

```

import java.sql.Timestamp;
import java.util.Calendar;

public class clsTicket {
    // private int numero=0; <=== NO RECUERDO QUE CONTROLABA lo he cambiado por
    idSesion
    //
    // YA ME ACUERDO ERA El nÂ° Orden cuando esto e
    ra para colas
    private String idTicket=""; // he cambiado idSesion por idTicket ya podria
    darse el caso que un usuario
    // - iniciara sesion adquiriera y un ticket
    ( con una idSesion )
    // - terminara esa sesion y despues establec
    iera otra(esta nueva sesion
    // no tendria ticket pero si que lo te
    ndria como una cookie), cuando
    // fuera a dejar la bici el idSesion s
    eria distinto, y eso no puede
    // ser a la hora de de grabar la inf d
    e dejar la bici ya que debe ser
    // el mismo ( el de coger y dejar )

    private Timestamp cuando;
    private String puesto="";
    private String numBici="";
    private String usuario="";

    /*
    public clsTicket(String idSesion, Timestamp cuando){
        this.setidTicket(Integer.parseInt(idSesion));
        this.setCuando(cuando);
    }
    */

    /*
    public clsTicket(int idSesion, Timestamp cuando){
        this.setidTicket(idSesion);
        this.setCuando(cuando);
    }
    */

    public clsTicket(String idSesion, Timestamp cuando,String puesto,String numB
    ici,String usuario){
        this.setidTicket(idSesion);
        this.setCuando(cuando);
        this.setPuesto(puesto);
        this.setnumBici(numBici);
        this.setUsuario(usuario);
    }

    public clsTicket(String idSesion, String cuando,String puesto,String numBici
    ,String usuario){
        this.setidTicket(idSesion);
        this.setCuando(Timestamp.valueOf(cuando));
        this.setPuesto(puesto);
        this.setnumBici(numBici);
        this.setUsuario(usuario);
    }
}
/*

```

```

    public clsTicket(int idSesion, Timestamp cuando,String puesto,String numBici
,String usuario){
        this.setidTicket(idSesion);
        this.setCuando(cuando);
        this.setPuesto(puesto);
        this.setnumBici(numBici);
        this.setUsuario(usuario);
    }
*/

/*
    private void setidTicket(String numSiguiete) {
        this.idSesion = Integer.parseInt(numSiguiete);
    }
*/

    private void setUsuario(String usuario) {
        this.usuario = usuario;
    }

    private void setidTicket(String idSesion) {
        this.idTicket=idSesion;
    }

    private void setCuando(Timestamp cuando) {
        this.cuando = cuando;
    }

    private void setPuesto(String puesto) {
        this.puesto = puesto;
    }

    private void setnumBici(String numBici) {
        this.numBici = numBici;
    }

    public String getidTicket() {
        return idTicket;
    }

    public Timestamp getCuando() {
        return cuando;
    }

    public String getPuesto() {
        return puesto;
    }

    public String getnumBici() {
        return numBici;
    }

    public String getUsuario() {
        return usuario;
    }

    @Override
    public String toString(){
        return " [ Ticket ] \n " +

```

```

        " usuario: "+ this.getUsuario() + "\n " +
        " numbici: "+ this.getnumBici() + "\n" +
        " puesto: "+ this.getPuesto() + "\n" +
        " cuando: "+ this.getCuando().toString()+ "\n" +
        " idT: "+ this.getIdTicket()+"\n";
    }
    @Override
    public boolean equals(Object Obj){
        //compara si los valores de dos Ticket son iguales
        // compara tambien si el hash de uno es igual al idT del otro y viceversa
a
        // asi se pueden comparar ticket y cookieTicket( que guarda el hash de
        ticket.getIdTicket)
        // y da igual cual es cual simepre funciona.

        clsTicket T2=(clsTicket) Obj;
        boolean dev=( T2.getnumBici().equals(this.numBici)
            && T2.getPuesto().equals(this.puesto)
            && T2.getCuando().equals(this.cuando)
            && ( T2.getIdTicket().equals(this.getIdTicket) ||
                T2.mismoHash(this.getIdTicket) ||
                this.mismoHash(T2.getIdTicket()) )
        );

        return dev;
    }

    // public boolean mismoHash(int HashIdTicket){
    //     //compara si los valores de dos Ticket son iguales
    //     return ( HashIdTicket==Integer.parseInt(this.getIdTicket) ) ;
    // }

    public boolean mismoHash(String HashIdTicket){
        //compara si los valores de dos Ticket son iguales
        try{
            return ( this.getIdTicket.hashCode()==Integer.parseInt(HashIdTicket) )
;
        }catch (NumberFormatException e) {
            //
        }
        return false;
    }

    public int compareToDate(Timestamp fecha){
        //compara la fecha del ticket con una externa
        //sigue la regla de compareTo
        // si fecha>dato objeto : ret <0
        // si fecha<dato objeto : ret >0
        // si fecha=dato objeto : ret 0
        Calendar propio=Calendar.getInstance(); // el del objeto actual (.get
year() esta deprecated)
        Calendar externo=Calendar.getInstance();

        propio.setTime(cuando);
        externo.setTime(fecha);
        return cuando.compareTo(fecha);
    }

```

```
        //return ( (propio.get(Calendar.DAY_OF_YEAR)-externo.get(Calendar.DAY
_OF_YEAR))- anys );
    }
}
```

```
@Deprecated
private Timestamp obtenerFechaSQL() {
    java.util.Date today = new java.util.Date();
    return new java.sql.Timestamp(today.getTime());
}
```

```
@Deprecated
private int convInteger(String valor) {
    try {
        return Integer.parseInt(valor);
    } catch (NumberFormatException e) {
        // no es un n mero o falla algo
        return 9997;
    }
}
```

```
}
```