```
import java.io.IOException;
import java.sql.Time;
import java.sql.Timestamp;
import java.text.SimpleDateFormat;
import java.util.Calendar;
import java.util.Date;

import javax.servlet.http.Cookie;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;
import javax.swing.JOptionPane;

import org.apache.catalina.util.MD5Encoder;

import sun.security.provider.MD5;

/*////////////////////////////////////////////////////
  //                                                //
  //                    ALQUILER                    //
  //                                                //
  ////////////////////////////////////////////////////*/

public class LCookie {

    //devuelve el array con TODAS las cookies, es necesario para buscar una cook
ie en particular
    public static Cookie[] dameArrayCookies(HttpServletRequest request){
        return request.getCookies();
    }


    //devuelve la cookie del array de cookies si existe si no devuelve null
    private static String dameCookieDelArray(Cookie[] arrayCookies,String mCooki
eBuscar  ){
        for(int i=0;i<arrayCookies.length;i++) {
            Cookie lCookie=arrayCookies[i];
////++//  Cadenas.alert("Cookie["+i+"]="+lCookie.getName());
            if(lCookie.getName().equals(mCookieBuscar) ) {
////++//  Cadenas.alert(mCookieBuscar+"\n cookie["+lCookie.getName()+"] = "+lCoo
kie.getValue());
////++//  Cadenas.alert( " cookie +:"+lCookie.getValue());
                return lCookie.getValue();
            }
        }
        return null;
    }

    //devuelve la cookie si existe si no devuelve null
    public static String dameCookie(HttpServletRequest request,String mCookieBus
car  ){
        Cookie[] arrayCookies=LCookie.dameArrayCookies(request);
        if (arrayCookies!=null){
////++//  Cadenas.alert("dameCookies ( "+mCookieBuscar+" )");
            String mCookie =dameCookieDelArray(arrayCookies,mCookieBuscar);
            if ( mCookie!=null)
                return mCookie;
         }
////++//  Cadenas.alert("Sin array de Cookies");
         return null;
```

```
    }

    public static void borraCookie(HttpServletResponse response, String txtClave
Cookie,String txtValorCookie){
        Cookie miCookie=new Cookie(txtClaveCookie,txtValorCookie);
        //miCookie.setMaxAge(0);
        //response.addCookie(miCookie);
        miCookie.setMaxAge(-1);
        response.addCookie(miCookie);

    }

    //Elimina la cookie del ticket
    public static void eliminaCookieTicket(clsTicket mTicekt,HttpServletResponse
 response){
        // TDOD
        // TDOD
        // TDOD
        String txtCookie="0~0~0~0~0";
        borraCookie(response,"ticketBC"+mTicekt.getUsuario(),txtCookie);
        GestionLogs.escribirLog(null, "Cookie eliminada : "+mTicekt.toString() )
;
    }


    //GRABA una cookie a traves del Objeto de respuesta
    public static void creaCookie(HttpServletResponse response, String txtClaveC
ookie,String txtValorCookie){
        Cookie miCookie=new Cookie(txtClaveCookie,txtValorCookie);
        miCookie.setMaxAge( 60*60*24*10 );
        response.addCookie(miCookie);

    }

    //GRABA una cookie con la informacion del Ticket
    public static void creaCookieTicket(clsTicket mTicekt,HttpServletResponse re
sponse){
        // TDOD
        // TDOD
        // TDOD
        String txtCookie=mTicekt.getidTicket().hashCode()+"~"+
                         mTicekt.getCuando().toString()+"~"+
                         mTicekt.getPuesto()+"~"+
                         mTicekt.getnumBici()+"~"+
                         mTicekt.getUsuario();
        creaCookie(response,"ticketBC"+mTicekt.getUsuario(),txtCookie);
        GestionLogs.escribirLog(null, "Cookie Ticket Creada : "+mTicekt.toString
() );
    }

    //crea y devuelve un ticket desde la cookie guardada si existe y es correcta
 si no devuelve null
    public static clsTicket creaTicketDesdeCookie(HttpSession sesion,HttpServlet
Request request){
        String usuario=(String) sesion.getAttribute("usuarioBC");
        String mCookieBuscar="ticketBC"+usuario;
        String mCookie=dameCookie(request, mCookieBuscar);
////++//  Cadenas.alert("cookie en crea :"+ mCookie);
        //si hay cookies
        if (mCookie!=null){
```

```
                String[] datos=mCookie.split("~");
                // si hay 5 datos
                //      0=idTicket (idSesion del alquiler);
                //      1=fecha de recogida
                //      2=numero de puesto
                //      3=numero de bici
                //      4=usuario
////++//  Cadenas.alert("creaTicketDesdeCookie 1 \n cookie.lenght :"+ datos.leng
th);

                if (datos.length==5){
                    if (datos[0].equals("0") || datos[1].equals("0") || datos[2]
.equals("0") || datos[4].equals("0") ){
                        // es un ticket borrado, esta aqui por si despues de dev
olver la bici el navegador no borra
                        //     el ticket, lo he puesto con todos sus datos a "0"
, asi se que ese ticket no es valido
                        //     aunque exista
                        return null;
                    }else{

                        // Â¿el ticket pertenece al usuario que se ha validado?
////++//  Cadenas.alert("creaTicketDesdeCookie 2 \n usuario :"+ usuario +"\n"+ "
en cookie:"+datos[4]);
                        if (! usuario.equalsIgnoreCase(datos[4])){
                            String txtError="ERROR : USUARIO DE COOKIE("+datos[4
]+") != USUARIO VALIDADO  ("+usuario+"), se usa usuario validado )";
                            txtError=txtError+"\n idticket="+datos[0]+"/cuando="
+datos[1]+"/puesto="+datos[2]+"/numbici="+datos[3];
                            GestionLogs.escribirLog(sesion, txtError);
                        }
                        try {
////++//  Cadenas.alert("creaTicketDesdeCookie 3 \n fecha Cookie antes de :"+ da
tos[1]);
                            Timestamp fechaCuandoCookie=Timestamp.valueOf(datos[
1]);
////++//  Cadenas.alert("creaTicketDesdeCookie 4 \n fecha Cookie despues de :"+
fechaCuandoCookie.toString());
                        } catch (IllegalArgumentException e) {
                            //la fecha guardada no es correcta
                           java.util.Date today = new java.util.Date();
                           Timestamp ahora= new java.sql.Timestamp(today.getTime
());

                           Calendar hoy=Calendar.getInstance(); // el del objeto
 actual (.getyear() esta deprecated)
                           hoy.setTime(ahora);
                           hoy.add(Calendar.HOUR_OF_DAY,-1);

                           //String res=String.valueOf(hoy.get(Calendar.YEAR))+"
-"+String.valueOf(hoy.get(Calendar.MONTH))+"-"+String.valueOf(hoy.get(Calendar.D
AY_OF_MONTH))+".txt";
                        }

                        //                      datos[0],datos[1], datos[2], da
tos[3], datos[4]);
                        //mTicket =new clsTicket(idTicket,  cuando,   puesto,  n
umBici, usuario )
//++//  Cadenas.alert("creaTicketDesdeCookie 5 \n isTicket:"+ datos[0]+" \n cuan
do: "+datos[1]+" \npuesto: "+ datos[2]+" \n numbici: "+ datos[3]+" \n usuario: "
```

```
+ usuario);

                        clsTicket ticket= new clsTicket( datos[0],datos[1], dato
s[2], datos[3], usuario);
//++//  Cadenas.alert(" [ Ticket desde Cookie ]  usuario:"+ ticket.toString() +
"\n "
                        GestionLogs.escribirLog(null, "Ticket Creado desde BD :
"+ticket.toString() );
                        return ticket;
                    } // del else de if (ticketCookie.getidTicket().equals("0")
|| ti ....
                }// del if (datos.length==5){
        }
        return null;
    }

/*
    //crea y devuelve un ticket desde la cookie guardada si existe si no devuelv
e null
    @Deprecated
    public static clsTicket creaTicketDesdeCookieConHash(HttpSession sesion,Http
ServletRequest request){
        Cookie[] arrayCookies=LCookie.dameArrayCookies(request);
        String mCookieBuscar="ticketBC"+ login;
        String mCookie="";
        //si hay cookies
        if (arrayCookies!=null){
            mCookie=dameCookie(request, mCookieBuscar);
            if (mCookie!=""){
                String[] datos=mCookie.split("~");
                // si hay 5 datos
                if (datos.length==5){
                    //obtenemos el hashcode para comprobar si existe y es igual
al guardado
                    String hashCookie=String.valueOf(mCookie.hashCode());
                    String mhashCookie=dameCookie(arrayCookies,hashCookie );
                    // si los hash code son iguales creamos el ticket
                    if (mhashCookie.equals(hashCookie)){
                        //mTicket =new clsTicket(idTicket,  cuando,   puesto,  n
umBici, usuario)
                        return new clsTicket(    datos[0],datos[1], datos[2], da
tos[3], datos[3]);
                    }
                }
            }
        }
        return null;
    }
*/

    //GRABA una cookie con la informacion del Ticket
    @Deprecated
    public static void creaCookieTicketConHash(clsTicket mTicket,HttpServletResp
onse response){
        // TDOD
        // TDOD
        // TDOD
        String txtCookie=mTicket.getidTicket()+"~"+
                         mTicket.getCuando().toString()+"~"+
                         mTicket.getPuesto()+"~"+
```

```
                        mTicket.getnumBici()+"~"+
                        mTicket.getUsuario();
        String hashCookie=String.valueOf(txtCookie.hashCode());

        creaCookie(response,"ticketBC"+ mTicket.getUsuario(),txtCookie);
        //se guarda un hascode de la cookie guardada para comprobar que no han e
xistido cambios una vez se guardo.
        creaCookie(response,hashCookie,hashCookie);

    }


}
```