**Java Spring**

**Dependências iniciais**

* **Spring Boot DevTools: Pra não precisar reiniciar a aplicação a cada alteração que fizer no código**
* **Lombok: Gera os códigos verbosos do java baseado em anotações**
* **Spring Web: modulo web para a aplicação**

**Dependencias para validação e persistencias no banco de dados**

* **Validation I/OBean Validation with Hibernate validator: Valida dados de entrada da aplicação**
* **MySQL Driver: driver do banco de dados mysql**
* **Spring Data JPA SQLPersist data in SQL stores with Java Persistence API using Spring Data and Hibernate.**

**controllers: Classes que mapeamos nossas rotas**

**Record é um recurso que permite representar uma classe imutável, esse tipo de classe se encaixa perfeitamente para representar classes DTO que servem apenas para representar dados que serão recebidos ou devolvidos pela API, sem nenhum tipo de comportamento.**

**Migrations com Flyway: Controle de versões do banco de dados**

**Utilizar o flyway para fazer o controle de criação e alterações das tabelas no banco de dados**

**As migrations ficam do diretorio de resources/db/migrations**

**Criar arquivo tipo sql com nome descritivo: V1\_\_create-table-medicos.sql**

**Migrations que já foram executadas não podem ser editadas!**

**Validação com Bean Validation**

**Aplicar anotaçoes de validação nos record/dto da aplicação**

**@NotBlack: String não pode ser nulo e nem vazio.**

**@Patter: recebe uma expressão regular como parâmetro**

**@Pattern(regexp = "\\d{8}"): Valida se está sendo enviado oito dígitos (numéricos)**

**@Pattern(regexp = "\\d{4,6}"): Valida se está sendo enviado de 4 a 6 dígitos (numéricos)**