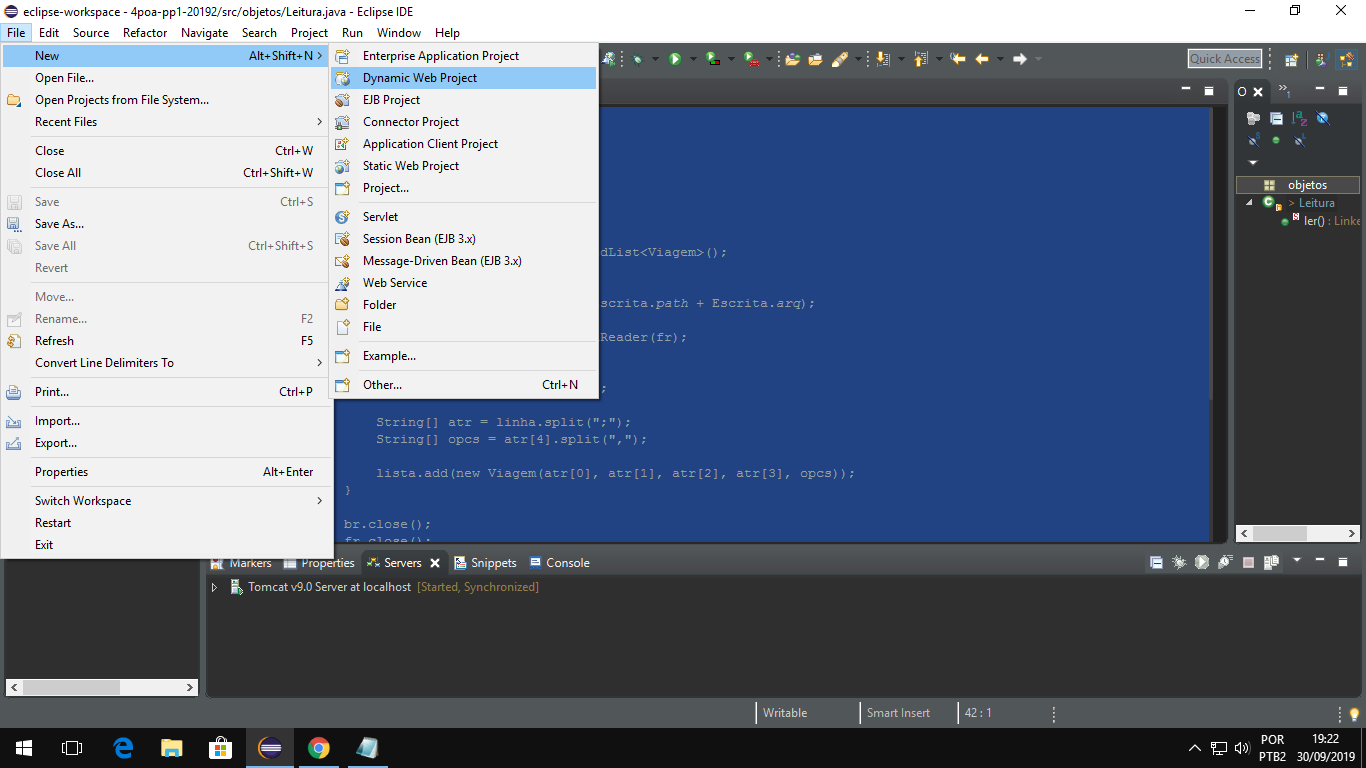
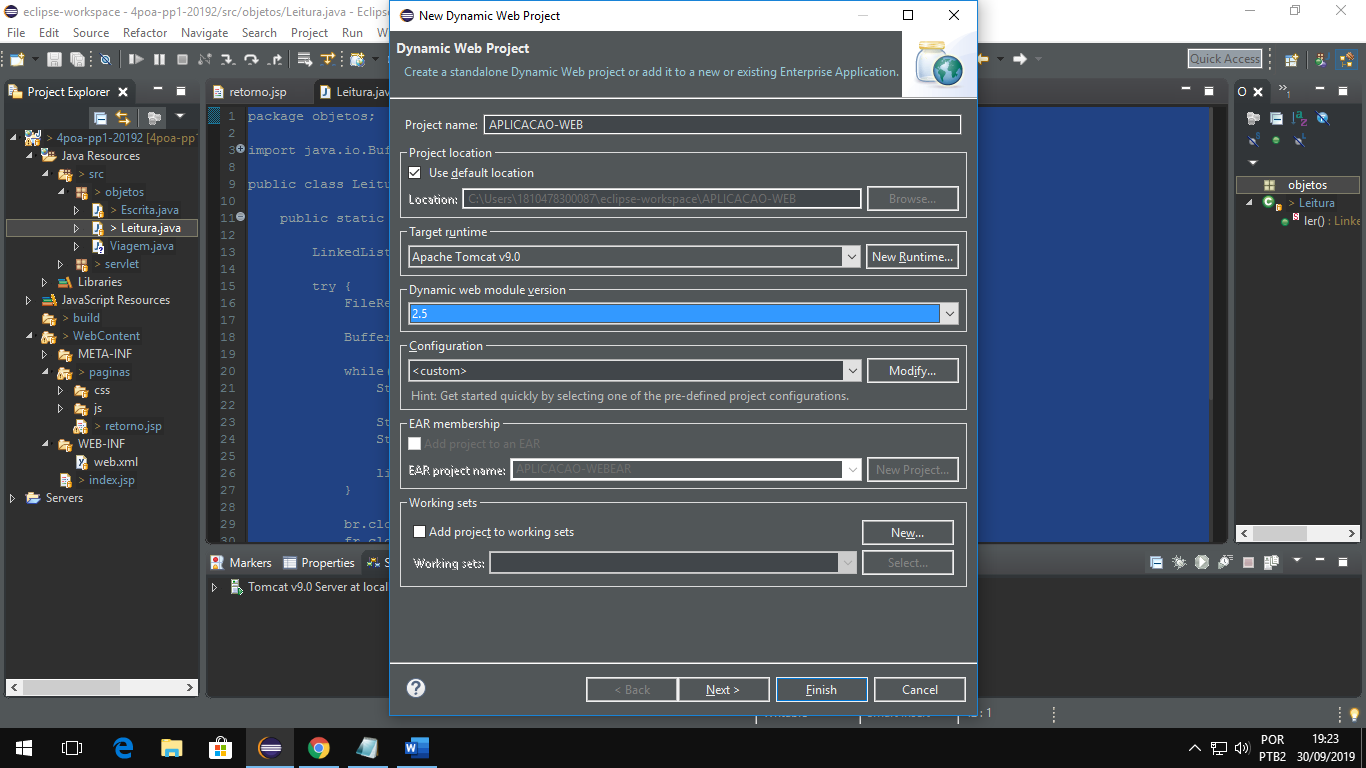
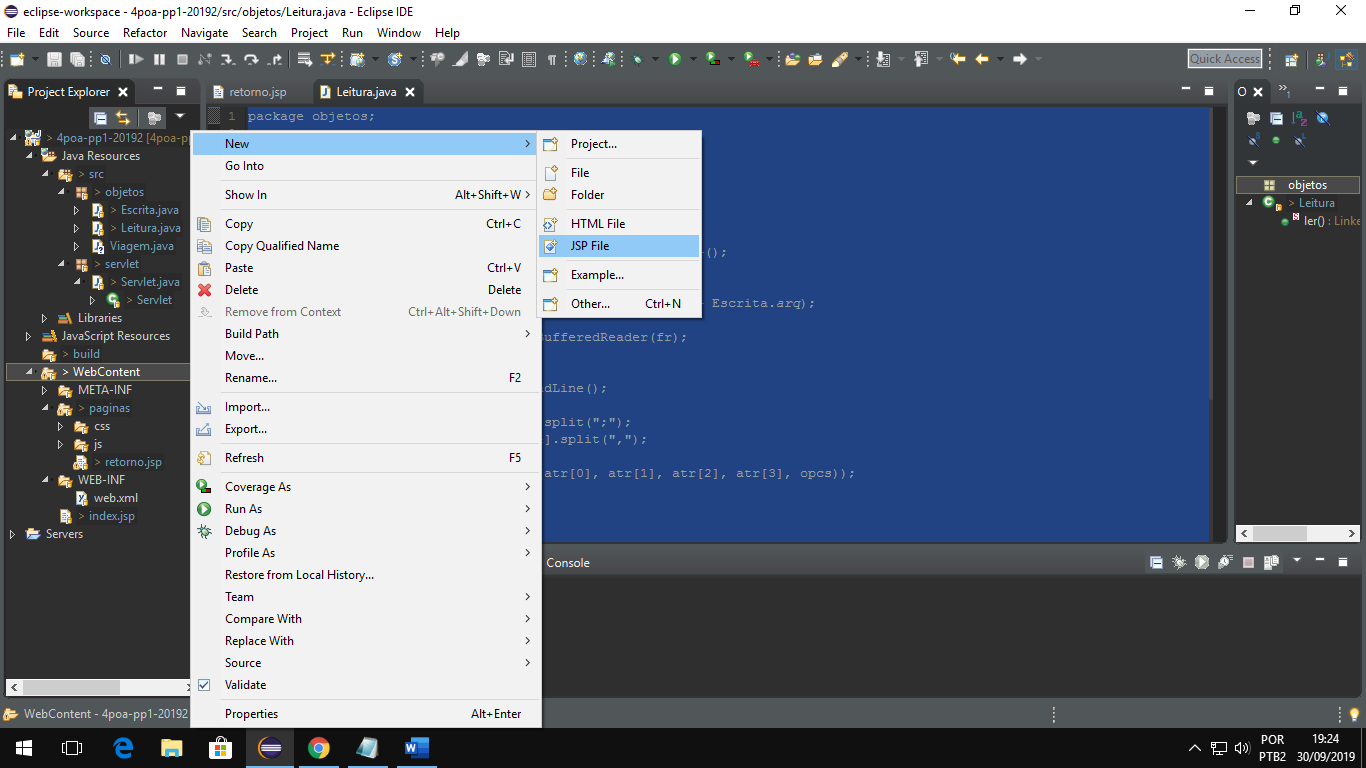
Tutorial para aplicação WEB ECLIPSE + TOMCAT 9.0 / 2.5

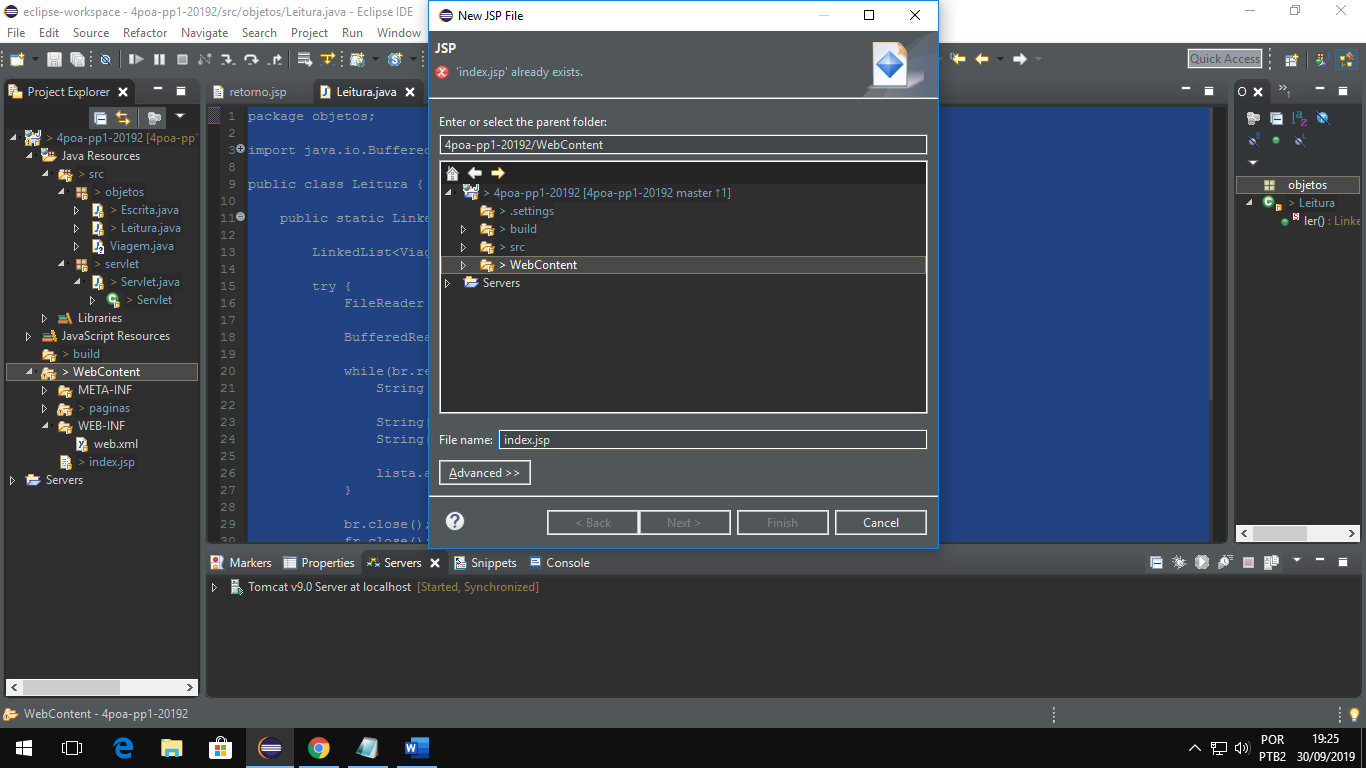
2019/2 – POA – MIGUEL

TÁCITO NUNES // JOÃO SÉRGIO // DIEGO









\WebContent\INDEX.JSP

<%@ page language="java" contentType="text/html; charset=ISO-8859-1"

pageEncoding="ISO-8859-1"%>

<!DOCTYPE html>

<html>

<head>

<meta charset="ISO-8859-1">

<link rel="stylesheet" type="text/css" href="paginas/css/style.css">

<title>TUTORIAL APLICAÇÃO WEB</title>

</head>

<body>

<!-- PASSO 1 - CRIAR O FORMULÁRIO -->

<!-- NO FINAL DESTE ARQUIVO TEM INSTRUÇÕES PARA O PASSO 2 -->

<form action="Servlet" method="post">

<!-- TIPO TEXT -->

<label for="placa">Placa</label><input type="text" class="placa" name="placa" id="placa" required />

<br /><br />

<!-- TIPO RADIO BUTTON -->

<fieldset>

<legend> Montadora </legend>

<input type="radio" name="montadora" id="montadora" value="fiat"> Fiat<br />

<input type="radio" name="montadora" id="montadora" value="ford"> Ford<br />

<input type="radio" name="montadora" id="montadora" value="honda"> Honda<br />

<input type="radio" name="montadora" id="montadora" value="toyota"> Toyota<br />

<input type="radio" name="montadora" id="montadora" value="volkswagen"> Volkswagen<br />

</fieldset>

<br />

<!-- TIPO TEXT -->

<label for="modelo">Modelo</label><input type="text" name="modelo" id="modelo" required />

<!-- TIPO SELECT -->

<label for="ano">Ano</label>

<select name="ano" id="ano">

<option value="2016">2014</option>

<option value="2016">2015</option>

<option value="2016">2016</option>

<option value="2017">2017</option>

<option value="2018">2018</option>

<option value="2019">2019</option>

</select>

<br /><br />

<!-- TIPO CHECKBOX (PERMITIDAS VÁRIAS SELEÇÕES -- RECUPERAR ARRAY-->

<fieldset>

<legend>Opcionais</legend>

<input type="checkbox" name="opcionais" id="opcionais" value="CÂMBIO AUTOM."> Câmbio Automático<br />

<input type="checkbox" name="opcionais" id="opcionais" value="AIRBAG"> Airbag<br />

<input type="checkbox" name="opcionais" id="opcionais" value="FREIO ABS"> Freio ABS<br />

<input type="checkbox" name="opcionais" id="opcionais" value="VIDRO ELÉT."> Vidro Elétrico<br />

</fieldset>

<br />

<!-- BOTÃO PARA ENVIAR INFORMAÇÕES VIA MÉTODO DO FORMULÁRIO -->

<input class="botao" type="submit" value="Enviar">

<input class="botao" style="color: red;" type="reset" value="Redefinir">

</form>

<!-- PASSO 2 - DEPOIS QUE CRIAR O FORMULÁRIO, CRIE UM PACOTE EM: JAVA RESOURCES -> SRC -> servlet

DENTRO DELE, CRIE UM ARQUIVO (FILE) SERVLET E ATRIBUA O NOME QUE VOCÊ DEU NO ACTION DESTE FORMULÁRIO!!!

INDEX TERMINADA!!!

VÁ PRA O SERVLET!!-->

<script type="text/javascript" src="https://code.jquery.com/jquery-latest.min.js"></script>

<script type="text/javascript" src="paginas/js/jquery.mask.min.js"></script>

<script>

$(document).ready(function(){

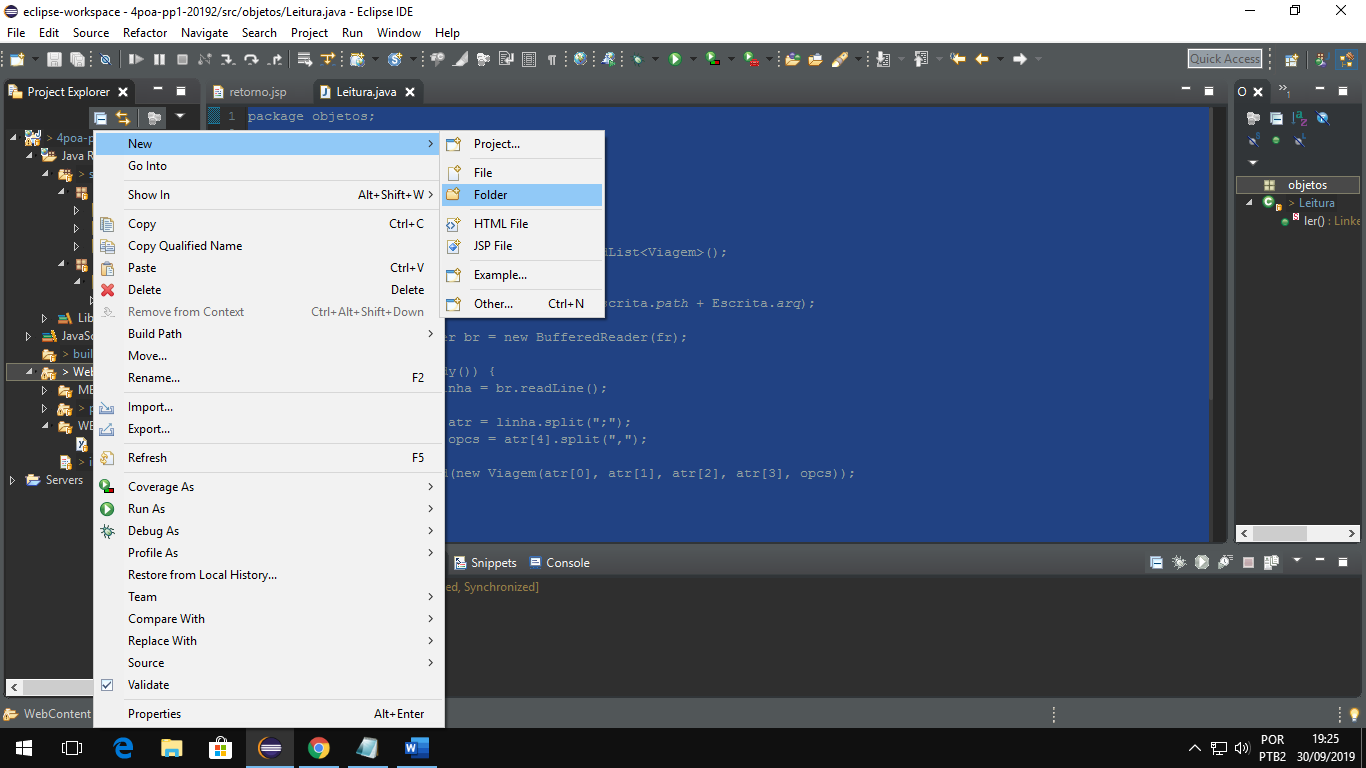
$('#placa').mask('SSS-0A00');

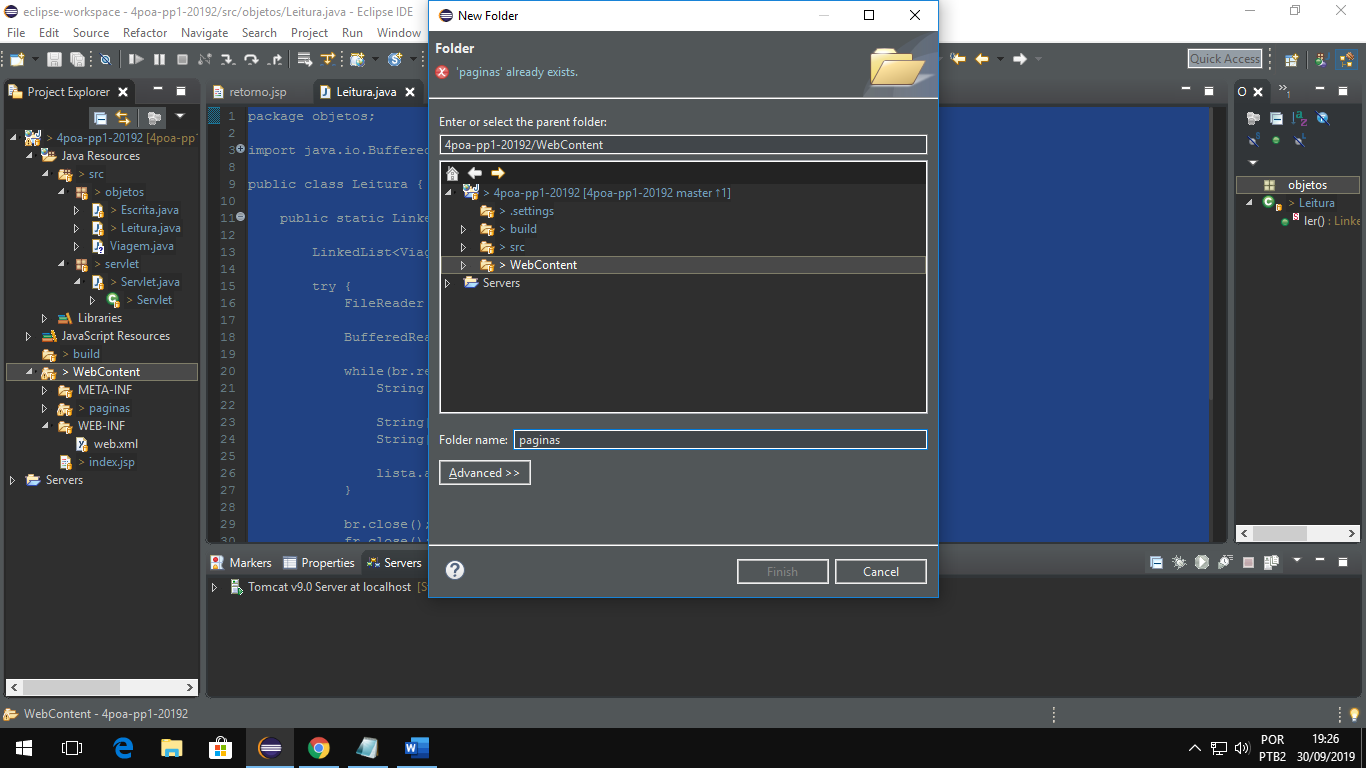
});

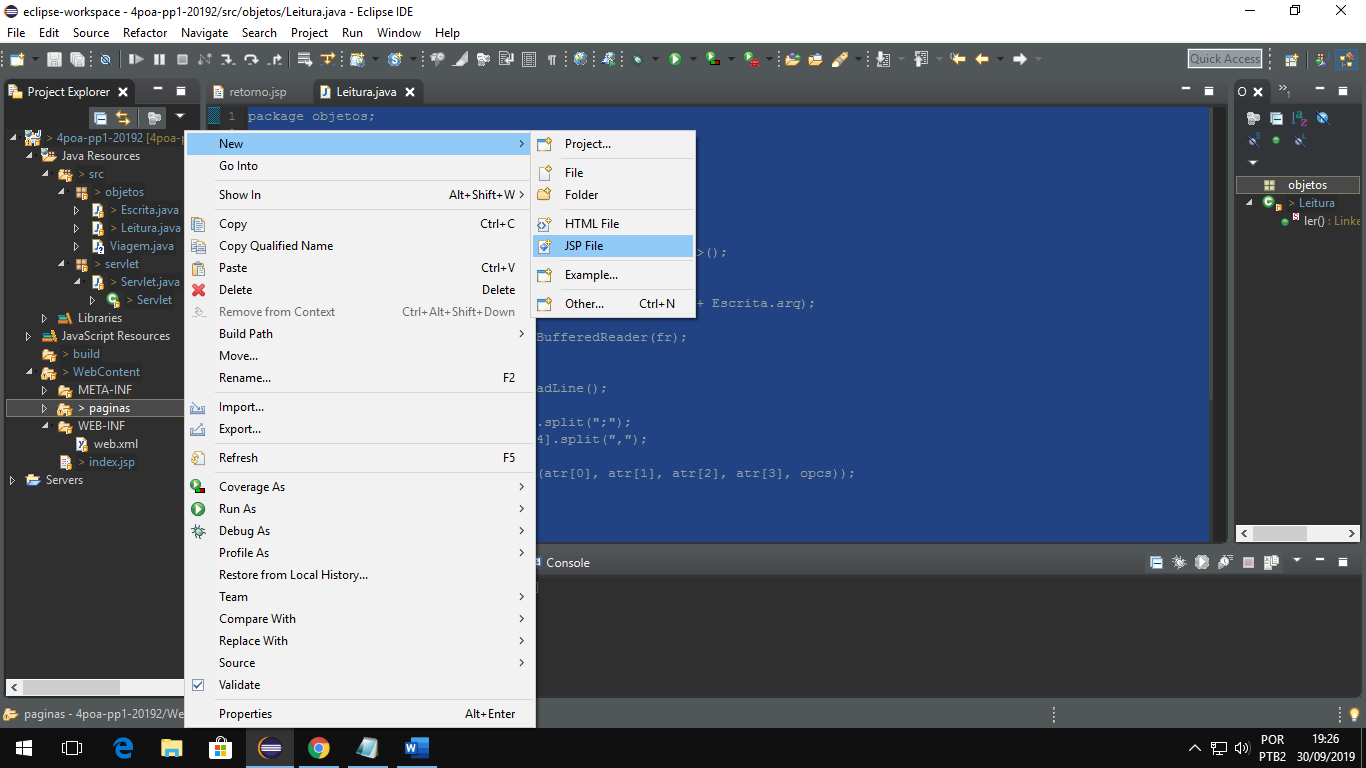
</script>

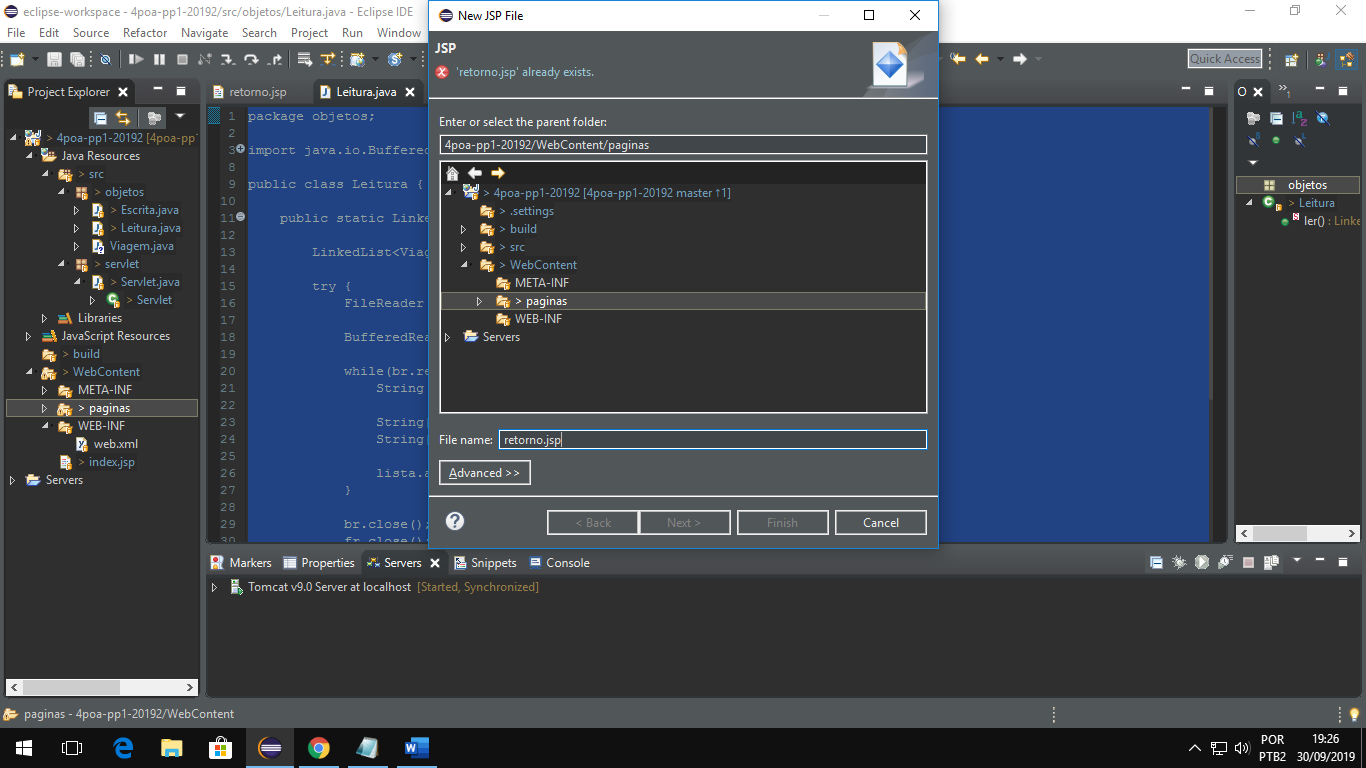
</body>

</html>









\WebContent\paginas\RETORNO.JSP

<%@ page language="java" contentType="text/html; charset=ISO-8859-1"

pageEncoding="ISO-8859-1"%>

<%//NÃO ESQUECER DE IMPORTAR OS OBJETOS UTILIZADOS %>

<%@ page import="java.util.LinkedList"%>

<%@ page import="objetos.Carro"%>

<!DOCTYPE html>

<html>

<head>

<meta charset="ISO-8859-1">

<!-- SE VOCÊ TIVER UM CSSS, IMPORTE-O ABAIXO! -->

<link rel="stylesheet" type="text/css" href="css/style.css">

<title>Listagem de Carros</title>

</head>

<body>

<%

@SuppressWarnings("unchecked")

LinkedList<Carro> listagemRet = (LinkedList<Carro>) session.getAttribute("lista");

%>

<a class="botao" href="../index.jsp">Cadastrar Carro</a>

<table>

<tr>

<th>Placa</th>

<th class="ordenado">Montadora</th>

<th>Modelo</th>

<th>Ano</th>

<th>Opcionais</th>

</tr>

<% for(Carro instancia: listagemRet){ %>

<%int a = instancia.getOpcionais().length + 1; %>

<tr>

<td rowspan="<%=a %>"><%=instancia.getPlaca()%></td>

<td rowspan="<%=a %>"><%=instancia.getMontadora()%></td>

<td rowspan="<%=a %>"><%=instancia.getModelo()%></td>

<td rowspan="<%=a %>"><%=instancia.getAno()%></td>

<%for (String opcional : instancia.getOpcionais()) { %>

<%if(opcional.equals("null")){%>

<tr><td><%="n/a"%></td></tr>

<%} else { %>

<tr><td><%=opcional%></td></tr>

<% } %>

<% } %>

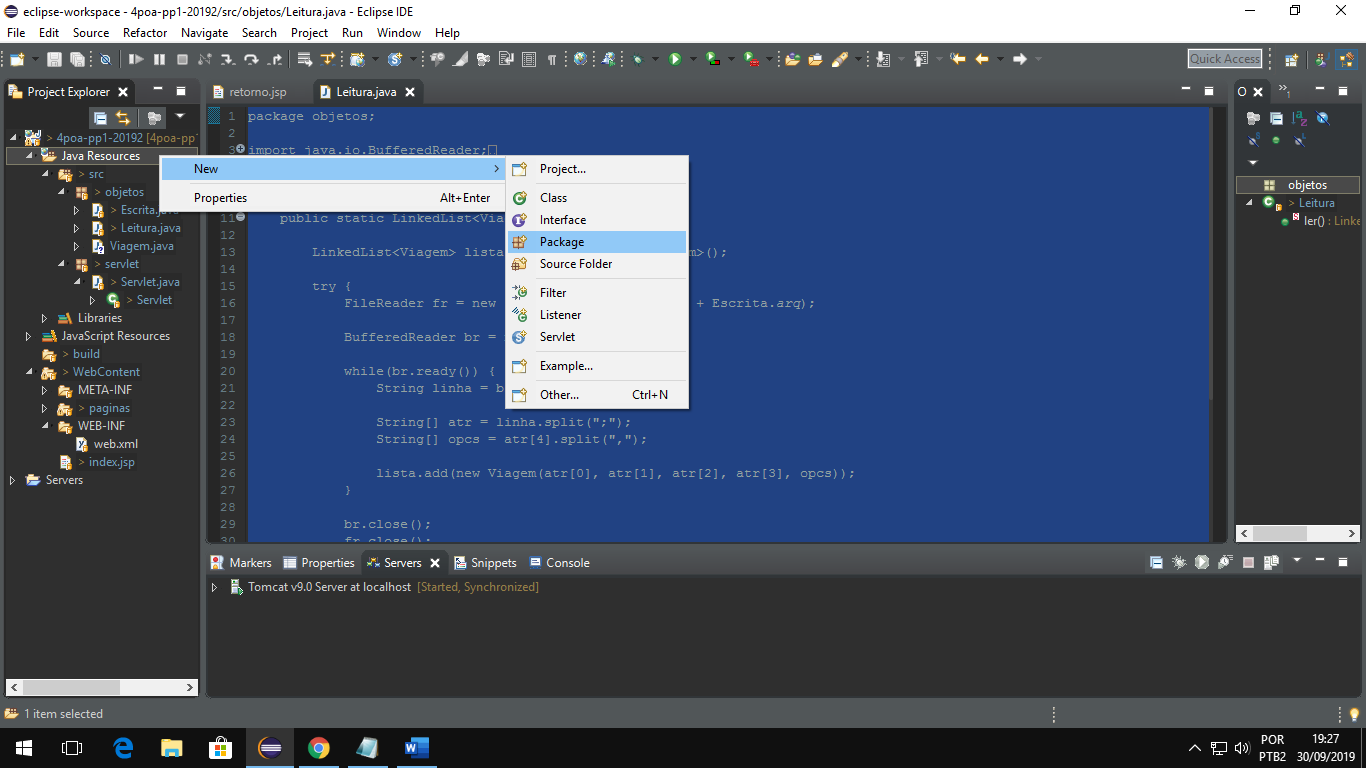
</tr>

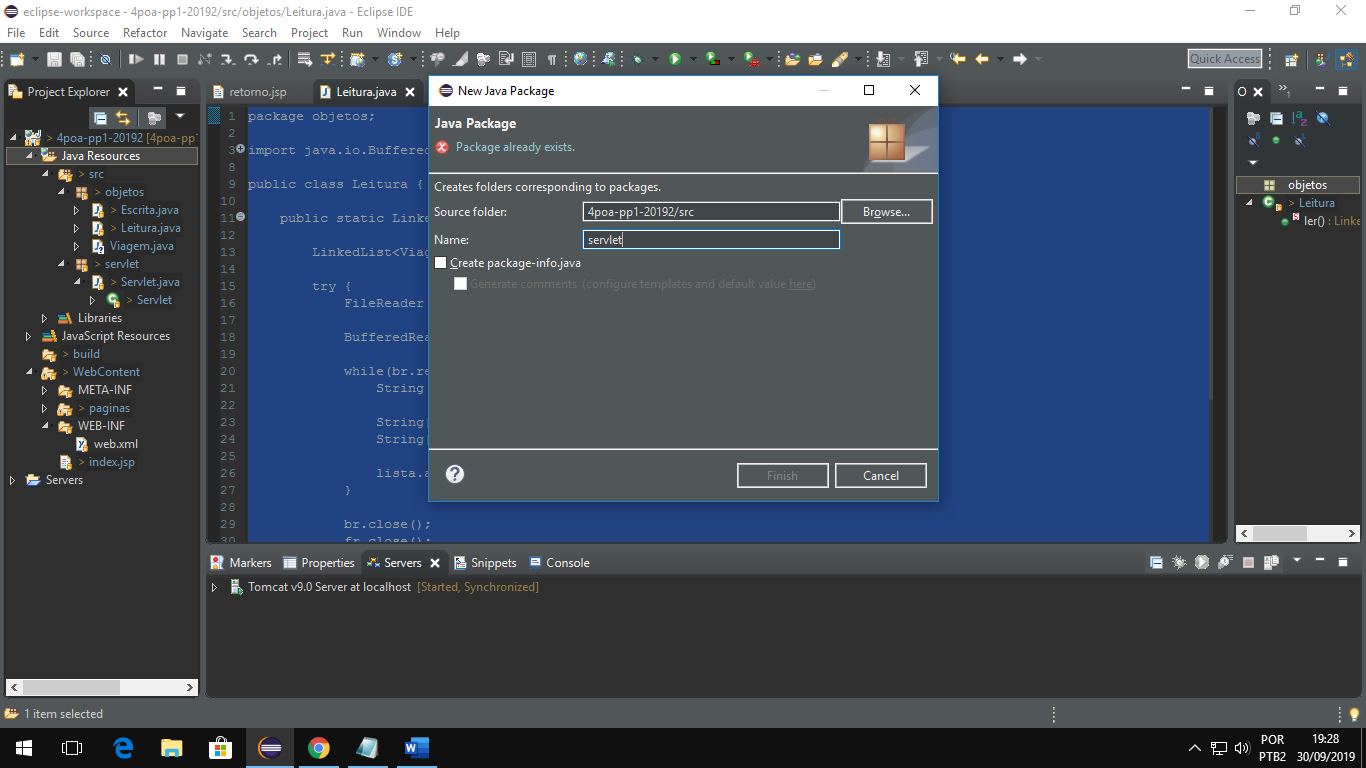
<%} %>

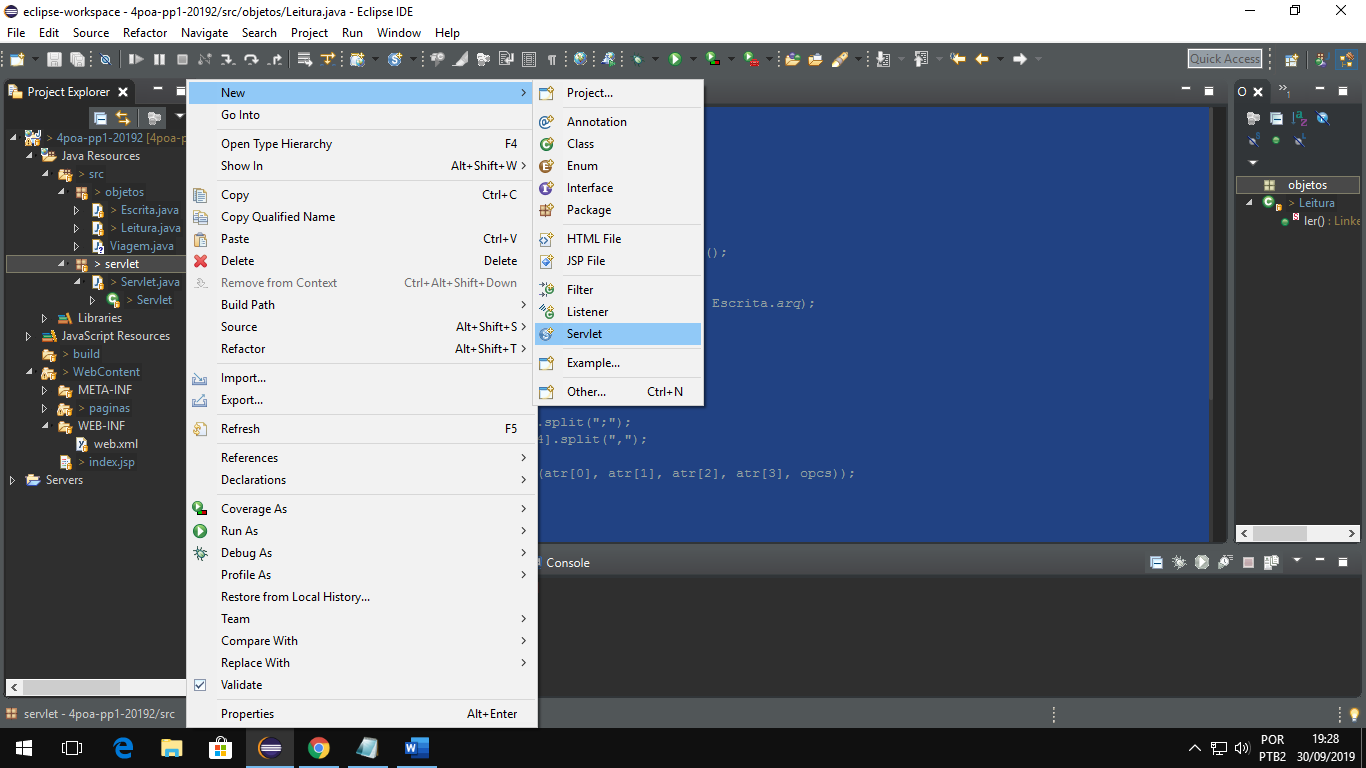
</table>

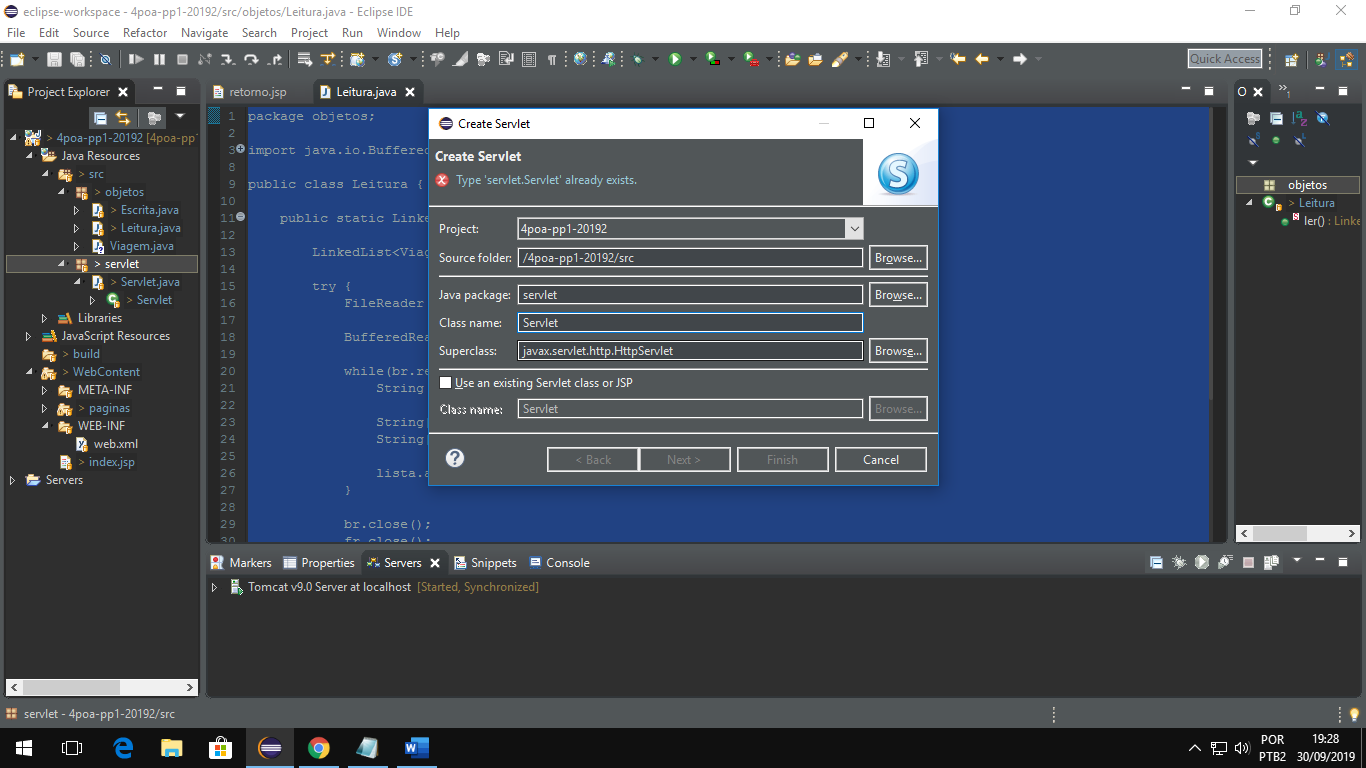
</body>

</html>









\src\servlet\SERVLET.JAVA

package servlet;

//NÃO ESQUECER DE IMPORTAR AS CLASSES: package.NomeDaClasse;

import objetos.Carro;

import objetos.Escrita;

import objetos.Leitura;

import java.util.LinkedList;

import java.util.Collections;

//IMPORTAÇÕES AUTOMÁTICAS (OU DEVERIA)

import java.io.IOException;

import javax.servlet.ServletException;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import javax.servlet.http.HttpSession;

public class Servlet extends HttpServlet {

private static final long serialVersionUID = 1L;

public Servlet() {

super();

}

protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

response.getWriter().append("Served at: ").append(request.getContextPath());

}

protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

// PASSO 2.1 - APAGAR OU COMENTAR O CODIGO ABAIXO

//doGet(request, response);

//PASSO 2.2 - PEGAR OS PARAMETROS DO INDEX

//getParameter("name=id") => PEGAR DO TYPE TEXT, RADIO OU SELECT

String pla = request.getParameter("placa");

String mon = request.getParameter("montadora");

String mod = request.getParameter("modelo");

String ano = request.getParameter("ano");

// getParameterValues("name=id") => PEGAR DO TYPE CHECKBOX (CRIAR UM ARRAY DE ATRIBUTOS "TICADOS"

String[] opc = request.getParameterValues("opcionais");

//PASSO 2.3 - INSTANCIAR HTTPSESSION (CRIAR OU BUSCAR SESSAO) E FAÇA A IMPORTAÇÃO

HttpSession novaSessao = request.getSession();

//import javax.servlet.http.HttpSession;

//PASSO 2.4 - CRIE UM PACOTE EM: "JAVA RESOURCES/SRC" CHAMADO objetos E, DENTRO DELE, UMA NOVA CLASSE

// ESTA SERÁ A SUA CLASSE PRINCIPAL. NO NOSSO CASO AQUI, SERÁ CARRO!!

//PASSO 2.5 - INSTANCIE UM NOVO OBJETO DA CLASSE CRIADA, USANDO O CONSTRUTOR E FAÇA A IMPORTAÇÃO DA CLASSE

//import objetos.Carro;

Carro carro = new Carro(pla, mon, mod, ano, opc);

//PASSO 2.6 - CRIE A CLASSE (PRONTA) ESCRITA E FAÇA A IMPORTAÇÃO DA CLASSE

//import objetos.Escrita;

Escrita.escrever(carro);

//PASSO 2.7 - CRIE A CLASSE (PRONTA) LEITURA E FAÇA A IMPORTAÇÃO

//import objetos.Leitura;

//PASSO 2.8 - INSTANCIE UMA LISTA LIGADA E ATRIBUA AO RETORNO DA FUNÇÃO LER DA CLASSE DE LEITURA E FAÇA A IMPORTAÇÃO

LinkedList<Carro> listagem = Leitura.ler();

//import java.util.LinkedList;

//PASSO 2.9 - ORDENE A SUA LISTAGEM, CONFORME O MÉTODO compareTo PRESENTE NA SUA CLASSE E FAÇA A IMPORTAÇÃO

Collections.sort(listagem);

//import java.util.Collections;

//PASSO 2.10 - ATRIBUA UM NOVO ATRIBUTO PARA CHAMAR NA PÁGINA DE RETORNO PELO MÉTODO DA SESSÃO CRIADA NO PASSO 2.3

novaSessao.setAttribute("lista", listagem);

//PASSO 2.11 - CRIE UMA PASTA DENTRO DE WEBCONTENT E, DENTRO DELA, UM ARQUIVO JSP (NO NOSSO CASO AQUI, retorno.jsp)

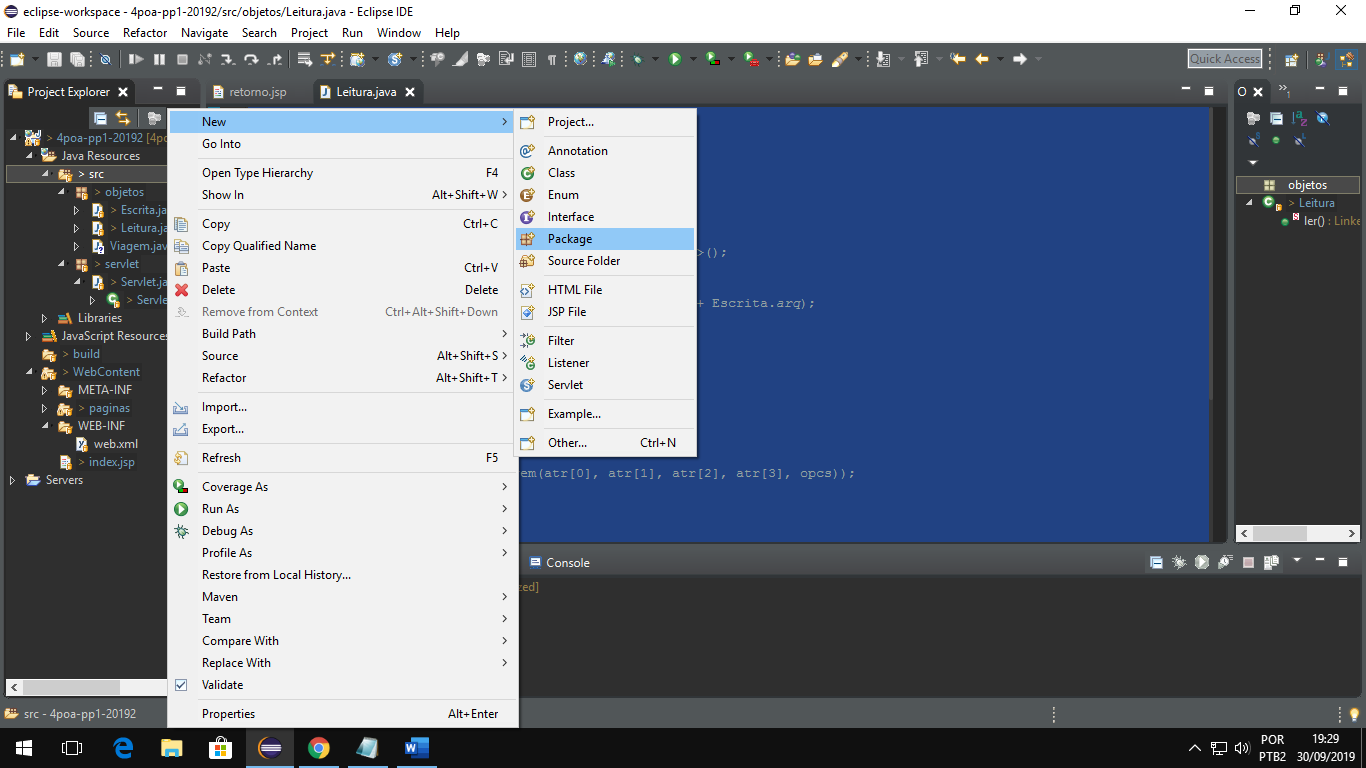
//PASSO 2.12 - CHAME O MÉTODO DE RESPOSTA DA REQUISIÇÃO COM REDIRECIONAMENTO PARA A PÁGINA CRIADA

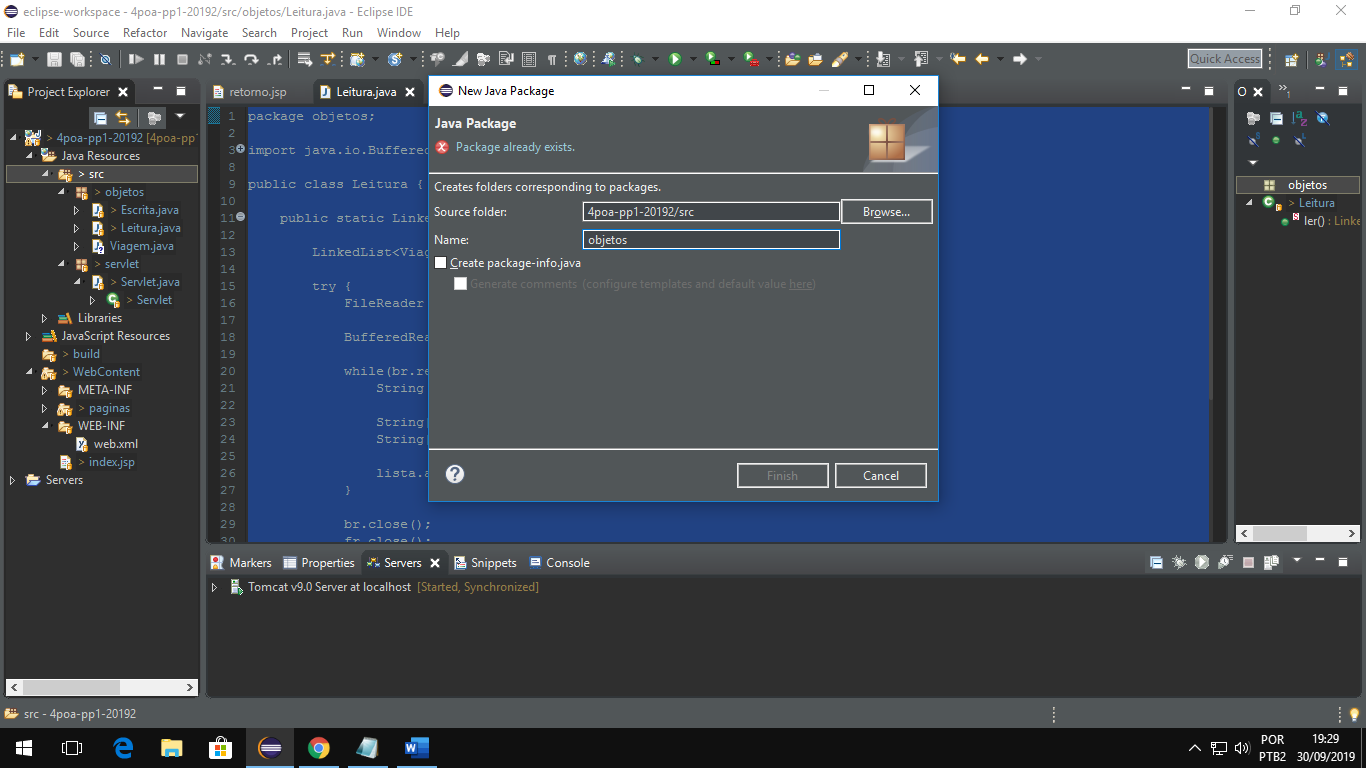
response.sendRedirect("paginas/retorno.jsp");

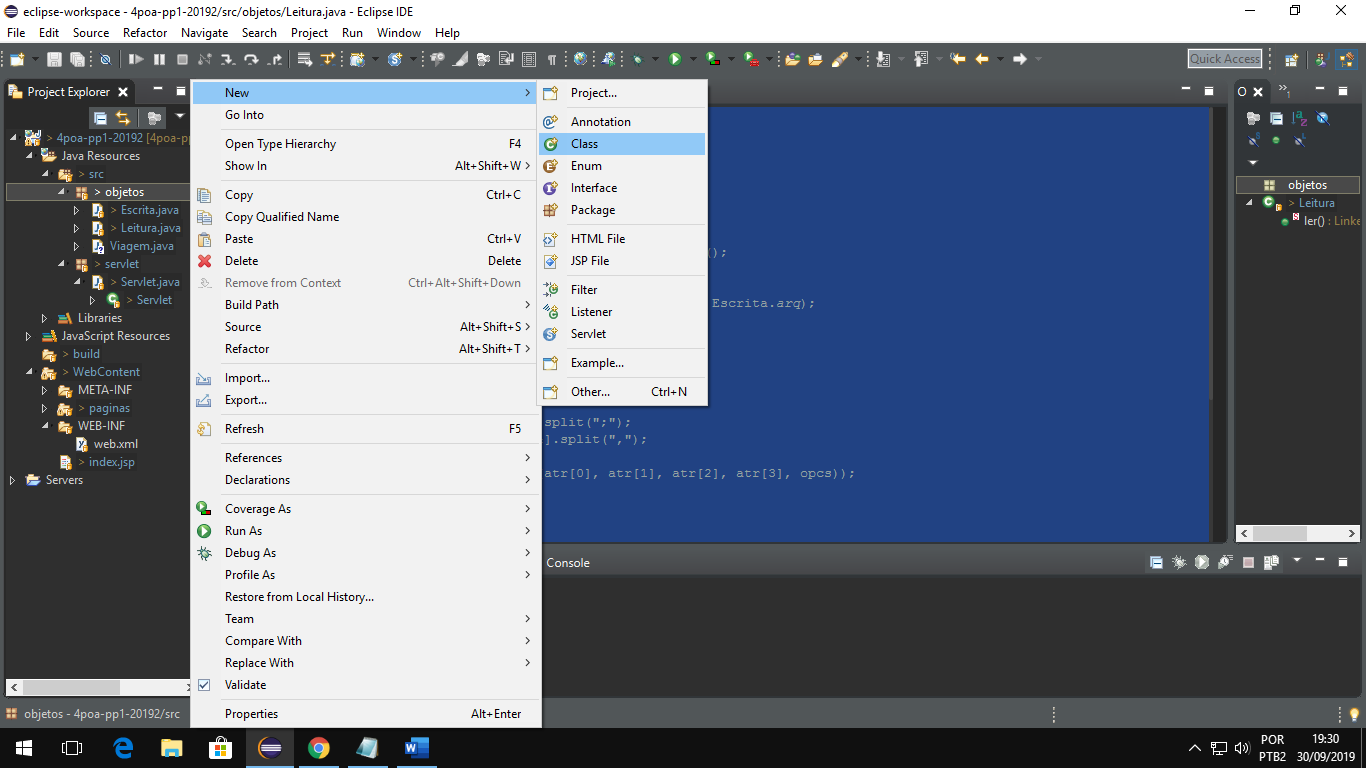
//PASSO 2.13 - ABRA-O. LÁ, TEREMOS MAIS INFORMAÇÕES! ESTÁ ACABANDO!

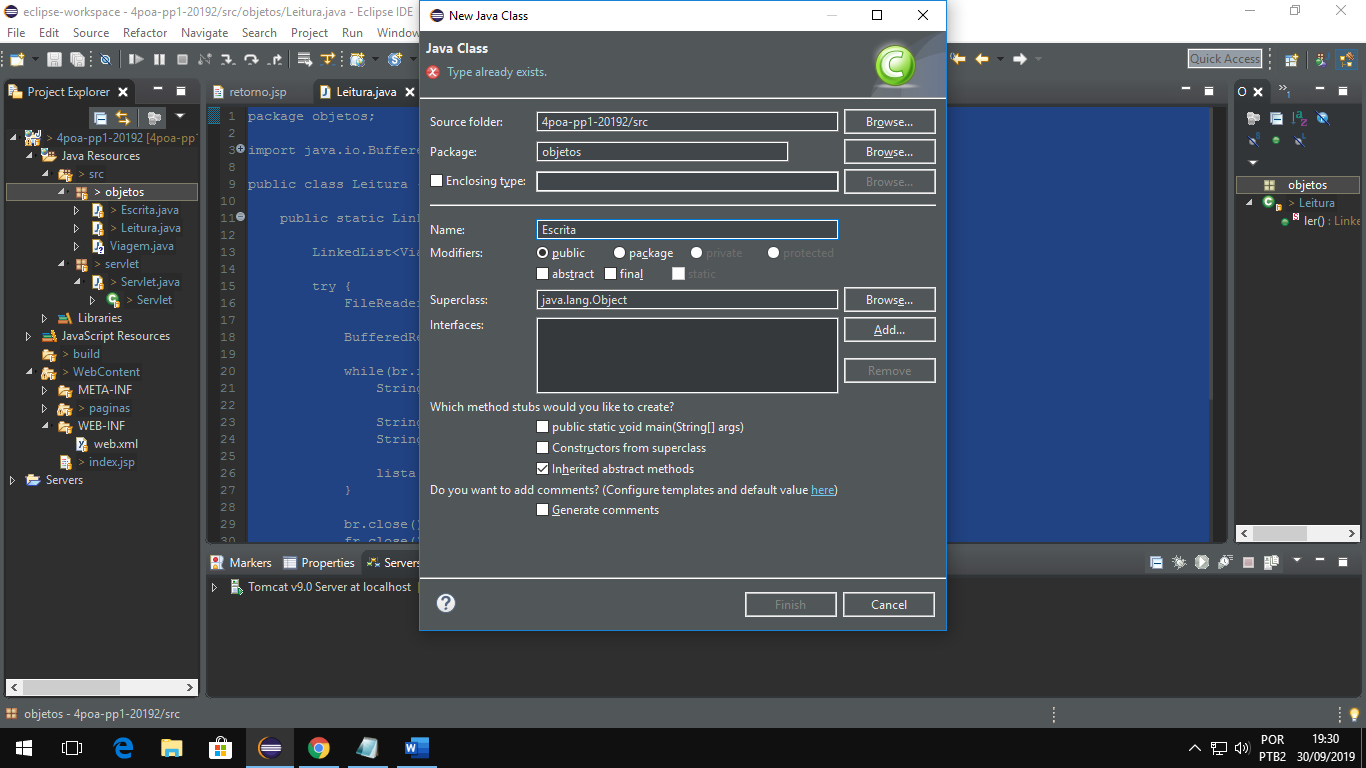
}

}









\src\objetos\ESCRITA.JAVA

package objetos;

//NÃƒO ESQUECER DE IMPORTAR

import java.io.BufferedWriter;

import java.io.File;

import java.io.FileWriter;

import java.io.IOException;

import java.util.Arrays;

public class Escrita {

// PASSO 2.6 - CRIAR O METODO "escrever" PÃšBLICO E ESTATICO, QUE RECEBA O OBJETO ENVIADO PELO SERVLET

static String path = "C:\\4poa\\";

static String arq = "arquivo.txt";

public static void escrever(Carro carro){

// PASSO 2.6.1 - INSTANCIAR NOVO DIRETÃ“RIO/ARQUIVO

File diretorio = new File(path);

// PASSO 2.6.2 - VERIFICAR SE O DIRETORIO JA EXISTE. SE NA, CRIAR!

if (!diretorio.exists()) {

diretorio.mkdir();

}

try {

FileWriter fw = new FileWriter(path + arq, true);

BufferedWriter bw = new BufferedWriter(fw);

// OBS 1: PERCEBA ABAIXO QUE ESCOLHI O ";" PRA SEPARAR OS ATRIBUTOS

//ESTE MESMO CARACTER (;) DEVERÃ� CONSTAR NA CLASSE DE LEITURA PARA O SPLIT

// OBS 2: OPCIONAIS ESCREVERA UM ARRAY DE ATRIBUTOS SEM OS COLCHETES NO ARQUIVO

bw.write(carro.getPlaca() + ";" + carro.getMontadora() + ";" + carro.getModelo() + ";" + carro.getAno() + ";" + Arrays.toString(carro.getOpcionais()).replace("[", "").replace("]", ""));

//import java.util.Arrays;

bw.newLine();

bw.close();

fw.close();

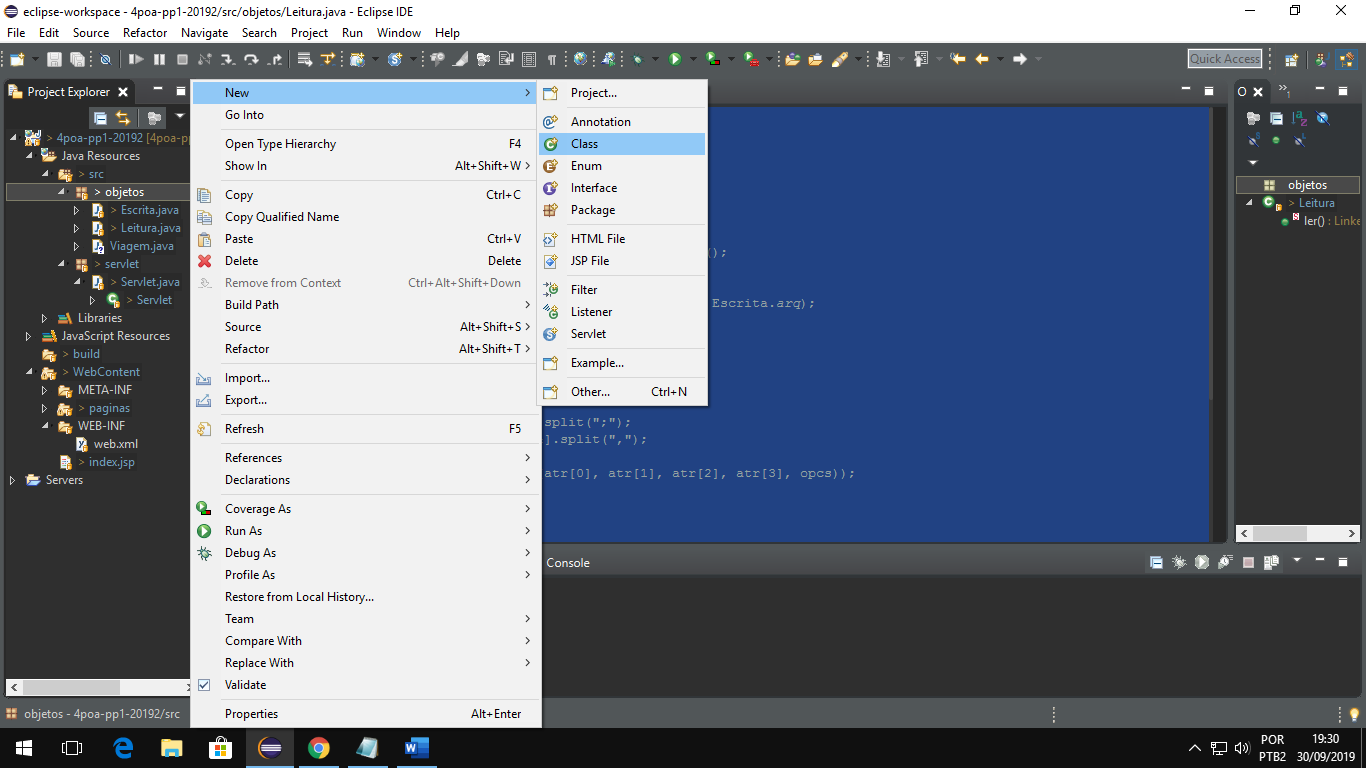
} catch(IOException erro){

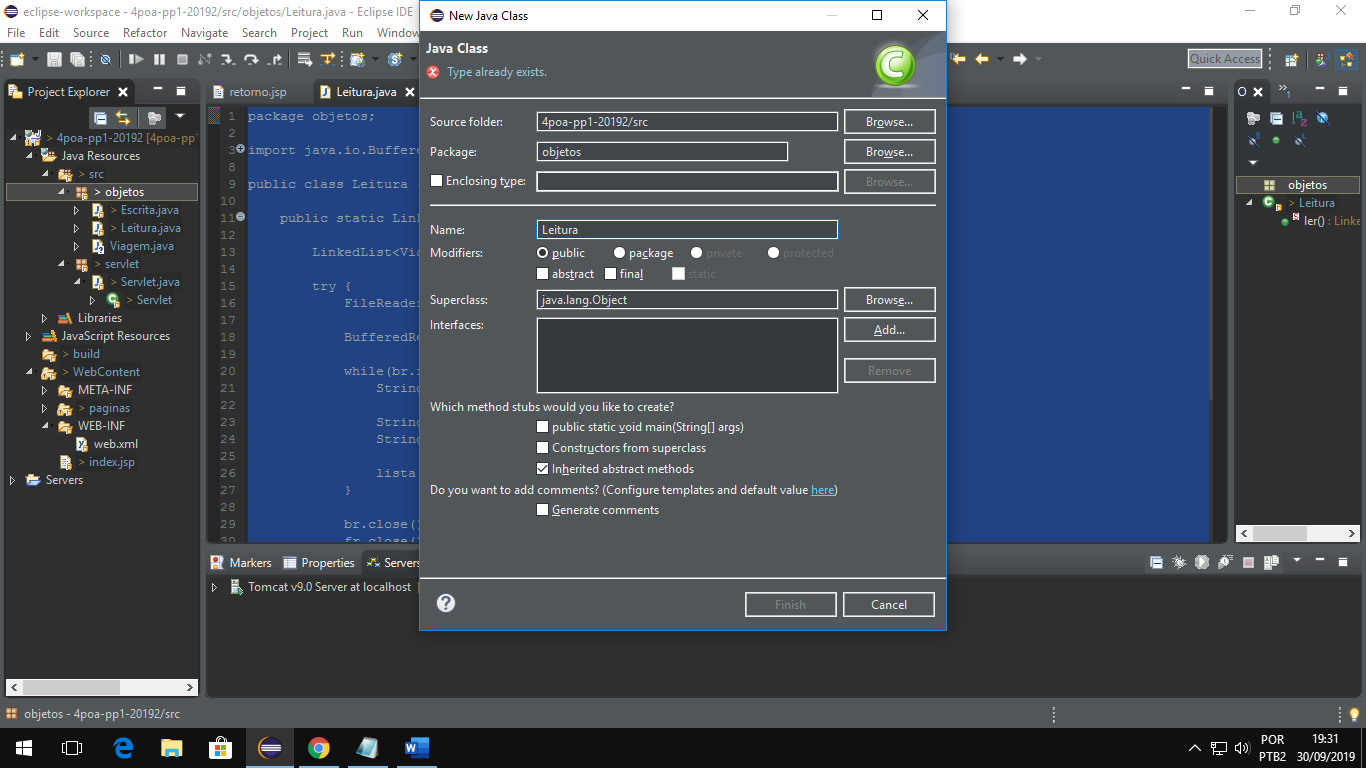
erro.printStackTrace();

}

}

}





\src\objetos\LEITURA.JAVA

package objetos;

import java.io.BufferedReader;

import java.io.FileNotFoundException;

import java.io.FileReader;

import java.io.IOException;

import java.util.LinkedList;

public class Leitura {

// PASSO 2.6.1 - CRIAR ESTE MÉTODO ESTÁTICO RETORNANDO UMA LISTA DE OBJETOS

//IMPORTE "import java.util.LinkedList;"

public static LinkedList<Carro> ler() {

// PASSO 2.6.2 - CRIE A CLASSE "Carro" no pacote "objetos", neste nosso caso

// PASSO 2.6.3 - INSTANCIE A SUA LISTA

LinkedList<Carro> lista = new LinkedList<Carro>();

// PASSO 2.6.4 - BLOCO PADRÃO PARA LEITURA

try {

//MESMO CAMINHO DA ESCRITA

FileReader fr = new FileReader(Escrita.path + Escrita.arq);

//CRIANDO UM BUFFER PARA PREVENIR PERDAS

BufferedReader br = new BufferedReader(fr);

while(br.ready()) {

//LER UMA LINHA DO ARQUIVO

String linha = br.readLine();

//LER OS ATRIBUTOS NO ARQUIVO, SEPARADOS PELO CARACTER DEFINIDO NA HORA DA ESCRITA

String[] atr = linha.split(";");

String[] opcs = atr[4].split(",");

//INSERE O OBJETO IMPORTADO DA LINHA DO ARQUIVO NA SUA LISTA INSTANCIADA NO PASSO 2.6.3

lista.add(new Carro(atr[0], atr[1], atr[2], atr[3], opcs));

}

br.close();

fr.close();

}

catch (FileNotFoundException e) {

e.printStackTrace();

}

catch (IOException e) {

e.printStackTrace();

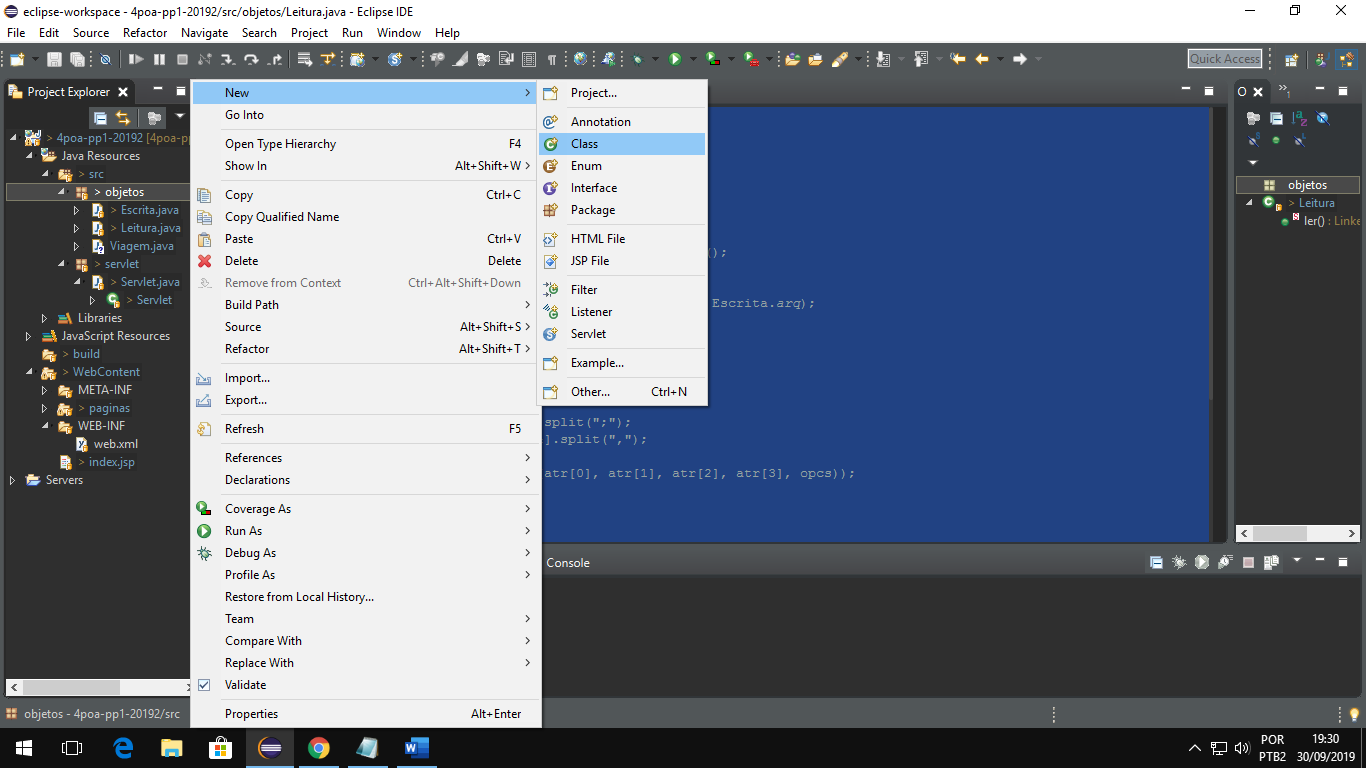
}

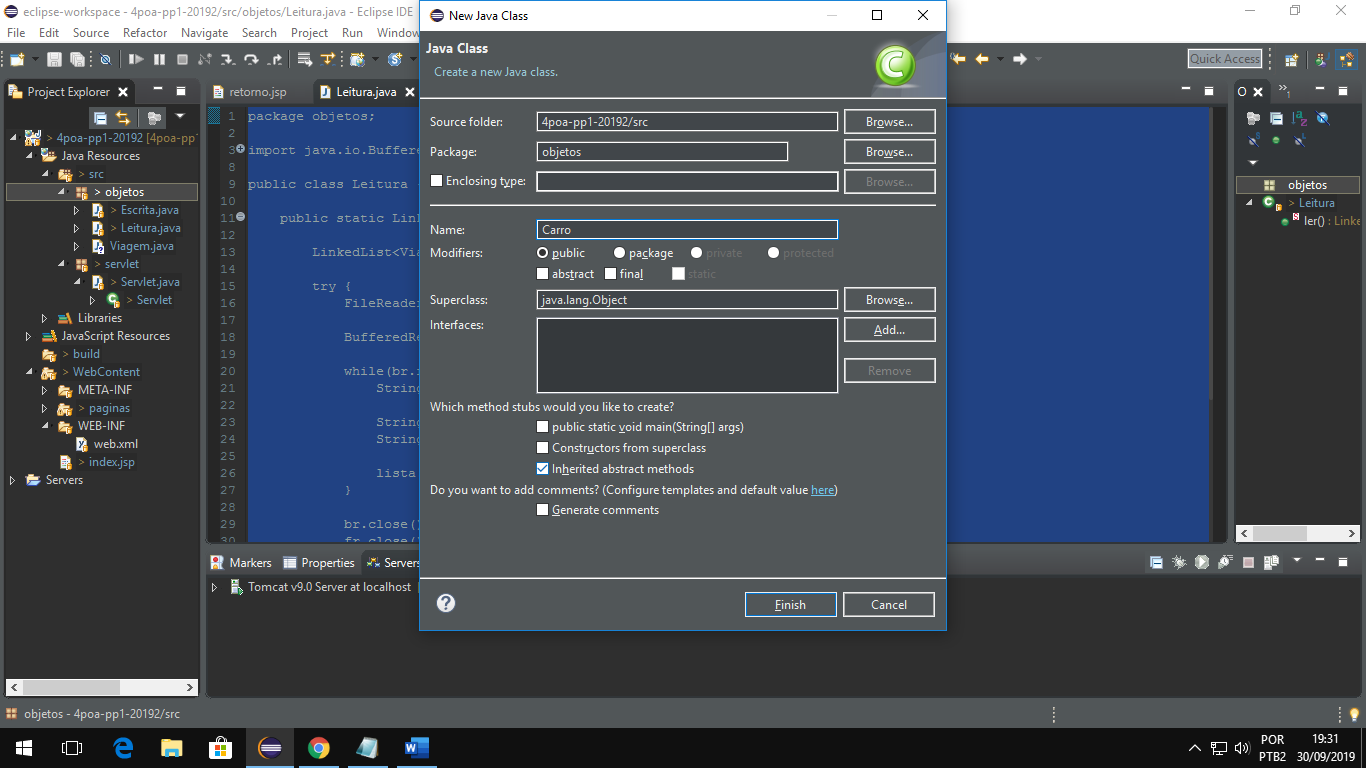
// RETORNA A LISTA CRIADA

return lista;

}

}





\src\objetos\CARRO.JAVA

package objetos;

/\* IMPORTANTE: ESTA CLASSE TERÁ SEUS OBJETOS ORDENADOS,

/\* LOGO É OBRIGATÓRIO IMPLEMENTAR A INTERFACE COMPARABLE\*/

public class Carro implements Comparable<Carro> {

//PASSO 2.4.1 - CRIE OS ATRIBUTOS RETORNADOS NO SERVLET;

private String placa;

private String montadora;

private String modelo;

private String ano;

private String[] opcionais;

//PASSO 2.4.1.1 - IMPLEMENTE SEUS GETTERS AND SETTERS (MENU SOURCE (ALT+SHIFT+S) --> GENERATE GETTERS AND SETTERS)

// PASSO 2.4.2 - CRIE O CONSTRUTOR DA CLASSE, RECEBENDO SEUS PARÂMETROS

public Carro(String pl, String mn, String md, String an, String[] opc){

setPlaca(pl);

setMontadora(mn);

setModelo(md);

setAno(an);

setOpcionais(opc); //PERCEBA QUE ESTE ATRIBUTO É PEGO COMO ARRAY

}

// PASSO 2.4.3 - (OBRIGATÓRIO) IMPLEMENTE O MÉTODO DA INTERFACE COMPARABLE

@Override

public int compareTo(Carro outro) {

/\*NO CASO ABAIXO, ESTOU COMPARANDO POR MONTADORA,

JÁ QUE É POR ISSO QUE ORDENAREI NO ARQUIVO DE RESPOSTA \*/

return montadora.compareTo(outro.montadora);

}

//2.4.1

public String getPlaca() {

return placa;

}

public void setPlaca(String placa) {

this.placa = placa.toUpperCase();

}

public String getMontadora() {

return montadora;

}

public void setMontadora(String montadora) {

this.montadora = montadora.toUpperCase();

}

public String getModelo() {

return modelo;

}

public void setModelo(String modelo) {

this.modelo = modelo.toUpperCase();

}

public String getAno() {

return ano;

}

public void setAno(String ano) {

this.ano = ano.toUpperCase();

}

public String[] getOpcionais() {

return opcionais;

}

public void setOpcionais(String[] opcionais) {

this.opcionais = opcionais;

}

}