PRÀCTICA 2 PTI. INFORME

En aquesta pràctica hem hagut de descarregar un servidor Apache Tomcat i fer-lo servir per gestionar una petita pàgina de gestió de lloguer de cotxes utilitzant servlets. En els fitxers de configuració s'ha utilitzat HTML i Java.

- Entorn i configuració de la pràctica.
L'entorn utilitzat ha estat un servidor Apache Tomcat i un navegador.
Primer hem començat comprovant si Java estava instal·lat. Com que no era aixi ho hem fet nosaltres:

#sudo apt-get update
#javac -version
#sudo apt-get install default-jdk

També hem instalat i iniciat un servidor Apache Tomcat.

#wget https://gitlab.fib.upc.edu/pti/pti/raw/master/p2servlets/apache-tomcat-9.0.5.tar.gz
#tar -xvzf apache-tomcat-9.0.5.tar.gz
#cd apache-tomcat-9.0.5
#./bin/startup.sh &

Després de comprovar que el servidor funcionava amb un senzill "Hello Word" en HTML hem creat el servlet web.xml amb el codi proporcionat al guió.

Finalment, un cop hem tingut l'entorn apunt hem decarregat els fitxers de la botiga de lloguer de cotxes, els hem descomprimit, situat a la carpeta que els corresponia i compilat:

#wget
https://gitlab.fib.upc.edu/pti/pti/raw/master/p2_servlets/carrental.tar.gz
#tar -xvzf carrental.tar.gz

*#mv *.html webapps/my_webapp/*
*#mv *.java webapps/my_webapp/WEB-INF/classes/mypackage*
#mkdir webapps/my_webapp/WEB-INF/lib
#mv json-simple-1.1.1.jar webapps/my_webapp/WEB-INF/lib
#javac -cp lib/servlet-api.jar:webapps/my_webapp/WEB-INF/lib/json-simple-1.1.1.jar webapps/my_webapp/WEB-INF/classes/mypackage/.java*

Ara ja ho tenim tot per començar a implementar el nostre codi.

- Alternatives per resoldre-la (si hi ha diferents opcions).
En tenir fitxers HTML per fer servir de plantilla, el codi del fitxer web.xml complert i un guió força detallat dels passos, la pràctica deixava poques alternatives. Només hem hagut d'indagar una mica més a l'hora de llegir i escriure en un fitxer JSON però la pràctica també proporcionava un enllaç amb la informació necessària:
http://www.mkyong.com/java/json-simple-example-read-and-write-json/

- Explicar la solució triada (es poden incloure fragments de codi rellevants per il·lustrar la solució). Cóm s'ha resolt el que es demana en l'enunciat?

Per realitzar la pràctica hem modificat els arxius CarRentalNew.java i CarRentalList.java.

En el primer, hem començat llegint la informació introduïda a través del document HTML i mostrant-la a l'usuari, així com també la data i el preu final del lloguer.

```
public void doGet(HttpServletRequest req, HttpServletResponse res)
                    throws ServletException, IOException {
  res.setContentType("text/html");
  PrintWriter out = res.getWriter();

  String model = req.getParameter("model_vehicle");
  String subModel = req.getParameter("sub_model_vehicle");
  String dies = req.getParameter("dies_lloguer");
  String numero = req.getParameter("num_vehicles");
  String descompte = req.getParameter("descompte");

  cont ++;
  out.println("<html><big>Hola</big><br>"+
              cont + " Accesos desde su carga.<br><br></html>");

  out.println("Car Model: " + model + "<br>");
  out.println("Engine: " + subModel + "<br>");
  out.println("Number of days: " + dies + "<br>");
  out.println("Number of units: " + numero + "<br>");
  out.println("Discount: " + descompte + "<br>");

  String preu = "Una millonada";
  out.println("Preu: " + preu + "<br>");

  LocalDateTime l = LocalDateTime.now();
  int dia = l.getDayOfMonth();
  int mes = l.getMonthValue();
  int any = l.getYear();
  out.println("Data: " + dia + "/" + mes + "/" + any + "<br>");

  JSONParser parser = new JSONParser();
```

En la segona part del document hem guardat tota aquesta informació dins d'un document JSON anomenat *BD.json*. Es comprova si existeix el document, es llegeix, s'afegeix la informació nova i es torna a guardar. En cas que no existeixi, el crea. Per realitzar aquesta part hem hagut d'afegir paquets nous de json.

```
JSONParser parser = new JSONParser();
JSONArray jsonLlogers = new JSONArray();

try {

    File bd = new File("./BD.json");

    if(!bd.exists()) {
        bd.createNewFile();
    }
    else if(bd.exists() && !bd.isDirectory()) {
        Object llogers = parser.parse(new FileReader(bd));
        jsonLlogers =  (JSONArray) llogers;
    }

    JSONObject nouLlogger = new JSONObject();
    nouLlogger.put("model", model);
    nouLlogger.put("subModel", subModel);
    nouLlogger.put("dies", dies);
    nouLlogger.put("numero", numero);
    nouLlogger.put("descompte", descompte);
    nouLlogger.put("preu", preu);

    jsonLlogers.add(nouLlogger);
}

catch (FileNotFoundException e) {
    out.println("No hi ha cap lloguer fet.");
}
catch (IOException e) {
    out.println("No s'han pogut carregar els lloguers.");
}
catch (ParseException e) {
    out.println("No s'han pogut carregar els lloguers.");
}

try (FileWriter file = new FileWriter("./BD.json")) {

    file.write(jsonLlogers.toJSONString());
    file.flush();


} catch (IOException e) {
    e.printStackTrace();
}
```

En el segon fitxer .java haviem de mostrar la informació guardada, només a l'usuari que disposa de l'ususari i la contrasenya correctes. Es comproven aquests dos camps, si coincideixen amb "admin" i "admin", llegeix del fixter *BD.json* i ho mostra per pantalla.

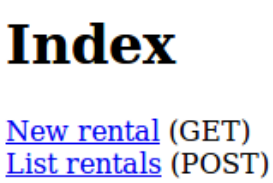
```
public void doGet(HttpServletRequest req, HttpServletResponse res)
                throws ServletException, IOException {
  res.setContentType("text/html");
  PrintWriter out = res.getWriter();
  String nombre = req.getParameter("userid");
  String contrasenya = req.getParameter("password");

  if (!nombre.equals("admin") && !contrasenya.equals("admin")) {
      out.println("<html><big> Hola! <br> No tienes acceso a este sitio :)</big></html>");
  }
  else {
      cont ++;
      out.println("<html><big>Hola "+ nombre + "!</big><br>"+
                  cont + " Acceso(s) desde su carga. <br> <br></html>");
```

(en aquesta primera part del codi mirem que l'usuari i la contrasenya siguin correctes)

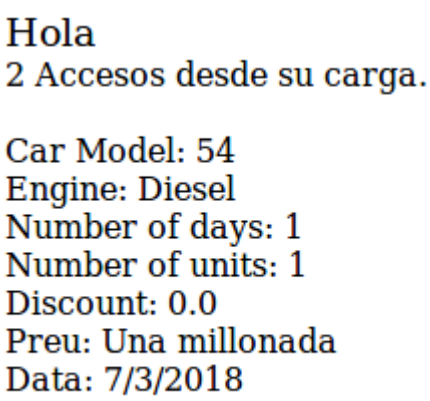
```
      JSONParser parser = new JSONParser();

      try {

          Object bd = parser.parse(new FileReader("./BD.json"));

          JSONArray list = (JSONArray) bd;
          Iterator<JSONObject> iterator = list.iterator();
          int i = 1;
          while (iterator.hasNext()) {
              JSONObject item = (JSONObject) iterator.next();

              out.println("Lloguer número " + i + "<br>");

              String model = (String) item.get("model");
              out.println("Car Model: " + model + "<br>");

              String subModel = (String) item.get("subModel");
              out.println("Engine: " + subModel + "<br>");

              String dies = (String) item.get("dies");
              out.println("Number of days: " + dies + "<br>");

              String numero = (String) item.get("numero");
              out.println("Number of units: " + numero + "<br>");

              String descompte = (String) item.get("descompte");
              out.println("Discount: " + descompte + "<br>");

              String preu = (String) item.get("preu");
              out.println("Price: " + preu + "<br> <br>");

              ++i;
          }
      }

      catch (FileNotFoundException e) {
          out.println("No hi ha cap lloguer fet.");
      }
      catch (IOException e) {
          out.println("No s'han pogut carregar els lloguers.");
      }
      catch (ParseException e) {
          out.println("No s'han pogut carregar els lloguers.");
      }
  }
}
```

(Aquí imprimim la informació guardada en el fitxer JSON)

- Avaluació de la solució. Explicar les proves realitzades per comprovar el funcionament de la solució i indicar el que s'ha observat (p.ex. amb captures de pantalla).
Per comprovar la solució hem accedit al link:
http://localhost:8080/my_webapp/carrental_home.html



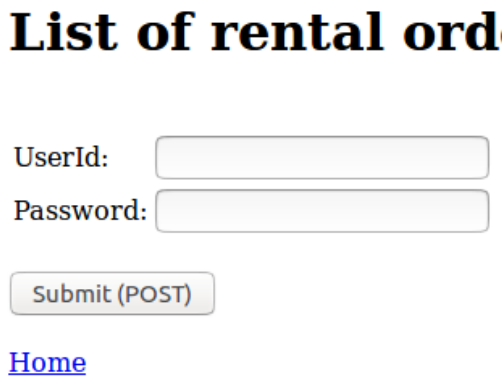
Primer hem provat que funcionés l'apartat de fer un nou lloguer (*carrental_form_new.html*) i que en fer "Submit" es mostrés una llista amb el lloguer demanat, incloent el preu i la data, i es guardés bé en l'arxiu *BD.json*.



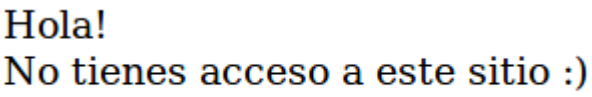

```
[{"numero":"1","subModel":"Diesel","preu":"Una
millonada","model":"54","dies":"1","descompte":"0.0"},
{"numero":"5","subModel":"Gasolina","preu":"Una
millonada","model":"139","dies":"2","descompte":"0.0"},
{"numero":"1","subModel":"Diesel","preu":"Una
millonada","model":"54","dies":"1","descompte":"0.0"},{"numero":"1","subModel":"Diesel","preu":"Una
millonada","model":"54","dies":"1","descompte":"0.0"}]
```


(Extracte del fitxer *BD.json* on es pot veure diverses reserves guardades)

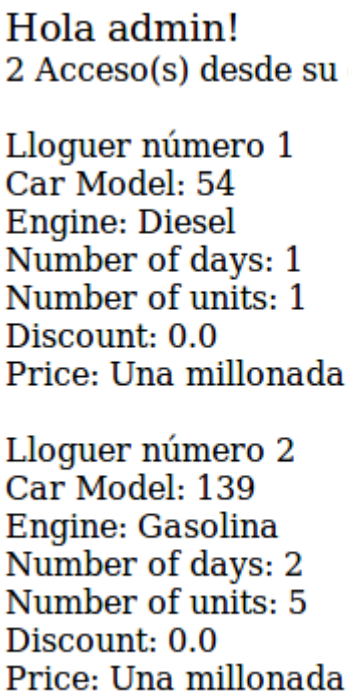
La segona part de la pràctica consisteix en que l'administrador de la pàgina pugui veure el llistat de lloguers realitzats.



Si l'usuari i la contrasenya no són correctes, la pàgina ens mostrarà el missatge següent:



En canvi, si les credencials són correctes podràs accedir i veure les reserves de la manera següent:



- Comentar/argumentar aspectes positius i negatius de la solució/tecnologia (p.ex. limitacions, flexibilitat, camps d'aplicació, comparar la solució/tecnologia amb altres tecnologies, etc.).

Aquest és un ús molt simple d'un servidor Apache i va molt bé per introduir-ho a qui no l'hagi fet servir mai. Tanmateix la mateixa solució es podria implementar amb un altre tipus de servidor web (per exemple NGINX, Cherokee...). També, en lloc de Java es podrien utilitzar altres llenguatges com Javascript o PHP i les

dades es podrien enviar amb XML, per exemple. Creiem que és molt insegur passar informació a través de l'URL perquè és susceptible de ser interceptada per usuaris que no ens interessen. Tanmateix es tracta d'una pràctica simple i va molt bé per entendre els conceptes.