

## Projeto 2

- Desenvolver uma aplicação (**API REST + frontend**) que atenda às seguintes especificações:

### Requisitos funcionais

1) Permitir envio de imagem (upload), informando o seu tipo:

- Para a imagem serão permitidos os formatos jpeg, png, tiff e pdf. Desta forma, será necessário fazer uma validação e notificar com status de erro e mensagem (feedback).
- O tipos de imagens suportados são CNH (Carteira nacional de habilitação), RG (Registro geral), CR (Comprovante de residência). Também será necessário validar e notificar (feedback) o tipo caso não seja um desses.
- A aplicação deve persistir em uma tabela de banco de dados as seguintes informações:
  - o Caminho do arquivo
  - o Tipo
  - o Data de criação
- O diretório que contém as imagens salvas no backend deve ter nome do tipo. Por exemplo, considere que foram cadastradas três imagens, um RG e duas CNH, os caminhos seriam os seguintes:
  - o C:/diretorioRaiz/CNH/imagem\_cnh\_000.tiff
  - o C:/diretorioRaiz/CNH/imagem\_cnh\_001.tiff
  - o C:/diretorioRaiz/RG/imagem\_rg\_000.tiff
- A requisição deve retornar um identificador único como resposta para cada envio realizado

2) Permitir download de imagem

- para um dado identificador único informado na requisição, deve-se retornar o binário da imagem, mapeada por esse identificador, como download na resposta
  - o o identificador usado como chave para busca e download é o mesmo retornado na resposta da requisição de upload
- caso não seja encontrada imagem para o identificador informado, deve-se notificar com status de erro mensagem (feedback)

3) Permitir consulta de imagens (com seus respectivos tipos)

- Consulta com filtros opcionais de tipo e data
- O resultado (em estrutura de JSON) deve conter todos itens que se enquadrem na pesquisa, ou seja:
  - o Se não for usado nenhum filtro, retorna tudo;
  - o Se usar apenas o filtro tipo, retorna todos itens do respectivo tipo;
  - o Se forem usados de data (data inicial e final), retornar todos os itens no intervalo definido.
    - Se for preenchida apenas a data inicial, retorna todos criados a partir da data inicial;
    - Se for preenchida apenas a data final, retorna todos criados até a data final;

- Se forem preenchidos ambos, retorna todos criados neste intervalo. Lembrando que data inicial não pode ser maior que a final, ou seja, será necessário validar e notificar (feedback)
- o Podem ser combinados os filtros de data com o de tipo. Neste caso se aplicam as regras de ambos.

**Observações:**

- A API **deve** ser desenvolvida na linguagem Java (versão 8);
- **Utilizar o framework Spring** (uso do Spring boot é opcional);
- **Utilizar banco de dados** (H2, MSSQL, MySQL ou PostgreSQL). As credenciais para conexão com BD devem ser configuráveis (via arquivo de propriedades);
- Diretório raiz, onde devem ser armazenadas as imagens, deve ser configurável via arquivo de propriedades;
- Frontend preferencialmente com framework angular...