

PEC 4

Fecha solución: 13/06/2007

**Solución Oficial**

## Presentación y Objetivos

La Prueba de Evaluación Continuada IV (PEC4) es la última de las cuatro PECs de las que consta el curso. Dicha PEC pesará un 35% en la nota final, y los conceptos en ella desarrollados son los correspondientes a los módulos “Bases de datos en MySQL”, “Bases de datos en PostgreSQL”, “Desarrollo de aplicaciones en conexión con Bases de datos” y “Casos de estudio”.

La PEC consta de tres ejercicios prácticos y de un complementario voluntario. La valoración de cada pregunta en el global de la PEC se incluye en la cabecera de cada una.

Puesto que la asignatura no tiene examen, es necesario realizar todas las PECs para tener nota final.

### 1. Disparadores (15%)

A efectos de controlar el uso de la aplicación Seguimiento Temperaturas que se ha realizado (según descripción de tablas de PEC3, ejercicio 1), se desea auditar el borrado de datos de la tabla SUSTITUTO.

Para ello se pide:

- a) Crear la tabla:  
AUDITORIA (id\_aud, nombre\_tabla, operacion, fecha\_hora, valor\_anterior, usuario)
- b) Crear el disparador que se debe ejecutar en el caso de borrado de datos de SUSTITUTO, a fin de que se inserten los datos en AUDITORIA, según la siguiente descripción de los campos:

**id\_aud:** identificador único, campo tipo secuencia.

**nombre\_tabla:** se almacenará el nombre de la tabla donde se ha producido la operación que se audita (en este ejercicio siempre el literal ‘SUSTITUTO’).

**operacion:** tipo de operación que se ha producido: i (de insert), u (de update) o d (de delete). En este caso siempre se insertará la ‘d’.

**fecha\_hora:** timestamp del momento de ejecución de la rutina de auditoria.

**valor\_anterior:** se indicará que es tipo varchar(100) y contendrá concatenados (separados por el signo ‘dos puntos’) los valores de id\_colaborador, id\_sustituto y id\_tipo\_sustitucion.

**usuario:** campo que almacenará el identificador del usuario de la base de datos que ha ejecutado la acción auditada.

### Con PostgreSQL:

*Creamos la tabla:*

```
CREATE TABLE AUDITORIA (
  ID_AUD SERIAL PRIMARY KEY,
  NOMBRE_TABLA VARCHAR(40) NOT NULL,
  OPERACION CHAR(1),
  FECHA_HORA TIMESTAMP DEFAULT NOW(),
  VALOR_ANTERIOR VARCHAR(100),
  USUARIO VARCHAR(40) NOT NULL,
```

```
CHECK (OPERACION = 'I' OR OPERACION = 'U' OR OPERACION = 'D'));
```

*Creamos la función que insertará los valores en los campos de la tabla AUDITORIA. Dicho código será llamado por el disparador:*

```
CREATE FUNCTION FU_AUD_SUSTITUTO_DEL() RETURNS TRIGGER AS '  
BEGIN  
  INSERT INTO AUDITORIA (NOMBRE_TABLA, OPERACION, VALOR_ANTERIOR, USUARIO)  
  VALUES (  
    "SUSTITUTO",  
    "D",  
    OLD.ID_COLABORADOR || ":" || OLD.ID_SUSTITUTO || ":" || OLD.ID_TIPO_SUSTITUCION,  
    USER);  
  RETURN NEW;  
END;  
' LANGUAGE 'plpgsql';
```

*Creamos el disparador que llamará a la función a cada borrado:*

```
CREATE TRIGGER TR_AUD_SUSTITUTO_DEL  
BEFORE DELETE ON SUSTITUTO  
FOR EACH ROW  
EXECUTE PROCEDURE FU_AUD_SUSTITUTO_DEL();
```

#### INDICACIONES:

- Nombre\_tabla y operación se insertaran como un literales desde el trigger (o función llamada por el trigger).
- Para cumplimentar el campo usuario se recuperará el nombre del usuario utilizando "user" (PostgreSQL) o "user()" (MySQL).

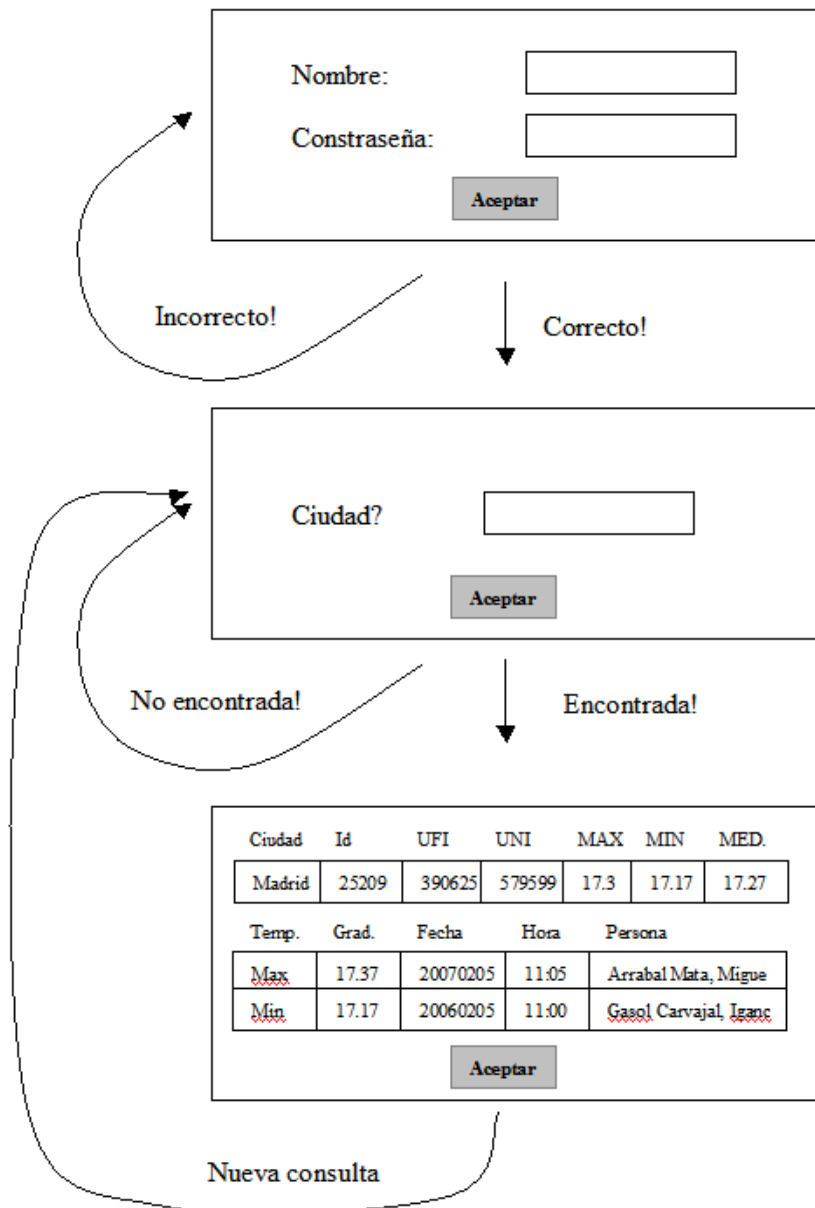
El ejercicio se puede realizar utilizando PostgreSQL o MySQL (en el caso de utilizar MySQL será preciso crear la tabla SUSTITUTO – no hace falta indicar las FK, ni crear tablas con las que se relacionaría). Adjuntar el trigger, el procedimiento (de ser necesario) y la prueba de funcionamiento.

## 2. Sistema de consultas Web (60%)

El diseño de la aplicación para el Seguimiento de las Temperaturas medioambientales ha conseguido finalmente la ayuda económica de la Comunidad Económica Europea, por lo que se decide empezar a codificar el modulo de autenticación al sistema y unas primeras pantallas para verificar su funcionamiento. Por ello partiendo del diseño y las tablas de la solución de la PEC3 se pide:

- a) Crear la tabla:  
USUARIOS (Usuario, Contraseña)  
El nombre de usuario no se puede repetir. La contraseña no puede estar vacía.
- b) Realizar una aplicación en PHP que permita a los usuarios dados de alta en el sistema consultar las temperatura máxima y mínimas de una ciudad.

Gráficamente:



### DETALLES:

#### - Pantalla inicial:

En ella se pedirá nombre de usuario y contraseña.

Se enviarán dichos datos por POST a una segunda pantalla que cotejará en la tabla USUARIOS la corrección del nombre de usuario y contraseña. En el caso de ser incorrectos se indicará y se volverá a la pantalla inicial. En el caso de ser correctos se pasará a la siguiente pantalla. En todas estas operaciones no hace falta tener en cuenta la seguridad de la aplicación.

#### - Pantalla de introducción de ubicación:

La pantalla a la que se accede una vez verificada la contraseña pedirá el nombre de una ubicación. Una vez introducido el dato, se pasará por POST a una nueva pantalla, y desde esta se buscará en la tabla UBICACION. Si no se encuentra se indicará y se volverá a la pantalla anterior. En el caso de localizarse, se deberá mostrar en un primer bloque de información:

- Nombre de la ciudad/ubicación.
- Código identificativo.
- Coordenadas UNI y UFI.
- Temperatura máxima, mínima y media.

En un segundo bloque de datos, se mostrará para la temperatura máxima;

- La temperatura en grados
- La fecha
- La hora en que se midió
- La persona que realizo dicha lectura.

Se mostraran los mismos datos para la temperatura mínima.

Para demostrar el correcto funcionamiento de la aplicación se deben adjuntar las siguientes capturas de pantalla:

- Pantalla inicial de autenticación.
- Pantalla donde se introduce la ubicación a buscar.
- Pantalla con los datos devueltos para 'Madrid'.
- Pantalla con los datos devueltos para 'Salou'.

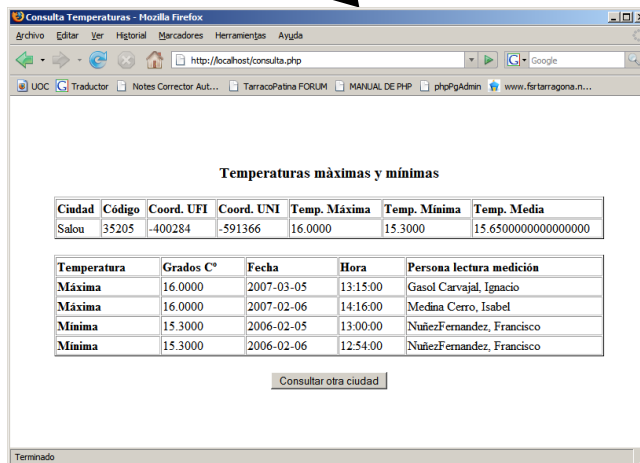
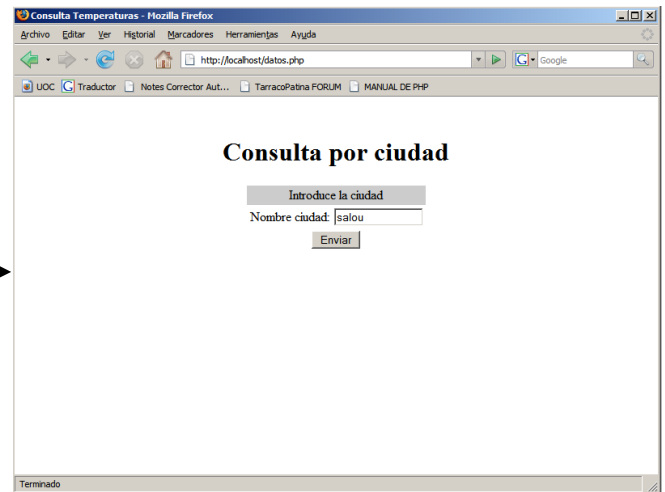
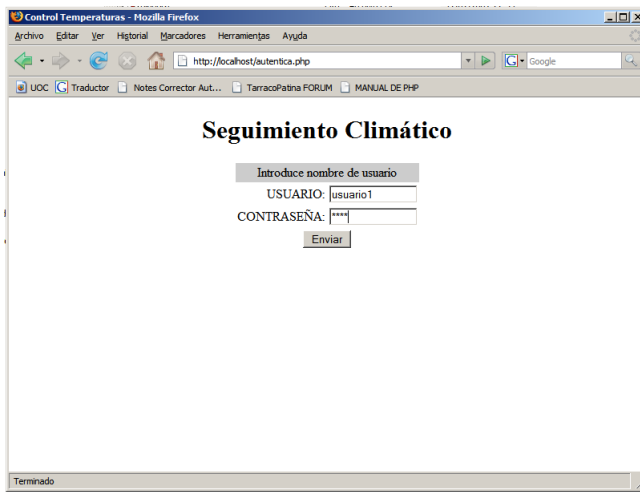
Se entregarán los programas PHP en un fichero comprimido, conjuntamente con el documento con la solución de la PEC.

Se valorará un pequeño resumen con las incidencias y/o el proceso seguido para la realización.

```
CREATE TABLE USUARIO (  
  NOMBRE VARCHAR(20) PRIMARY KEY,  
  CONTRASEÑA VARCHAR(20) NOT NULL);
```

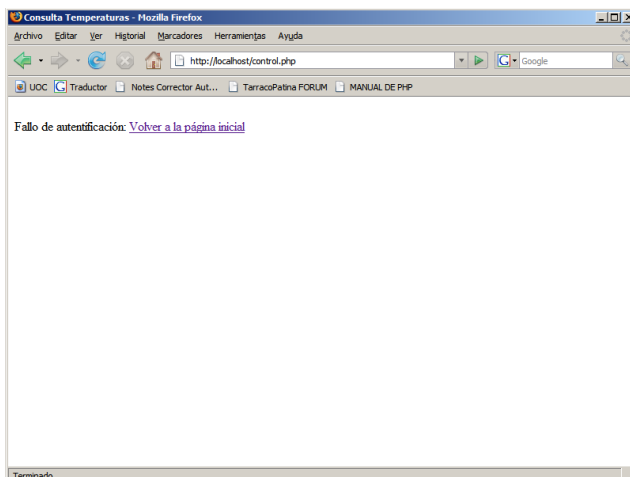
```
INSERT INTO USUARIO VALUES ('usuario1', 'abc');
```

### Pantallas aplicación:

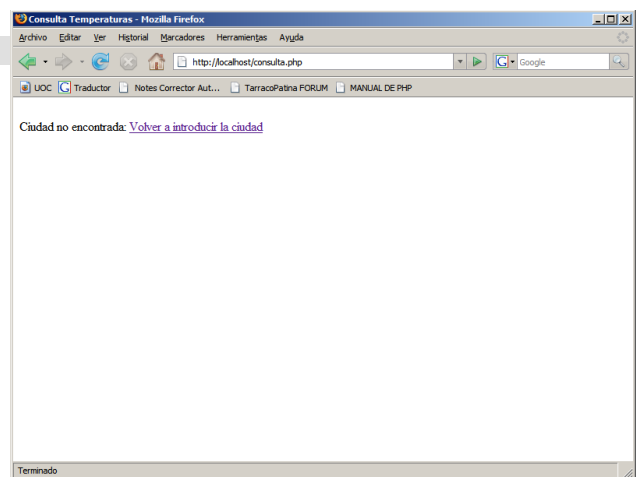


### Control de errores:

#### Fallo de autenticación:



#### No se encuentra ciudad:



```
<html>
<head>
  <title>Control Temperaturas</title>
</head>
<body>
  <center><h1>Seguimiento Climático</h1></center>
  <form action="control.php" method="post">
  <table align="center" width="225" cellspacing="2" cellpadding="2" border="0">
  <tr>
    <td colspan="2" align="center" bgcolor=#cccccc>Introduce nombre de usuario</td>
  </tr>
  <tr>
    <td align="right">USUARIO: </td>
    <td><input type="Text" name="usuario" size="14" maxlength="50"></td>
  </tr>
  <tr>
    <td align="right">CONTRASEÑA: </td>
    <td><input type="password" name="contrasena" size="14" maxlength="50"></td>
  </tr>
  <tr>
    <td colspan="2" align="center"><input type="Submit" value="Enviar"></td>
  </tr>
  </table>
</form>
</body>
</html>
```

```
<html>
<head>
  <title>Consulta Temperaturas</title>
</head>
<body>
  <br>
  <?php
    $conexion = pg_connect("host=localhost port=5432 dbname=postgres password=1234
user=postgres") or die("<BR>ERROR accediendo a la BD!<BR>");
    $sql = "select count(*) from usuario where nombre = '".$_POST['usuario']."' and
contrasena = '".$_POST['contrasena']."'";
    $resultado_set = pg_Exec ($conexion, $sql);
    $TotalRegistros = pg_result($resultado_set, 0);
    pg_close($conexion);
    if ($TotalRegistros == 0) {
      echo "Fallo de autenticación: ";
      ?> <a href="autentica.php">Volver a la página inicial</a>
    <?php
    }
  }
  else {
    header("location: datos.php");
  }
  ?>
</body>
</html>
```

```
<html>
```

```

<head>
  <title>Consulta Temperaturas</title>
</head>
<body>
<br>
<?php
  $conexion = pg_connect("host=localhost port=5432 dbname=postgres password=1234
user=postgres") or die("<BR>ERROR accediendo a la BD!<BR>");
  $sql = "SELECT UB.ID_UBICACION, UB.NOMBRE, UFI, UNI, MAX(GRADOS), MIN(GRADOS),
  AVG(GRADOS)
  FROM
  TEMPERATURA TE, UBICACION UB
  WHERE
  TE.ID_UBICACION = UB.ID_UBICACION AND
  UPPER(NOMBRE) = UPPER('".$_POST['ciudad']. "')
  GROUP BY UB.NOMBRE, UB.ID_UBICACION, UFI, UNI";
  $resultado_set = pg_Exec ($conexion, $sql);
  if (pg_num_rows ($resultado_set) == 0){
    echo "Ciudad no encontrada: ";
    ?> <a href="datos.php">Volver a introducir la ciudad</a>
  }
  <?php
  }
  else {
    // Coloca en variables los resultados de la select (ID_UBICACION, NOMBRE, UFI, UNI,
    G_MAX, G_MIN i G_MEDIA
    $ID_UBICACION = pg_result($resultado_set, 0, 0);
    $NOMBRE = pg_result($resultado_set, 0, 1);
    $G_MAX = pg_result($resultado_set, 0, 4);
    $G_MIN = pg_result($resultado_set, 0, 5);
    ?><BR><BR>
    <CENTER><H3>Temperaturas màximas y mínimas</H3></TD>
    <TABLE width="700" cellpadding="1" cellspacing="1" border="1"
bordercolor="#000000"><TR>
    <TR>
    <TD><B>Ciudad</B></TD>
    <TD><B>Código</B></TD>
    <TD><B>Coord. UFI</B></TD>
    <TD><B>Coord. UNI</B></TD>
    <TD><B>Temp. Màxima</B></TD>
    <TD><B>Temp. Mínima</B></TD>
    <TD><B>Temp. Media</B></TD>
    </TR>
    <TR>
    <TD><?php ECHO $NOMBRE; ?></TD>
    <TD><?php ECHO $ID_UBICACION; ?></TD>
    <TD><?php ECHO pg_result($resultado_set, 0, 2) ?></TD>
    <TD><?php ECHO pg_result($resultado_set, 0, 3) ?></TD>
    <TD><?php ECHO $G_MAX ?></TD>
    <TD><?php ECHO $G_MIN ?></TD>
    <TD><?php ECHO pg_result($resultado_set, 0, 6) ?></TD>
    </TR>
    </TABLE>
    <BR>
    <TABLE width="700" cellpadding="1" cellspacing="1" border="1"
bordercolor="#000000"><TR>
    <TR border="0">
    <TD><B>Temperatura</B></TD>
    <TD><B>Grados C°</B></TD>
    <TD><B>Fecha</B></TD>
    <TD><B>Hora</B></TD>
    <TD><B>Persona lectura medición</B></TD>
    </TR>
    <?php
    // Con parte de los datos recuperados anteriormente, recupera la fecha y hora, y
  
```

```

los datos de la persona que realizo la medición de temperatura MAXIMA
$sql = "SELECT TE.FECHA, TE.HORA, CO.APELLIDO1 || ' ' || CO.APELLIDO2 || ', ' ||
CO.NOMBRE
FROM
TEMPERATURA TE, COLABORADOR CO
WHERE
TE.GRADOS = '". $G_MAX.'" AND
TE.ID_UBICACION = ".$ID_UBICACION." AND
CO.ID_COLABORADOR = TE.ID_COLABORADOR";
$resultado_set = pg_Exec ($conexion, $sql);
$MAX_resultado_rows = pg_NumRows($resultado_set);
for ($contador=0; $contador < $MAX_resultado_rows; $contador++) {
    ?>
    <TR>
    <TD><B>Máxima</B></TD>
    <TD><?php ECHO $G_MAX; ?></TD>
    <TD><?php ECHO pg_result($resultado_set, $contador, 0) ?></TD>
    <TD><?php ECHO pg_result($resultado_set, $contador, 1) ?></TD>
    <TD><?php ECHO pg_result($resultado_set, $contador, 2) ?></TD>
    </TR> <?php
}
// Con parte de los datos recuperados anteriormente, recupera la fecha y hora, y
los datos de la persona que realizo la medición de temperatura MINIMA
$sql = "SELECT TE.FECHA, TE.HORA, CO.APELLIDO1 || CO.APELLIDO2 || ', ' ||
CO.NOMBRE
FROM
TEMPERATURA TE, COLABORADOR CO
WHERE
TE.GRADOS = '". $G_MIN.'" AND
TE.ID_UBICACION = ".$ID_UBICACION." AND
CO.ID_COLABORADOR = TE.ID_COLABORADOR";
$resultado_set = pg_Exec ($conexion, $sql);
$MIN_resultado_rows = pg_NumRows($resultado_set);
for ($contador=0; $contador < $MIN_resultado_rows; $contador++) { ?>
<TR>
<TD><B>Mínima</B></TD>
<TD><?php ECHO $G_MIN; ?></TD>
<TD><?php ECHO pg_result($resultado_set, $contador, 0) ?></TD>
<TD><?php ECHO pg_result($resultado_set, $contador, 1) ?></TD>
<TD><?php ECHO pg_result($resultado_set, $contador, 2)?></TD>
</TR> <?php
}
pg_close($conexion); ?>
</TABLE>
<BR>
<form action="datos.php" method="post">
<input type="Submit" name="VolverAConsultar" value="Consultar otra ciudad"/>
</form>
<?php
} ?>
</body>
</html>

```

### 3. Consulta datos cliente/servidor (25%)

Una vez vista la calidad y cantidad de datos recogidos por el sistema de seguimiento de temperaturas desarrollado, el Centro Europeo de Estudio del Cambio Climático (CEECC) pide poder acceder a los mismos desde sus sistemas (desarrollados con tecnología cliente/servidor con Java), a fin de integrarlos en sus modelos de predicción.

Como el grid de supercomputación de dicho organismo utiliza una base de datos MySQL, se decide que el traspaso de datos entre los dos sistemas se realizará nocturnamente de forma asíncrona. Para ello se exportarán los datos de PostgreSQL en formato CSV, y posteriormente se importarán en el sistema del CEECC.

Se pide:

- a) Cargar los datos de los ficheros *20060112\_habitantes* y *20070114\_habitantes.txt* en la tabla NUM\_HABITANTES (en el PostgreSQL), utilizando si es preciso una tabla temporal.

*Creamos la tabla que utilizaremos para realizar la carga de habitantes.*

```
CREATE TABLE LOAD_HABITANTES (
  ID_LH INTEGER PRIMARY KEY,
  NOMBRE_CIUADAD VARCHAR(40),
  NUM_HABITANTES INT4
);
```

*Realizamos una carga del primer fichero:*

```
COPY LOAD_HABITANTES FROM 'c:/20060112_habitantes.txt' WITH DELIMITER '\t';
```

*Trasparamos el contenido a num\_habitantes:*

```
INSERT INTO NUM_HABITANTES
  SELECT ID_LH, '2006-01-12', NUM_HABITANTES
  FROM LOAD_HABITANTES;
```

*Borramos el contenido de la tabla temporal:*

```
DELETE FROM LOAD_HABITANTES;
```

*Repetimos el proceso para el segundo fichero:*

```
COPY LOAD_HABITANTES FROM 'c:/20070114_habitantes.txt' WITH DELIMITER '\t';
```

```
INSERT INTO NUM_HABITANTES
```

```
SELECT ID_LH, '2007-01-14', NUM_HABITANTES  
FROM LOAD_HABITANTES;
```

```
DELETE FROM LOAD_HABITANTES;
```

- b) Realizar una consulta SQL sobre el sistema 'Seguimiento Temperaturas' (PostgreSQL) que realice la exportación de datos, entre dos fecha indicadas, de las tablas:
- TEMPERATURAS
  - NUM\_HABITANTES
  - Y los datos completos de UBICACION relacionadas.

*Como se indica que se realizará la exportación de forma desatendida, se prepara la consulta para realizarse desde consola, a efectos de que se pueda programar su ejecución con un CRON (o similar):*

```
\COPY (  
  SELECT  
    NH.ID_UBICACION,  
    NH.FECHA,  
    NH.NUM_HABITANTES  
  FROM NUM_HABITANTES NH  
  WHERE  
    NH.FECHA BETWEEN '2006-12-01' AND '2007-12-31' AND  
    NH.ID_UBICACION IN  
    (SELECT TE.ID_UBICACION  
     FROM TEMPERATURA TE, NUM_HABITANTES NH  
     WHERE TE.ID_UBICACION = NH.ID_UBICACION);  
)  
TO 'C:/UNLOAD_H.TXT'
```

```
\COPY (  
  SELECT  
    TE.ID_UBICACION,  
    TE.FECHA, TE.HORA,  
    TE.ID_COLABORADOR,  
    TE.GRADOS  
  FROM TEMPERATURA TE  
  WHERE  
    TE.FECHA BETWEEN '2006-12-01' AND '2007-12-31' AND  
    TE.ID_UBICACION IN  
    (SELECT TE.ID_UBICACION  
     FROM  
       TEMPERATURA TE,  
       NUM_HABITANTES NH  
     WHERE  
       TE.ID_UBICACION = NH.ID_UBICACION AND  
       NH.FECHA BETWEEN '2006-12-01' AND '2007-12-31' AND  
       TE.FECHA BETWEEN '2006-12-01' AND '2007-12-31'  
     )  
)  
TO 'C:/UNLOAD_T.TXT'
```

```
\COPY (
  SELECT UB.ID_UBICACION, UB.NOMBRE, UB.UFI, UB.UNI, UB.LAT, UB.LONG
  FROM UBICACION UB
  WHERE UB.ID_UBICACION IN
  (SELECT TE.ID_UBICACION
  FROM
    TEMPERATURA TE,
    NUM_HABITANTES NH
  WHERE
    TE.ID_UBICACION = NH.ID_UBICACION AND
    NH.FECHA BETWEEN '2006-12-01' AND '2007-12-31' AND
    TE.FECHA BETWEEN '2006-12-01' AND '2007-12-31')
)
TO 'C:/UNLOAD_U.TXT'
```

- c) Cargar los datos exportados en el MySQL, creando para ello las tablas TEMPERATURAS, NUM\_HABITANTES y UBICACION.

*Crearemos las tablas en utilizando InnoDB:*

```
CREATE TABLE UBICACION (
  ID_UBICACION INT4 NOT NULL,
  NOMBRE VARCHAR(60),
  UFI CHAR(8) NOT NULL,
  UNI CHAR(8) NOT NULL,
  LAT DECIMAL(9,7) NOT NULL,
  LONGI DECIMAL(9,7) NOT NULL,
  PRIMARY KEY (ID_UBICACION),
  UNIQUE INDEX (UFI, UNI)
)
ENGINE=INNODB;
```

```
CREATE TABLE TEMPERATURA (
  ID_UBICACION INT4 NOT NULL,
  FECHA DATE NOT NULL,
  HORA TIME NOT NULL,
  ID_COLABORADOR SMALLINT NOT NULL,
  GRADOS DECIMAL(6,4) NOT NULL,
  PRIMARY KEY (ID_UBICACION, FECHA, HORA),
  FOREIGN KEY (ID_UBICACION) REFERENCES UBICACION (ID_UBICACION)
)
ENGINE=INNODB;
```

```
CREATE TABLE NUM_HABITANTES (
  ID_UBICACION INT4 NOT NULL,
  FECHA DATE NOT NULL,
  NUM_HABITANTES INT4 NOT NULL,
  PRIMARY KEY (ID_UBICACION, FECHA),
  FOREIGN KEY (ID_UBICACION) REFERENCES UBICACION (ID_UBICACION)
)
ENGINE=INNODB;
```

*Cargamos los ficheros de texto con los datos que hemos descargados del PostgreSQL, en las tabla correspondientes.*

```
LOAD DATA INFILE 'c:/unload_u.txt'
INTO TABLE UBICACION
FIELDS TERMINATED BY '\t'
LINES TERMINATED BY '\n';
```

```
LOAD DATA INFILE 'c:/unload_t.txt'
INTO TABLE TEMPERATURA
FIELDS TERMINATED BY '\t'
LINES TERMINATED BY '\n';
```

```
LOAD DATA INFILE 'c:/unload_h.txt'
INTO TABLE NUM_HABITANTES
FIELDS TERMINATED BY '\t'
LINES TERMINATED BY '\n';
```

- d) Realizar un programa en Java que accediendo a la base de datos MySQL por JDBC (según ejemplo facilitado), retorne por pantalla un listado con los siguiente datos tabulados (para las ciudades de las que disponemos de datos de habitantes y de temperatura):
- a. Nombre de ciudad.
  - b. Número máximo habitantes 2006
  - c. Número máximo habitantes 2007
  - d. Temperatura máxima 2006
  - e. Temperatura máxima 2007

*Empezamos componiendo la consulta que devuelve los datos pedidos:*

```
SELECT
  UB.NOMBRE,
  MAX(NH2006.NUM_HABITANTES),
  MAX(NH2007.NUM_HABITANTES),
  MAX(TE2006.GRADOS),
  MAX(TE2007.GRADOS)
FROM
  UBICACION UB,
  TEMPERATURA TE2006,
  TEMPERATURA TE2007,
  NUM_HABITANTES NH2006,
  NUM_HABITANTES NH2007
WHERE
  UB.ID_UBICACION = TE2006.ID_UBICACION AND
  UB.ID_UBICACION = TE2007.ID_UBICACION AND
  UB.ID_UBICACION = NH2006.ID_UBICACION AND
  UB.ID_UBICACION = NH2007.ID_UBICACION AND
  YEAR(TE2006.FECHA) = 2006 AND
  YEAR(TE2007.FECHA) = 2007 AND
  YEAR(NH2006.FECHA) = 2006 AND
  YEAR(NH2007.FECHA) = 2007 AND
  UB.NOMBRE IS NOT NULL GROUP BY UB.NOMBRE;
```

*Para luego introducirla en la codificación Java que se ha preparado:*

```
//
// Consulta de datos de ubicaciones con temperaturas y población por años.
// 20070510
//

import java.io.*;
import java.lang.*;
import java.sql.*;
public class pec4_3 {
public static void main(String[] args) throws Exception {
    String driver = "com.mysql.jdbc.Driver";
    String url = "jdbc:mysql://localhost:3306/test";
    String user = "root";
    String passwd = "1234";
    Class.forName(driver);
    Connection conexion = DriverManager.getConnection(url, user, passwd);

    // Lanzamos la consulta SQL
    Statement clima = conexion.createStatement();
    ResultSet rsclima = clima.executeQuery ("SELECT UB.NOMBRE,
        MAX(NH2006.NUM_HABITANTES), MAX(NH2007.NUM_HABITANTES), MAX(TE2006.GRADOS),
        MAX(TE2007.GRADOS) FROM UBICACION UB, TEMPERATURA TE2006, TEMPERATURA TE2007,
        NUM_HABITANTES NH2006, NUM_HABITANTES NH2007 WHERE UB.ID_UBICACION =
        TE2006.ID_UBICACION AND UB.ID_UBICACION = TE2007.ID_UBICACION AND UB.ID_UBICACION =
        NH2006.ID_UBICACION AND UB.ID_UBICACION = NH2007.ID_UBICACION AND
        YEAR(TE2006.FECHA) = 2006 AND YEAR(TE2007.FECHA) = 2007 AND YEAR(NH2006.FECHA) =
        2006 AND YEAR(NH2007.FECHA) = 2007 AND UB.NOMBRE IS NOT NULL GROUP BY UB.NOMBRE");

    // Recorremos los datos devueltos mostrándolos
    int num_registro = 1;
    System.out.println("Num_Reg Hab Max 2006 Hab Max 2007 T.Max06 T.Max07 Ciudad");
    while (rsclima.next()) {
        System.out.println( num_registro + "\t " + rsclima.getString(2) + " \t" +
rsclima.getString(
num_registro++;
    }

    // Cerramos la conexión a la base de datos.
    conexion.close();
    }
}
```

*Al ejecutar el programa obtenemos una salida similar a:*

```

C:\PEC4\CAS>java pec4_3
Num_Reg Hab Max 2006 Hab Max 2007 T.Max06 T.Max07 Ciudad
1 161895 174516 16.00 16.20 Albacete
2 138840 60176 18.00 18.80 Andujar
3 129837 135232 18.00 18.40 Basauri
4 618496 395018 23.00 23.20 Bilbao
5 65022 194452 34.00 34.60 Burgos
6 66833 105392 19.00 19.90 Caceres
7 482608 71021 15.19 15.79 Eibar
8 76334 82998 34.00 34.70 Ejido
9 201578 162871 16.00 16.90 El Ferrol
10 111390 434807 23.00 23.20 El Puerto de Santa Maria
11 379680 318223 13.00 13.90 Elche
12 186902 168434 18.00 18.90 Elda
13 81788 70669 25.00 25.70 Figueras
14 259158 296840 18.00 18.40 Gijon
15 68264 79417 16.00 16.20 Irun
16 89761 173105 17.00 17.40 Lleida
17 171406 187738 14.00 14.20 Logrono
18 66871 3711042 17.17 17.37 Madrid
19 477558 768570 23.00 23.70 Malaga
20 124202 67648 14.50 15.30 Merida
21 1246275 63527 18.00 18.90 Mieres
22 176522 151777 19.00 19.40 Mollet
23 189242 62476 28.65 28.95 Mondragon
24 125116 55347 23.00 23.60 Montilla
25 53198 454930 23.00 23.90 Mostoles
26 5178690 109485 19.00 19.90 Motril
27 55202 599668 24.00 24.90 Murcia
28 424738 144448 18.00 18.30 Ourense
29 40806 245022 12.00 12.20 Oviedo
30 87984 105708 15.19 15.79 Palencia
31 54578 328753 18.00 18.70 Pamplona
32 56258 119517 24.60 25.20 Sagunto
33 105133 212070 22.72 23.32 Salanaca
34 442698 337740 15.20 16.10 San Sebastian
35 57790 109379 33.70 34.10 Santa Lucia
36 60050 269627 12.60 12.90 Santander
37 101604 77015 30.60 31.40 Segovia
38 182066 1296624 17.80 18.70 Sevilla
39 147343 57985 30.80 31.70 Siero
40 82376 42454 13.75 13.95 Soria
41 138090 70813 22.32 23.02 Sueca
42 203836 96898 24.80 25.60 Talavera de la Reina
43 147730 129219 27.61 28.51 Torrelavega
44 167729 142098 20.11 20.51 Torrevecija
C:\PEC4\CAS>_

```

## Formato de entrega

El documento a entregar consistirá en un fichero con las siguientes características:

- Formato documento: ODT (preferente), PDF, DOC, RTF o SXW + ZIP o RAR con los programas PHP y java.
- Nombre del documento: BD\_PEC4\_Apellido1\_Nombre.extensión (p. ejemplo: BD\_PEC4\_Hernández\_Ramirez.odt)
- Asignatura, nombre y apellidos estudiante en la página inicial o portada.

El documento se entregará a través del apartado del Campus: Evaluación -> Actividades

La fecha de entrega máxima es el **viernes día 08/06/2007**.