Qual é o desafio?

Criar uma API Rest utilizando Java e Spring + Hibernate que realizará o controle da aplicação de vacinas entre a população brasileira, ou seja, realizará o gerenciamento do cadastro dos usuários e do cadastro de aplicação da vacina.

Nome: Laís Lima Santos

E-mail: lais\_lima@usp.br / laissp25@gmail.com

Linkedin: https://www.linkedin.com/in/lais-lima-santos/

GitHub: https://github.com/LaisLimaSantos /

Repositório do Projeto: <https://github.com/LaisLimaSantos/ControleVacinas>

Apresentação

Olá, prazer em conhecer!!

Me chamo Laís Lima, tenho 26 anos, moro na Cidade de São Paulo e sou Desenvolvedora Full Stack Java Jr.

O texto apresentado aqui mostra a criação e implementação de uma API Rest, criada usando Java e Spring+Hibernate, que visa realizar o controle de aplicação de uma vacina entre os brasileiros! #VacinaSim #VacinasSalvamVidas #VivaaCiência #ATerraNãoéPlana :D

Esse desafio faz parte do processo seletivo do programa “Orange Talents” da consultoria Zup.

Mas o que é uma API Rest?

Uma API é uma expressão do inglês “Application Programing Interface” (quem em português significa Interface de Programação de Aplicações). Rest também é um termo do inglês Representational state transfer (Transferência de Estado Representacional) criado no ano 2000 por Roy Fielding em sua tese de doutoramento na qual ele descreve um design de arquitetura de software construído para servir aplicações em rede.

Então uma “API Rest” trata-se de um conjunto de rotinas e padrões estabelecidos e documentados por uma aplicação A, para que outras aplicações consigam utilizar as funcionalidades desta aplicação A, sem precisar conhecer detalhes da implementação do software.

Pode ser um pouco confuso de entender esses conceitos, então vou dar alguns exemplos:

A API do Facebook, que permite que você se autentique em aplicações externas ao Facebook;

API’s de previsão do tempo, como a ClimaTempo e a OpenWeatherMap;

API’s do Google, como o Google Drive e Google Heart;

As API’s de Inteligência Artificial Watson, da IBM.

Agora começarei a abordar a criação do Projeto, então vamos lá?

Tecnologias utilizadas:

Linguagem de programação: Java

Ecossistema: Spring

Spring Boot

Sistema gerenciador de banco de dados: Mysql

Dependências: Spring Boot Dev Tools, Spring Web, Spring Data JPA, Mysql Driver, Spring Security

Gerenciador de dependências: Maven;

Deploy: Heroku

Mas afinal, para que servem essas ferramentas?

* - Linguagem de programação Java: Uma linguagem de programação é um conjunto de símbolos e códigos que serve, resumidamente, para fornecer “instruções” para o computador executar determinadas tarefas. É o “idioma” que usamos para nos comunicar com a máquina.
* Existem várias linguagens de programação, porém neste projeto iremos utilizar apenas a linguagem Java, que se trata de umas linguagens mais usadas atualmente, por ser muito versátil, segura e já possuir diversas ferramentas associadas que otimizam o processo de desenvolvimento;
* - Sistema Gerenciador de Banco de Dados: É um conjunto de softwares que possui recursos para manipularmos o nosso Banco de Dados, como criar novas tabelas, colunas, deletar conteúdos, entre outras funções;
* - Spring: é um framework (um framework é, resumidamente, um conjunto de bibliotecas ou componentes que são utilizados para criar um sistema base, otimizando o desenvolvimento de uma api) de código aberto baseado na “injeção” de dependências, que são algumas funcionalidades prontas. As que utilizaremos nesse projeto são:
  + Spring Boot Dev Tools: ajuda a tornar a reinicialização dos aplicativos mais rápida e outras configurações para desenvolvimento
  + Spring Web: dependência específica para projetos web, incluindo RestFull, usando o Spring MVC (framework que ajuda construção de aplicações web, tornando-as mais flexíveis)
  + Spring Data JPA: é um framework que auxilia o gerenciamento dos nossos repositórios.
  + MySQL Driver: faz a nossa conexão com o banco de dados.
  + Validation: para validação dos dados da nossa aplicação.
* Maven: é um gerenciador de dependências, que realiza automaticamente atualizações de nossas dependências.
* Heroku: ferramenta que realizará o deploy (implantação) de nossa API na web.

//\*Seu desafio:

Dado que você fosse implementar esse sistema utilizando Java como linguagem e Spring + Hibernate como stacks de tecnologia fundamentais da aplicação: Escreva um post de blog explicando de maneira detalhada tudo que você faria para implementar esse código.

No texto, queremos que você:

Explique quais as tecnologias do mundo Spring você usaria;

Conte qual o papel das tecnologias escolhidas e quais benefícios elas trazem para a implementação do código;

Diga quais classes seriam criadas nesse processo e traga trechos autorais explicando a construção de cada classe do código;

Explique as etapas do processo de construção do seu código e como faria a implementação do sistema na Web;

Traga trechos de códigos autorais que justifiquem a implementação de cada tecnologia (não precisamos do código inteiro);

Os itens acima são obrigatórios, mas não se limite. Se quiser escrever mais, fique à vontade (não existe quantidade máxima de páginas ou caracteres).

Dicas:

Não precisamos saber como instalar cada ferramenta, queremos saber como você implementaria o seu código.

Ao justificar as tecnologias, mostre a sua visão e entendimento de cada uma e não conceitos prontos já conhecidos.

Cite as tecnologias que você utilizaria em seu projeto e porquê elas fazem parte do escopo/construção do seu código, pois ficaremos de olho nesse ponto.

Suas classes e atributos podem ser em português ou em inglês, não se apegue a isso, queremos um bom resultado final.

Não queremos um manual de programação e sim um post explicando como você faria a implementação da API passo a passo.

Mais do que respostas certas, queremos ver o quão nerd você consegue ser e o que considera relevante. O importante é mostrar que consegue aprender, se fazer entender e ainda transmitir conhecimento.

Se ficou fácil, te desafiamos a:

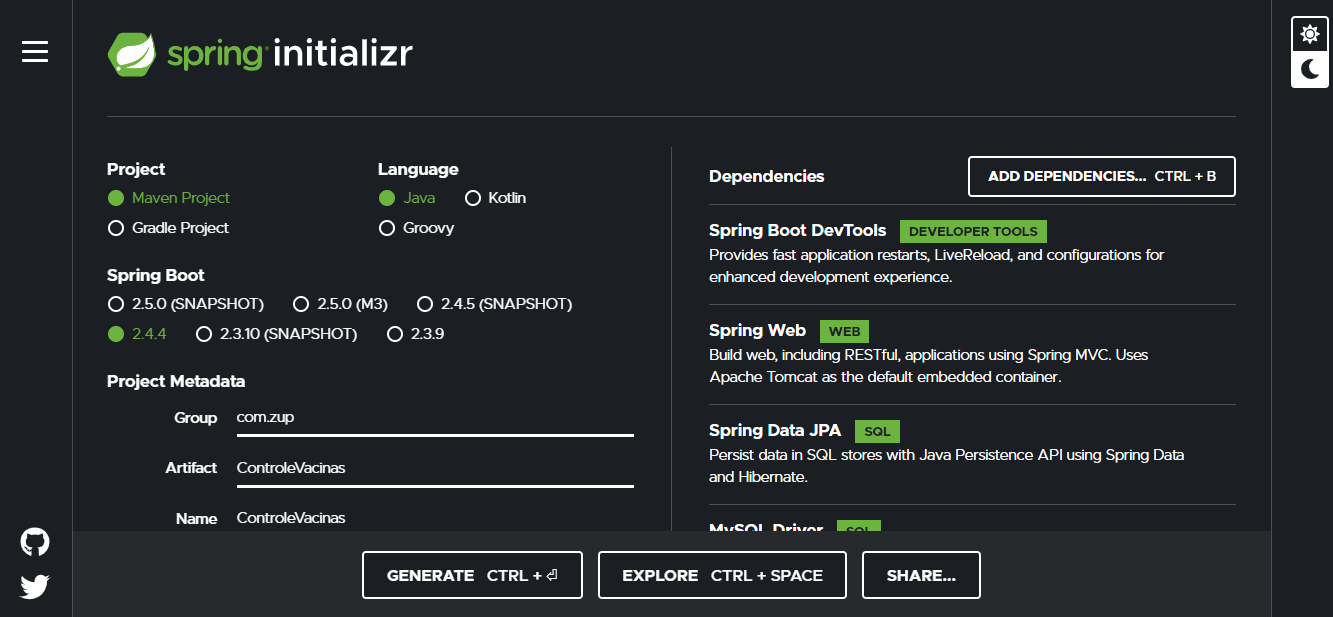
Construir a aplicação sem utilizar Lombok;

Substituir o e-mail do usuário na tabela de vacinação para uma chave estrangeira associada ao id do usuário.

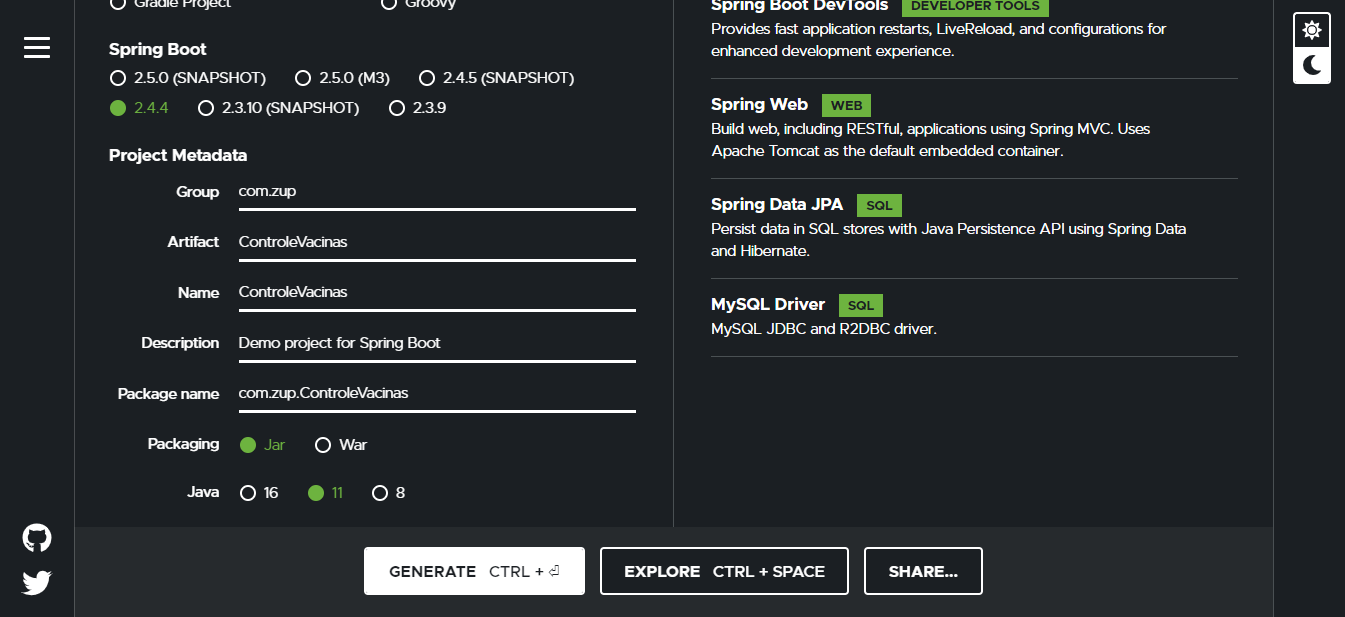
Seu texto poderá conter imagens e deverá ser enviado em formato PDF.

O prazo para envio é de 7 dias após o recebimento desse e-mail. \*/

Podemos criar nosso template de diversas formas, nesse caso acessar o gerador de Templates em <https://start.spring.io/>, alterar as configurações de acordo com a necessidade do projeto . Essa ferramenta é muito interessante, pois de forma rápida e fácil já podemos configurar todas as dependências que desejamos injetar em nossa aplicação.



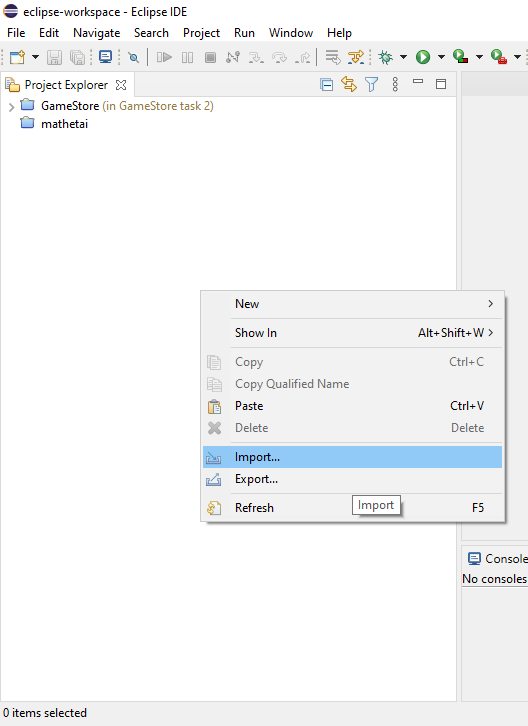
Em seguida gerar e baixar o arquivo:

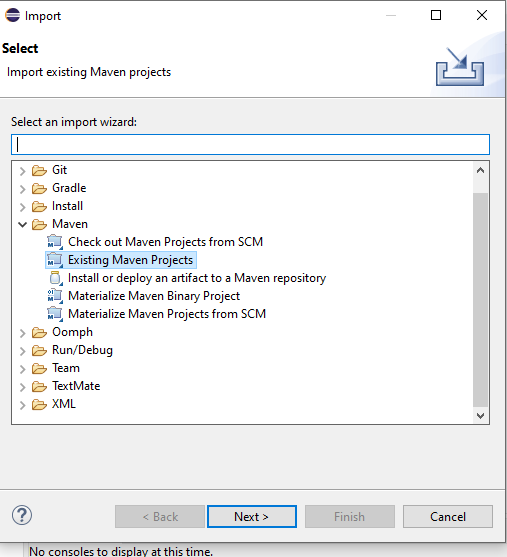


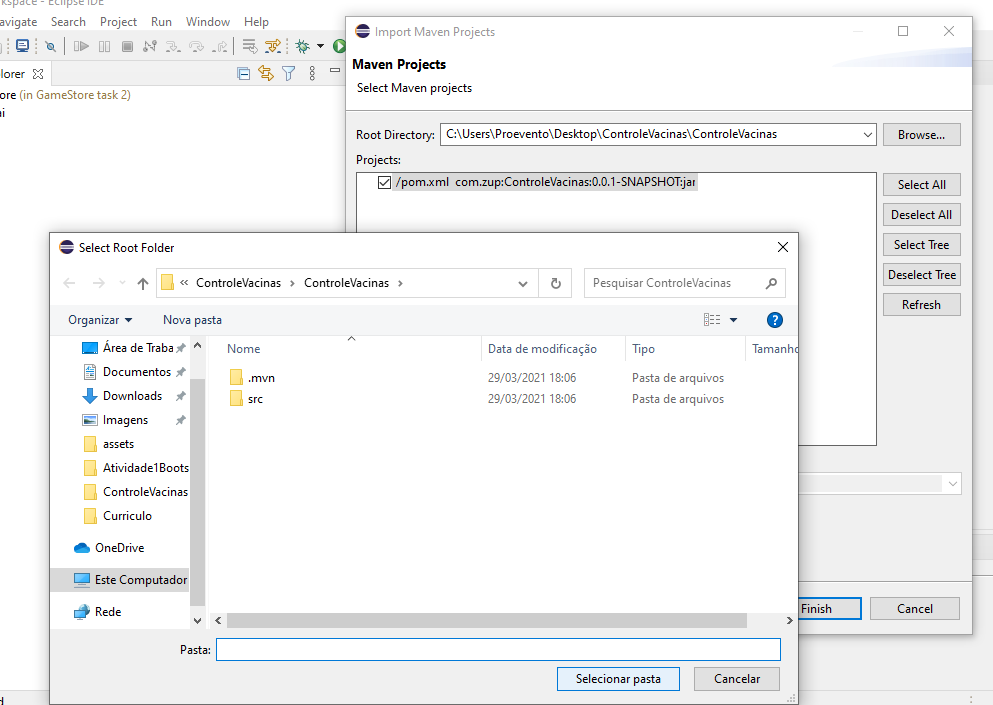
Após gerar esse template, e importar o arquivo ZIP com nosso projeto, que pode ser importado para qualquer IDE (ambiente de desenvolvimento), nesse caso iremos utilizar o Eclipse.

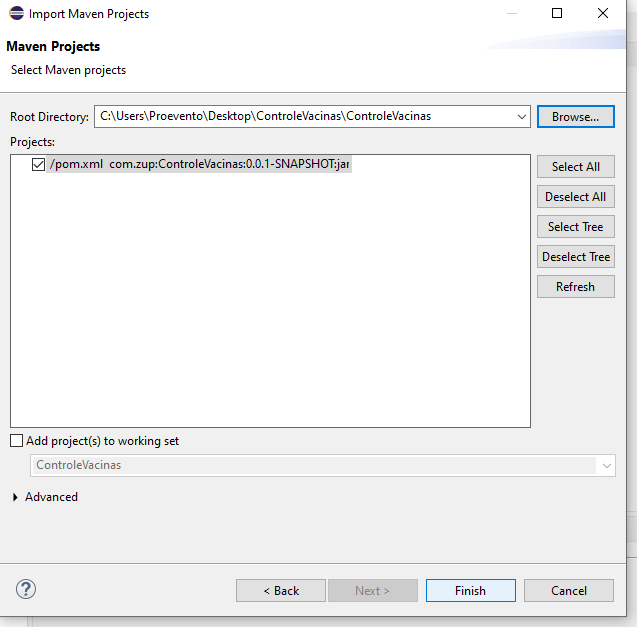
Podemos importar o projeto pelo seguinte caminho: caminho Clicar com o botão direito do mouse -> File -> Import -> Existing Maven Projects e selecionar a pasta onde extraímos o

Projeto, selecionar o arquivo pom.xml, que é a pasta na qual iremos extrair as dependências e clicar em “finish”.

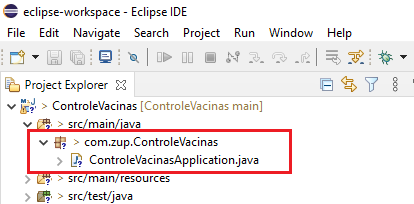








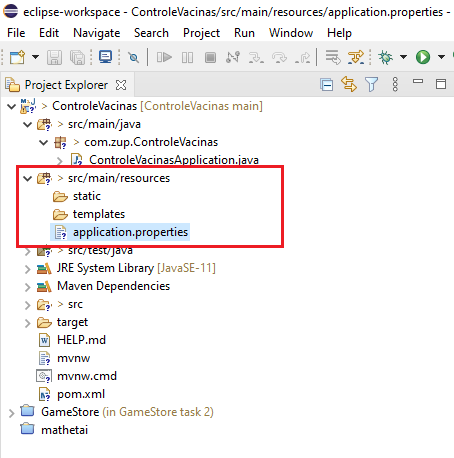
Feito isso, nós já teremos nossa classe principal (main) criada! (Observação: é importante que evitarmos fazer alterações nessa classe, apenas se realmente for necessário)



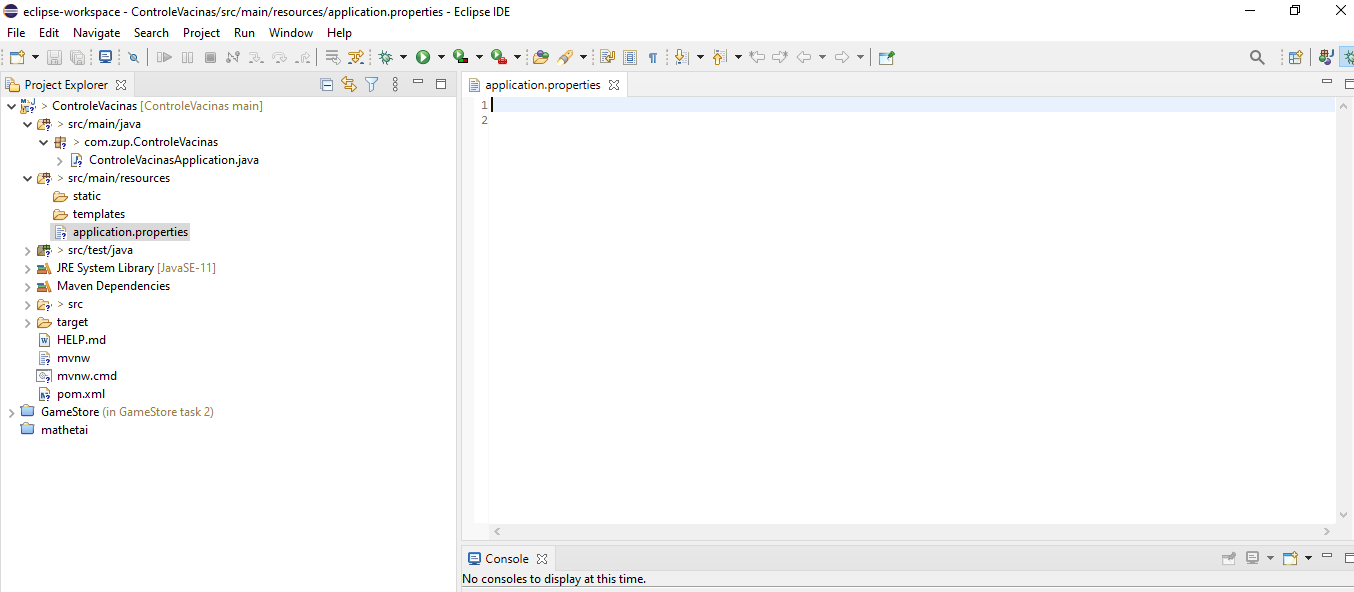
Configurando nosso projeto:

Após importarmos nosso projeto, chegou a hora de configurá-lo para podermos gerenciar os dados da nossa API. Vamos fazer isso pelo arquivo application.properties que

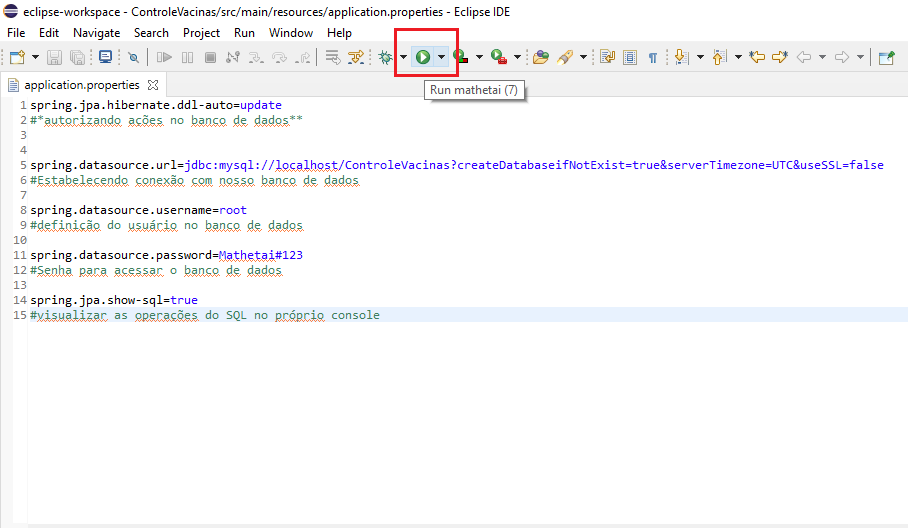
está no caminho src -> main -> resources:



Ao abrirmos esse arquivo ele estará totalmente em branco:



Após isso iremos digitar os seguintes comandos e clicar em “run”:



Após configurar nosso banco de dados, nós iremos configurar as demais classes de nossa API: