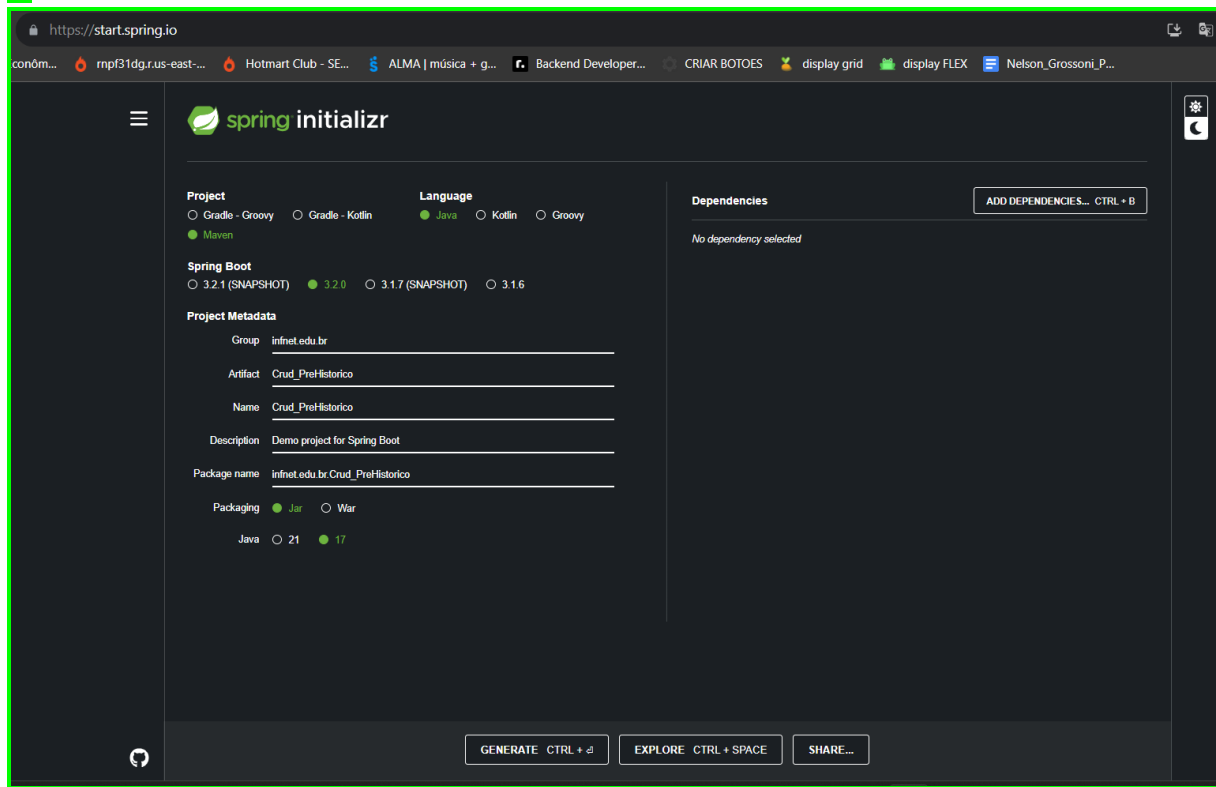


*O aluno utilizou o Spring Initalizr para criar seu projeto?

R:



*O aluno utilizou as Goals do maven para iniciar e pausar seu projeto?

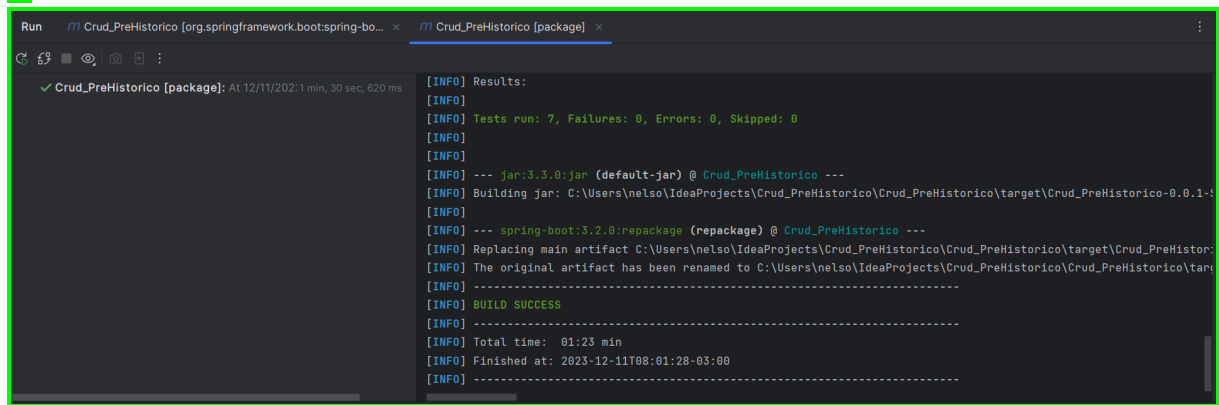
R:

```
"C:\Program Files\Java\jdk-20\bin\java.exe" -Dmaven.multiModuleProjectDirectory=C:\Users\nelso\IdeaProjects\Crud_PreHistorico
[INFO] Scanning for projects...
[INFO]
[INFO] -----< infnet.edu.br:Crud_PreHistorico >-----
[INFO] Building Crud_PreHistorico 0.0.1-SNAPSHOT
[INFO]    from pom.xml
[INFO] -----[ jar ]-----
[INFO]
[INFO] --- spring-boot:3.2.0:start (default-cli) @ Crud_PreHistorico ---
[INFO] Attaching agents: []

"C:\Program Files\Java\jdk-20\bin\java.exe" -Dmaven.multiModuleProjectDirectory=C:\Users\nelso\IdeaProjects\Crud_PreHistorico
[INFO] Scanning for projects...
[INFO]
[INFO] -----< infnet.edu.br:Crud_PreHistorico >-----
[INFO] Building Crud_PreHistorico 0.0.1-SNAPSHOT
[INFO]    from pom.xml
[INFO] -----[ jar ]-----
[INFO]
[INFO] --- spring-boot:3.2.0:stop (default-cli) @ Crud_PreHistorico ---
[INFO] Stopping application...
[INFO]
[INFO] BUILD SUCCESS
[INFO]
```

*O aluno utilizou o maven para gerar um fatjar do seu projeto?

R:



```
Run  m Crud_PreHistorico [org.springframework.boot:spring-bo... x  m Crud_PreHistorico [package] x
✓ Crud_PreHistorico [package]: At 12/11/2023:1 min, 30 sec, 620 ms

[INFO] Results:
[INFO]
[INFO] Tests run: 7, Failures: 0, Errors: 0, Skipped: 0
[INFO]
[INFO]
[INFO] --- jar:3.3.0:jar (default-jar) @ Crud_PreHistorico ---
[INFO] Building jar: C:\Users\nelso\IdeaProjects\Crud_PreHistorico\Crud_PreHistorico\target\Crud_PreHistorico-0.0.1-SNAPSHOT.jar
[INFO]
[INFO] --- spring-boot:3.2.0:repackage (repackage) @ Crud_PreHistorico ---
[INFO] Replacing main artifact C:\Users\nelso\IdeaProjects\Crud_PreHistorico\Crud_PreHistorico\target\Crud_PreHistorico-0.0.1-SNAPSHOT.jar with repackaged one
[INFO] The original artifact has been renamed to C:\Users\nelso\IdeaProjects\Crud_PreHistorico\Crud_PreHistorico\target\Crud_PreHistorico-0.0.1-SNAPSHOT.jar.original
[INFO]
[INFO] BUILD SUCCESS
[INFO]
[INFO] Total time: 01:23 min
[INFO] Finished at: 2023-12-11T08:01:28-03:00
[INFO]
```

*O aluno importou o projeto Lombok?

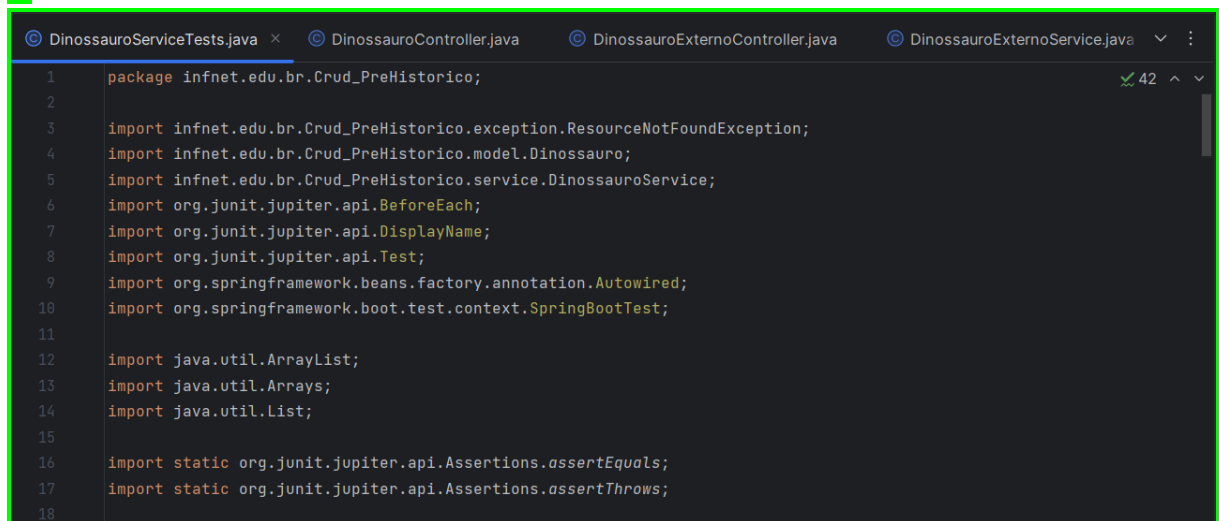
R:



```
Dinossauero.java x  DinossaueroExterno.java  m pom.xml (Crud_PreHistorico)  DinossaueroService.java  DinossaueroService.java
1 package infnet.edu.br.Crud_PreHistorico.model;
2
3 import lombok.AllArgsConstructor;
4 import lombok.Builder;
5 import lombok.Data;
6 import lombok.NoArgsConstructor;
7
8 import java.util.ArrayList;
9
10 @Data@AllArgsConstructor@NoArgsConstructor@Builder
11 public class Dinossauero {
12     private String id;
13     private String nome;
14     private Double altura;
15     private ArrayList<String> caracteristica;
16 }
```

*O aluno importou a biblioteca do JUNIT5?

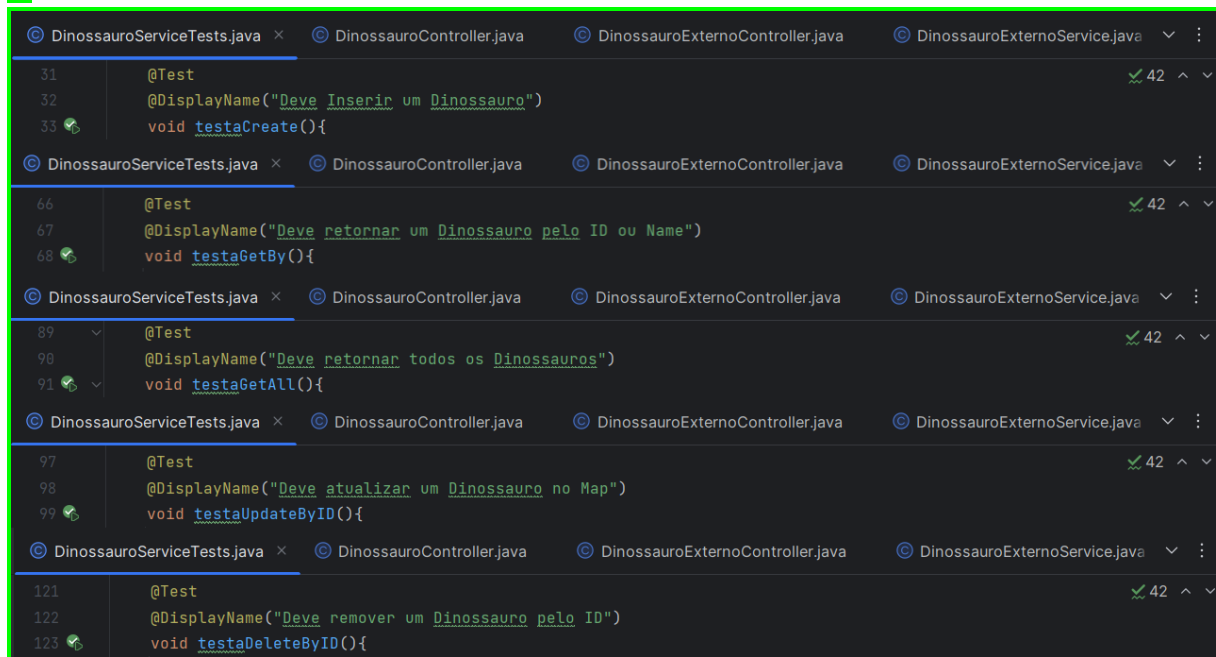
R:



```
DinossaueroServiceTests.java x  DinossaueroController.java  DinossaueroExternoController.java  DinossaueroExternoService.java
1 package infnet.edu.br.Crud_PreHistorico;
2
3 import infnet.edu.br.Crud_PreHistorico.exception.ResourceNotFoundException;
4 import infnet.edu.br.Crud_PreHistorico.model.Dinossauero;
5 import infnet.edu.br.Crud_PreHistorico.service.DinossaueroService;
6 import org.junit.jupiter.api.BeforeEach;
7 import org.junit.jupiter.api.DisplayName;
8 import org.junit.jupiter.api.Test;
9 import org.springframework.beans.factory.annotation.Autowired;
10 import org.springframework.boot.test.context.SpringBootTest;
11
12 import java.util.ArrayList;
13 import java.util.Arrays;
14 import java.util.List;
15
16 import static org.junit.jupiter.api.Assertions.assertEquals;
17 import static org.junit.jupiter.api.Assertions.assertThrows;
18
```

*O aluno testou seus endpoints com as Assertions do Junit?

R:



```
31 @Test
32 @DisplayName("Deve Inserir um Dinossauro")
33 void testaCreate(){

66 @Test
67 @DisplayName("Deve retornar um Dinossauro pelo ID ou Name")
68 void testaGetBy(){

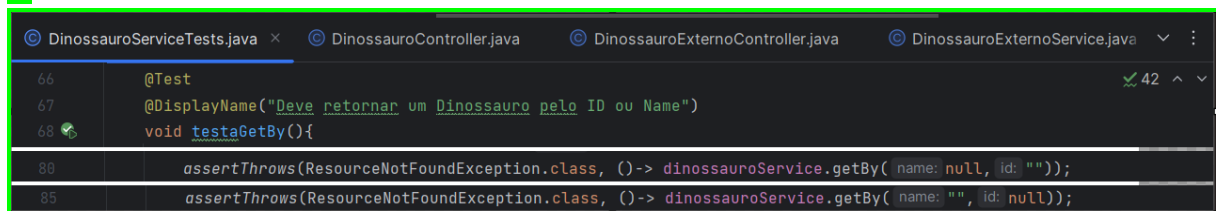
89 @Test
90 @DisplayName("Deve retornar todos os Dinossauros")
91 void testaGetAll(){

97 @Test
98 @DisplayName("Deve atualizar um Dinossauro no Map")
99 void testaUpdateByID(){

121 @Test
122 @DisplayName("Deve remover um Dinossauro pelo ID")
123 void testaDeleteByID(){
```

*O aluno utilizou a Assert Throws do Junit?

R:



```
66 @Test
67 @DisplayName("Deve retornar um Dinossauro pelo ID ou Name")
68 void testaGetBy(){

80 assertThrows(ResourceNotFoundException.class, ()-> dinossauroService.getBy( name: null, id: ""));
85 assertThrows(ResourceNotFoundException.class, ()-> dinossauroService.getBy( name: "", id: null));
```

*O aluno utilizou o Junit ao invés de criar o método main para testar seu código?

R: Sim, conforme visto nos prints acima.

*O aluno utilizou a api de Requests do JAVA 11 para realizar uma chamada a uma api externa?

```
DinossauoExternoService.java x m pom.xml (Crud_PreHistorico) DinossauoService.java CrudPreHistoricoApplicationTests.java
19 @Service
20 public class DinossauoExternoService {
21     private static final Logger logger = LoggerFactory.getLogger(DinossauoExternoService.class);
22     public DinossauoExterno getRandomDinosaur(){
23         try {
24             HttpRequest request = HttpRequest.newBuilder()
25                 .GET()
26                 .version(HttpClient.Version.HTTP_2)
27                 .uri(new URI("https://dinosaur-facts-api.shultzlab.com/dinosaurs/random"))
28                 .build();
29
30             HttpClient client = HttpClient.newBuilder().build();
31             HttpResponse<String> response = client.send(request, HttpResponse.BodyHandlers.ofString());
32
33             logger.info("StatusCode de GET:DinossauoExterno: " + response.statusCode());
34             if(response.statusCode() == 404){
35                 throw new ResourceNotFoundException("Localizacao nao encontrada");
36             }
37             ObjectMapper mapper = JsonMapper.builder().addModule(new JavaTimeModule()).build();
38
39             return mapper.readValue(response.body(), DinossauoExterno.class);
40         } catch (URISyntaxException | IOException | InterruptedException e) {
41             throw new RuntimeException(e);
42         }
43     }
44 }
45
46 }
```

*O aluno imprimiu com LOG o status code da Resposta?

R:

```
DinossauoExternoService.java x m pom.xml (Crud_PreHistorico) DinossauoService.java CrudPreHistoricoApplicationTests.java
19 @Service
20 public class DinossauoExternoService {
21     private static final Logger logger = LoggerFactory.getLogger(DinossauoExternoService.class);
22
34     logger.info("StatusCode de GET:DinossauoExterno: " + response.statusCode());
```

*O aluno converteu o resultado da requisição de um JSON para um objeto JAVA?

R:

```
DinossauoExterno.java x DinossauoExternoService.java m pom.xml (Crud_PreHistorico) DinossauoService.java Cr
1 package infnet.edu.br.Crud_PreHistorico.model;
2 import ...
3
4 5 usages
7 @Data@AllArgsConstructor@NoArgsConstructor@Builder
8 public class DinossauoExterno {
9     @JsonProperty("Name")
10     private String name;
11     @JsonProperty("Description")
12     private String description;
13 }
14
15 DinossauoExternoService.java x m pom.xml (Crud_PreHistorico) DinossauoService.java CrudPreHistoricoApplicationTests.java
19 @Service
20 public class DinossauoExternoService {
21
22     public DinossauoExterno getRandomDinosaur(){
23
40         return mapper.readValue(response.body(), DinossauoExterno.class);
```

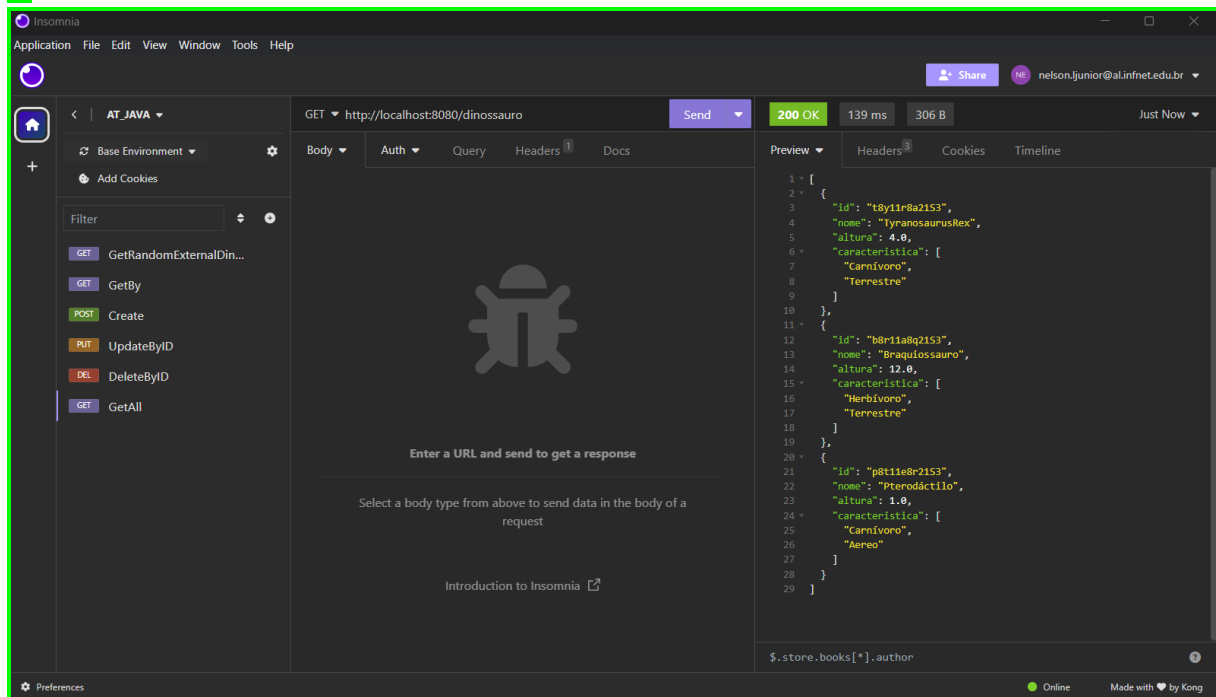
*O aluno testou a requisição externa?

R:

```
© DinossauroExternoServiceTests.java x © DinossauroServiceTests.java m pom.xml (Crud_PreHistorico) © DinossauroService.java v :
1 package infnet.edu.br.Crud_PreHistorico;
2
3 > import ...
10
11 @SpringBootTest
12 public class DinossauroExternoServiceTests
13 {
14     @Autowired
15     DinossauroExternoService dinossauroExternoService;
16
17     @Test
18     @DisplayName("Não deve retornar nulo na chamada pra API Externa")
19     public void testaGetRandomDinosaur()
20     {
21         assertEquals(unexpected: null, dinossauroExternoService.getRandomDinosaur());
22     }
23 }
```

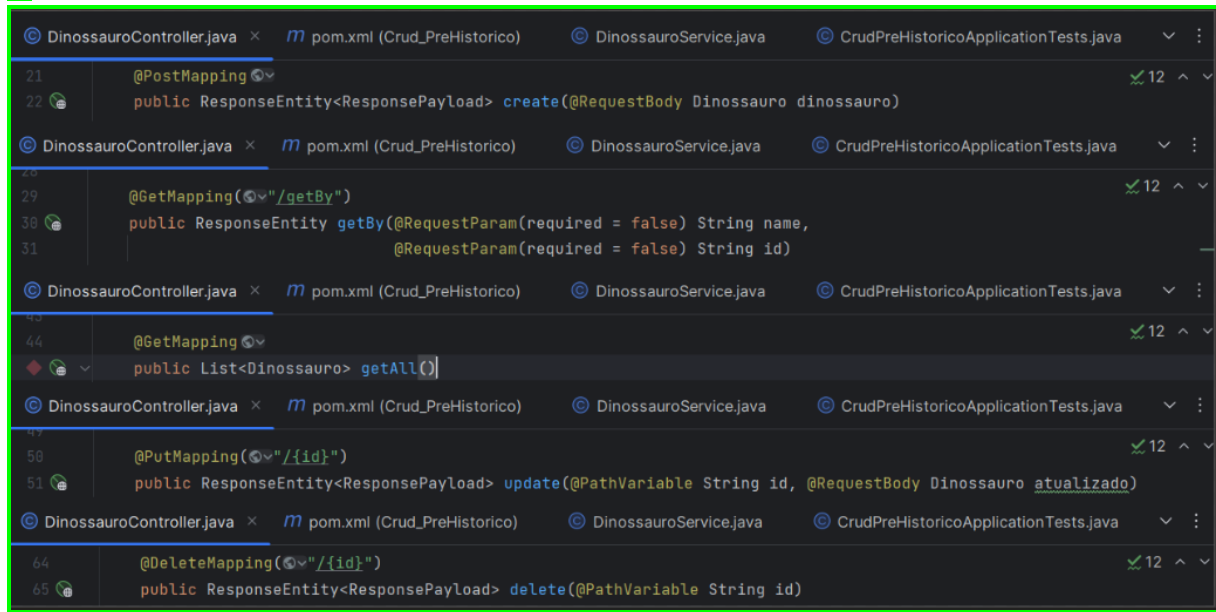
*O aluno criou um serviço web ouvindo na porta 8080?

R:



*O aluno criou pelo menos um endpoint para cada verbo HTTP(GET,POST,DELETE,PUT)?

R:



```
DinossauroController.java x pom.xml (Crud_PreHistorico) DinossauroService.java CrudPreHistoricoApplicationTests.java
21 @PostMapping
22 public ResponseEntity<ResponsePayload> create(@RequestBody Dinossauro dinossauro)
23
DinossauroController.java x pom.xml (Crud_PreHistorico) DinossauroService.java CrudPreHistoricoApplicationTests.java
28 @GetMapping("/{getBy}")
30 public ResponseEntity getBy(@RequestParam(required = false) String name,
31 @RequestParam(required = false) String id)
32
DinossauroController.java x pom.xml (Crud_PreHistorico) DinossauroService.java CrudPreHistoricoApplicationTests.java
44 @GetMapping
45 public List<Dinossauro> getAll()
46
DinossauroController.java x pom.xml (Crud_PreHistorico) DinossauroService.java CrudPreHistoricoApplicationTests.java
50 @PutMapping("/{id}")
51 public ResponseEntity<ResponsePayload> update(@PathVariable String id, @RequestBody Dinossauro atualizado)
52
DinossauroController.java x pom.xml (Crud_PreHistorico) DinossauroService.java CrudPreHistoricoApplicationTests.java
64 @DeleteMapping("/{id}")
65 public ResponseEntity<ResponsePayload> delete(@PathVariable String id)
```

*O aluno desenvolveu um código de erro como tratamento de um método problemático?

R:



```
ResourceNotFoundException.java x DinossauroServiceTests.java pom.xml (Crud_PreHistorico) DinossauroService.java
1 package infnet.edu.br.Crud_PreHistorico.exception;
2
3 public class ResourceNotFoundException extends RuntimeException {
4     public ResourceNotFoundException(String message) { super(message); }
5 }
6
7
8
```