

FIAP GRADUAÇÃO

DIGITAL BUSINESS ENABLEMENT

Prof. THIAGO T. I. YAMAMOTO

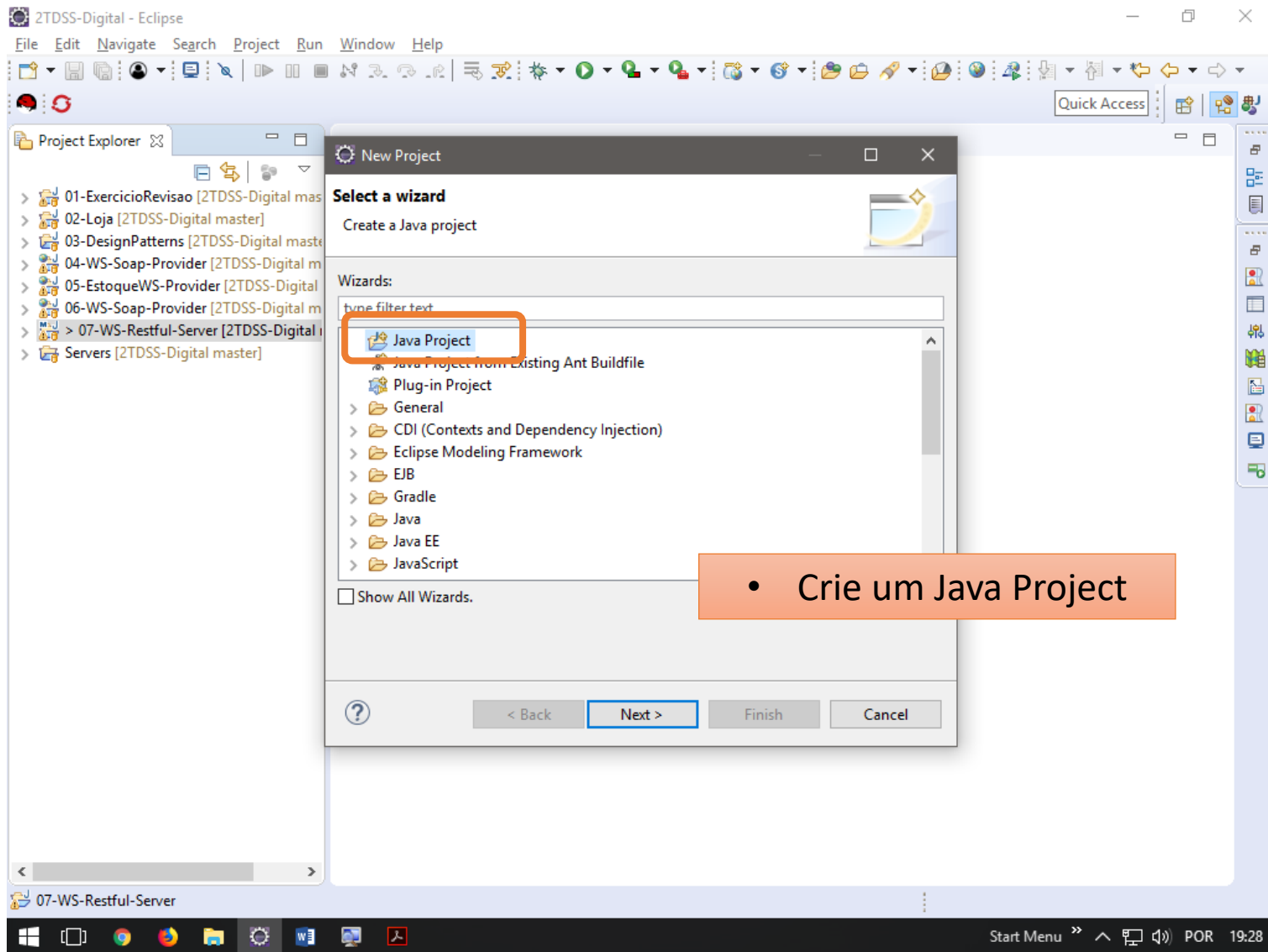
PROJETO RESTFUL CLIENT

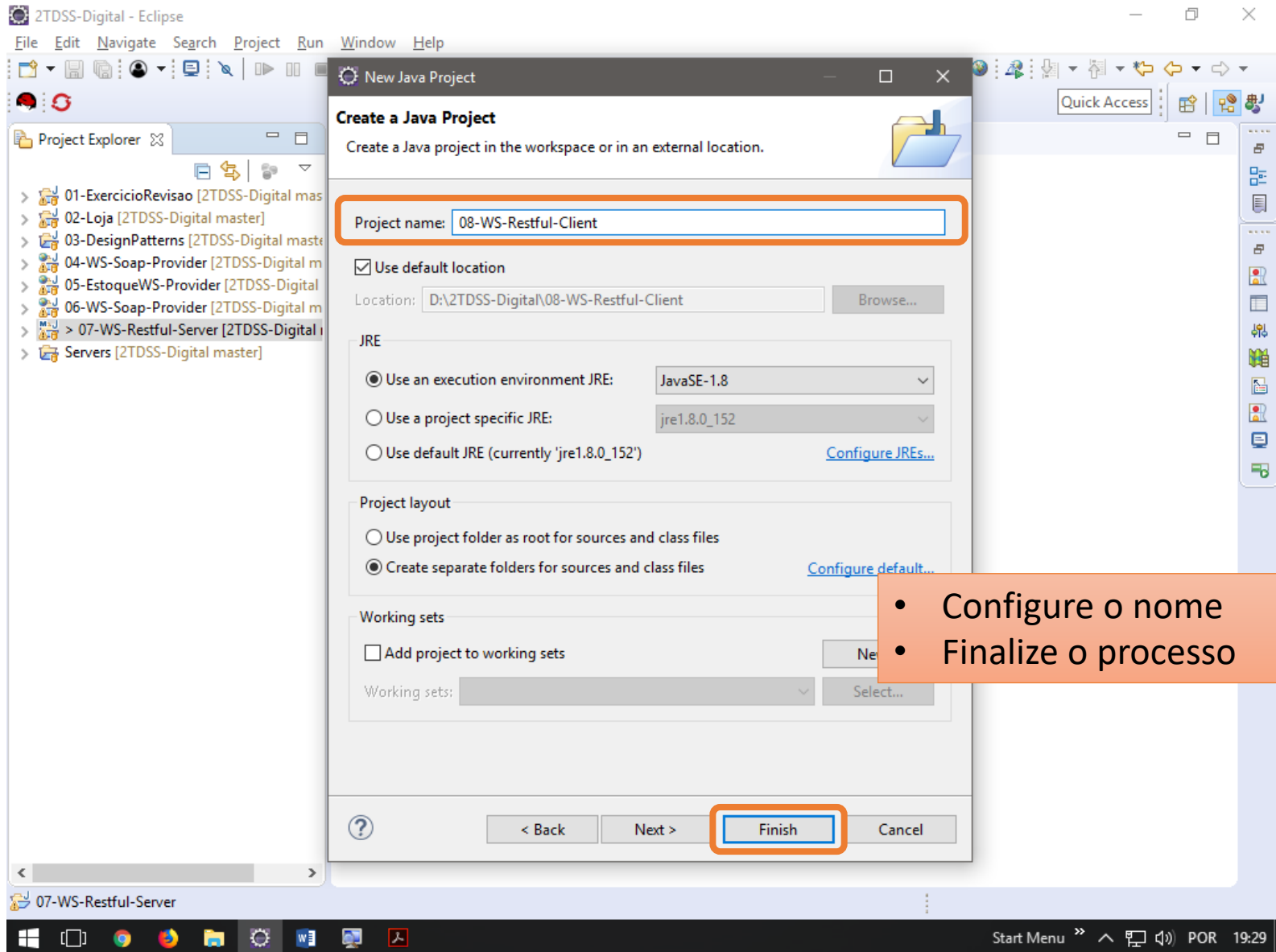


thiagoyama

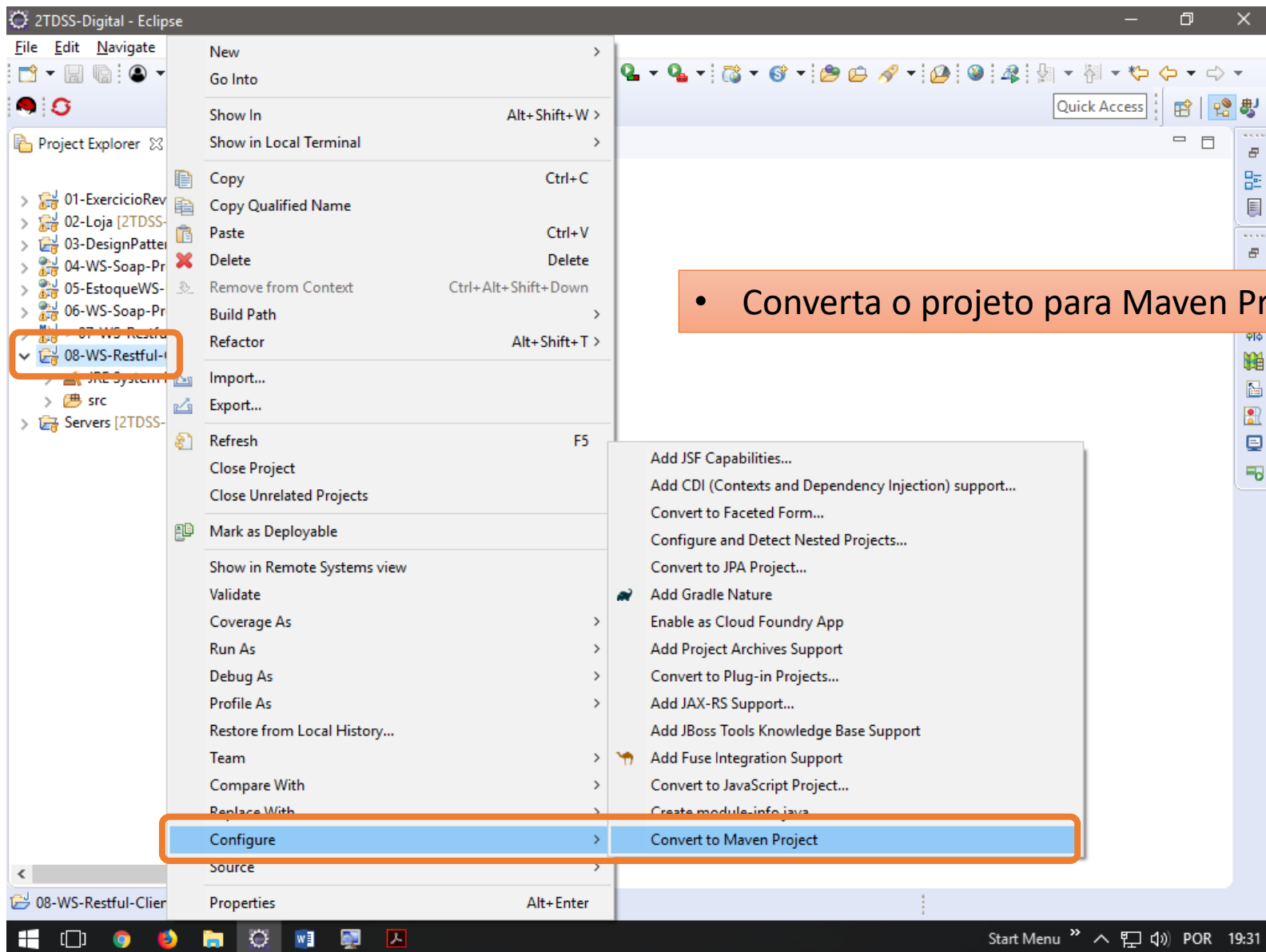


thiagoyama@gmail.com

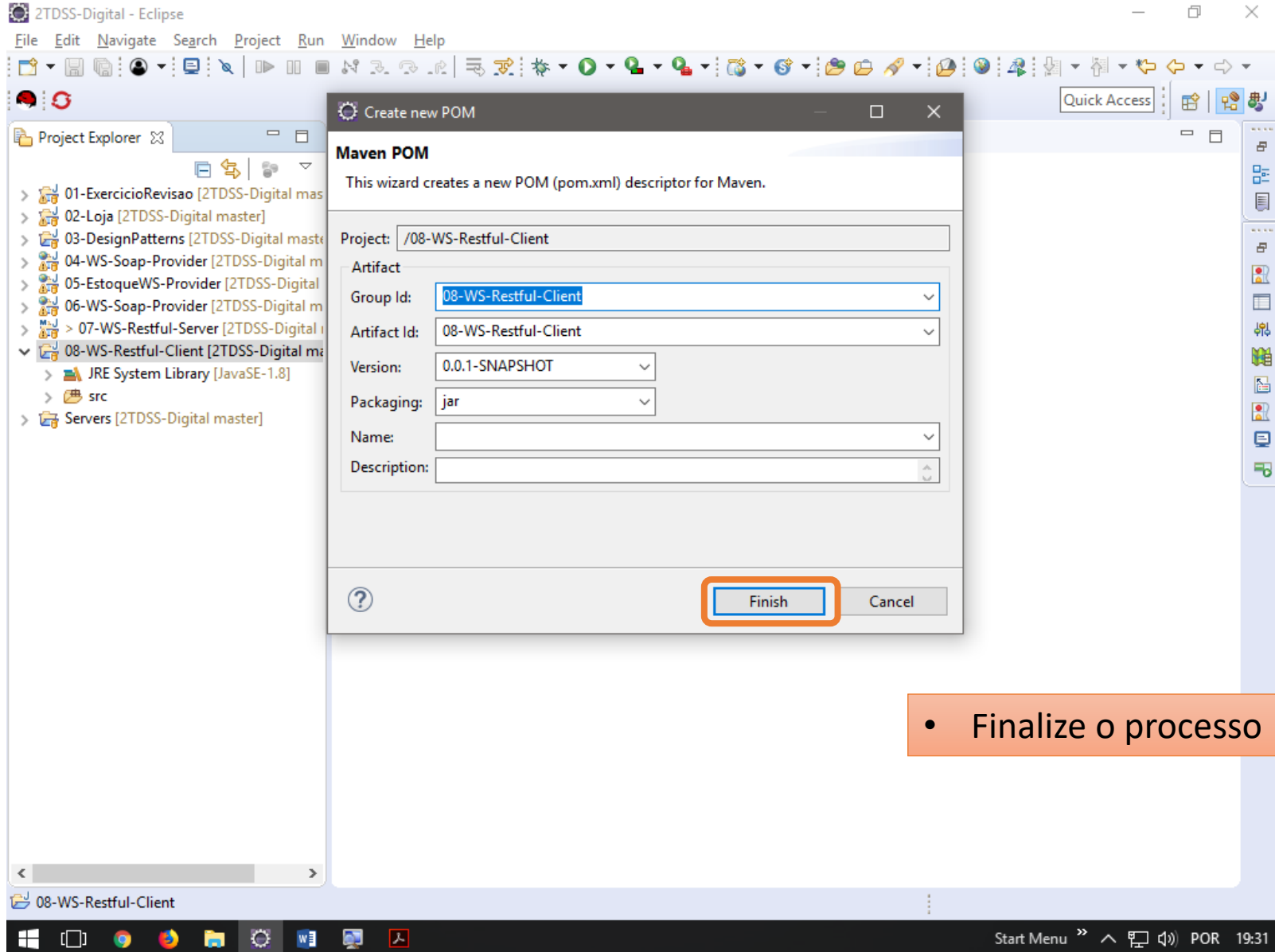




MAVEN PROJECT



MAVEN PROJECT



- Finalize o processo

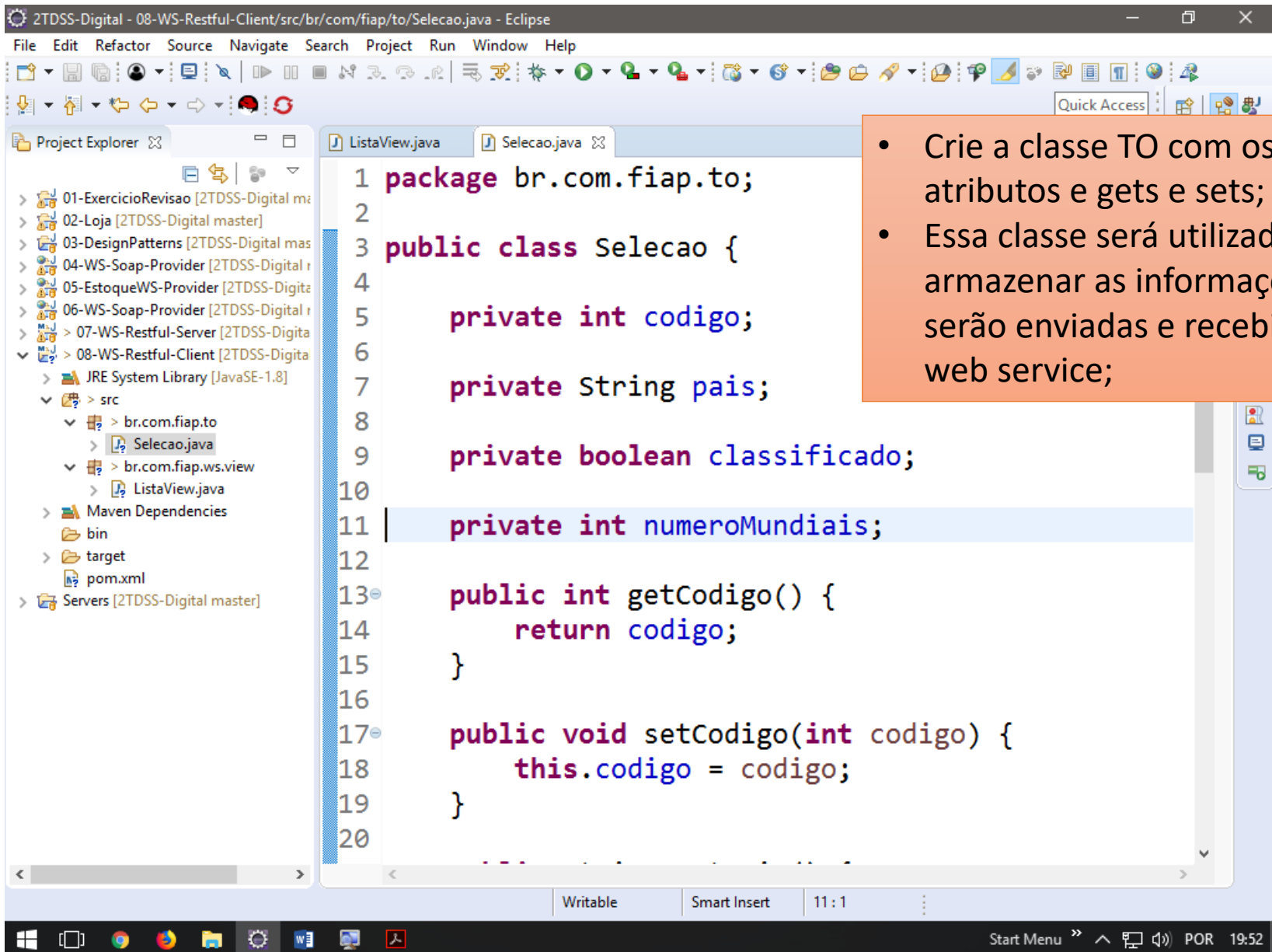
DEPENDÊNCIAS DO RESTFUL CLIENT – POM.XML

- Configure as dependências no pom.xml

```
16         </configuration>
17     </plugin>
18 </plugins>
19 </build>
20 <dependencies>
21     <dependency>
22         <groupId>com.sun.jersey</groupId>
23         <artifactId>jersey-client</artifactId>
24         <version>1.8</version>
25     </dependency>
26     <dependency>
27         <groupId>org.glassfish.jersey.media</groupId>
28         <artifactId>jersey-media-json-jackson</artifactId>
29         <version>2.25.1</version>
30     </dependency>
31 </dependencies>
32 </project>
```

Overview | Dependencies | Dependency Hierarchy | Effective POM | pom.xml

Building workspace: (75%)



2TDSS-Digital - 08-WS-Restful-Client/src/br/com/fiap/to/Selecao.java - Eclipse

File Edit Refactor Source Navigate Search Project Run Window Help

Project Explorer

- 01-ExercicioRevisao [2TDSS-Digital master]
- 02-Loja [2TDSS-Digital master]
- 03-DesignPatterns [2TDSS-Digital master]
- 04-WS-Soap-Provider [2TDSS-Digital master]
- 05-EstoqueWS-Provider [2TDSS-Digital master]
- 06-WS-Soap-Provider [2TDSS-Digital master]
- 07-WS-Restful-Server [2TDSS-Digital master]
- 08-WS-Restful-Client [2TDSS-Digital master]
 - JRE System Library [JavaSE-1.8]
 - src
 - br.com.fiap.to
 - Selecao.java
 - br.com.fiap.ws.view
 - ListView.java
 - Maven Dependencies
 - bin
 - target
 - pom.xml
 - Servers [2TDSS-Digital master]

```
1 package br.com.fiap.to;
2
3 public class Selecao {
4     private int codigo;
5
6     private String pais;
7
8     private boolean classificado;
9
10    private int numeroMundiais;
11
12
13    public int getCodigo() {
14        return codigo;
15    }
16
17    public void setCodigo(int codigo) {
18        this.codigo = codigo;
19    }
20
```

Writable Smart Insert 11:1

Start Menu ^ POR 19:52

- Crie a classe TO com os atributos e gets e sets;
- Essa classe será utilizada para armazenar as informações que serão enviadas e recebidas do web service;

CLASSE SERVICE

The screenshot shows the Eclipse IDE interface. The title bar indicates the file path: 2TDSS-Digital - 08-WS-Restful-Client/src/br/com/fiap/ws/service/SelecaoService.java - Eclipse. The menu bar includes File, Edit, Refactor, Source, Navigate, Search, Project, Run, Window, and Help. The toolbar contains various icons for file operations, editing, and running. The Project Explorer on the left shows a project structure with several packages and files. The main editor window displays the code for SelecaoService.java, which is currently empty except for the package declaration and class definition. A text box on the right explains the purpose of the class.

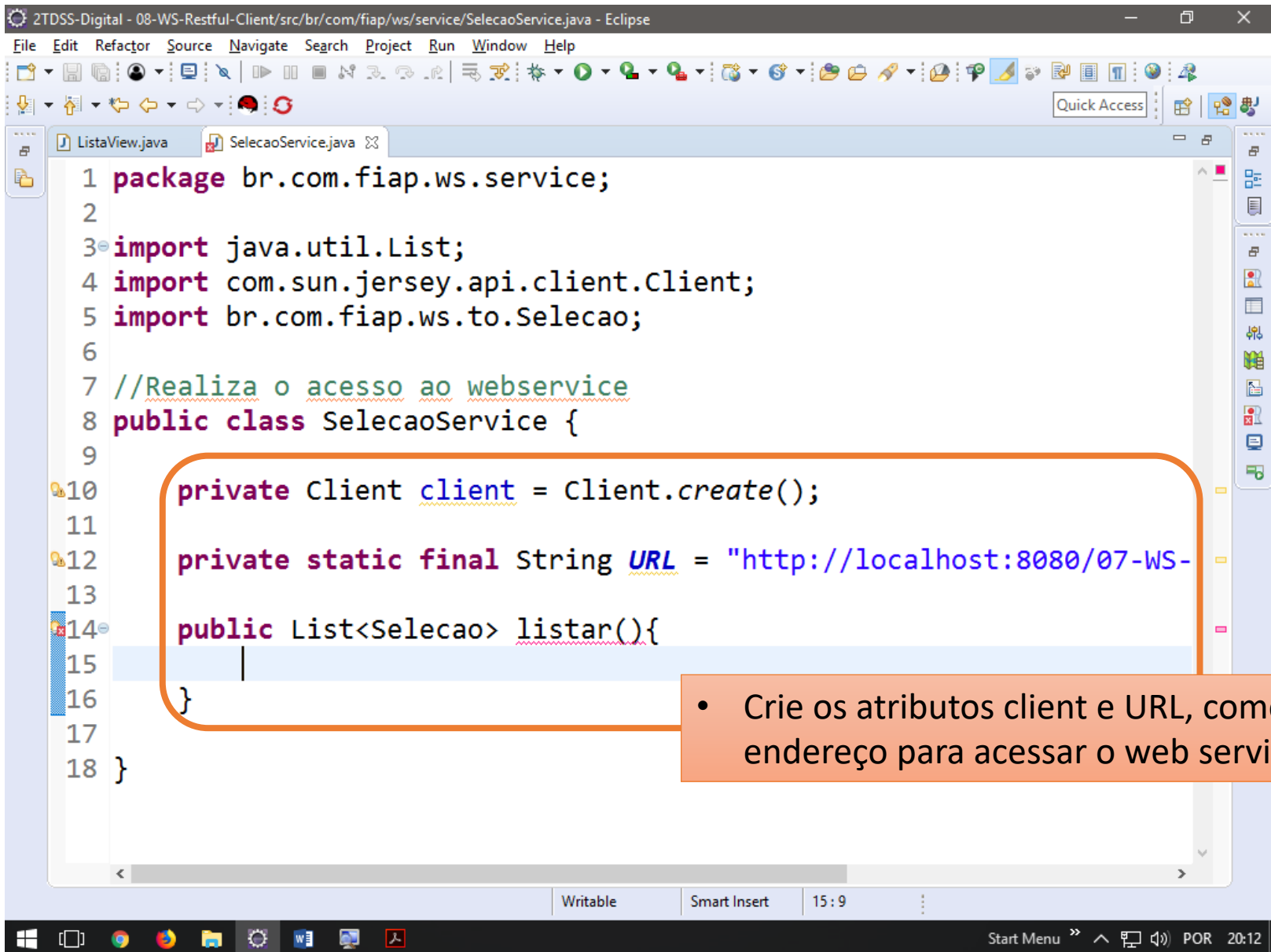
```
1 package br.com.fiap.ws.service;
2
3 public class SelecaoService {
4
5
6
7 }
```

- Cria a classe service, que irá acessar o web service;

Writable Smart Insert 5 : 5

Start Menu » ^ POR 20:06

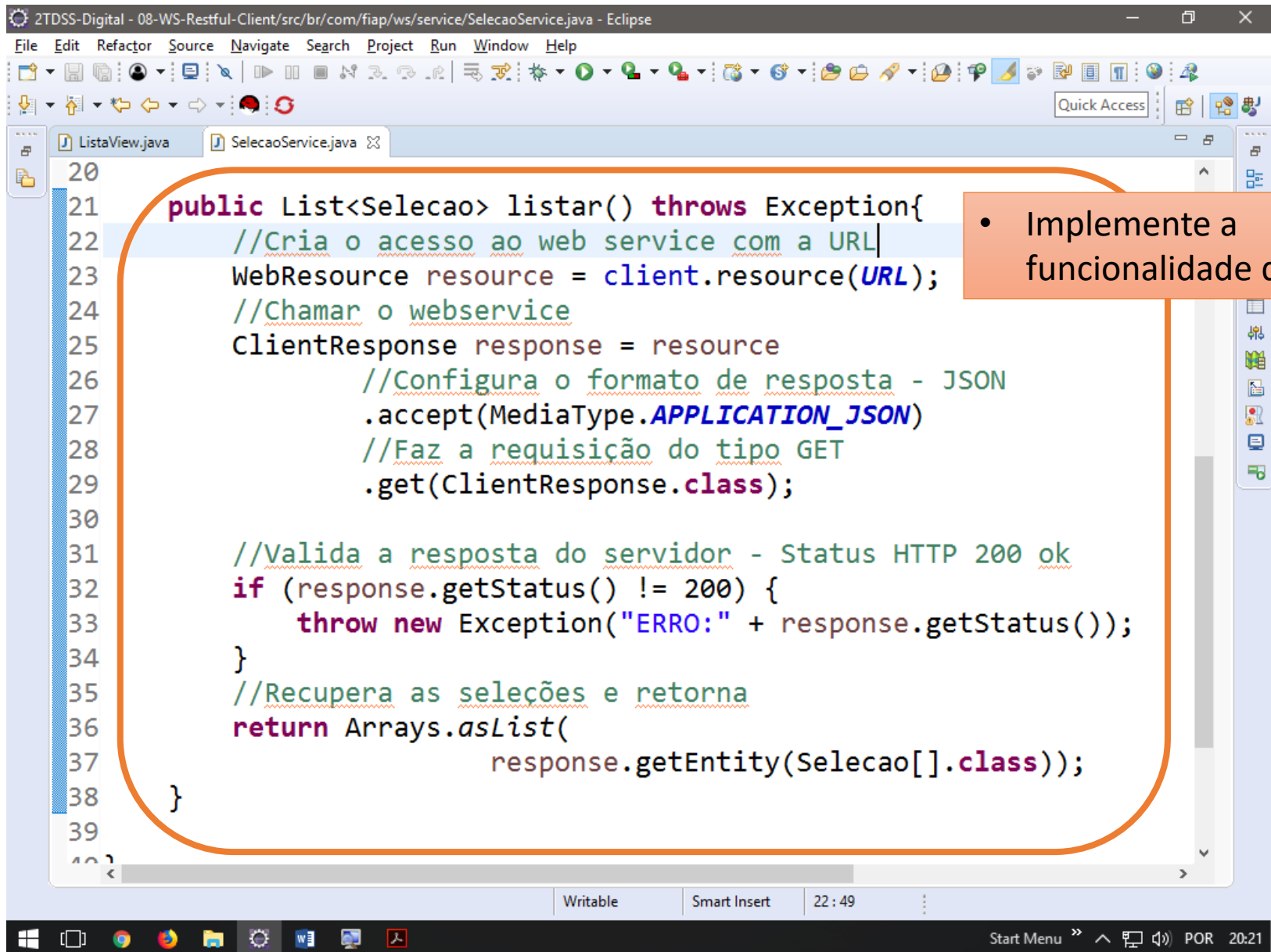
CLASSE SERVICE



```
1 package br.com.fiap.ws.service;
2
3 import java.util.List;
4 import com.sun.jersey.api.client.Client;
5 import br.com.fiap.ws.to.Selecao;
6
7 //Realiza o acesso ao webservice
8 public class SelecaoService {
9
10     private Client client = Client.create();
11
12     private static final String URL = "http://localhost:8080/07-WS-
13
14     public List<Selecao> listar(){
15         |
16     }
17
18 }
```

- Crie os atributos client e URL, como o endereço para acessar o web service;

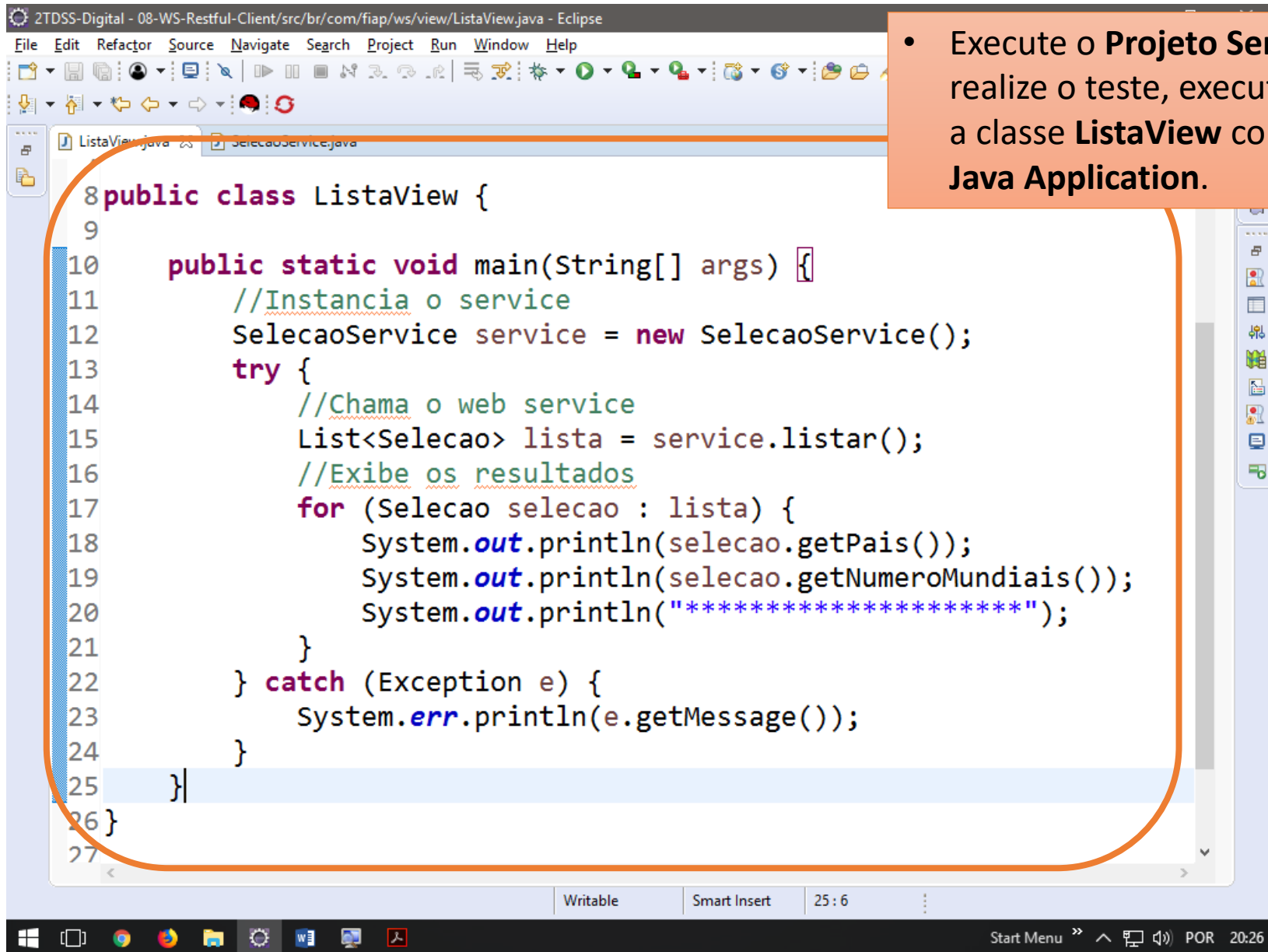
FUNCIONALIDADE DE LISTAR



```
20
21 public List<Selecao> listar() throws Exception{
22     //Cria o acesso ao web service com a URL
23     WebResource resource = client.resource(URL);
24     //Chamar o webservice
25     ClientResponse response = resource
26         //Configura o formato de resposta - JSON
27         .accept(MediaType.APPLICATION_JSON)
28         //Faz a requisição do tipo GET
29         .get(ClientResponse.class);
30
31     //Valida a resposta do servidor - Status HTTP 200 ok
32     if (response.getStatus() != 200) {
33         throw new Exception("ERRO:" + response.getStatus());
34     }
35     //Recupera as seleções e retorna
36     return Arrays.asList(
37         response.getEntity(Selecao[].class));
38 }
39
```

• Implemente a funcionalidade de listar;

TESTE DA LISTAGEM!

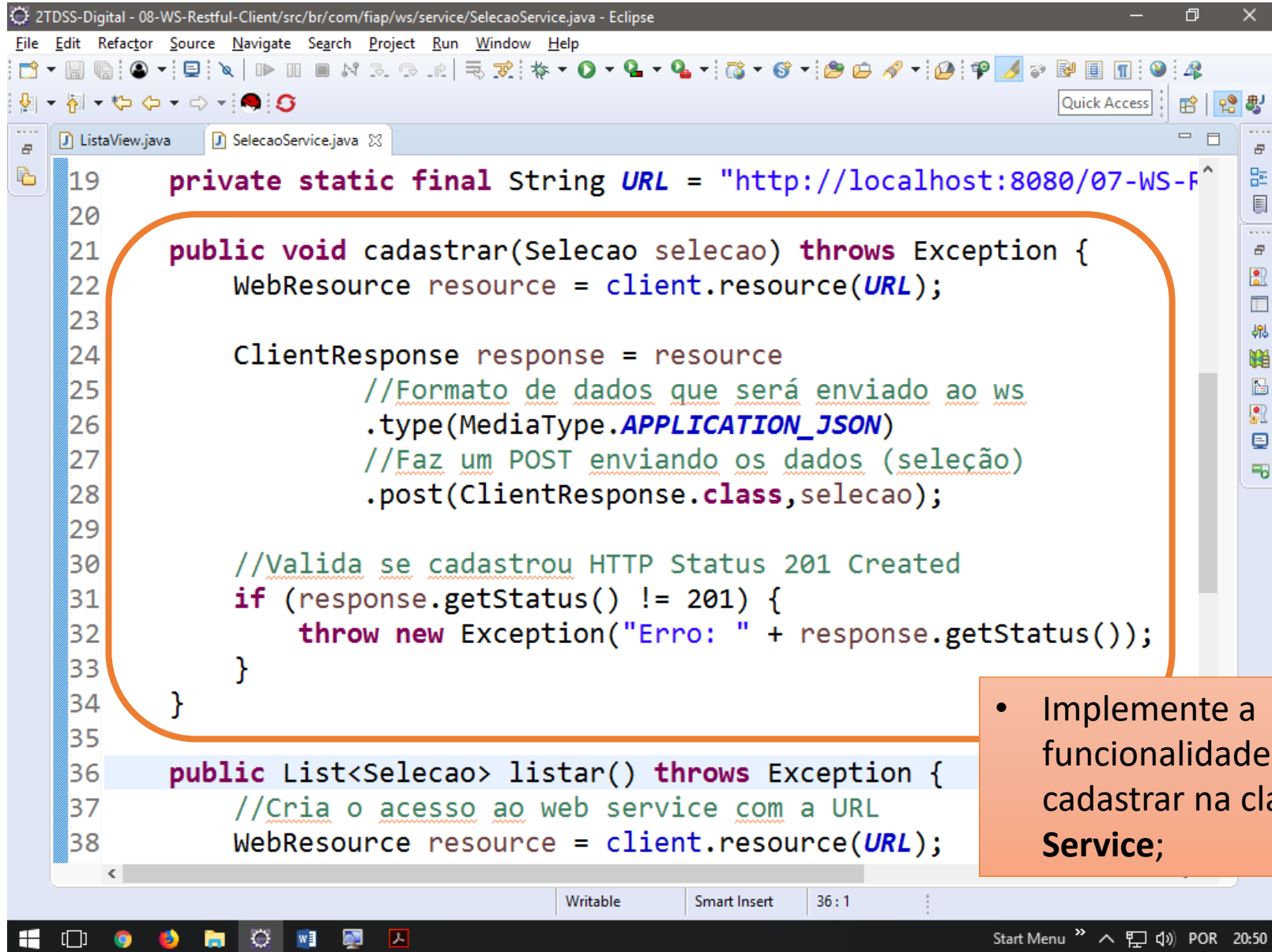


```
2DSS-Digital - 08-WS-Restful-Client/src/br/com/fiap/ws/view/ListView.java - Eclipse
File Edit Refactor Source Navigate Search Project Run Window Help
ListView.java SelecaoService.java
8 public class ListView {
9
10     public static void main(String[] args) {
11         //Instancia o service
12         SelecaoService service = new SelecaoService();
13         try {
14             //Chama o web service
15             List<Selecao> lista = service.listar();
16             //Exibe os resultados
17             for (Selecao selecao : lista) {
18                 System.out.println(selecao.getPais());
19                 System.out.println(selecao.getNumeroMundiais());
20                 System.out.println("*****");
21             }
22         } catch (Exception e) {
23             System.err.println(e.getMessage());
24         }
25     }
26 }
27
```

- Execute o **Projeto Server** e realize o teste, executando a classe **ListView** como **Java Application**.

Writable Smart Insert 25 : 6 Start Menu ^ POR 20:26

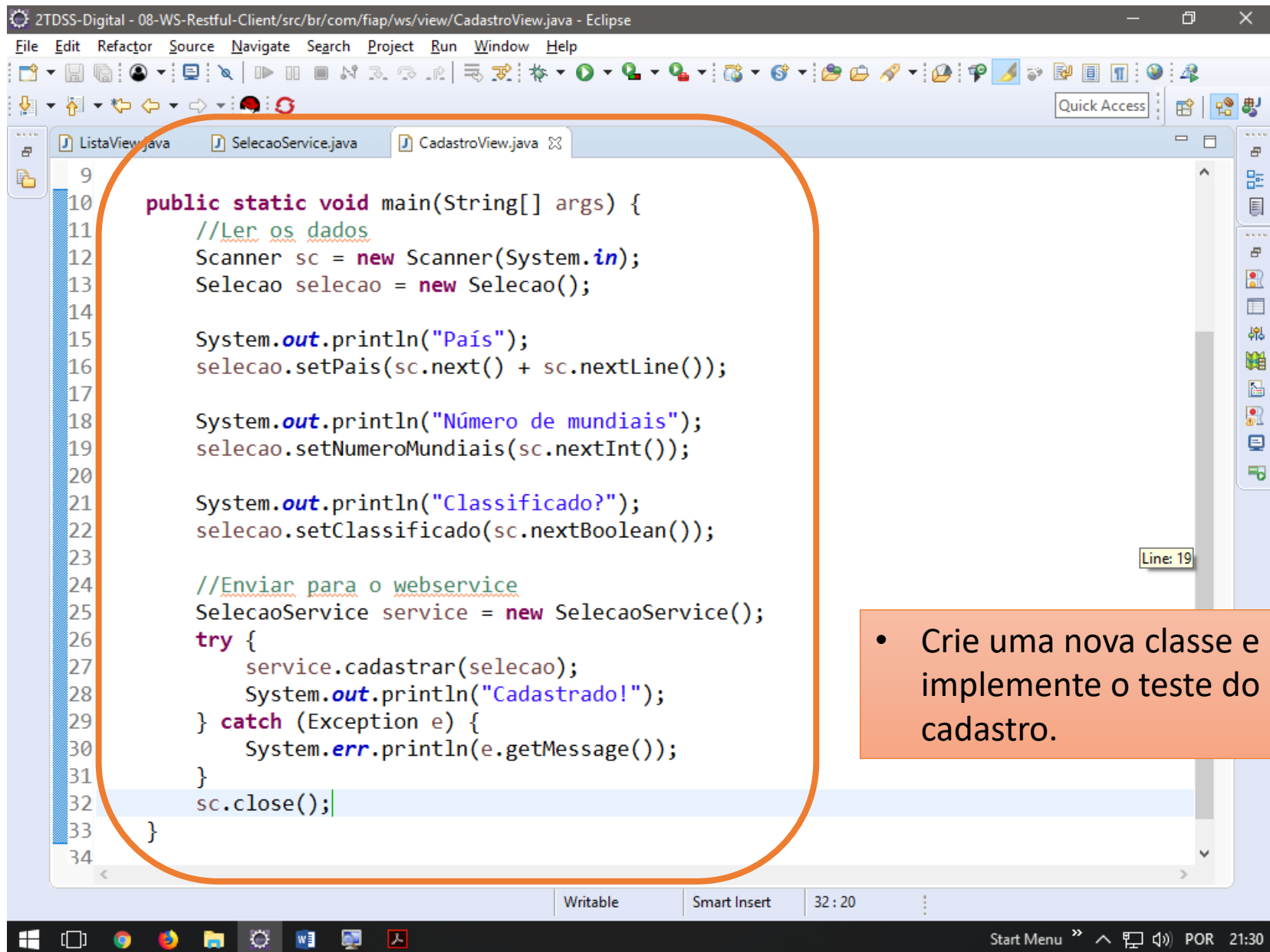
FUNCIONALIDADE DE CADASTRO



```
21 private static final String URL = "http://localhost:8080/07-WS-F
22
23 public void cadastrar(Selecao selecao) throws Exception {
24     WebResource resource = client.resource(URL);
25
26     ClientResponse response = resource
27         //Formato de dados que será enviado ao ws
28         .type(MediaType.APPLICATION_JSON)
29         //Faz um POST enviando os dados (seleção)
30         .post(ClientResponse.class, selecao);
31
32     //Valida se cadastrou HTTP Status 201 Created
33     if (response.getStatus() != 201) {
34         throw new Exception("Erro: " + response.getStatus());
35     }
36 }
37
38 public List<Selecao> listar() throws Exception {
39     //Cria o acesso ao web service com a URL
40     WebResource resource = client.resource(URL);
```

- Implemente a funcionalidade de cadastrar na classe Service;

TESTE DO CADASTRO!



```
2TDSS-Digital - 08-WS-Restful-Client/src/br/com/fiap/ws/view/CadastroView.java - Eclipse
File Edit Refactor Source Navigate Search Project Run Window Help
Quick Access
ListaView.java SelecaoService.java CadastroView.java
9
10 public static void main(String[] args) {
11     //Ler os dados
12     Scanner sc = new Scanner(System.in);
13     Selecao selecao = new Selecao();
14
15     System.out.println("País");
16     selecao.setPais(sc.next() + sc.nextLine());
17
18     System.out.println("Número de mundiais");
19     selecao.setNumeroMundiais(sc.nextInt());
20
21     System.out.println("Classificado?");
22     selecao.setClassificado(sc.nextBoolean());
23
24     //Enviar para o webservice
25     SelecaoService service = new SelecaoService();
26     try {
27         service.cadastrar(selecao);
28         System.out.println("Cadastrado!");
29     } catch (Exception e) {
30         System.err.println(e.getMessage());
31     }
32     sc.close();
33 }
34
Line: 19
Writable Smart Insert 32:20
Start Menu POR 21:30
```

- Crie uma nova classe e implemente o teste do cadastro.

FIAP

15

TESTE DA BUSCA POR CÓDIGO!

```

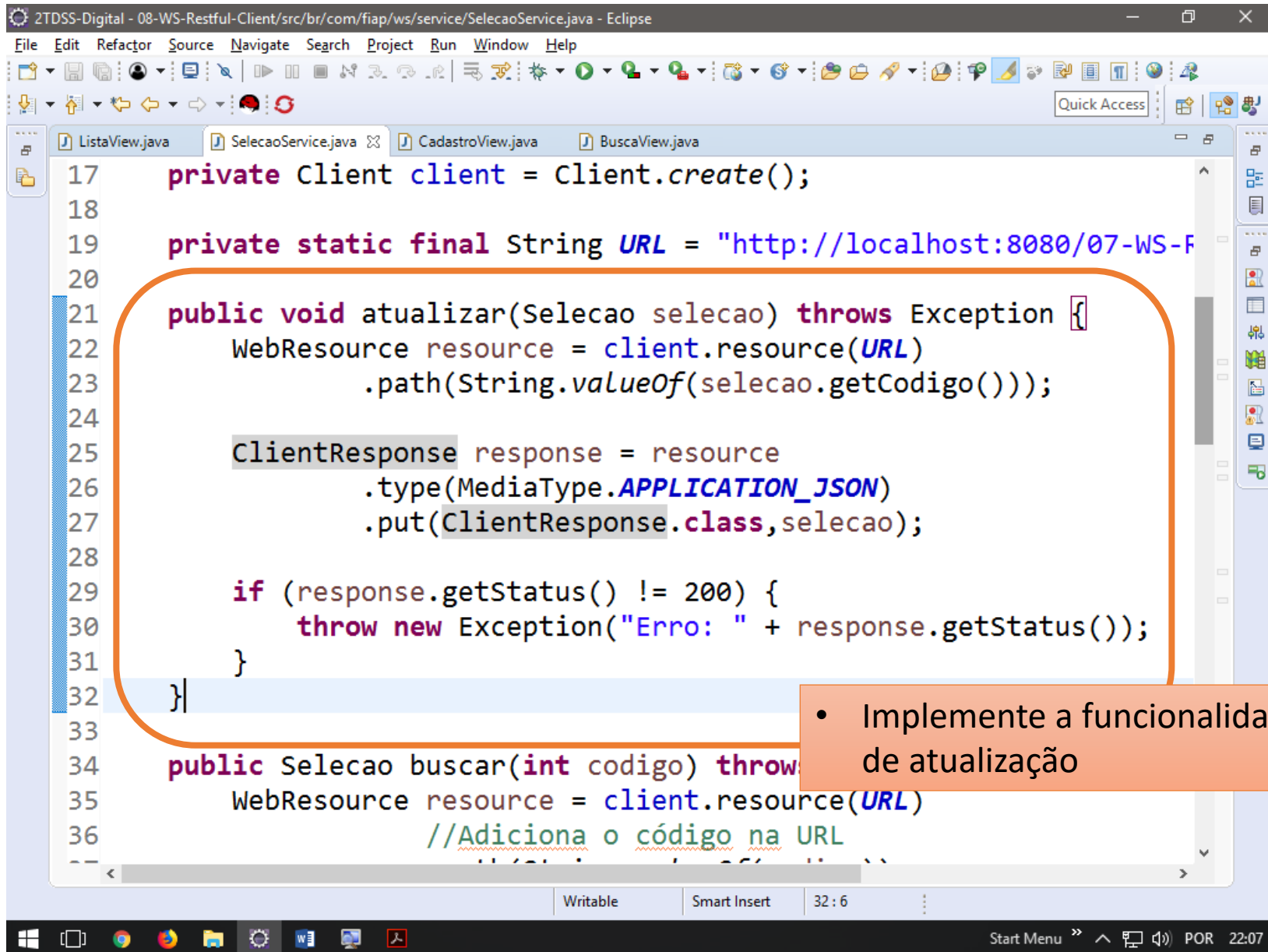
9
10 public static void main(String[] args) {
11     // Ler o código
12     Scanner sc = new Scanner(System.in);
13     System.out.println("Código");
14     int codigo = sc.nextInt();
15
16     try {
17         // Buscar a Seleção no web service
18         SelecaoService service = new SelecaoService();
19         Selecao selecao = service.buscar(codigo);
20         // Exibir os dados
21         System.out.println(selecao.getPais());
22         System.out.println(selecao.getNumeroMundiais());
23         System.out.println(selecao.isClassificado());
24     } catch (Exception e) {
25         System.err.println(e.getMessage());
26     }
27
28     sc.close();
29

```

- Crie uma nova classe para o teste de busca por código.

- Crie uma nova classe para o teste de busca por código.

FUNCIONALIDADE DE ATUALIZAÇÃO



```
2DSS-Digital - 08-WS-Restful-Client/src/br/com/fiap/ws/service/SelecaoService.java - Eclipse
File Edit Refactor Source Navigate Search Project Run Window Help

ListView.java SelecaoService.java CadastroView.java BuscaView.java

17 private Client client = Client.create();
18
19 private static final String URL = "http://localhost:8080/07-WS-F
20
21 public void atualizar(Selecao selecao) throws Exception {
22     WebResource resource = client.resource(URL)
23         .path(String.valueOf(selecao.getCodigo()));
24
25     ClientResponse response = resource
26         .type(MediaType.APPLICATION_JSON)
27         .put(ClientResponse.class, selecao);
28
29     if (response.getStatus() != 200) {
30         throw new Exception("Erro: " + response.getStatus());
31     }
32 }
33
34 public Selecao buscar(int codigo) throw
35     WebResource resource = client.resource(URL)
36     //Adiciona o código na URL
--
```

- Implemente a funcionalidade de atualização

Writable Smart Insert 32 : 6 Start Menu POR 22:07

TESTE DA ATUALIZAÇÃO!

2TDSS-Digital - 08-WS-Testful-Client/src/br/com/fiap/ws/AtualizaView.java - Eclipse

File Edit Refactor Source Navigate Search Project Run Window Help

Quick Access

List View.java SelecaoService.java CadastroView.java Buscaview.java AtualizaView.java

```

8 public class AtualizaView {
9
10     public static void main(String[] args) {
11         Scanner sc = new Scanner(System.in);
12         //Ler os dados
13         Selecao selecao = new Selecao();
14
15         System.out.println("Codigo");
16         selecao.setCodigo(sc.nextInt());
17
18         System.out.println("País");
19         selecao.setPais(sc.next() + sc.nextLine());
20
21         System.out.println("Número Mundiais");
22         selecao.setNumeroMundiais(sc.nextInt());
23
24         System.out.println("Classificado?");
25         selecao.setClassificado(sc.nextBoolean());
26
27         //Atualializar através do webservice
28         SelecaoService service = new SelecaoService();
29         try {
30             service.atualizar(selecao);
31             System.out.println("Atualizado!");
32         } catch (Exception e) {
33             System.err.println(e.getMessage());
34         }
35
36         sc.close();
37     }

```

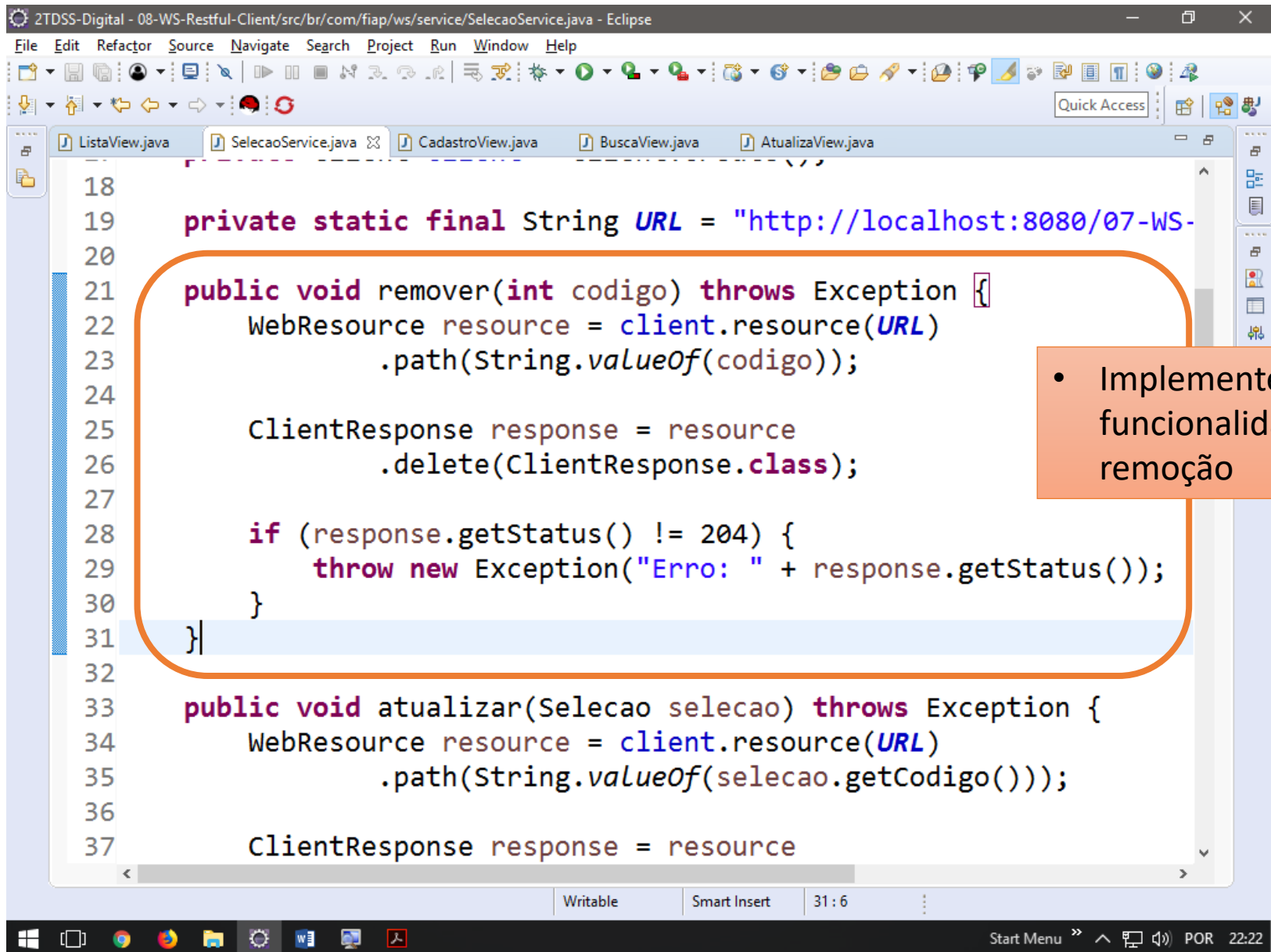
• Crie uma nova classe para realizar o teste de atualização.

Writable Smart Insert 30 : 18

Start Menu ^ > POR 22:17

- Crie uma nova classe para realizar o teste de atualização.

FUNCIONALIDADE DE REMOÇÃO



2TDSS-Digital - 08-WS-Restful-Client/src/br/com/fiap/ws/service/SelecaoService.java - Eclipse

File Edit Refactor Source Navigate Search Project Run Window Help

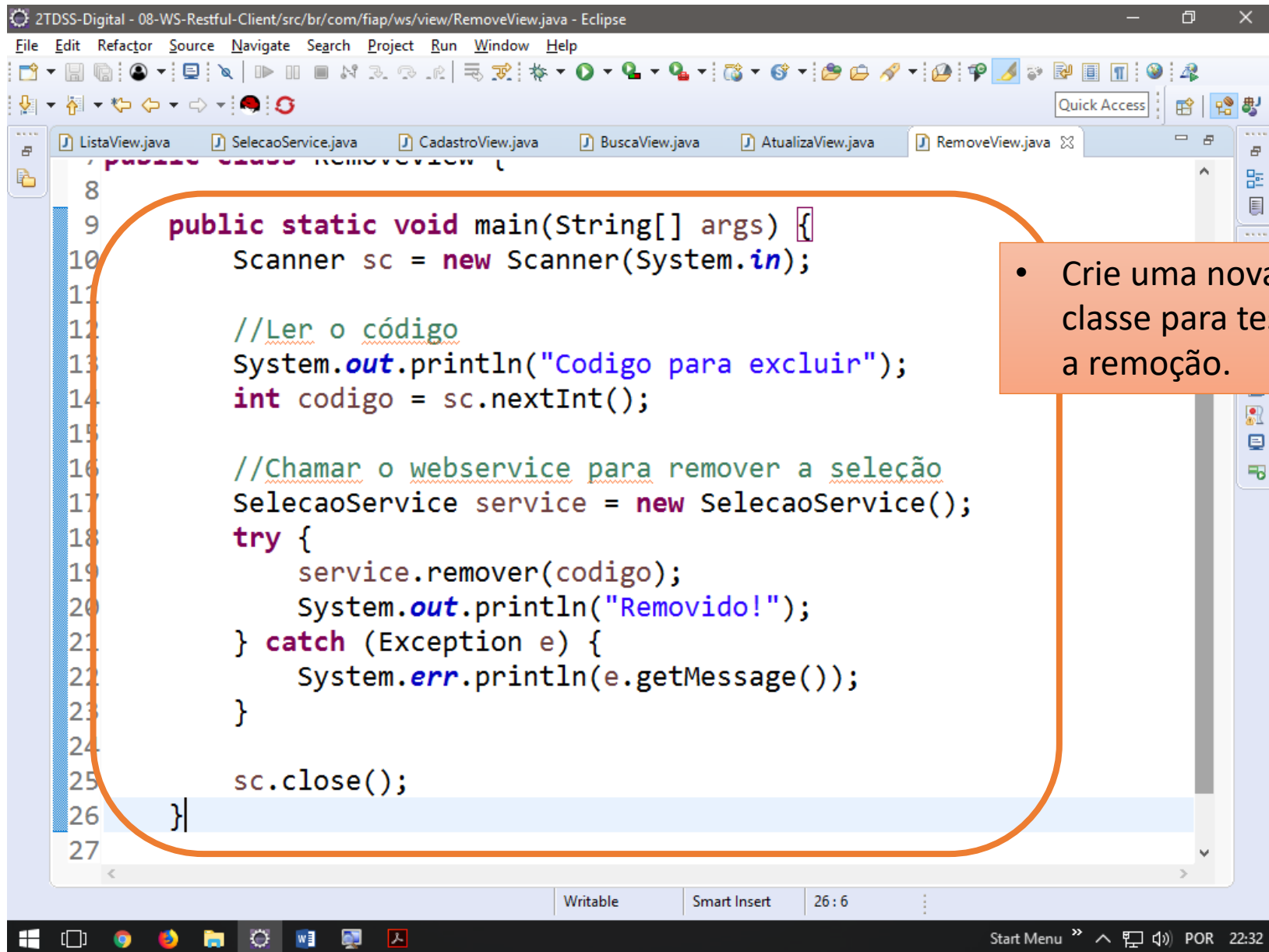
ListView.java SelecaoService.java CadastroView.java BuscaView.java AtualizaView.java

```
18
19 private static final String URL = "http://localhost:8080/07-WS-
20
21 public void remover(int codigo) throws Exception {
22     WebResource resource = client.resource(URL)
23         .path(String.valueOf(codigo));
24
25     ClientResponse response = resource
26         .delete(ClientResponse.class);
27
28     if (response.getStatus() != 204) {
29         throw new Exception("Erro: " + response.getStatus());
30     }
31 }
32
33 public void atualizar(Selecao selecao) throws Exception {
34     WebResource resource = client.resource(URL)
35         .path(String.valueOf(selecao.getCodigo()));
36
37     ClientResponse response = resource
```

- Implemente a funcionalidade de remoção

Writable Smart Insert 31 : 6 Start Menu ^ POR 22:22

TESTE DE REMOÇÃO!



```
2TDS-Digital - 08-WS-Restful-Client/src/br/com/fiap/ws/view/RemoveView.java - Eclipse
File Edit Refactor Source Navigate Search Project Run Window Help
ListaView.java SelecaoService.java CadastroView.java BuscaView.java AtualizaView.java RemoveView.java
8
9 public static void main(String[] args) {
10     Scanner sc = new Scanner(System.in);
11
12     //Ler o código
13     System.out.println("Codigo para excluir");
14     int codigo = sc.nextInt();
15
16     //Chamar o webservice para remover a seleção
17     SelecaoService service = new SelecaoService();
18     try {
19         service.remover(codigo);
20         System.out.println("Removido!");
21     } catch (Exception e) {
22         System.err.println(e.getMessage());
23     }
24
25     sc.close();
26 }
27
```

- Crie uma nova classe para testar a remoção.

Writable Smart Insert 26 : 6 Start Menu ^ POR 22:32

Copyright © 2013 - 2018 Prof. Me. Thiago T. I. Yamamoto

Todos direitos reservados. Reprodução ou divulgação total ou parcial deste documento é expressamente proibido sem o consentimento formal, por escrito, do Professor (autor).

"Você provavelmente terá de entrar em uma batalha mais de uma vez para vencê-la"
Margaret Thatcher