

DESAFIOS NA TECNOLOGIA E O MOMENTO ATUAL. VOCÊS ESTÃO PRONTOS? ENTÃO, VAMOS NESSA!

Me chamo Adriana, tenho 29 anos e vou contar para vocês como tem sido me tornar uma desenvolvedora alinhada às necessidades do mercado de trabalho atual.

Em janeiro de 2021 topei participar de um desafio no qual era necessário pensar um projeto para a criação de uma API REST capaz de gerar números aleatórios para lances de loteria, em que cada aposta estivesse associada a um usuário. Assim, a pergunta que não quer calar é: quais tecnologias aproveitei para encarar esta etapa?

Bom, devido à fama de “queridinho” do *Spring Boot*, entre os desenvolvedores Java, esta foi a ferramenta utilizada durante o meu trabalho, pois ela simplifica o processo das aplicações realizadas dentro do universo Spring, agindo como um *template* facilitador para a configuração e finalização do programa. Além disso, o Spring Boot também pode ser aproveitado para o desenvolvimento de microserviços, pois simplifica o processo de configurações e, consequentemente, o processo do *deploy* destas atividades independentes que compõe, juntas, o software.

Chamo atenção para a palavra “facilidade”, porque é a partir dela que devemos compreender a importância da otimização de tempo e do aumento da produtividade ao construirmos aplicações via Spring. Além do *Spring Boot* ser a “caixinha de ferramentas” que oferece recursos essenciais para isso, o *Spring* oferece a página *Spring Initializr*, na qual o desenvolvedor organiza todo o seu projeto com as dependências requeridas por ele, de modo que, depois disso, ele precisará apenas descompactar a pasta (projeto *Maven*, por exemplo) gerada e importá-la dentro da IDE Spring Tools. Daí para frente, é só começar a codar!

A seguir podemos visualizar a tela do *Spring Initializr* para entender melhor como o projeto pode ser criado em apenas alguns clicks:

The screenshot shows the Spring Initializr web application interface. It is a dark-themed form for creating a new Spring project. The interface is divided into several sections:

- Project:** Includes radio buttons for **Maven Project** (selected) and **Gradle Project**.
- Language:** Includes radio buttons for **Java** (selected), **Kotlin**, and **Groovy**.
- Spring Boot:** Includes radio buttons for versions **2.5.0 (SNAPSHOT)**, **2.4.3 (SNAPSHOT)**, **2.4.2** (selected), and **2.3.9 (SNAPSHOT)**.
- Project Metadata:** A form with fields for **Group** (com.example), **Artifact** (demo), **Name** (demo), **Description** (Demo project for Spring Boot), and **Package name** (com.example.demo). It also has a **Packaging** section with **Jar** (selected) and **War** options, and a **Java** version section with **15**, **11** (selected), and **8** options.
- Dependencies:** A list of dependencies with checkboxes and labels: **MySQL Driver** (SQL), **Spring Data JPA** (SQL), **Spring Web** (WEB), **Spring Boot DevTools** (DEVELOPER TOOLS), **Validation** (I/O), and **H2 Database** (SQL). Each dependency has a brief description.

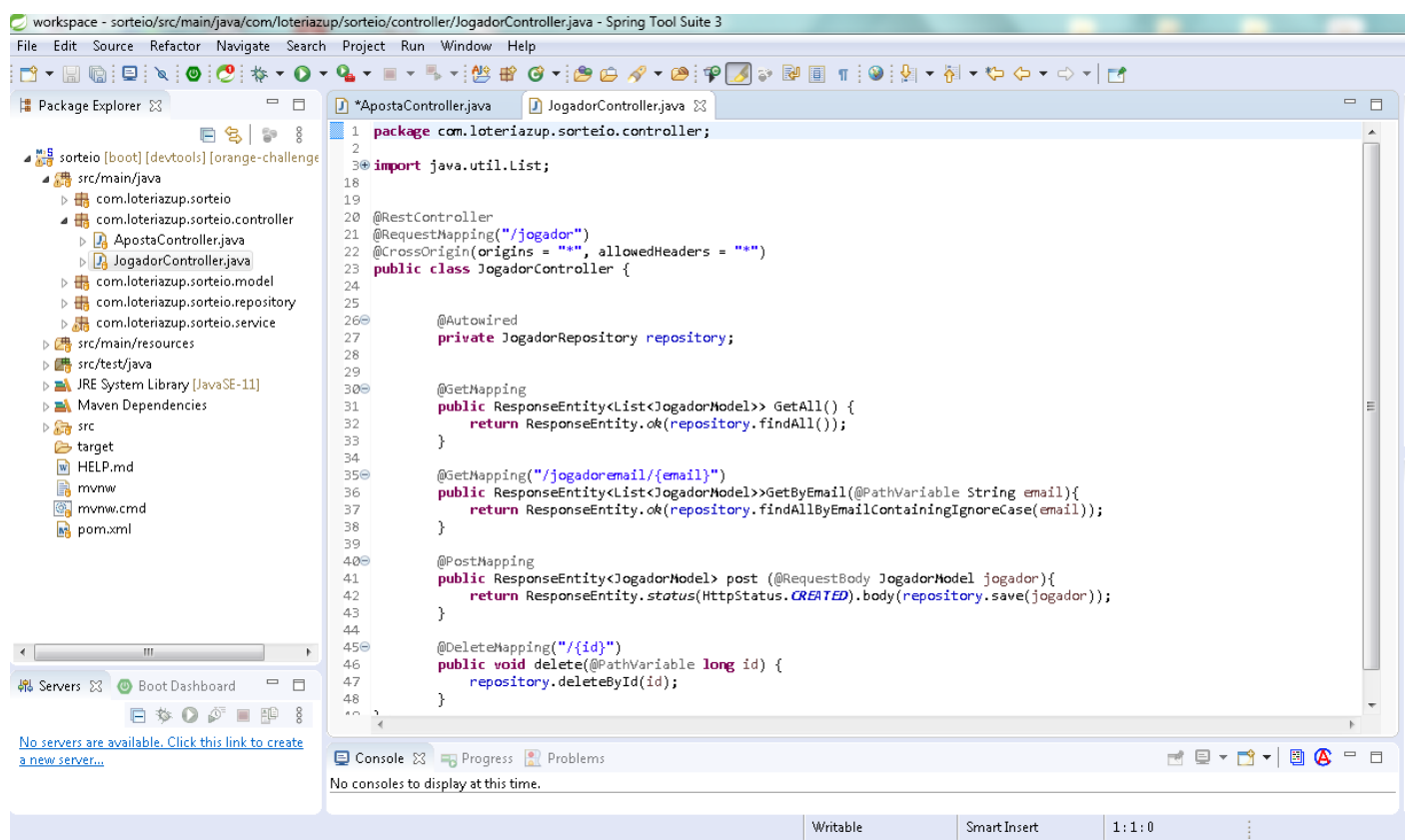
At the bottom, there are three buttons: **GENERATE** (CTRL + G), **EXPLORE** (CTRL + SPACE), and **SHARE...**. The Spring Initializr logo is in the top left, and a settings icon is in the top right.

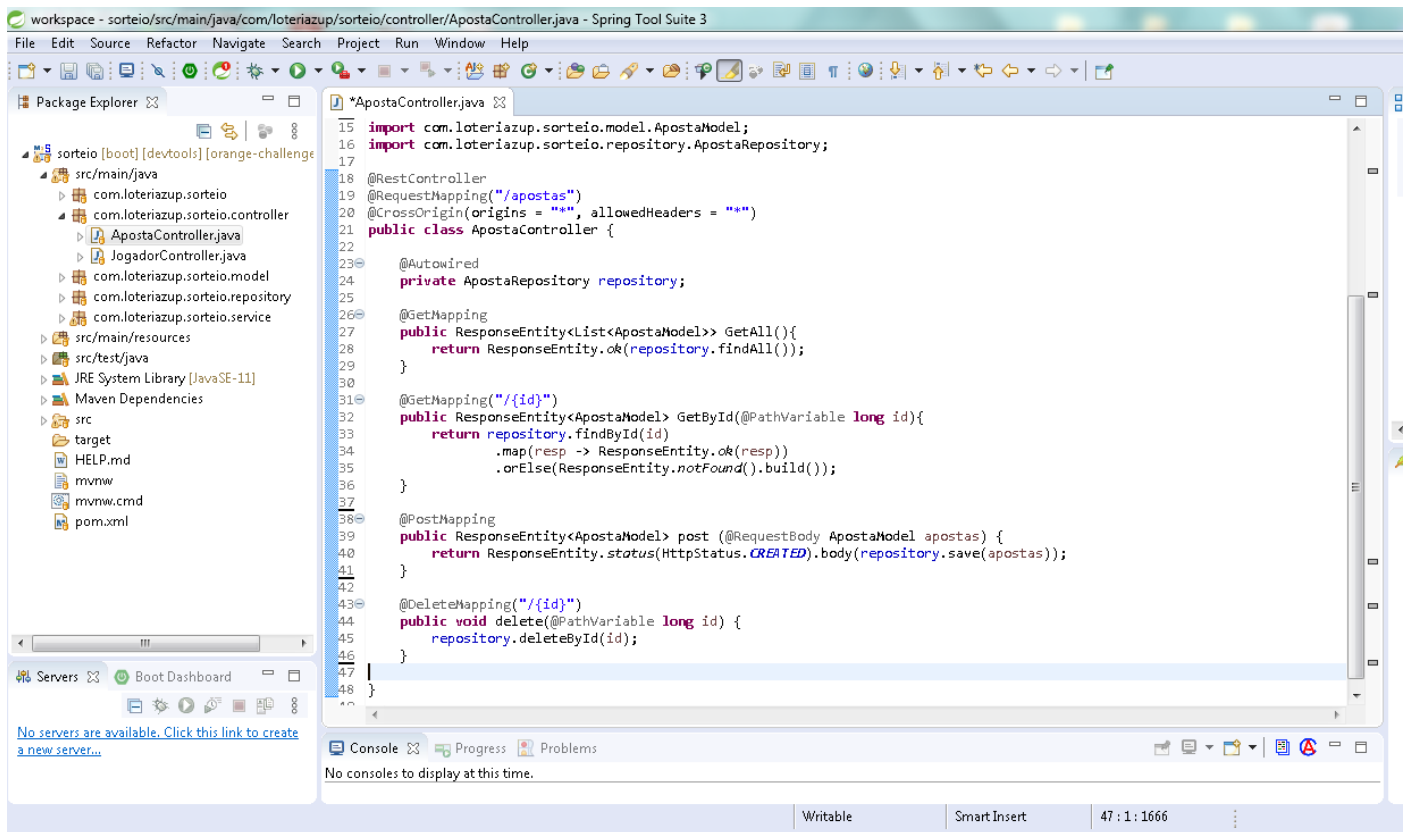
Desta forma, chega, então, a hora mais esperada pelos programadores de todo o mundo, o momento ilustre de “mãos ao código!” e de aproveitar a praticidade oferecida pelo *Spring Boot*, lembrando que ele já terá pré-configurado o seu projeto a partir de suas convenções contidas em seus módulos, além de já possuir um servidor (por *default*) denominado *TomCat*. Ou seja, uma verdadeira “mão-na-roda” dos desenvolvedores web.

Assim, para a criação de uma API REST capaz de gerar números aleatórios para um sorteio de loteria, onde cada aposta encontra-se associada a um jogador (identificado pelo seu endereço de e-mail), foi utilizada como Linguagem de Programação Java e como *Stacks* de tecnologia *Spring (Spring Boot Framework) + Hibernate Framework* (que, por sua vez, é um grande facilitador do mapeamento de objetos relacionais, já que é capaz de importar o conteúdo das bibliotecas de Java através do JPA – Java Persistence API – com o uso de anotações que permitem que a criação do conteúdo que será armazenado em um banco de dados seja realizado diretamente na classe).

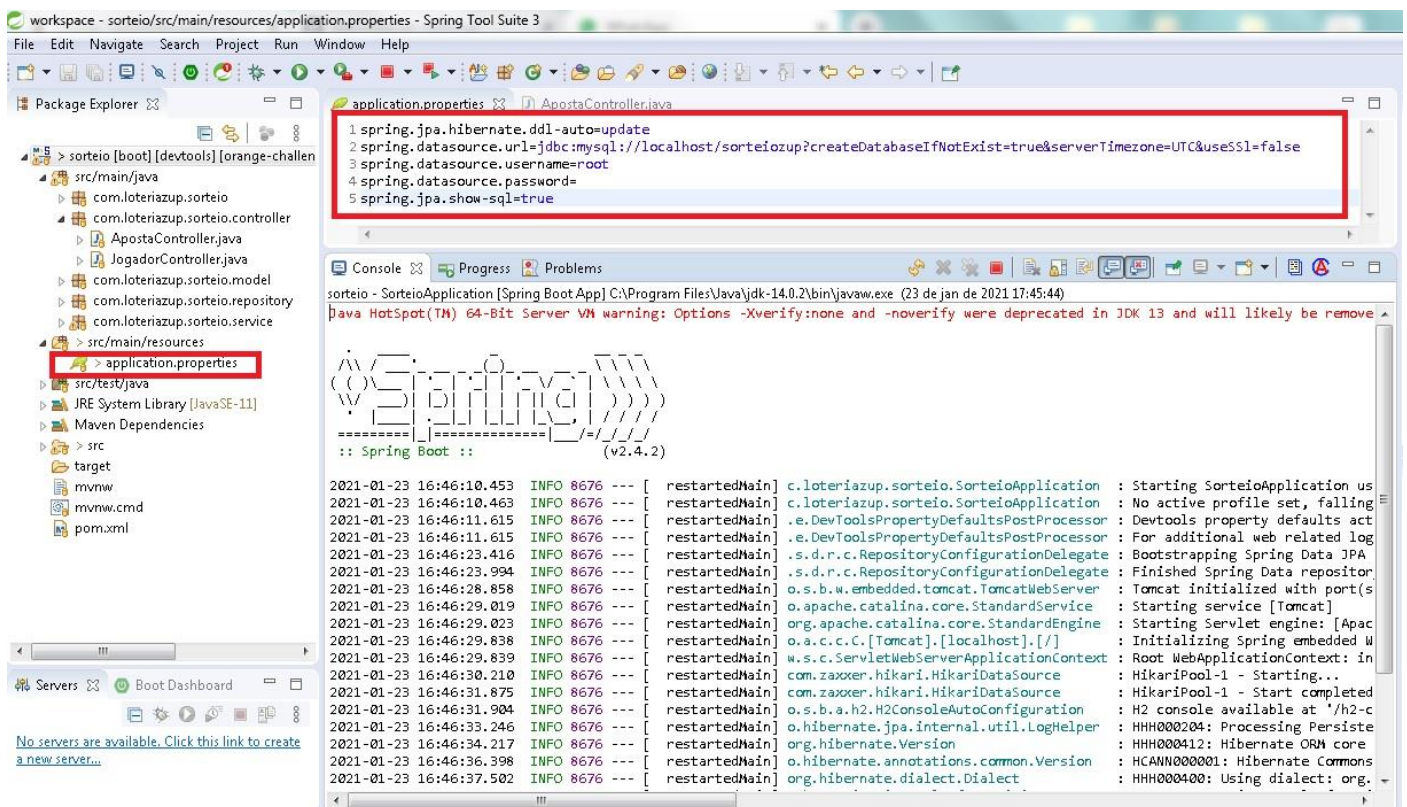
O gerenciador de banco de dados (relacional) utilizado para esta tarefa foi o MySQL Workbench, que recebeu e armazenou os dados gerados no ato em que registramos um jogador e à ele associamos uma aposta aleatória de seis números (que foi calculada graças a criação de um objeto “Random”).

O servidor web utilizado foi o **XAMPP** (*strat to Apache e MYSQL*) e o consumo da API foi feito através do POSTMAN (ferramenta utilizada para testar serviços de *Web APIs*, por meio do envio de requisições HTTP e da análise do seu retorno) que permitiu que os *Ends Points* fossem acessados, tanto para a inclusão de novos jogadores, como para criação de novas rodadas feitas pelos jogadores, mostrados nas imagens do programa a seguir:



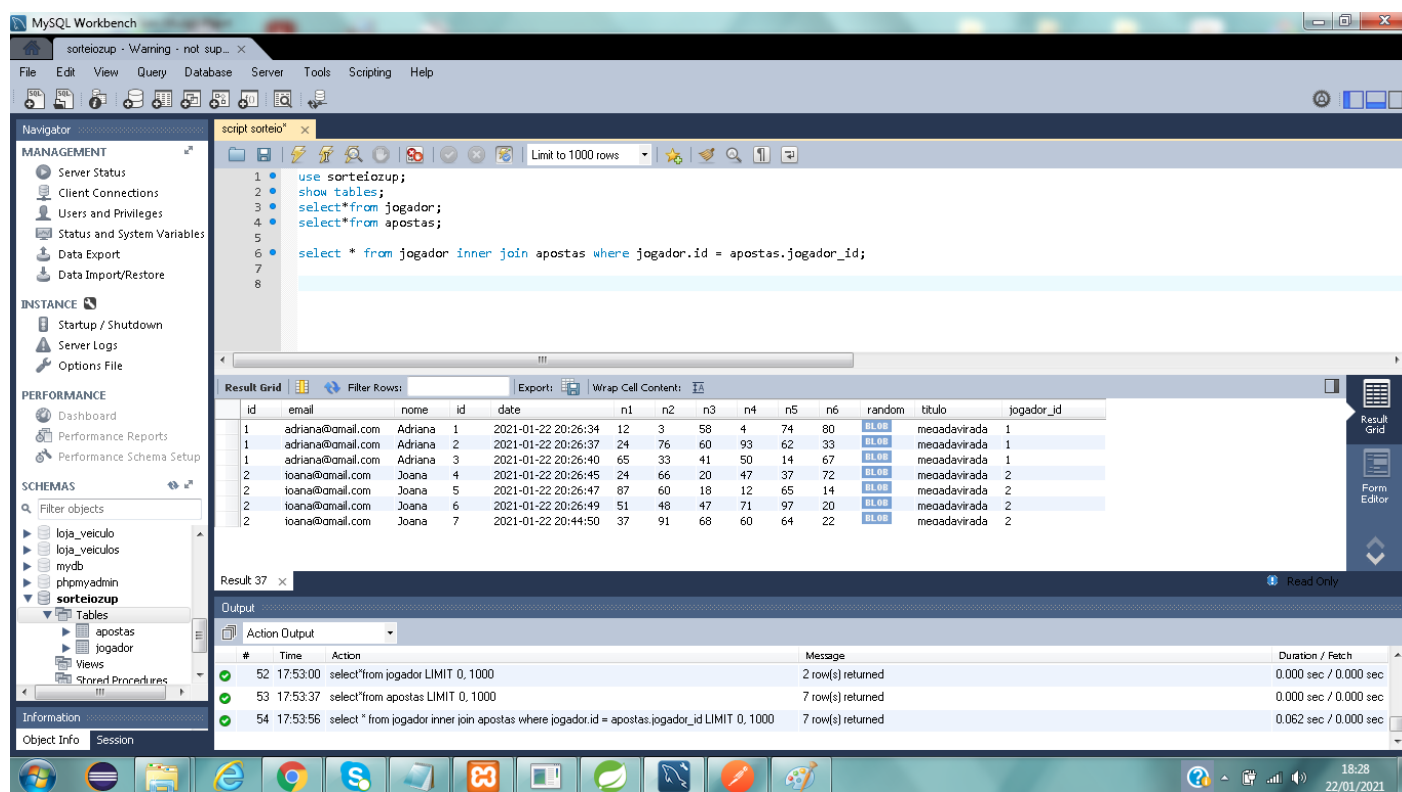


É importante entender, ainda, como o banco de dados está configurado ao programa criado, lembrando que os comandos são incluídos no application.properties, responsável por esta conexão. Vejamos a imagem explicativa a baixo.



Assim sendo, o projeto em questão foi feito a partir modelo padrão *Spring MVC* (***Model***, ***View***, ***Controller***), onde a camada *Model* compõe a estrutura das tabelas para as classes ‘Jogador’ e ‘Apostas’. A persistência dos dados (injeção no banco de dados) desse conteúdo é viabilizada pela interface *Repository* de cada classe mencionada e a camada *Service* para a validação desses dados (execução da regra de negócio). Já o *Controller* é criado para que possamos acessar os *End Points* e que conectará o programa ao *View* (*html/xml*) que, posteriormente, será consumido pelo *front-end*, fazendo com que a página web ganhe vida e se torne inteligível aos usuários comuns.

O resultado da associação de uma jogada aleatória para cada jogador foi armazenada no banco de dados apresentando o seguinte resultado:



The screenshot shows the MySQL Workbench interface. The query editor contains the following SQL script:

```
1 use sorteiozap;
2 show tables;
3 select*from jogador;
4 select*from apostas;
5
6 select * from jogador inner join apostas where jogador.id = apostas.jogador_id;
7
8
```

The result grid displays the following data:

	id	email	nome	id	date	n1	n2	n3	n4	n5	n6	random	titulo	jogador_id
1	1	adriana@gmail.com	Adriana	1	2021-01-22 20:26:34	12	3	58	4	74	80	BL 08	meoadavirada	1
1	1	adriana@gmail.com	Adriana	2	2021-01-22 20:26:37	24	76	60	93	62	33	BL 08	meoadavirada	1
1	1	adriana@gmail.com	Adriana	3	2021-01-22 20:26:40	65	33	41	50	14	67	BL 08	meoadavirada	1
2	2	ioana@gmail.com	Joana	4	2021-01-22 20:26:45	24	66	20	47	37	72	BL 08	meoadavirada	2
2	2	ioana@gmail.com	Joana	5	2021-01-22 20:26:47	87	60	18	12	65	14	BL 08	meoadavirada	2
2	2	ioana@gmail.com	Joana	6	2021-01-22 20:26:49	51	48	47	71	97	20	BL 08	meoadavirada	2
2	2	ioana@gmail.com	Joana	7	2021-01-22 20:44:50	37	91	68	60	64	22	BL 08	meoadavirada	2

The output pane shows the execution log:

#	Time	Action	Message	Duration / Fetch
52	17:53:00	select*from jogador LIMIT 0, 1000	2 row(s) returned	0.000 sec / 0.000 sec
53	17:53:37	select*from apostas LIMIT 0, 1000	7 row(s) returned	0.000 sec / 0.000 sec
54	17:53:56	select * from jogador inner join apostas where jogador.id = apostas.jogador_id LIMIT 0, 1000	7 row(s) returned	0.062 sec / 0.000 sec

Muito bacana, não é mesmo? Ao finalizar nosso código, devemos lembrar que ele pode ser disponibilizado através de um repositório *on line*, que servirá, inclusive, como um *back-up* das versões dos seus arquivos.

Neste caso, utilizamos o **GitHub** que, além de ser o site que comportará nosso código, também é uma excelente plataforma desenvolvida como uma rede social em que podemos seguir outros desenvolvedores e, assim, compartilhar e difundir novas tecnologias.

Para enviar nosso código ao **GitHub** é necessário possuir conhecimento dos comandos básicos do **GIT**, que é a ferramenta que está por trás das vantagens do **GitHub**, ou melhor, é o Sistema de Controle de Versionamento de código utilizado para enviar nossos arquivos para a plataforma *on line*.

Caso queiram visitar o repositório público da **API REST** desenvolvida para ilustrar este artigo, acesse: <https://github.com/Adrici/orange-challenge>

POR ADRIANA CIRELLI, JANEIRO DE 2021.
DESENVOLVEDORA WEB FULLSTACK JAVA JÚNIOR



 [linkedin.com/in/adrianacirelli/](https://www.linkedin.com/in/adrianacirelli/)

 github.com/Adrici