```
    1: package packvolcado;
    2:
    3: public class Pral {
    4:
    5:
    6:         public static void main(String[] args) {
    7:
    8:
    9:                 Conexion con = new Conexion();  //COnexiÃ³n es la clase principal !!!
   10:                 con.leerconf(con.config);               //Cargo la configuracion del archivo conf_volcado.txt
   11:
   12:                 while (true){
   13:                         con.conectar();                                 //Conecto con las BBDD
   14:                         con.conectal();
   15:
   16:                         con.tarea();                                    //Compruebo si hay tareas pendientes !
   17:                         //con.programa();
   18:
   19:
   20:                         try {
   21:                                 Thread.sleep(con.config.getTiempo()*1000);      //Aki el proceso duerme lo que diga el
archivo de configuraciÃ³n
   22:                         } catch (InterruptedException e) {
   23:                                 // TODO Auto-generated catch block
   24:                                 e.printStackTrace();
   25:                         }
   26:                         //Cierro conexion con BBDD
   27:                         //con.cierrar();
   28:                         con.cierral();
   29:
   30:                         if (con.connr == null) {
   31:                                 con.conectado=false;
   32:                         }
   33:
   34:                         //Invoco al garbage collector para liberar memoria !!
   35:                         Runtime r = Runtime.getRuntime();
   36:                         r.gc();
   37:                 }
   38:
   39:         }
   40:
   41: }
```

```
    1: package packvolcado;
    2:
    3: import java.io.BufferedReader;
    4: import java.io.FileNotFoundException;
    5: import java.io.FileReader;
    6: import java.io.IOException;
    7: import java.sql.Connection;
    8: import java.sql.DriverManager;
    9: import java.sql.PreparedStatement;
   10: import java.sql.ResultSet;
   11: import java.sql.SQLException;
   12:
   13:
   14: public class Conexion {
   15:
   16:         //ParÃ¡metros de conexiÃ³n y conexiones
   17:          Connection connr = null;
   18:          Connection connl = null;
   19:          Conf config = new Conf();
   20:          Tarea tarea = new Tarea();
   21:          Valvula valv = new Valvula();
   22:          Programacion prog = new Programacion();
   23:          public boolean conectado = false;
   24:          private final String rutaconf = "/home/mikel/conf_volcado.txt"; //Ruta al archivo de configuraciÃ³n
   25:
   26:
   27:
   28:         ////////////////////////////////////////////////////////////////////////
   29:         //
   30:         //Comprueba TAREAS pendientes y si las hay las copia a la local y las borra de la pasarela
   31:         protected void tarea(){
   32:
   33:                 PreparedStatement sentenciapre, sentenciaprel;
   34:                 ResultSet resultadosr, resultadosl;
   35:
   36:                 try {
   37:                                 //Consulto tareas pendientes de ejecutar en remoto
   38:                                 sentenciapre = connr.prepareStatement("select * from v02_tareaexec where CODPROG="+conf
ig.getIdrasp());
   39:                                 resultadosr = sentenciapre.executeQuery();
   40:
   41:                                 while (resultadosr.next()) {
   42:
   43:                                 //Cojo sÃ³lo Ã©sto de la tarea para evitar campos nulls y controlar si es tarea manual
o programaciÃ³n.
   44:                                 tarea.setIddstarea(resultadosr.getInt(4));
   45:                                 tarea.setIdtareaexec(resultadosr.getInt(1));
   46:
   47:
```

```
   48:                                  //Si la tarea es leer una programaciÃ³n (3) llamo a programa para que la gestione.
   49:                                  if(tarea.getIddstarea()==3)
   50:                                          programa(tarea);
   51:                                  //Si no es una tarea manual
   52:                                  if ((tarea.getIddstarea())==1 || (tarea.getIddstarea())==2){
   53:
   54:                                          //aki el resto de los campos de tarea
   55:                                                  tarea.setCodprog(resultadosr.getString(2));
   56:                                                  tarea.setCodelecvalv(resultadosr.getString(3));
   57:                                                  tarea.setFcexec(resultadosr.getTimestamp(5));
   58:                                                  tarea.setValor(resultadosr.getInt(6));
   59:                                                  tarea.setCodtarea(resultadosr.getString(7));
   60:                                                  tarea.setDstarea(resultadosr.getString(8));
   61:
   62:                                                  //Compruebo que la tarea no existe en local !!
   63:                                                  //Cojo la Ãºltima tarea de la tabla local
   64:                                                  sentenciaprel = connl.prepareStatement("SELECT MAX(idtarea),fcexec FROM
 tarea");
   65:                                                  resultadosl = sentenciaprel.executeQuery();
   66:                                                  while (resultadosl.next()) {
   67:                                                  tarea.setIdtarealocal(resultadosl.getInt(1));
   68:                                                  tarea.setFcloc(resultadosl.getTimestamp(2));
   69:
   70:                                                  //Si la tarealocal es menor es que la tareaexec no estÃ¡ en la bbdd loc
al, con lo que la inserto
   71:                                                  //Comparao las fechas tambiÃ©n por si la pasarela se ha reiniciado
   72:                                                  if (tarea.getIdtarealocal()<tarea.getIdtareaexec() || tarea.getFcloc().
before(tarea.getFcexec()))
   73:                                                  {
   74:                                                          if(Insertarea(sentenciaprel))
   75:                                                          {
   76:                                                                  //Aki tendrÃ-a que borrar de la t02_
   77:                                                                  if (Borratarea(tarea));
   78:                                                                          System.out.println("Tarea borrada de la pasarel
a correctamente !");
   79:                                                          }
   80:                                                          else{
   81:                                                                  System.out.println("Error al insertar la tarea de la bb
dd remota a la local !!");
   82:                                                          }
   83:                                                  }
   84:
   85:                                           }
   86:
   87:                                          }
   88:                                  }
   89:
   90:
   91:          } catch (SQLException e) {
```

```
   92:                 //
   93:                 e.printStackTrace();
   94:                 System.out.println("Error 93");
   95:         }
   96:
   97:
   98:
   99: }
  100:         protected boolean Borratarea(Tarea tar){
  101:                 PreparedStatement sentenciapre;
  102:                 System.out.println("Borro tarea en la t02 con ID " + tar.getIdtareaexec());
  103:                 try {
  104:                         sentenciapre = connr.prepareStatement("DELETE FROM t02_tareaexec WHERE idtareaexec="+tar.getIdt
areaexec());
  105:                         sentenciapre.executeUpdate();
  106:                         sentenciapre.close();
  107:                         return true;
  108:                 } catch (SQLException e) {
  109:                         e.printStackTrace();
  110:                         System.out.println("No se ha podido borrar la tarea en la BBDD remota.");
  111:                         return false;
  112:                 }
  113:         }
  114:         protected boolean Insertarea(PreparedStatement sentenciaprel){
  115:
  116:                 System.out.println("Inserto tarea "+tarea.getIdtareaexec()+" en la local");
  117:                 try {
  118:                         sentenciaprel = connl.prepareStatement("INSERT INTO tarea (idtarea, codprog, codelecvalv, iddst
area, " +
  119:                                                                                                                 "fcexec, valor, codt
area,dstarea) " +
  120:                                                                                                                 "VALUES (’"+tarea.ge
tIdtareaexec()  +
  121:                                                                                                                      "’,’"+tarea.get
Codprog()+
  122:                                                                                                                      "’,’"+tarea.get
Codelecvalv()+
  123:                                                                                                                      "’,’"+tarea.get
Iddstarea()+
  124:                                                                                                                      "’,’"+tarea.get
Fcexec()+
  125:                                                                                                                      "’,’"+tarea.get
Valor()+
  126:                                                                                                                      "’,’"+tarea.get
Codtarea()+
  127:                                                                                                                      "’,’"+tarea.get
Dstarea()+"’ )") ;
  128:                         sentenciaprel.executeUpdate();
  129:                         return true;
```

```
  130:                 } catch (SQLException e) {
  131:                         //
  132:                         e.printStackTrace();
  133:                         System.out.println("No se ha podido actualizar la tarea en la BBDD local.");
  134:                         return false;
  135:                 }
  136:
  137:
  138:
  139:         }
  140:
  141:
  142:         ////////////////////////////////////////////////////////////////////////
  143:         //
  144:         //Comprueba PROGRAMACIONES pendientes y si las hay las copia a la local y borra la tarea y demases de la pasere
la
  145:         protected  void programa(Tarea tar){
  146:                 PreparedStatement sentenciapre, sentenciaprel;
  147:                 ResultSet resultadosr, resultadosl;
  148:
  149:                 try {
  150:                         //Consulto programaciones pendientes de ejecutar en remoto
  151:                         sentenciapre = connr.prepareStatement("select * from v10_programaexec where CODPROG="+config.ge
tIdrasp());
  152:                         resultadosr = sentenciapre.executeQuery();
  153:
  154:
  155:                         while (resultadosr.next()) {
  156:
  157:                                 prog.setIdprograma(resultadosr.getInt(1));
  158:                                 prog.setCodprograma(resultadosr.getString(2));
  159:                                 prog.setDsprograma(resultadosr.getString(3));
  160:                                 prog.setFcinicio(resultadosr.getDate(4));
  161:                                 prog.setFcFin(resultadosr.getDate(5));
  162:                                 prog.setActivo(resultadosr.getString(6));
  163:                                 prog.setCodprog(resultadosr.getString(7));
  164:                                 prog.setTipo(resultadosr.getString(8));
  165:                                 prog.setDial(resultadosr.getInt(9));
  166:                                 prog.setDiam(resultadosr.getInt(10));
  167:                                 prog.setDiax(resultadosr.getInt(11));
  168:                                 prog.setDiaj(resultadosr.getInt(12));
  169:                                 prog.setDiav(resultadosr.getInt(13));
  170:                                 prog.setDias(resultadosr.getInt(14));
  171:                                 prog.setDiad(resultadosr.getInt(15));
  172:                                 prog.setModo(resultadosr.getString(16));
  173:                                 prog.setModoini(resultadosr.getString(17));
  174:                                 prog.setPbloque(resultadosr.getInt(18));
  175:                                 prog.setCuota(resultadosr.getInt(19));
  176:                                 prog.setLeido(resultadosr.getString(20));
```

```
177:                                    prog.setEnmarcha(resultadosr.getString(21));
178:
179:
180:                                    System.out.println("\nDescripcion: "  + prog.getDsprograma());
181:                                    System.out.println(prog.getCodprograma());
182:                                    System.out.println(prog.getFcinicio());
183:
184:
185:                                    //Compruebo que la programacion no existe en local !!
186:                                    //Cojo la Ãºltima progamacion de la tabla
187:                                    sentenciaprel = connl.prepareStatement("SELECT MAX(idprograma)  FROM programa");
188:                                    resultadosl = sentenciaprel.executeQuery();
189:                                    while (resultadosl.next()) {
190:                                    prog.setIdprogramal(resultadosl.getInt(1));
191:
192:                                            //Cojo el dia
193:                                            sentenciapre = connr.prepareStatement("select fecha from v11_progdiasexec where
 idprograma="+prog.getIdprograma());
194:                                            resultadosr = sentenciapre.executeQuery();
195:                                            while (resultadosr.next()) {
196:                                                    prog.setDia(resultadosr.getDate(1));
197:                                                    System.out.println("Volcado guarda el dia " + prog.getDia());
198:                                                    //Aki llamo a las inserts de las dias, horas y valvulas;
199:                                                    if (!insertadia(sentenciaprel)){
200:                                                            System.out.println("Error al insertar el dia en la local");
201:                                                    }else{
202:                                                            //borradia(sentenciapre);
203:                                                    }
204:                                            }
205:
206:                                            //Cojo las horas y las guardo
207:                                            sentenciapre = connr.prepareStatement("select hrinicio,hrfin from v12_proghoras
exec where idprograma="+prog.getIdprograma());
208:                                            resultadosr = sentenciapre.executeQuery();
209:                                            while (resultadosr.next()) {
210:                                                    prog.setHoraini(resultadosr.getString(1));
211:                                                    prog.setHorafin(resultadosr.getString(2));
212:                                                    System.out.println("Volcado guarda las horas " + prog.getHoraini() + "
"+prog.getHorafin());
213:                                                    if (!insertahoras(sentenciaprel)){
214:                                                            System.out.println("Error al insertar las horas en la local");
215:                                                    }else{
216:                                                            //borrahoras(sentenciapre);
217:                                                    }
218:
219:                                            }
220:
221:                                            //Cojo las valvulas y las guardo
222:                                            sentenciapre = connr.prepareStatement("select CODELECVALV,DURACION,BLOQUE from
```

```
v13_progvalvexec where idprograma="+prog.getIdprograma());
  223:                                          resultadosr = sentenciapre.executeQuery();
  224:                                          while (resultadosr.next()) {
  225:                                                  valv.setCodelecvalv(resultadosr.getString(1));
  226:                                                  valv.setDuracion(resultadosr.getInt(2));
  227:                                                  valv.setBloque(resultadosr.getInt(3));
  228:                                                 System.out.println("Volcado guarda las valvulas " + valv.getCodelecvalv
());
  229:
  230:                                                 if (!insertavalv(sentenciaprel)){
  231:                                                         System.out.println("Error al insertar las horas en la local");
  232:                                                 }else{
  233:                                                         //borravalv(sentenciapre);
  234:                                                 }
  235:
  236:                                          }
  237:
  238:
  239:                                                 //Si la proglocal es menor es que la programaexec no estÃ¡ en la bbdd l
ocal, con lo que la inserto
  240:                                                 if (prog.getIdprogramal()<prog.getIdprograma())
  241:                                                 {
  242:                                                         if(insertaprog(sentenciaprel))
  243:                                                         {
  244:                                                                 //DeberÃ-a borrar la tarea de la programaciÃ³n una vez
EJECUTADA si acaso!!!!
  245: //                                                                      if (Borratarea(sentenciapre))
  246: //                                                                              System.out.println("Tarea de programaci
on borrada de la tabla t02 en la pasarela correctamente !");
  247:                                                                       borraprog(sentenciapre);
  248:                                                                       Borratarea(tar);
  249:                                                         }
  250:                                                         else{
  251:                                                                 System.out.println("Error al insertar la tarea de la bb
dd remota a la local !!");
  252:                                                         }
  253:                                                 }
  254:
  255:                                           }
  256:                           }
  257:
  258:         } catch (SQLException e) {
  259:                 e.printStackTrace();
  260:                 System.out.println("Error 261");
  261:         }
  262:
  263: }
  264:         protected boolean insertaprog(PreparedStatement sentenciaprel){
  265:
```

```
  266:                 System.out.println("Aki inserto programacion en la local");
  267:                 try {
  268:                         sentenciaprel = connl.prepareStatement("INSERT INTO programa (idprograma, codprograma, dsprogra
ma, fcinicio," +
  269:                                                                "fcfin, activo, codp
rog,tipo, dial,diam,diax,diaj,diav,dias,diad, " +
  270:                                                                "modo,modoini,pbloqu
e,cuota,leido,enmarcha)"+
  271:                                                                "VALUES (’"+prog.get
Idprograma()  +
  272:                                                                     "’,’"+prog.getC
odprograma()+
  273:                                                                     "’,’"+prog.getD
sprograma()+
  274:                                                                     "’,’"+prog.getF
cinicio()+
  275:                                                                     "’,’"+prog.getF
cFin()+
  276:                                                                     "’,’"+prog.getA
ctivo()+
  277:                                                                     "’,’"+prog.getC
odprog()+
  278:                                                                     "’,’"+prog.getT
ipo()+
  279:                                                                     "’,’"+prog.getD
ial()+
  280:                                                                     "’,’"+prog.getD
iam()+
  281:                                                                     "’,’"+prog.getD
iax()+
  282:                                                                     "’,’"+prog.getD
iaj()+
  283:                                                                     "’,’"+prog.getD
iav()+
  284:                                                                     "’,’"+prog.getD
ias()+
  285:                                                                     "’,’"+prog.getD
iad()+
  286:                                                                     "’,’"+prog.getM
odo()+
  287:                                                                     "’,’"+prog.getM
odoini()+
  288:                                                                     "’,’"+prog.getP
bloque()+
  289:                                                                     "’,’"+prog.getC
uota()+
  290:                                                                     "’,’"+prog.getL
eido()+
  291:                                                                     "’,’"+prog.getE
```

```
nmarcha()+"' )") ;
  292:                          sentenciaprel.executeUpdate();
  293:
  294:                          //sentenciaprel = connl
  295:                          return true;
  296:                  } catch (SQLException e) {
  297:                          //
  298:                          e.printStackTrace();
  299:                          System.out.println("No se ha podido actualizar la programacion en la BBDD local.");
  300:                          return false;
  301:                  }
  302:
  303:
  304:
  305:          }
  306:          protected boolean borraprog(PreparedStatement sentenciapre){
  307:
  308:                  try {
  309:                          //Actualizo a leido en la t10 para que me la borre de pendientes !!
  310:                          sentenciapre = connr.prepareStatement("UPDATE t10_programa set leido='S' WHERE idprograma="+pro
g.getIdprograma());
  311:                          sentenciapre.executeUpdate();
  312:                          System.out.println("Programa ya leido y borrado de la pasarela");
  313:                          return true;
  314:                  } catch (SQLException e) {
  315:                          e.printStackTrace();
  316:                          System.out.println("No se ha podido borrar el programa en la BBDD remota.");
  317:                          return false;
  318:                  }
  319:          }
  320:
  321:          protected boolean insertadia(PreparedStatement sentenciaprel){
  322:                  System.out.println("Aki inserto dias en la local");
  323:                  try {
  324:                          sentenciaprel = connl.prepareStatement("INSERT INTO dias (codprog,idprograma,fecha)"+
  325:                                                                                                  "VALUES ('"+config.g
etIdrasp()  +
  326:                                                                                                       "','"+prog.getI
dprograma()+
  327:                                                                                                       "','"+prog.getD
ia()+"' )") ;
  328:                          sentenciaprel.executeUpdate();
  329:
  330:                          return true;
  331:                  } catch (SQLException e) {
  332:                          //
  333:                          e.printStackTrace();
  334:                          System.out.println("No se ha podido actualizar el dia en la BBDD local.");
  335:                          return false;
```

```
  336:                 }
  337:
  338:         }
  339:
  340:
  341:         protected boolean insertahoras(PreparedStatement sentenciaprel){
  342:                 System.out.println("Aki inserto horas en la local");
  343:                 try {
  344:
  345:                         sentenciaprel = connl.prepareStatement("INSERT INTO horas (codprog,idprograma,hrinicio,hrfin)"+
  346:                                                                                                                 "VALUES ('"+config.g
etIdrasp()  +
  347:                                                                                                                  "','"+prog.getIdpro
grama()+
  348:                                                                                                                      "','"+prog.getH
oraini()+
  349:                                                                                                                      "','"+prog.getH
orafin()+"' )") ;
  350:                         sentenciaprel.executeUpdate();
  351:
  352:                         return true;
  353:                 } catch (SQLException e) {
  354:                         //
  355:                         e.printStackTrace();
  356:                         System.out.println("No se ha podido actualizar las horas en la BBDD local.");
  357:                         return false;
  358:                 }
  359:
  360:         }
  361:
  362:         protected boolean insertavalv(PreparedStatement sentenciaprel){
  363:                 System.out.println("Aki inserto valvulas en la local");
  364:                 try {
  365:
  366:                         sentenciaprel = connl.prepareStatement("INSERT INTO valvulas (codprog,idprograma,codelecvalv,du
racion,bloque)"+
  367:                                                                                                                 "VALUES ('"+config.g
etIdrasp()  +
  368:                                                                                                                 "','"+prog.getIdprog
rama()+
  369:                                                                                                                 "','"+valv.getCodele
cvalv()+
  370:                                                                                                                 "','"+valv.getDuraci
on()+
  371:                                                                                                                 "','"+valv.getBloque
()+"' )") ;
  372:                         sentenciaprel.executeUpdate();
  373:
  374:                         return true;
```

```
  375:                 } catch (SQLException e) {
  376:                         e.printStackTrace();
  377:                         System.out.println("No se ha podido actualizar valvulas en la BBDD local.");
  378:                         return false;
  379:                 }
  380:
  381:         }
  382:
  383:
  384:
  385:
  386:         ////////////////////////////////////////////////////////////////////////////
  387:         //
  388:         // GestiÃ³n conexiones a las BBDD
  389:         protected void conectar (){
  390:
  391:
  392:                 while (conectado==false)
  393:                 {
  394:                         try {
  395:                                 Class.forName("com.mysql.jdbc.Driver");
  396:
  397:                                 // REMOTA
  398:                                 String urlremota = "jdbc:mysql://"+config.getHostr()+":"+config.getPuertor()+"/"+config
.getDbr();
  399:                                 connr = DriverManager.getConnection(urlremota,config.getUserr(),config.getPassr());
  400:                                 System.out.println("ConexiÃ³n con la pasarela establecida !!");
  401:                                 conectado = true;
  402:                         }
  403:                         catch (Exception e) {
  404:                                 //e.printStackTrace();
  405:
  406:                                 System.out.println("\n");
  407:                                 System.out.println("No se ha podido conectar con la BBDD remota!!");
  408:                                 System.out.println("Reintentando conexion en 10 segundos.");
  409:                                 int i=0;
  410:                                 while ( i<10){
  411:                                         try {
  412:                                                 Thread.sleep(500);
  413:                                                 System.out.print(".");
  414:                                                 Thread.sleep(500);
  415:                                                 System.out.print(".");
  416:                                                 i++;
  417:                                         } catch (InterruptedException e1) {
  418:                                                 System.out.println("La BBDD remota estÃ¡ caida! ContÃ¡cte con su admini
strador.");
  419:                                         }
  420:                                 }
  421:                         }
```

```
  422:                    }
  423:            }
  424:
  425:            protected void conectal (){
  426:
  427:                    boolean conectado = false;
  428:
  429:                    while (conectado==false)
  430:                    {
  431:                            try {
  432:                                    Class.forName("com.mysql.jdbc.Driver");
  433:
  434:                                    // LOCAL
  435:                                    String urllocal = "jdbc:mysql://"+config.getHostl()+":"+config.getPuertol()+"/"+config.
getDbl();
  436:                                    connl = DriverManager.getConnection(urllocal, config.getUsuariol(), config.getPassl());
  437:                                    System.out.println("ConexiÃ³n con la bbdd local establecida !!");
  438:                                    conectado = true;
  439:                            }
  440:                            catch (Exception e) {
  441:                                    //e.printStackTrace();
  442:
  443:                                    System.out.println("\n");
  444:                                    System.out.println("No se ha podido conectar con la BBDD local!!");
  445:                                    System.out.println("Reintentando conexion en 10 segundos.");
  446:                                    int i=0;
  447:                                    while ( i<10){
  448:                                            try {
  449:                                                    Thread.sleep(500);
  450:                                                    System.out.print(".");
  451:                                                    Thread.sleep(500);
  452:                                                    System.out.print(".");
  453:                                                    i++;
  454:                                            } catch (InterruptedException e1) {
  455:                                                    System.out.println("La BBDD local estÃ¡ caida! ContÃ¡cte con su adminis
trador.");
  456:                                            }
  457:                                    }
  458:                            }
  459:                    }
  460:
  461:            }//Fin COnectaL
  462:
  463:            protected void cierrar (){
  464:                    if (connr != null) {
  465:                            try {
  466:                                    connr.close();
  467:                                    System.out.println("ConexiÃ³n terminada con la pasarela.");
  468:                            } catch (Exception e) { /* ignore close errors */
```

```
469:                            }
470:                    }
471:            }
472:
473:            protected void cierral (){
474:                    if (connl != null) {
475:                            try {
476:                                    connl.close();
477:                                    System.out.println("ConexiÃ³n terminada con la bbdd local.");
478:                            } catch (Exception e) { /* ignore close errors */
479:                            }
480:                    }
481:            }
482:
483:
484:            ////////////////////////////////////////////////////////////////////////
485:            //
486:            //OJO Lee archivo de configuraciÃ³n conf_volcado.txt !!!
487:            protected void leerconf(Conf config){
488:
489:                            FileReader fr = null;
490:                            String linea = null;
491:
492:
493:                            try {
494:                                    fr = new FileReader(rutaconf);  //Ruta al archivo de configuraciÃ³n
495:
496:                            } catch (FileNotFoundException e) {
497:                                    e.printStackTrace();
498:                                    System.out.println("No existe el fichero de configuraciÃ³n !! \n Se sale !" );
499:
500:                            }
501:                            BufferedReader bf = new BufferedReader(fr);
502:
503:
504:                            //
505:                            //Leo lÃ-nea lÃ-nea el fichero de configuraciÃ³n y lo guardo todo en la clase conf
506:                            //
507:                            try {
508:                                    while ((linea = bf.readLine())!=null) {
509:
510:                                            if (linea.contains("ID")){
511:                                                    int i = linea.indexOf(":") +1;
512:                                                    config.setIdrasp(Integer.parseInt(linea.substring(i)));
513:                                                    //System.out.println(config.getIdrasp());
514:                                            }
515:                                            else if (linea.contains("HOSTR")){
516:                                                    int i = linea.indexOf(":") +1;
517:                                                    config.setHostr((linea.substring(i)));
```

```
518:                                    //System.out.println(config.getHostr());
519:                                }
520:                                else if (linea.contains("PUERTOR")){
521:                                    int i = linea.indexOf(":") +1;
522:                                    config.setPuertor(Integer.parseInt(linea.substring(i)));
523:                                    //System.out.println(config.getPuertor());
524:                                }
525:                                else if (linea.contains("DBR")){
526:                                    int i = linea.indexOf(":") +1;
527:                                    config.setDbr((linea.substring(i)));
528:                                    //System.out.println(config.getDbr());
529:                                }
530:                                else if (linea.contains("USUARIOR")){
531:                                    int i = linea.indexOf(":") +1;
532:                                    config.setUserr((linea.substring(i)));
533:                                    //System.out.println(config.getUserr());
534:                                }
535:                                else if (linea.contains("PASSR")){
536:                                    int i = linea.indexOf(":") +1;
537:                                    config.setPassr((linea.substring(i)));
538:                                    //System.out.println(config.getPassr());
539:                                }
540:                                else if (linea.contains("HOSTL")){
541:                                    int i = linea.indexOf(":") +1;
542:                                    config.setHostl((linea.substring(i)));
543:                                    //System.out.println(config.getHostl());
544:                                }
545:                                else if (linea.contains("PUERTOL")){
546:                                    int i = linea.indexOf(":") +1;
547:                                    config.setPuertol(Integer.parseInt(linea.substring(i)));
548:                                    //System.out.println(config.getPuertol());
549:                                }
550:                                else if (linea.contains("DBL")){
551:                                    int i = linea.indexOf(":") +1;
552:                                    config.setDbl((linea.substring(i)));
553:                                    //System.out.println(config.getDbl());
554:                                }
555:                                else if (linea.contains("USUARIOL")){
556:                                    int i = linea.indexOf(":") +1;
557:                                    config.setUsuariol((linea.substring(i)));
558:                                    //System.out.println(config.getUsuariol());
559:                                }
560:                                else if (linea.contains("PASSL")){
561:                                    int i = linea.indexOf(":") +1;
562:                                    config.setPassl((linea.substring(i)));
563:                                    //System.out.println(config.getPassl());
564:                                }
565:                                else if (linea.contains("TIEMPO")){
566:                                    int i = linea.indexOf(":") +1 ;
```

```
567:                                                    config.setTiempo(Integer.parseInt(linea.substring(i)));
568:                                                    //System.out.println(config.getTiempo());
569:                                            }
570:
571:                                    }
572:
573:                            } catch (IOException e) {
574:                                    //
575:                                    e.printStackTrace();
576:                                    System.out.println("No se ha podido leer la linea ! \n Se sale !" );
577:
578:                            }
579:                            try {
580:                                    fr.close();
581:                                    bf.close();
582:                            } catch (IOException e) {
583:                                    //
584:                                    e.printStackTrace();
585:                                    System.out.println("No se ha podido cerrar la lectura de archivo !!" );
586:                            }
587:                    }
588:
589:
590:
591: }
```

```
    1: package packvolcado;
    2:
    3:
    4:
    5:
    6: import java.sql.Timestamp;
    7:
    8:
    9:
   10:
   11: public class Tarea {
   12:
   13:         // Nueva tarea que tenga que realizar
   14:         private int Idtareaexec;
   15:         private String Codprog;
   16:         private String Codelecvalv;
   17:         private int Iddstarea;
   18:         private Timestamp Fcexec;
   19:         private int Valor;
   20:         private String Codtarea;
   21:         private String Dstarea;
   22:
   23:         //Para comparar con la remota
   24:         private int Idtarealocal;
   25:         private Timestamp Fcloc;
   26:
   27:
   28:
   29:         public int getIdtareaexec() {
   30:                 return Idtareaexec;
   31:         }
   32:         public void setIdtareaexec(int idtareaexec) {
   33:                 Idtareaexec = idtareaexec;
   34:         }
   35:         public String getCodprog() {
   36:                 return Codprog;
   37:         }
   38:         public void setCodprog(String codprog) {
   39:                 Codprog = codprog;
   40:         }
   41:         public String getCodelecvalv() {
   42:                 return Codelecvalv;
   43:         }
   44:         public void setCodelecvalv(String codelecvalv) {
   45:                 Codelecvalv = codelecvalv;
   46:         }
   47:         public int getIddstarea() {
   48:                 return Iddstarea;
   49:         }
```

```
50:         public void setIddstarea(int iddstarea) {
51:                 Iddstarea = iddstarea;
52:         }
53:         public Timestamp getFcexec() {
54:                 return Fcexec;
55:         }
56:         public void setFcexec(Timestamp fcexec) {
57:                 // si es null pongo la fecha actual !!!
58:                 if (fcexec == null){
59:                         java.util.Date date= new java.util.Date();
60:                         Fcexec = new Timestamp(date.getTime());
61:
62:                 }else{
63:                         Fcexec = fcexec;
64:                 }
65:
66:         }
67:         public int getValor() {
68:                 return Valor;
69:         }
70:         public void setValor(int valor) {
71:                 Valor = valor;
72:         }
73:         public int getIdtarealocal() {
74:                 return Idtarealocal;
75:         }
76:         public void setIdtarealocal(int idtarealocal) {
77:                 Idtarealocal = idtarealocal;
78:         }
79:         public String getCodtarea() {
80:                 return Codtarea;
81:         }
82:         public void setCodtarea(String codtarea) {
83:                 Codtarea = codtarea;
84:         }
85:         public String getDstarea() {
86:                 return Dstarea;
87:         }
88:         public void setDstarea(String dstarea) {
89:                 Dstarea = dstarea;
90:         }
91:
92:         public Timestamp getFcloc() {
93:                 return Fcloc;
94:         }
95:         public void setFcloc(Timestamp fcloc) {
96:                 Fcloc = fcloc;
97:         }
98:
```

```
   99: }
```

```
 1: package packvolcado;
 2:
 3: public class Valvula {
 4:
 5:
 6:           private String Codelecvalv;
 7:           private int Duracion;
 8:           private int Bloque;
 9:
10:
11:                public String getCodelecvalv() {
12:                        return Codelecvalv;
13:                }
14:                public void setCodelecvalv(String codelecvalv) {
15:                        Codelecvalv = codelecvalv;
16:                }
17:                public int getDuracion() {
18:                        return Duracion;
19:                }
20:                public void setDuracion(int duracion) {
21:                        Duracion = duracion;
22:                }
23:                public int getBloque() {
24:                        return Bloque;
25:                }
26:                public void setBloque(int bloque) {
27:                        Bloque = bloque;
28:                }
29:
30:
31: }
```

```
    1: package packvolcado;
    2:
    3: public class Conf {
    4:
    5:         private int idrasp;
    6:
    7:         //Datos conexion remota
    8:         private String hostr;
    9:         private int puertor;
   10:         private String dbr;
   11:         private String userr;
   12:         private String passr;
   13:
   14:         //Datos conexion local
   15:         private String hostl;
   16:         private int puertol;
   17:         private String dbl;
   18:         private String usuariol;
   19:         private String passl;
   20:
   21:         //Tiempo de espera a la conexiÃ³n
   22:         private int tiempo;
   23:
   24:
   25:
   26:
   27:         public int getIdrasp() {
   28:                 return idrasp;
   29:         }
   30:
   31:         public void setIdrasp(int idrasp) {
   32:                 this.idrasp = idrasp;
   33:         }
   34:
   35:         public String getHostr() {
   36:                 return hostr;
   37:         }
   38:
   39:         public void setHostr(String hostr) {
   40:                 this.hostr = hostr;
   41:         }
   42:
   43:         public int getPuertor() {
   44:                 return puertor;
   45:         }
   46:
   47:         public void setPuertor(int puertor) {
   48:                 this.puertor = puertor;
   49:         }
```

```
50:
51:         public String getDbr() {
52:                 return dbr;
53:         }
54:
55:         public void setDbr(String dbr) {
56:                 this.dbr = dbr;
57:         }
58:
59:         public String getUserr() {
60:                 return userr;
61:         }
62:
63:         public void setUserr(String userr) {
64:                 this.userr = userr;
65:         }
66:
67:         public String getPassr() {
68:                 return passr;
69:         }
70:
71:         public void setPassr(String passr) {
72:                 this.passr = passr;
73:         }
74:
75:         public String getHostl() {
76:                 return hostl;
77:         }
78:
79:         public void setHostl(String hostl) {
80:                 this.hostl = hostl;
81:         }
82:
83:         public int getPuertol() {
84:                 return puertol;
85:         }
86:
87:         public void setPuertol(int puertol) {
88:                 this.puertol = puertol;
89:         }
90:
91:         public String getDbl() {
92:                 return dbl;
93:         }
94:
95:         public void setDbl(String dbl) {
96:                 this.dbl = dbl;
97:         }
98:
```

```
 99:         public String getUsuariol() {
100:                 return usuariol;
101:         }
102:
103:         public void setUsuariol(String usuariol) {
104:                 this.usuariol = usuariol;
105:         }
106:
107:         public String getPassl() {
108:                 return passl;
109:         }
110:
111:         public void setPassl(String passl) {
112:                 this.passl = passl;
113:         }
114:
115:         public int getTiempo() {
116:                 return tiempo;
117:         }
118:
119:         public void setTiempo(int tiempo) {
120:                 this.tiempo = tiempo;
121:         }
122:
123:
124:
125: }
```

```
    1: package packvolcado;
    2:
    3: import java.sql.Date;
    4: //import java.sql.Timestamp;
    5:
    6: public class Programacion {
    7:
    8:         private int Idprograma;
    9:         private String Codprograma;
   10:         private String Dsprograma;
   11:         private Date Fcinicio;
   12:         private Date FcFin;
   13:         private String Activo;
   14:         private String Codprog;
   15:         private String tipo;
   16:         private int dial,diam,diax,diaj,diav,dias,diad;
   17:         private String modo;
   18:         private String modoini;
   19:         private int Pbloque;
   20:         private int Cuota;
   21:         private String Leido;
   22:         private String Enmarcha;
   23:
   24:         //local y tal
   25:         private int Idprogramal;
   26:         //private Timestamp Fcloc;
   27:         private Date dia;
   28:         private String horaini,horafin;
   29:
   30:
   31:
   32:         //Getters y Setters
   33:
   34:         public int getIdprograma() {
   35:                 return Idprograma;
   36:         }
   37:         public void setIdprograma(int idprograma) {
   38:                 Idprograma = idprograma;
   39:         }
   40:         public String getCodprograma() {
   41:                 return Codprograma;
   42:         }
   43:         public void setCodprograma(String codprograma) {
   44:                 Codprograma = codprograma;
   45:         }
   46:         public String getDsprograma() {
   47:                 return Dsprograma;
   48:         }
   49:         public void setDsprograma(String dsprograma) {
```

```
50:                 Dsprograma = dsprograma;
51:         }
52:         public Date getFcinicio() {
53:                 return Fcinicio;
54:         }
55:         public void setFcinicio(Date fcinicio) {
56:                 Fcinicio = fcinicio;
57:         }
58:         public Date getFcFin() {
59:                 return FcFin;
60:         }
61:         public void setFcFin(Date fcFin) {
62:                 FcFin = fcFin;
63:         }
64:         public String getActivo() {
65:                 return Activo;
66:         }
67:         public void setActivo(String activo) {
68:                 Activo = activo;
69:         }
70:         public String getCodprog() {
71:                 return Codprog;
72:         }
73:         public void setCodprog(String codprog) {
74:                 Codprog = codprog;
75:         }
76:         public String getTipo() {
77:                 return tipo;
78:         }
79:         public void setTipo(String tipo) {
80:                 this.tipo = tipo;
81:         }
82:         public int getDial() {
83:                 return dial;
84:         }
85:         public void setDial(int dial) {
86:                 this.dial = dial;
87:         }
88:         public int getDiam() {
89:                 return diam;
90:         }
91:         public void setDiam(int diam) {
92:                 this.diam = diam;
93:         }
94:         public int getDiax() {
95:                 return diax;
96:         }
97:         public void setDiax(int diax) {
98:                 this.diax = diax;
```

```
 99:         }
100:         public int getDiaj() {
101:                 return diaj;
102:         }
103:         public void setDiaj(int diaj) {
104:                 this.diaj = diaj;
105:         }
106:         public int getDiav() {
107:                 return diav;
108:         }
109:         public void setDiav(int diav) {
110:                 this.diav = diav;
111:         }
112:         public int getDias() {
113:                 return dias;
114:         }
115:         public void setDias(int dias) {
116:                 this.dias = dias;
117:         }
118:         public int getDiad() {
119:                 return diad;
120:         }
121:         public void setDiad(int diad) {
122:                 this.diad = diad;
123:         }
124:         public String getModo() {
125:                 return modo;
126:         }
127:         public void setModo(String modo) {
128:                 this.modo = modo;
129:         }
130:         public String getModoini() {
131:                 return modoini;
132:         }
133:         public void setModoini(String modoini) {
134:                 this.modoini = modoini;
135:         }
136:         public int getPbloque() {
137:                 return Pbloque;
138:         }
139:         public void setPbloque(int pbloque) {
140:                 Pbloque = pbloque;
141:         }
142:         public int getCuota() {
143:                 return Cuota;
144:         }
145:         public void setCuota(int cuota) {
146:                 Cuota = cuota;
147:         }
```

```
148:         public String getLeido() {
149:                 return Leido;
150:         }
151:         public void setLeido(String leido) {
152:                 Leido = leido;
153:         }
154:         public String getEnmarcha() {
155:                 return Enmarcha;
156:         }
157:         public void setEnmarcha(String enmarcha) {
158:                 Enmarcha = enmarcha;
159:         }
160:         public int getIdprogramal() {
161:                 return Idprogramal;
162:         }
163:         public void setIdprogramal(int idprogramal) {
164:                 Idprogramal = idprogramal;
165:         }
166:         public Date getDia() {
167:                 return dia;
168:         }
169:         public void setDia(Date dia) {
170:                 this.dia = dia;
171:         }
172:         public String getHoraini() {
173:                 return horaini;
174:         }
175:         public void setHoraini(String horaini) {
176:                 this.horaini = horaini;
177:         }
178:         public String getHorafin() {
179:                 return horafin;
180:         }
181:         public void setHorafin(String horafin) {
182:                 this.horafin = horafin;
183:         }
184:
185:
186: }
```