

## Projeto individual (“Recurso”)

Updated: 2022-12-16.

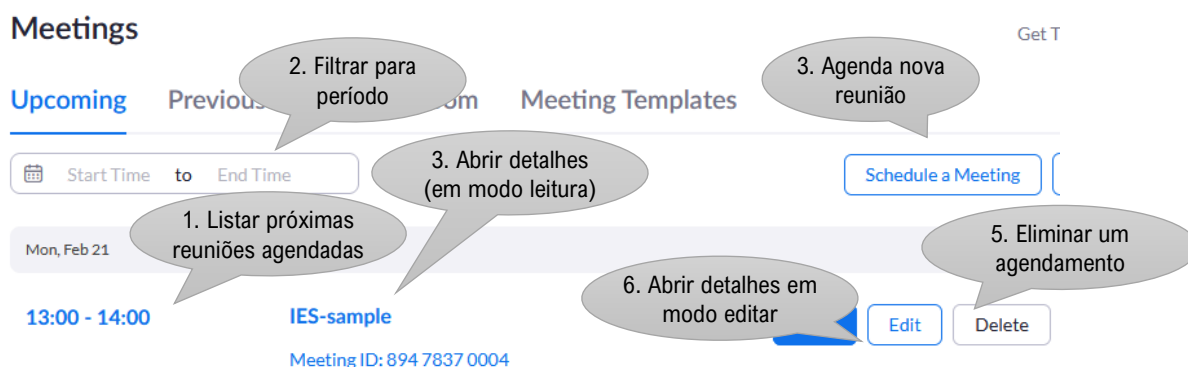
O Recurso da componente prática consiste no desenvolvimento de uma pequena aplicação multicamada em Spring Boot, com apresentação e defesa.

### Âmbito funcional da aplicação a desenvolver

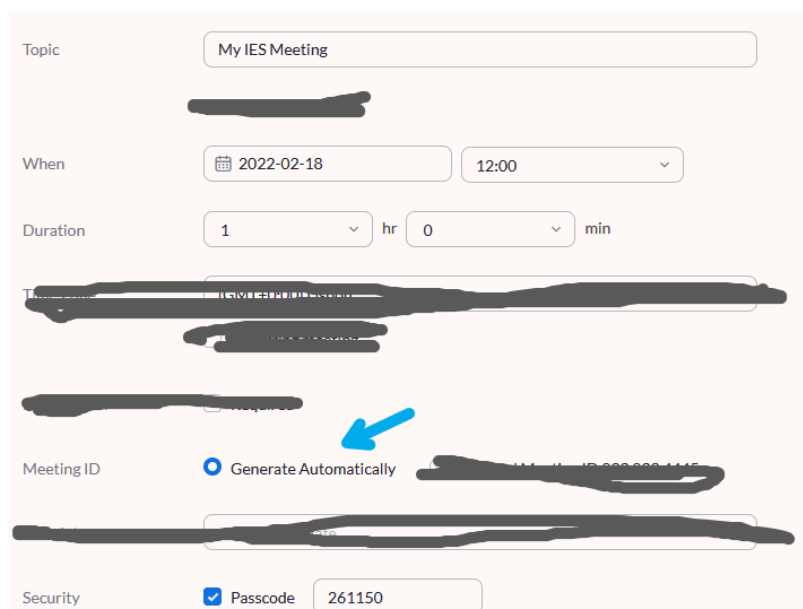
O projeto pretendido é um sistema de informação para agendamento de reuniões virtuais (relacionado: reuniões Zoom, Webex,...). Não é pretendido que a aplicação realize/controle as videoconferências; pretende-se apenas tratar do sistema de informação do agendamento.

**Conjunto mínimo** de casos de utilização da solução:

#### a) Gestão de marcações<sup>1</sup>



#### b) Detalhes (informação) para uma reunião



The screenshot shows a form for creating or editing a meeting. Fields include: 'Topic' (My IES Meeting), 'When' (2022-02-18, 12:00), 'Duration' (1 hr, 0 min), 'Meeting ID' (with a radio button for 'Generate Automatically' highlighted by a blue arrow), and 'Security' (checked 'Passcode' with value 261150). Several fields are redacted with black bars.

#### c) Para além das funcionalidades anteriores, a solução deve ainda:

- permitir criar e gerir (fichas de) Contactos (e.g.: nome e email)

<sup>1</sup> O *screenshot* incluído é para contextualizar; a solução não precisa de se parecer com esta interface.

- associar contactos a uma reunião, i.e., definir a lista de participantes, “escolhendo-os” da lista de contactos.
- duplicar (“*clone*”) um agendamento, criando uma nova entrada repetindo a informação de uma existente, podendo ser alterada.

**d) [Opcional] Valorização**

- integrar com um sistema de videoconferência [externo], isto é, permitir iniciar uma reunião da lista, transportando a informação possível (nome, pin de acesso,...)

## **Requisitos técnicos da implementação**

A solução deve seguir uma arquitetura de camadas, usando o *framework* Spring Boot:

- O armazenamento deve ser feito numa **base de dados** persistente.
- Deve implementar uma **API (REST) para possível integração** com outras aplicações, permitindo a **gestão completa** da solução através de um cliente externo (e.g.: via Postman).
- O *front-end Web*, não precisa de ser feito necessariamente com a tecnologia Java/Spring Boot, podendo ser usado um *framework* baseado em JavaScript.

**Práticas necessárias:**

- Gestão de versões do código num repositório Git, com *feature-branching*.
- Ambiente de execução baseado em containers Docker. Incluir no repositório configurações/scripts necessários.
- A instalação pode ser local, mas deve ser (facilmente) replicável noutro ambiente.