# **JAVA / SPRING BOOT / DOCKER / KUBERNETES**

1. Classes DTO serão utilizadas como retornos JSON nos métodos da camada Controller.
2. NullPointerException Ocorre quando chamo um método de uma classe não instanciada.
3. Quando um atributo ou lista for utilizado em mais de um método, define-se o atributo/lista como estático.
4. O @PostConstruct faz com que o método seja executado logo depois que o contêiner inicializa a classe que o invoca.
5. Todos os Beans do Spring são instanciados na inicialização da aplicação;
6. Comlo converter entidades para DTOs sem código boilerplate com ModelMapper:
   1. <https://www.youtube.com/watch?v=HU7bfKG8nV4> ou,
   2. List<CategoryDTO> listDTO = list.stream().map(obj -> service.fromDTO(obj)).collect(Collectors.toList());
7. \*\* retirado do Projeto: Api-library ou,

<https://www.youtube.com/watch?v=B1taCPYZvJM>

1. **Atributos Statics** :
   1. Para todos os objetos dessa classe, o valor do atributo static esta apontado para o mesmo endereço de memória, ou seja, o valor do atributo é o mesmo para todos os Objetos.
   2. Quando em algum objeto, o atributo static for alterado, essa alteração valerá para todos os demais objetos.
   3. Não é necessário instanciar o objeto para utilizar o atributo;
2. **Método Static:** 
   1. Não é necessário instanciar o objeto para utilizar o método;
   2. Um exemplo são os métodos da classe Math;
3. **Para verificar se uma porta esta executando:**
   1. Abra o CMD como Adm;
   2. execute: netstat -o -n -a;
   3. Procure na coluna Endereco local o Ip que esta executando a porta. exemplo:
      1. Protocolo                End Local                                                  PID
      2. TCP                         0.0.0.0:3306                                             6336
   4. Para Alterar a porta que irá rodar o projeto:
      1. No application.properties incluir : server.port=8081
4. Para Para converter a lista de objetos do tipo Categoria para CategoriaDTO:
   1. \*\* o objeto categoriaList é uma Lista do tipo <Categoria>
   2. List<CategoriaDTO> categoriaDtoList = categoriaList.stream().map(obj -> new CategoriaDTO(obj)).collect(Collectors.toList());
5. GIT
   1. Git Init
   2. Git config –global user.name “FlvAvsGit”
   3. Git config –global user.email [flvavsgit@gmail.com](mailto:flvavsgit@gmail.com)
   4. Git remote add origin <https://github.com/FlvAvsGit/nome-do-repositorio-remoto>
   5. Git Status
   6. Git Add .
   7. Git commit -m “Commit inicial”
   8. Git push -u origin master