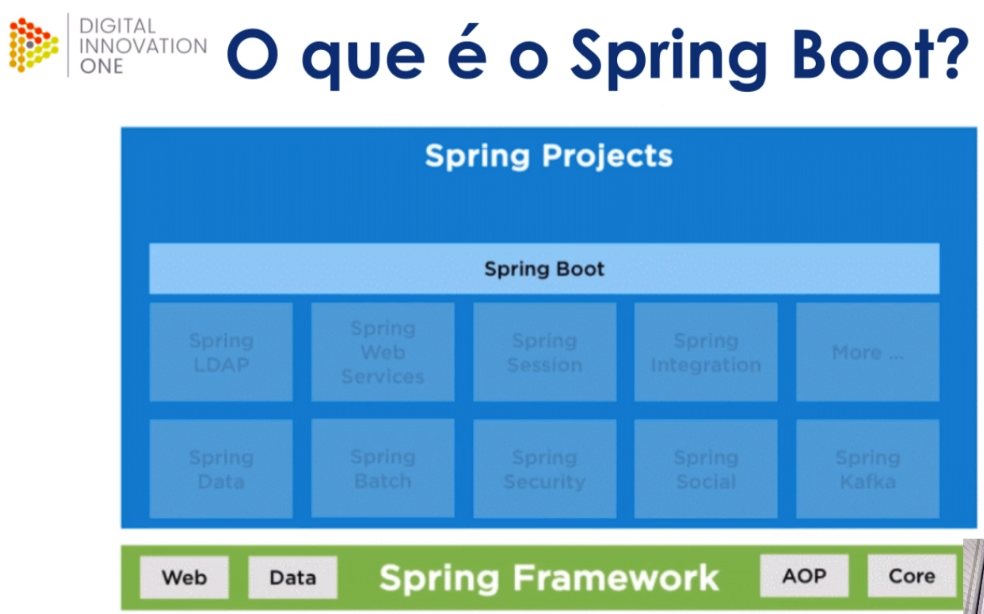
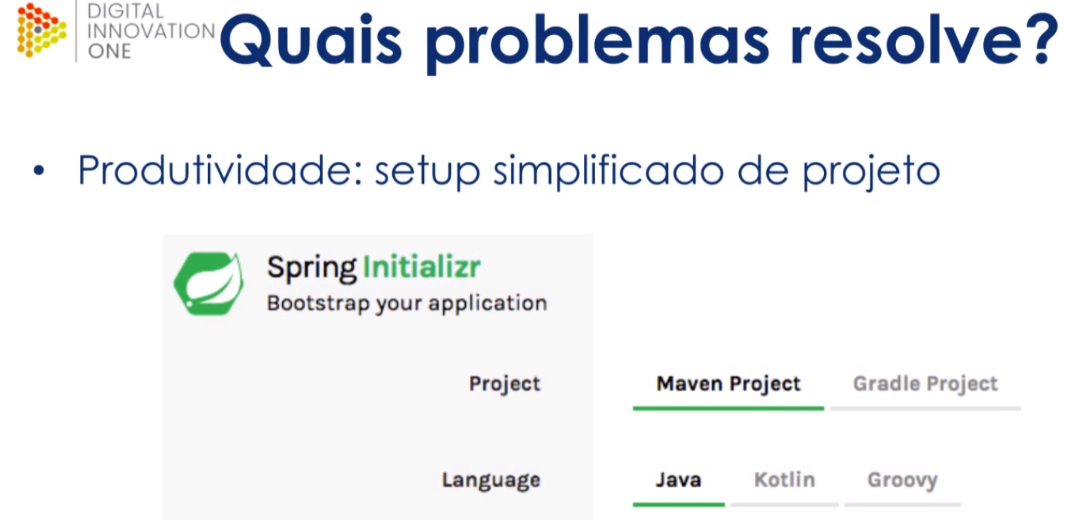
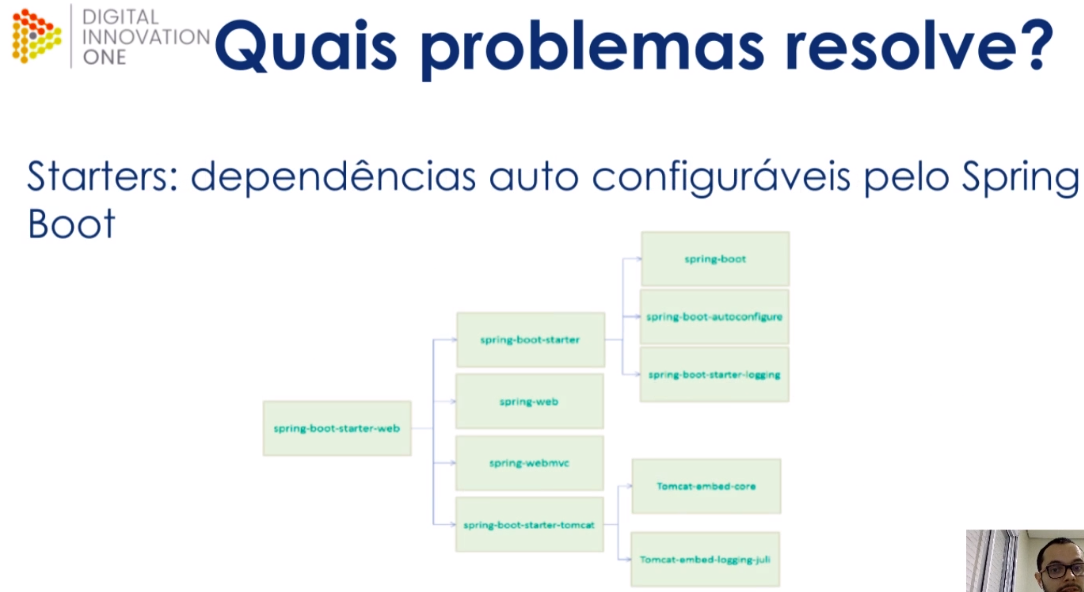
**SPRINGBOOT**

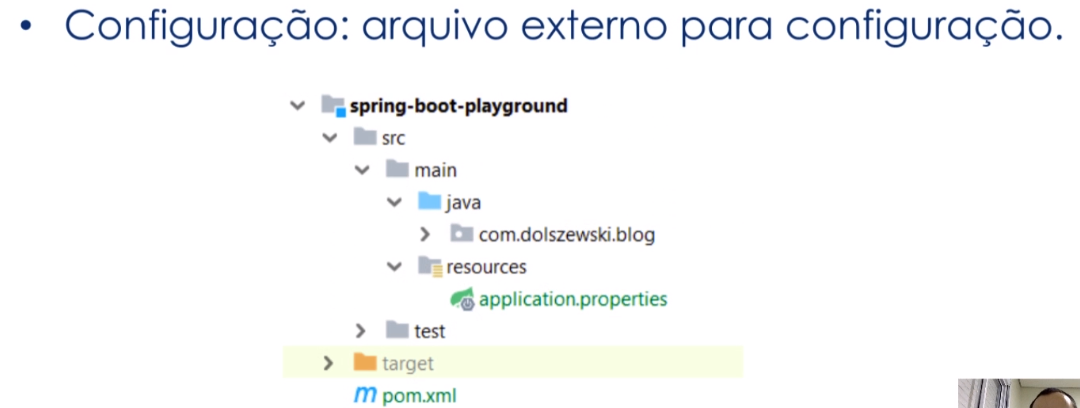
****

****

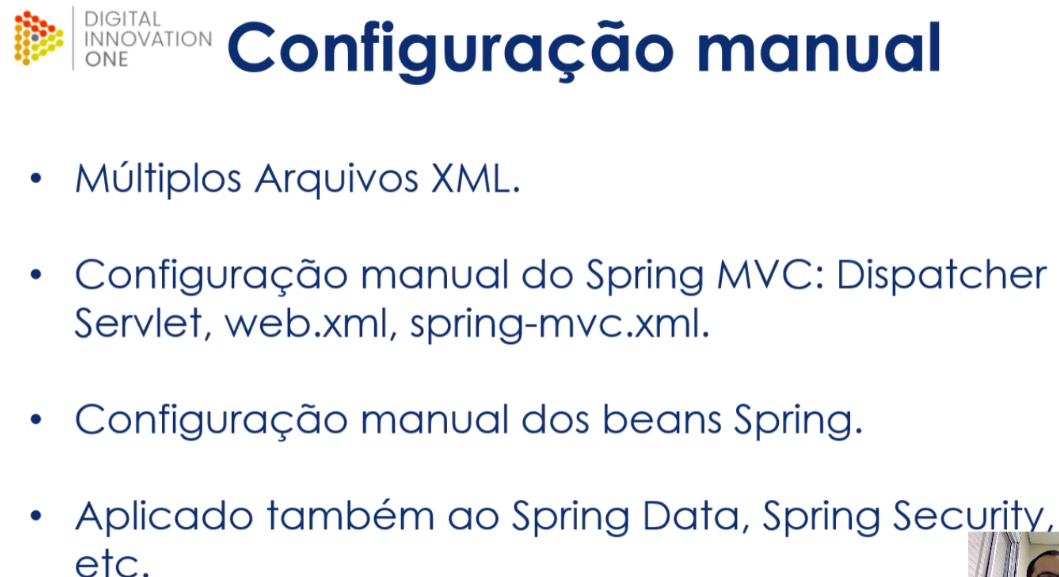
****

****

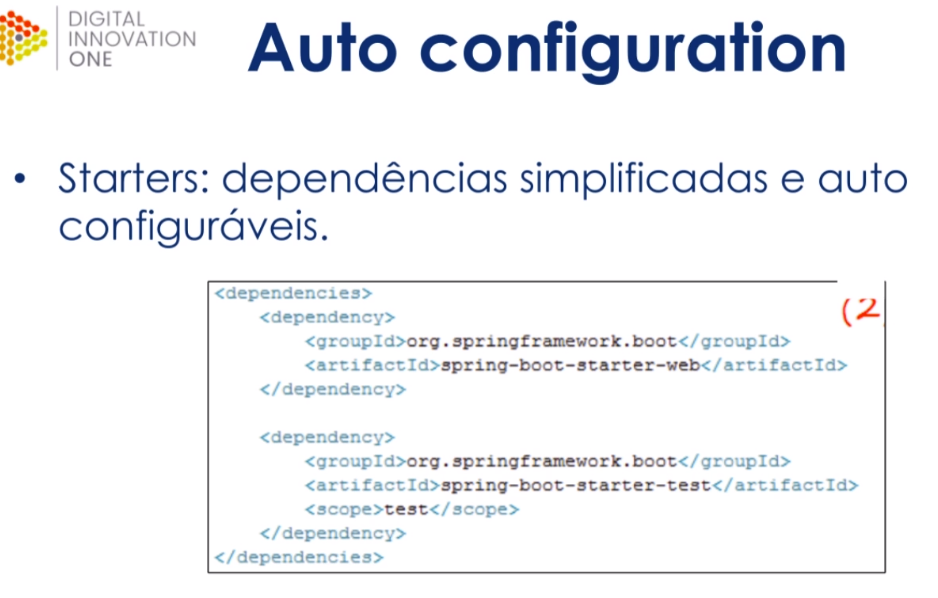
**Configuração do projeto na classe aplication.properties**

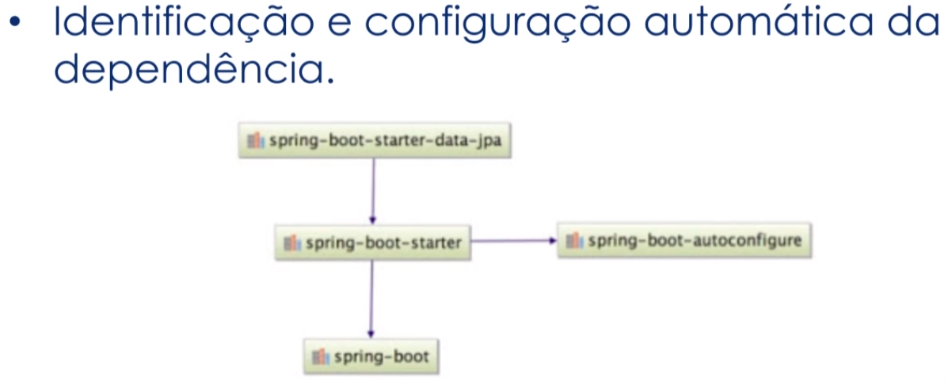
****

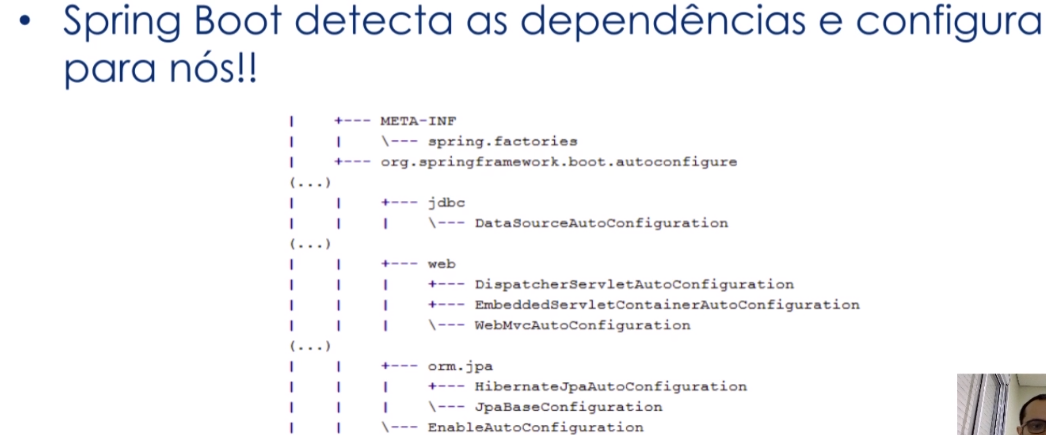
**Configuração manual**

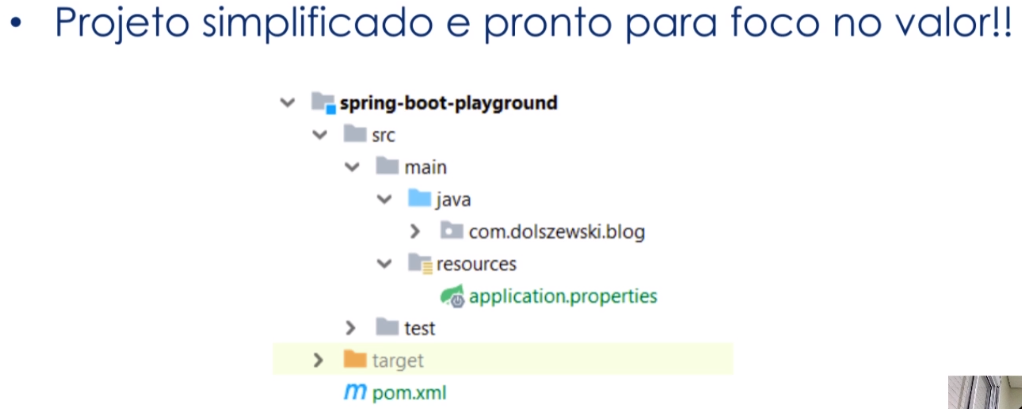
****

**Auto configuration**

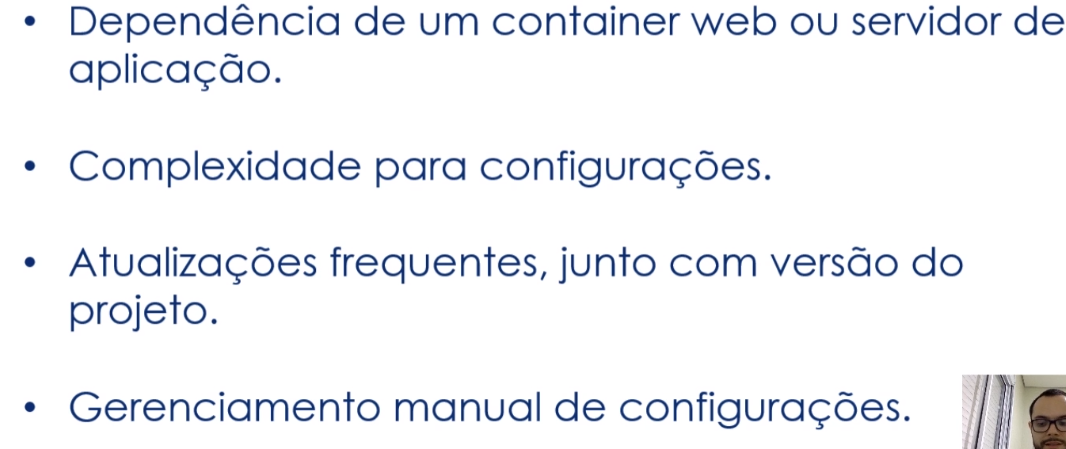
****

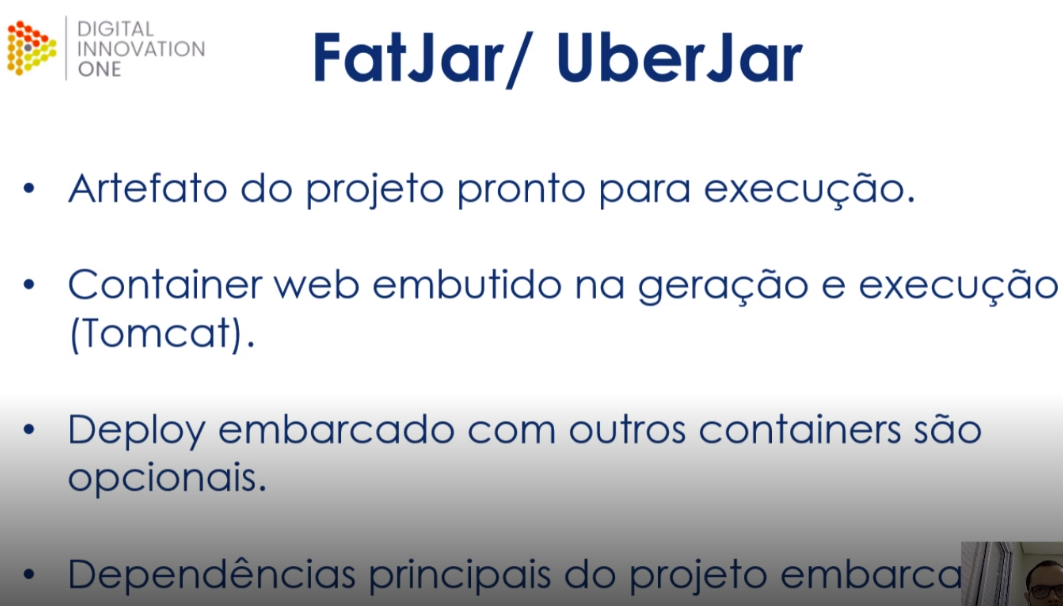
****

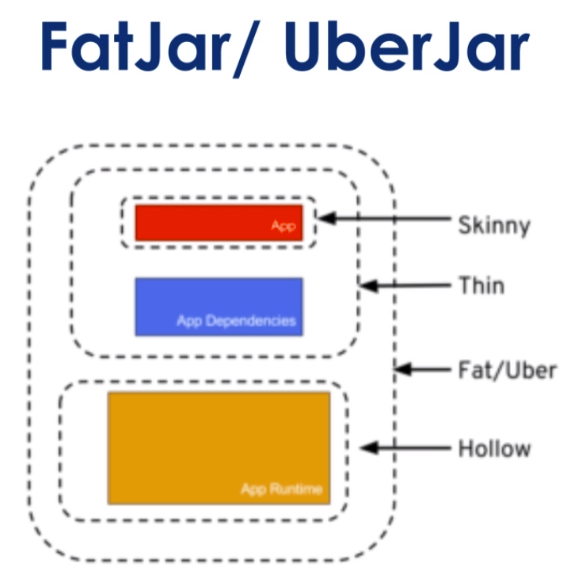
****

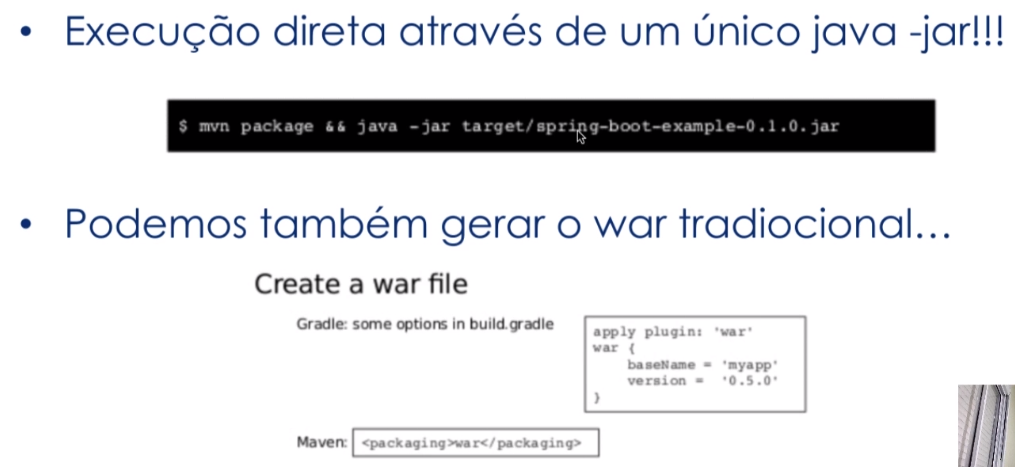
****

****

****

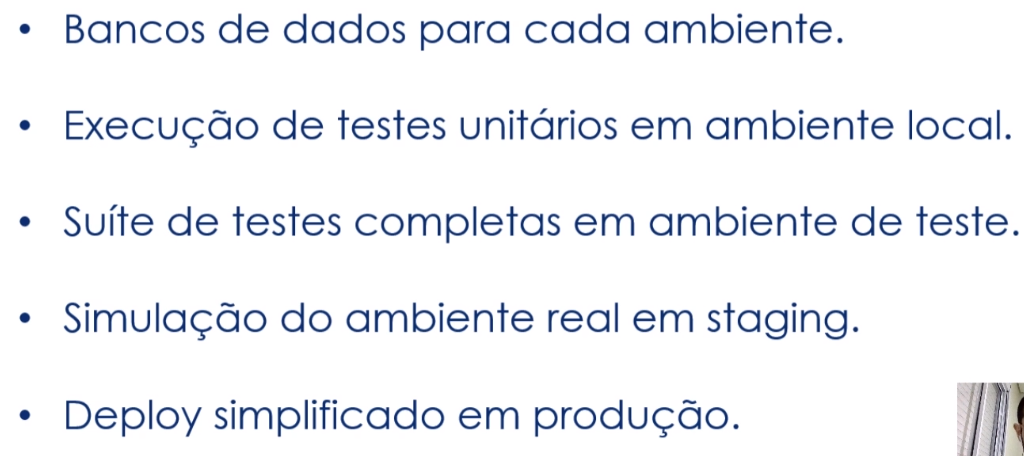
****

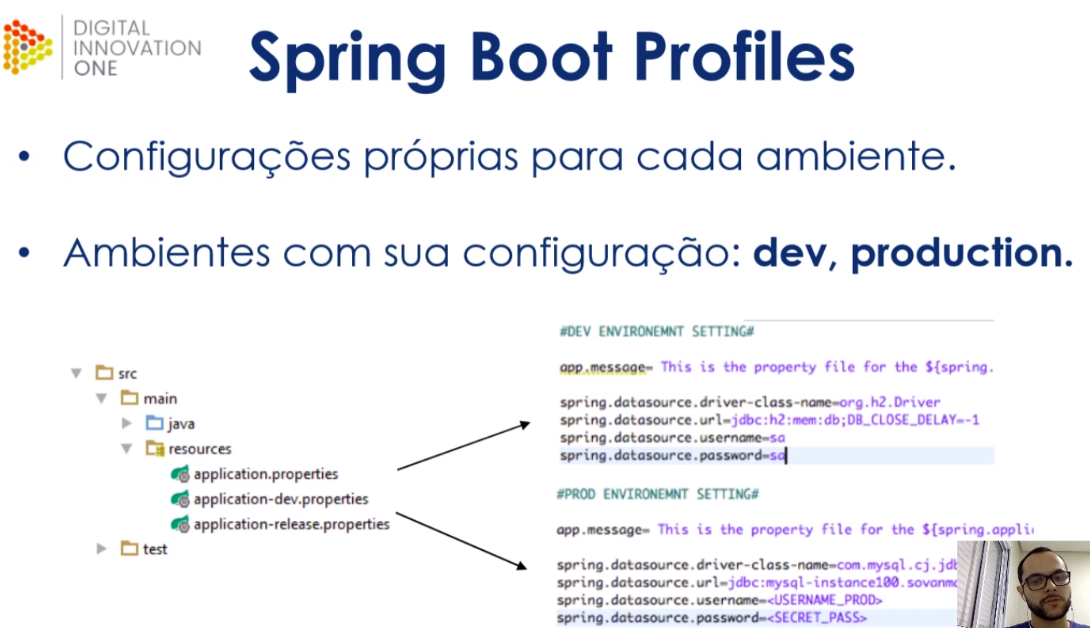
****

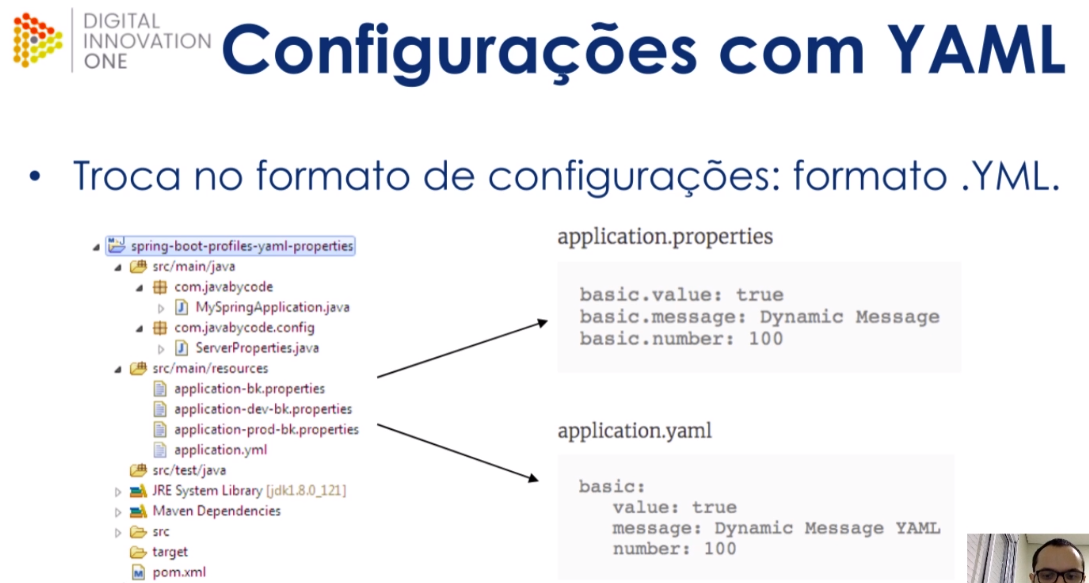
****

**Profiles e Configurações**

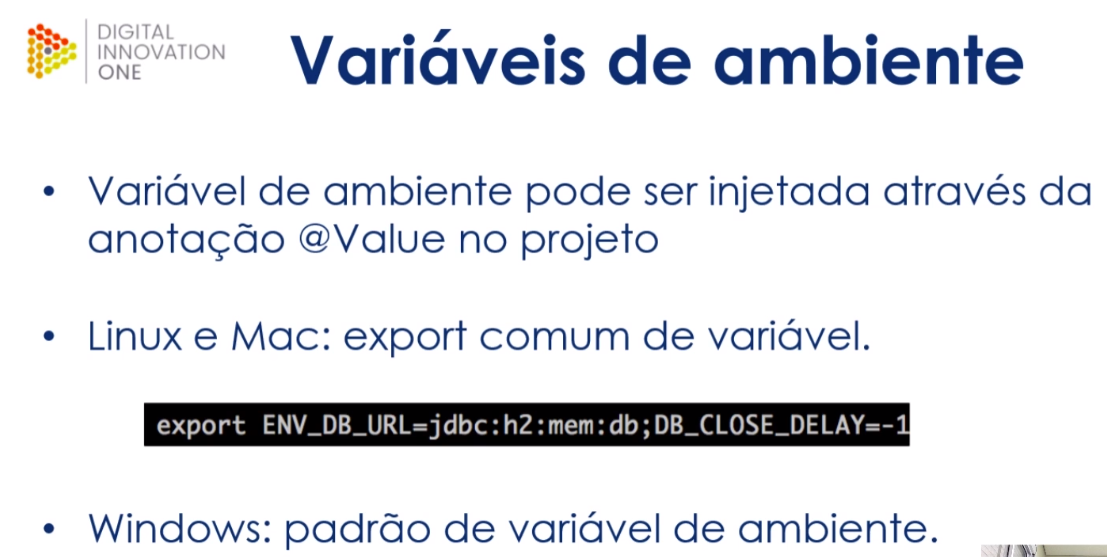
****

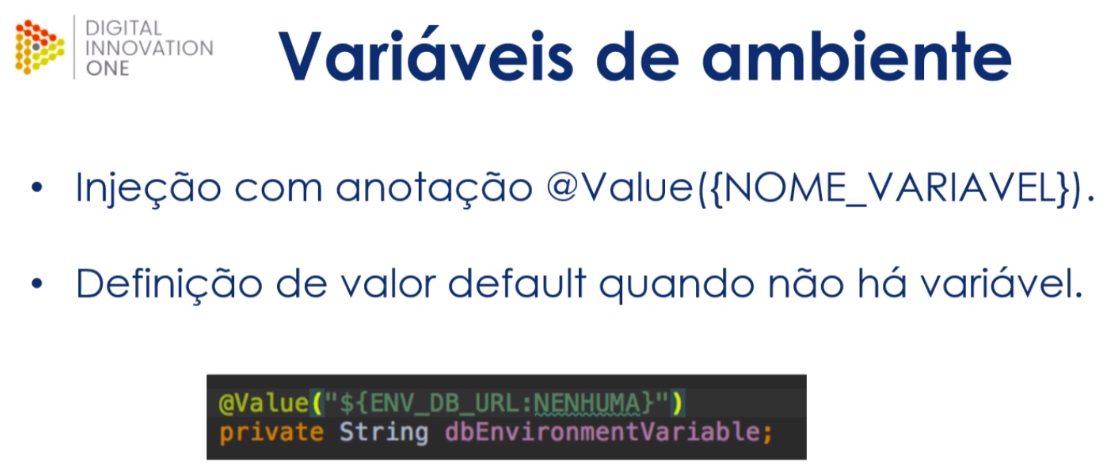
****

****

****

**Injetando variáveis de ambiente**

****

****

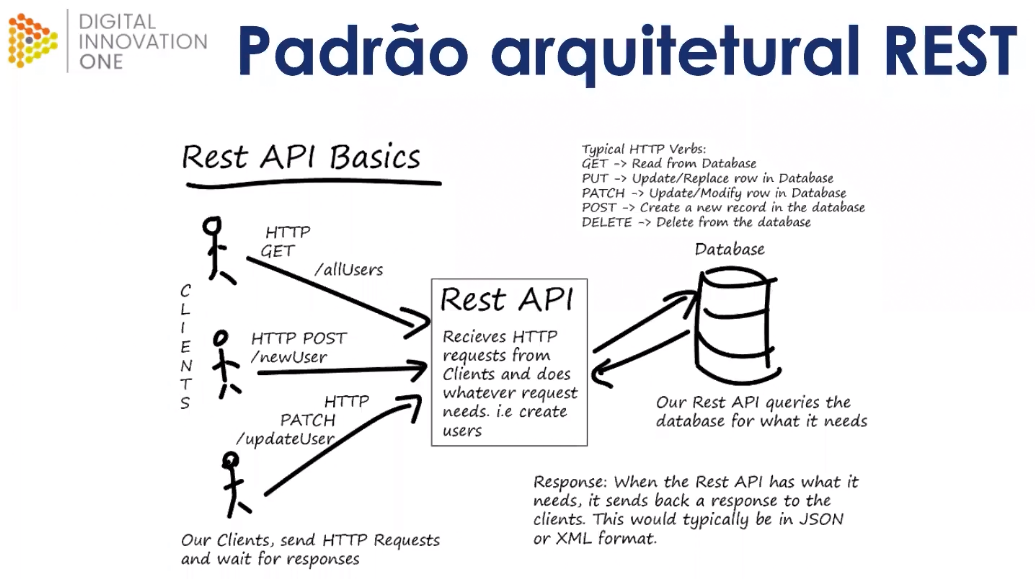
**SPRING REST API**

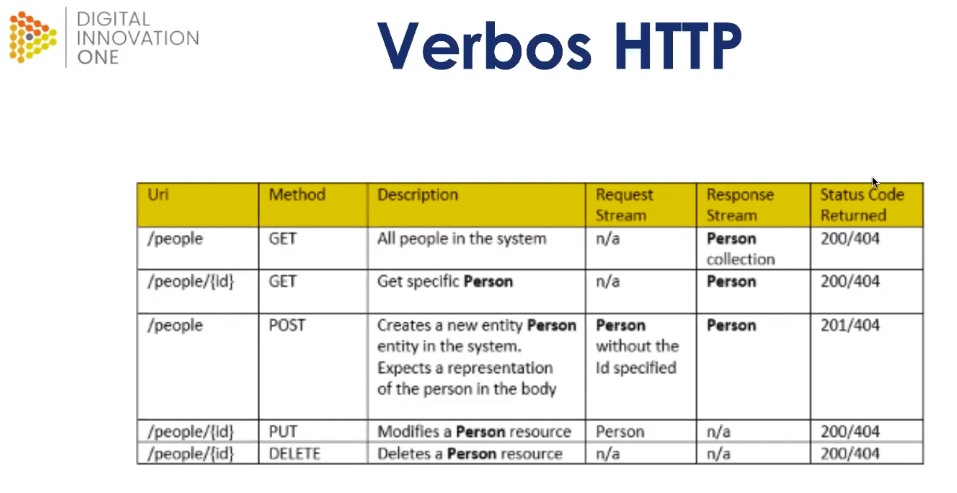
* **Entity:** Essa classe se refere a entidade (tabela) do nosso banco de dados, onde contem todos os seus atributos. Utilizamos a anotação @GeneretedValue para gerar automaticamente em nossa chave primária o seu autoincremento utilizando o tipo Generation.Type.IDENTITY, a anotação @Column passamos parâmetros de configuração, como nullable que informa que o campo não pode ser null, ou o name da coluna, caso seja inserido um nome diferente do que foi cadastrado no banco. Usamos também a anotação @Data para gerar nossos construtores, getters e setters, métodos equals e haschode, usamos a anotação @Entity para informar ao Spring que nossa classe é do tipo entidade.
* **Repository:** Classe responsável pela comunicação com o banco de dados, deve entender a classe JpaRepositry receber como parâmetros a entidade e o tipo da chave (ex: <Usuario, Integer>), essa classe nos traz alguns métodos prontos para criarmos nosso projeto, como o save, delete, findById que são da Interface CrudRepository. Essa classe utiliza o padrão DAO, faz o CRUD e demais operações.
* **Service:** Classe responsável por gerenciar as regras de negócio, utilizamos a anotação @Service para indicar que esta classe será gerenciada pelo Spring e que é uma classe de serviços, usamos a classe @AllArgsConstructor para que o Spring possa instanciar implicitamente os construtores e passando como parâmetros o @Autowired ele injeta as dependências automaticamente no objeto instanciado.
* **Controller:** Classe responsável por controlar as requisições em seus endpoints, usamos a anotação @RestController indicando ao Spring que esta classe é uma controladora que tem comportamento específico para API Rest e usamos também a anotação @RequestMapping passando o caminho de acesso a nossa API e usamos também a anotação @AllArgsConstructor para que o Spring possa instanciar implicitamente os construtores e passando como parâmetros o @Autowired ele injeta as dependências automaticamente no objeto instanciado.

**Nos métodos para o CRUD, usamos as anotações abaixo**

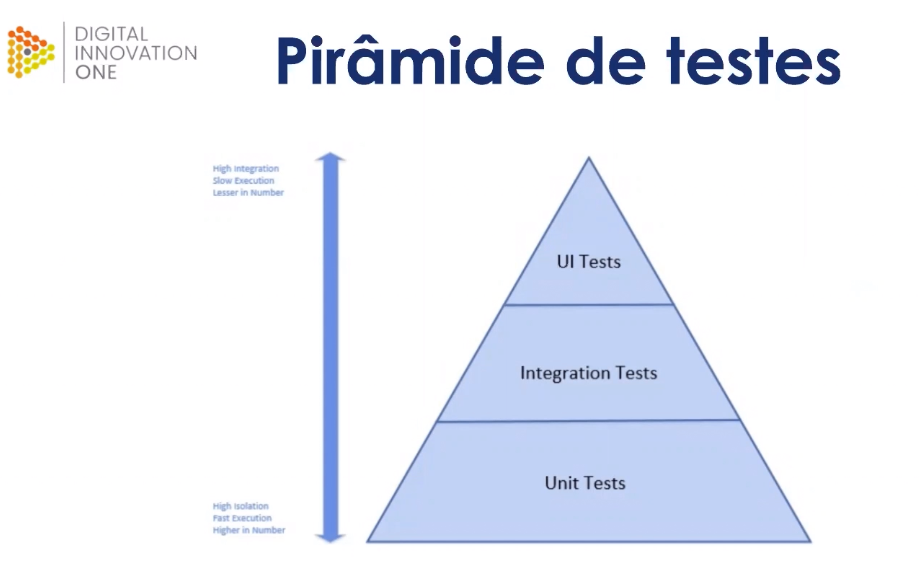
* **@Postmapping:** Anotação responsável por fazer a criação do objeto informado como parâmetro para o método que a utiliza, é usada para mapear solicitações HTTP POST.
* **@ Getmapping:** Anotação responsável por fazer a busca do objeto informado como parâmetro para o método que a utiliza,é usada para mapear solicitações HTTP GET em métodos manipuladores específicos.
* **@DeleteMapping: Anotação responsável por fazer o delete** do objeto informado como parâmetro para o método que a utiliza**,** é usada para mapear solicitações HTTP DELETE em métodos manipuladores específicos.
* **@Putmapping:** Anotação responsável por fazer a atualização do objeto informado como parâmetro para o método que a utiliza, é usada para mapear solicitações HTTP PUT em métodos manipuladores específicos.
* **@RequestBody: Anotação usada para indicar** que um parâmetro de método deve ser associado ao valor do corpo da solicitação HTTP.
* **@PathVariable:**Esta anotação é usada para anotar os argumentos do método do manipulador de solicitações. A anotação @RequestMapping pode ser usada para manipular alterações dinâmicas no URI, onde determinado valor de URI atua como um parâmetro.

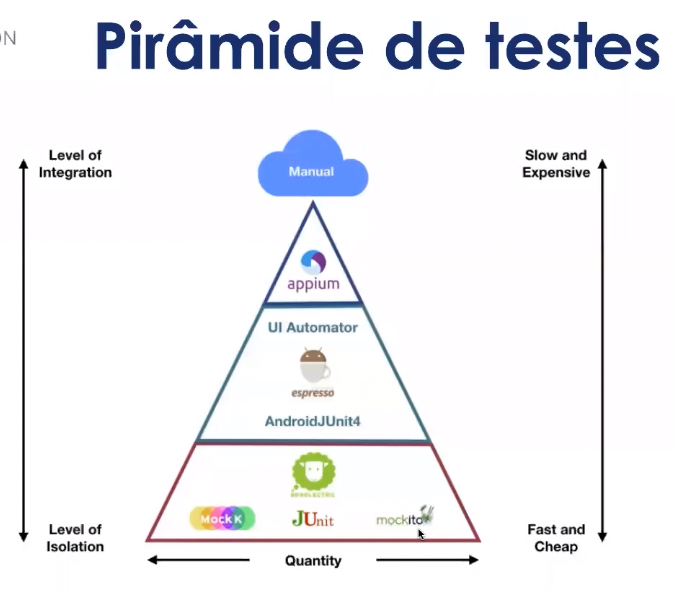
**Padrão de Arquitetura REST**

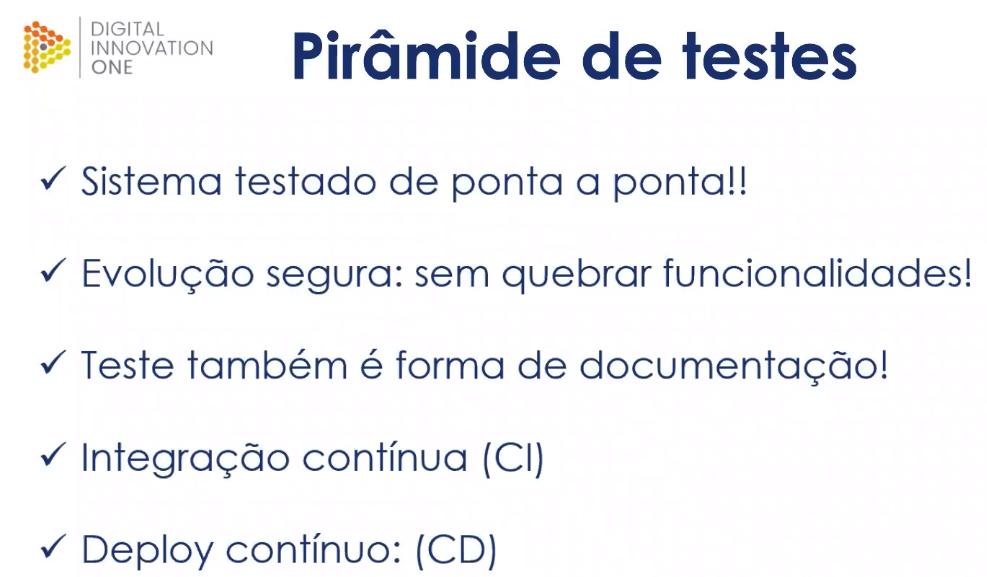
****

****

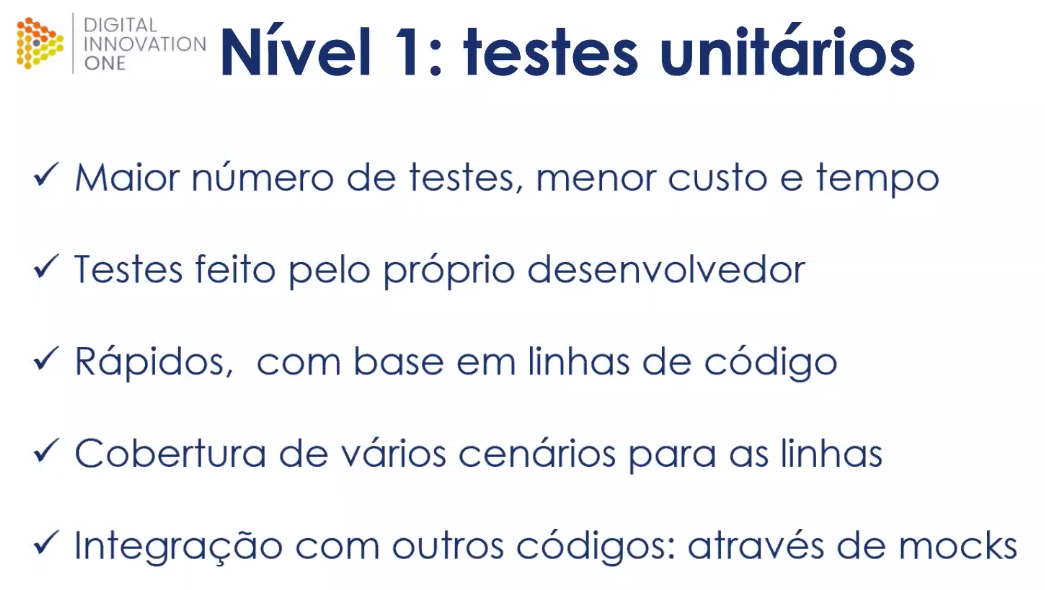
**Pirâmides de Testes**

****

****

****

**Níveis de Testes**

****