

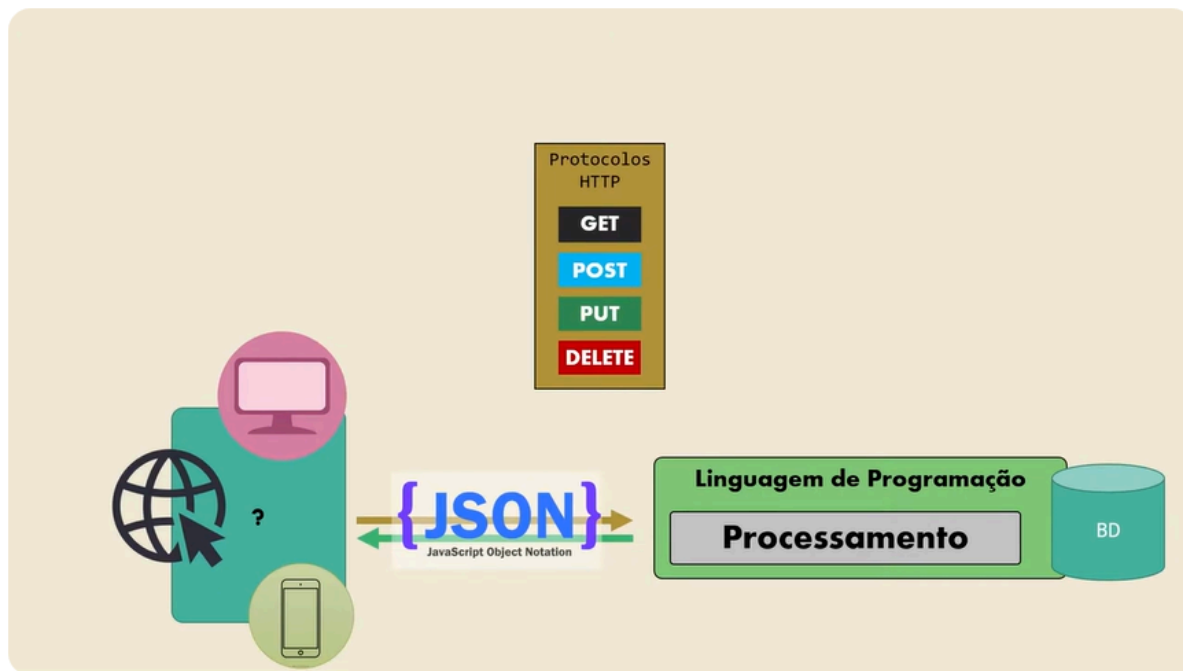
REST é um modelo de arquitetura que processa e se necessário armazena os dados

Veja que é só o back-end sem as telas do projeto.

Assim, podemos acoplar diferentes plataformas como web, desktop, mobile.

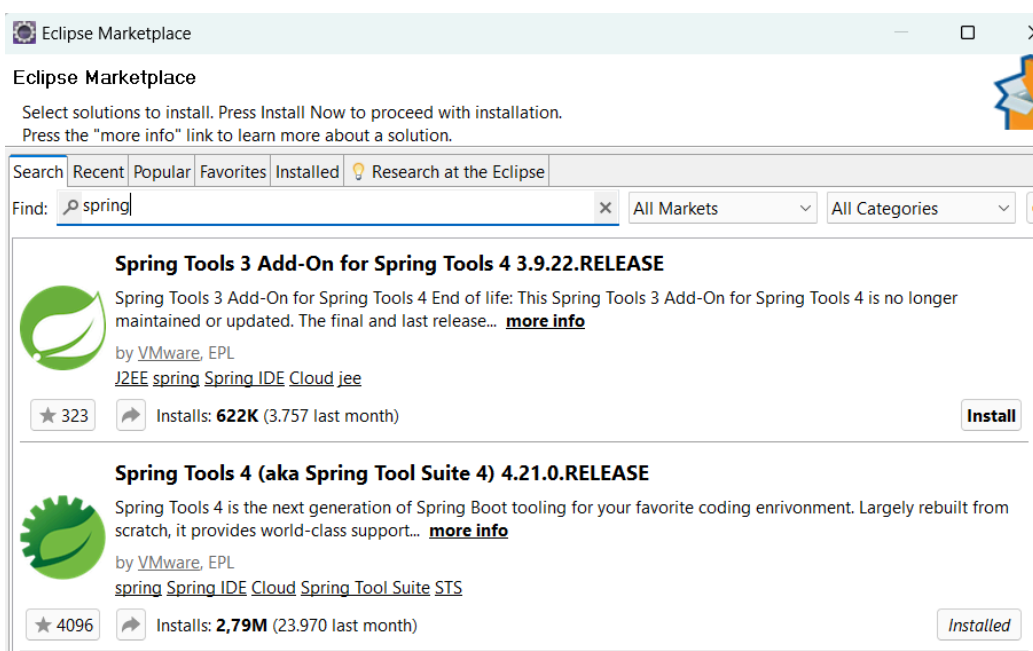
Para que isso seja possível, os dados entre as interfaces e o servidor REST são transferidos em um formato padrão, que atualmente se utiliza o JSON.

Mas a grande sacada do REST é o uso dos protocolos Http



Procedimentos para implementação

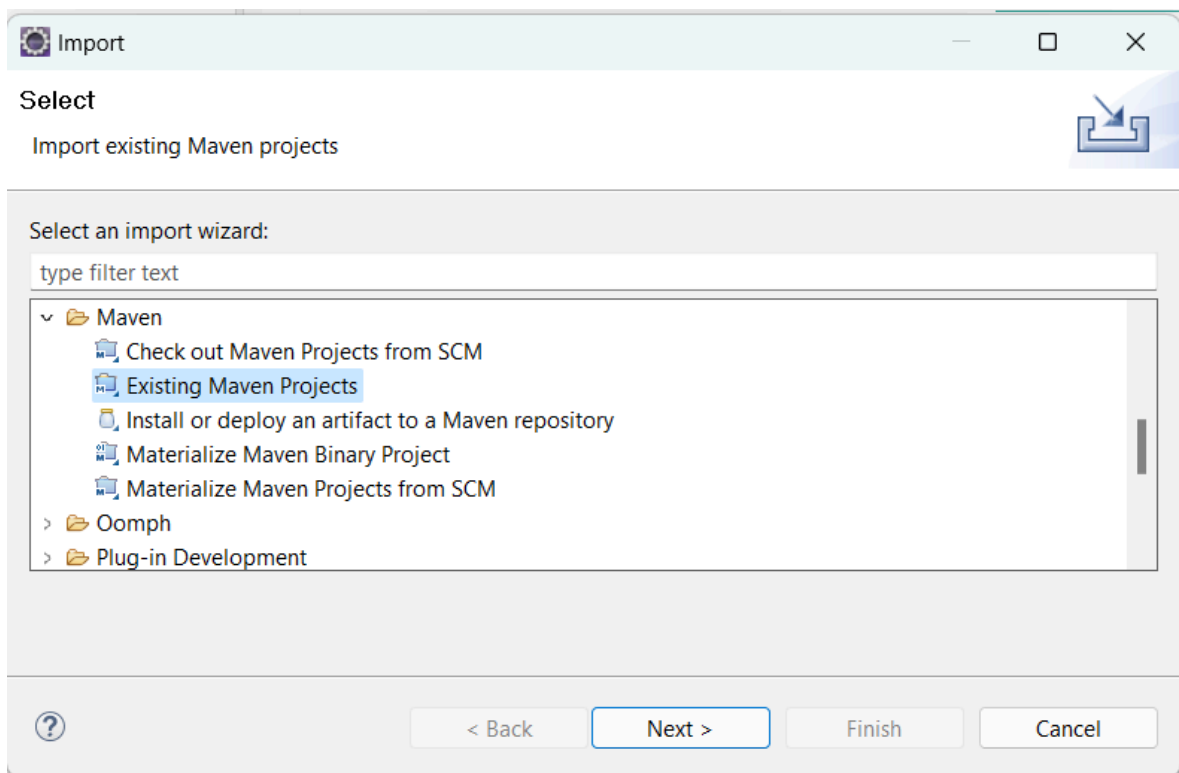
- 1) Usamos a IDE do Eclipse
- 2) Será preciso baixar a ferramenta Spring. Menu Help >> Marketplace >> Spring



3) Criando um projeto padrão maven com spring boot.

- site: <https://start.spring.io/> (Spring Initializr)
- Configurações:
 - Project: Maven (maven é um gerenciador de dependências)
 - Spring Boot: a última versão
 - Packaging: Jar
 - Java: Até o momento, versão 17
- Adicione as dependências
 - Spring Data JPA (ferramenta do banco de dados)
 - Rest Repositories
 - Mysql Driver (gerenciamento do banco de dados)
- Após, baixe o projeto e descompacte no workspace do eclipse

4) No Eclipse, importe o projeto Maven, na opção Maven >> Existing Maven Projects



5) Implementando o modelo de domínio.

Neste exemplo:

Contato
id
nome
telefone
e-mail
urlAvatar

5.1) Definindo as entidades

Pacote: com.dominio.nomeprojeto.entities (Neste pacote colocaremos todas as entidades)

Classe: Contact

```
public class Contact {  
    private Long id;  
    private String nome;  
    private String telefone;  
    private String email;  
    private String urlAvatar;  
}
```

CheckList após a construção da classe:

- Métodos construtores
- Métodos getters e setters

5.2) Definindo as anotações do framework

```
8  
9 @Entity // define que a classe será um entidade e framework cria uma tabela  
10 public class Contact {  
11     @Id //avisa quem é o identificador  
12     @GeneratedValue(strategy = GenerationType.IDENTITY) // informa que a chave será auto-increment  
13     private Long id;  
14     @Column(nullable = false) // informa que a coluna(campo) não pode ser vazia  
15     private String nome;  
16     @Column(nullable = false)  
17     private String telefone;  
18     @Column(nullable = false)  
19     private String email;  
20     private String urlAvatar;  
21 }
```

Estas anotações são da dependência do JPA

6) Definindo os repositórios

Pacote: com.dominio.nomeprojeto.repositories

Interface: ContactRepository

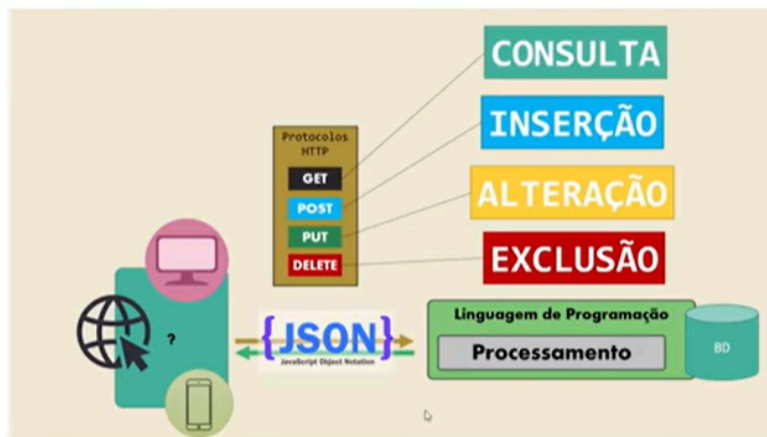
```
public interface ContactRepository extends JpaRepository<Contact, Long>  
{  
    // onde <contact, Long> tipo da entidade e tipo do identificador  
}
```

7) Definindo os controladores

Pacote: com.dominio.nomeprojeto.controllers

Classe: ContactController

Agora vamos inserir os métodos para completar o acesso de dados:



```
18
19 @RestController // define que a classe será um rest
20 @RequestMapping("/contact") // para vincular a uma url
21 public class ContactController {
22     @Autowired // cria o objeto e gerencia o ciclo de vida
23     private ContactRepository repositorio;
24
25     // busca todos os contatos do banco de dados
26     @GetMapping
27     public List<Contact> listar() {
28         return repositorio.findAll();
29     }
30
31     // salvar // @RequestBody converte o conteúdo Json em objeto java
32     @PostMapping
33     public void salvar(@RequestBody Contact contato) {
34         repositorio.save(contato);
35     }
36
37     // alterar
38     @PutMapping
39     public void alterar(@RequestBody Contact contato) {
40         if (contato.getId() > 0)
41             repositorio.save(contato);
42     }
43
44     // excluir
45     @DeleteMapping
46     public void excluir(@RequestBody Contact contato) {
47         repositorio.delete(contato);
48     }
49
50 }
51
52
```

8) Configurando o acesso do motor do banco de dados mysql

Arquivo: application.properties

```
spring.jpa.hibernate.ddl-auto=update #Estratégia
spring.datasource.url=jdbc:mysql://localhost:3306/nomebancodados
spring.datasource.username=root #usuario
spring.datasource.password= #senha
spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver
```

9) Faça os testes com o postman