os requisitos funcionais foram cumpridos.
- Em que medida os requisitos não-funcionais foram cumpridos.

5.3 Testes

- Cobertura dos casos de uso pelos testes escritos.
- Cobertura dos casos extraordinários pelos testes escritos.

5.4 Relatório

- Qualidade do Modelo de Domínio.
- Capacidade de perceber a organização em camadas da vossa aplicação.
- Qualidade do Diagrama de Classes.
- Qualidade do SSD pedido.
- Se é perceptível perceber o porquê das decisões tomadas no mapeamento para modelo relacional.

5.5 Design

- Qualidade da modelação do domínio.
- Qualidade da partição das classes em camadas.
- Qualidade da escrita dos casos de uso.
- Escolha acertada dos mapeamentos para a aplicação em causa.
- Uso correcto das primitivas do JPA/Spring Data.
- Uso correcto de queries complexas, em vez de manipulação de objectos um a um.
- Desenho de APIs REST que sigam a semântica REST.
- Uso correcto dos padrões de desenho e arquitecturais.
- Automatização dos testes (sem STDIN).

5.6 Interfaces

- Usabilidade das interfaces desenhadas.
- Capacidade de domínio do toolkit nativo.
- Uso correcto da semântica HTML.