

BASES DE DATOS

GP10 y 11

Servicios Telemáticos/Ingeniería de redes y servicios

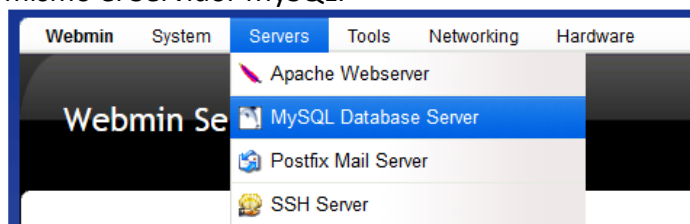
Nombre: Manuel Montoya Catalá

Nombre: Andrés Beato Ollero

Grupo:

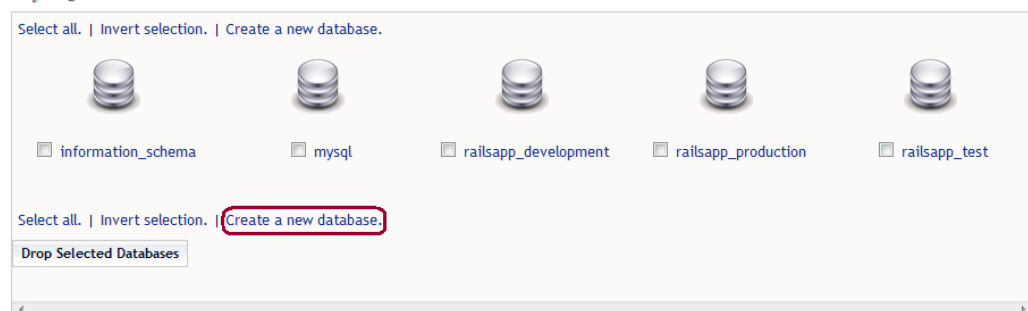
Apartado 1. Creación de la base de datos mediante Gestor web

Para la creación de la base de datos mediante el Gestor Web primero seleccionamos dentro del mismo el Servidor MySQL:



- Después le damos a crear una BBDD:

MySQL Databases



- Una vez dentro elegimos el nombre de la BBDD y le damos a crear:

New database options

Database name:

Character set:

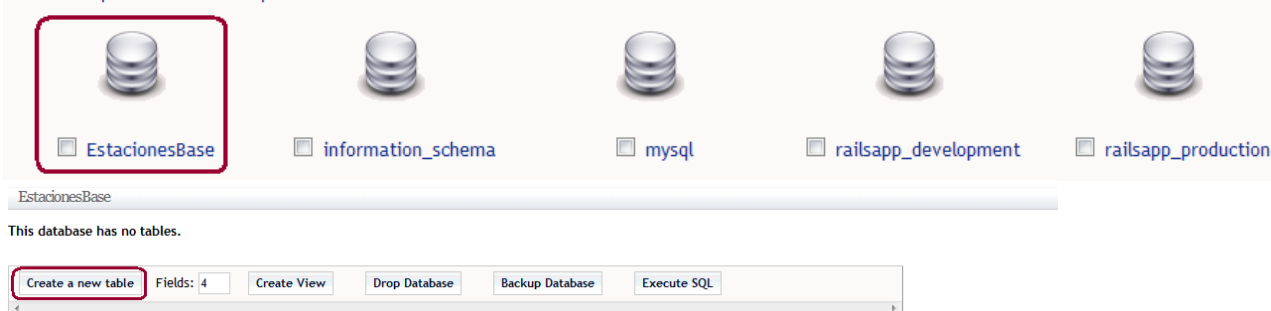
Collation order:

Initial table: ☒ None ☐ Named with fields below...

Field name	Data type	Type width	Key?	Auto-increment?	Allow nulls?	Unsigned?	Default value
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="checkbox"/> Yes <input type="checkbox"/> No	<input type="checkbox"/> Yes <input type="checkbox"/> No	<input checked="" type="checkbox"/> Yes <input type="checkbox"/> No	<input type="checkbox"/> Yes <input type="checkbox"/> No	<input type="text"/>
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="checkbox"/> Yes <input type="checkbox"/> No	<input type="checkbox"/> Yes <input type="checkbox"/> No	<input checked="" type="checkbox"/> Yes <input type="checkbox"/> No	<input type="checkbox"/> Yes <input type="checkbox"/> No	<input type="text"/>
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="checkbox"/> Yes <input type="checkbox"/> No	<input type="checkbox"/> Yes <input type="checkbox"/> No	<input checked="" type="checkbox"/> Yes <input type="checkbox"/> No	<input type="checkbox"/> Yes <input type="checkbox"/> No	<input type="text"/>
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="checkbox"/> Yes <input type="checkbox"/> No	<input type="checkbox"/> Yes <input type="checkbox"/> No	<input checked="" type="checkbox"/> Yes <input type="checkbox"/> No	<input type="checkbox"/> Yes <input type="checkbox"/> No	<input type="text"/>

- Para crear las tablas nos introducimos en dicha BBDD y le damos a create TABLE rellenando los campos según corresponda.

Select all. | Invert selection. | Create a new database.



- Rellenamos los campos:

New table options

Table name:

Copy fields from table:

Type:

Field name	Data type	Type width	Key?	Auto-increment?	Allow nulls?	Unsigned?	Default value
<input type="text" value="Country"/>	<input type="text" value="varchar"/>	<input type="text" value="20"/>	<input type="checkbox"/> Yes <input type="checkbox"/> No	<input type="checkbox"/> Yes <input type="checkbox"/> No	<input type="checkbox"/> Yes <input type="checkbox"/> No	<input type="checkbox"/> Yes <input type="checkbox"/> No	<input type="text"/>
<input type="text" value="City"/>	<input type="text" value="varchar"/>	<input type="text" value="20"/>	<input checked="" type="checkbox"/> Yes <input type="checkbox"/> No	<input type="checkbox"/> Yes <input type="checkbox"/> No	<input type="checkbox"/> Yes <input type="checkbox"/> No	<input type="checkbox"/> Yes <input type="checkbox"/> No	<input type="text"/>
<input type="text" value="Location"/>	<input type="text" value="varchar"/>	<input type="text" value="20"/>	<input type="checkbox"/> Yes <input type="checkbox"/> No	<input type="checkbox"/> Yes <input type="checkbox"/> No	<input checked="" type="checkbox"/> Yes <input type="checkbox"/> No	<input type="checkbox"/> Yes <input type="checkbox"/> No	<input type="text"/>
<input type="text" value="temperature"/>	<input type="text" value="int"/>	<input type="text" value="11"/>	<input type="checkbox"/> Yes <input type="checkbox"/> No	<input type="checkbox"/> Yes <input type="checkbox"/> No	<input checked="" type="checkbox"/> Yes <input type="checkbox"/> No	<input type="checkbox"/> Yes <input type="checkbox"/> No	<input type="text"/>

- Ya tenemos una tabla creada !

Apartado 2. Continuación del apartado anterior mediante consola

Para la utilización de la consola nos conectamos al puerto 12320 e iniciamos sesión como root. Después del login, ejecutamos el comando **mysql -u root -p** para entrar como el usuario root del servidor MySQL y acto seguido creamos la BBDD, las tablas y las inicializamos.

Aquí adjuntamos una captura de pantalla con los comandos y pasos:

```
root@rails ~# mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 56
Server version: 5.1.66-0+squeezel (Debian)

Copyright (c) 2000, 2012, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> CREATE DATABASE estaciones_base;
Query OK, 1 row affected (0.00 sec)

mysql> USE estaciones_base;
Database changed
mysql> CREATE TABLE estacion_base (
  -> country VARCHAR(20) NOT NULL,
  -> city VARCHAR(20) NOT NULL,
  -> location VARCHAR(20) ,
  -> temperatura INT ,
  -> PRIMARY KEY (city)
  -> );
Query OK, 0 rows affected (0.02 sec)

mysql> INSERT INTO estaciones_base VALUES ("Spain", "Alcala de Henares", NULL, 25);
ERROR 1146 (42S02): Table 'estaciones_base.estaciones_base' doesn't exist
mysql> INSERT INTO estacion_base VALUES ("Spain", "Alcala de Henares", NULL, 25);
Query OK, 1 row affected (0.00 sec)
```

Apartado 3. Inserción de datos.

Una vez implementado el esquema en la base de datos, se deben introducir los datos, para ello vamos a ver tres formas diferentes:

1) Desde la consola de MySQL:

Usando el comando

INSERT INTO *NombreTabla* **VALUES** (*valor1, valor2, valor3,...valorN*);

```
mysql> INSERT INTO estacion_base VALUES ("Spain", "Alcala de Henares", NULL, 25);
Query OK, 1 row affected (0.00 sec)
```

2) Generar un archivo de texto con la información en columnas separadas por un tabulador:

Para ello generamos la tabla con el formato indicado, el ejemplo creado es para la tabla "parámetro"

1	3	'temperatura'	'K'	5	1	100	100	100	100	50	50	50	50
2	2	'presion'	'atm'	12	10	10	10	10	10	5	5	5	5
3	1	'velocidad'	'km/h'	32	32	120	120	120	120	20	20	20	20
5	2	'lluvia'	'cm3'	43	12	300	300	300	300	50	50	50	50

- Para cargarla le damos a "Execute SQL" y después a la pestaña de Import Text file:

Select a text data file to import into MySQL database EstacionesBase ..
This file must contain one database record per line, with the fields in either tab separated or CSV format.

Text file import options

Text file source

☐ From local file
 ☒ From uploaded file
 init.txt

Table to import data into:

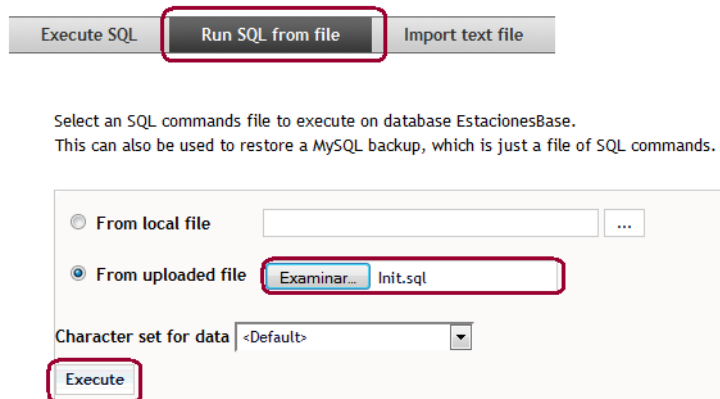
Delete data in table first? ☐ Yes ☒ No

Ignore duplicate rows? ☐ Yes ☒ No

File format ☐ CSV with quotes ☐ CSV without quotes ☒ Tab separated

Character set for data:

3) Generar un archivo de texto que contenga comandos SQL:
 Simplemente creamos un fichero con extensión .sql que contenga comandos SQL separados por punto y coma y lo cargamos en la web mediante:



Como ejemplo de código de inicialización tenemos:

```
INSERT INTO capturaImagen VALUES (1, 'Algo.jpg', 320, 240, 5, '20/6/2010', 1, 2);
INSERT INTO capturaImagen VALUES (2, 'Algo4.jpg', 320, 240, 5, '13/6/2001', 3, 4);
INSERT INTO capturaImagen VALUES (3, 'Algo6.jpg', 320, 240, 5, '12/1/2007', 4, 1);
INSERT INTO capturaImagen VALUES (4, 'Algo2.jpg', 320, 240, 5, '4/6/2010', 2, 3);
```

Apartado 4. Realización de búsquedas.

Para la realización de las búsquedas utilizamos la estructura SELECT y FROM, combinada mediante condiciones impuestas con WHERE para mostrar la información pedida.

1) Para que nos muestre solo el sensor y parámetro asociados a la imagen usamos las condiciones de que solo muestre aquellos con el id contenido en parametroId y sensorId de la imagen.

```
SELECT capturaImagen.id, imagen, capturaImagen.fecha, umbralUp1, umbralDown1, sensor.nombre
FROM capturaImagen, parametro, sensor, alarma
WHERE (parametroId = parametro.id) AND (capturaImagen.sensorId = sensor.id)
AND (alarma.sensorId = sensor.id);
```

2) Hemos considerado que para que se considere que un sensor produce una alarma, el campo sensor.num tiene que ser mayor que el campo alarma.criticidad por lo que basta con añadir dicha condición:

```
SELECT capturaImagen.id, imagen, capturaImagen.fecha, umbralUp1, umbralDown1, sensor.nombre
FROM capturaImagen, parametro, sensor, alarma
WHERE (parametroId = parametro.id) AND (capturaImagen.sensorId = sensor.id)
AND (alarma.sensorId = sensor.id) AND (alarma.criticidad < sensor.num);
```

Para demostrar su correcto funcionamiento mostramos diferentes tablas con que comprobamos la correcta asociación de las tablas:

id	umbralUp1	umbralDown1
1	100	50
2	10	5
3	120	20
5	300	50
4	145	50

id	nombre
1	Temperatura
2	Luz
3	Presion
4	Viento

id	sensorId	criticidad
4	1	32
1	3	43
2	4	65
3	2	76

id	imagen	fecha	parametroId	umbralUp1	umbralDown1	sensorId	nombre	num	criticidad
1	Algo.jpg	20/6/2010	2	10	5	1	Temperatura	35	32
2	Algo4.jpg	13/6/2001	4	145	50	3	Presion	1	43
3	Algo6.jpg	12/1/2007	1	100	50	4	Viento	70	65
4	Algo2.jpg	4/6/2010	3	120	20	2	Luz	3	76

id	imagen	fecha	umbralUp1	umbralDown1	nombre
1	Algo.jpg	20/6/2010	10	5	Temperatura
3	Algo6.jpg	12/1/2007	100	50	Viento

Apartado 5. Realización de Backup y restauración de BDs

En este apartado hemos ejecutado los comandos de BackUp en la Shell y obtenido el fichero correspondiente que adjuntamos en la memoria.

```

root@rails ~# mysqldump -u root -p BaseWay> fichero.sql
Enter password:
root@rails ~# ls
fichero.sql
root@rails ~# cat fichero.sql
-- MySQL dump 10.13 Distrib 5.1.66, for debian-linux-gnu (i486)
--
-- Host: localhost    Database: BaseWay
--
-- Server version      5.1.66-0+squeezel
--
/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
/*!40101 SET NAMES utf8 */;
/*!40103 SET @OLD_TIME_ZONE=@@TIME_ZONE */;
/*!40103 SET TIME_ZONE='+00:00' */;
/*!40014 SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0 */;
/*!40014 SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS, FOREIGN_KEY_CHECKS=0 */;
/*!40101 SET @OLD_SQL_MODE=@@SQL_MODE, SQL_MODE='NO_AUTO_VALUE_ON_ZERO' */;
/*!40111 SET @OLD_SQL_NOTES=@@SQL_NOTES, SQL_NOTES=0 */;

--
-- Table structure for table `alarma`
--
DROP TABLE IF EXISTS `alarma`;
/*!40101 SET @saved_cs_client      = @@character_set_client */;
/*!40101 SET character_set_client = utf8 */;
CREATE TABLE `alarma` (
  `id` int(11) NOT NULL,
  `sensorId` int(11) NOT NULL,
  `criticidad` int(11) NOT NULL,
  `tipo` varchar(20) NOT NULL,

```

Apartado 6. Acciones adicionales.

1) Creación de ficheros de comandos .sql para la creación, eliminación e inicialización directa de la BBDD pedida, estos ficheros se adjuntan en la memoria y son:

- Base.sql: Crea las tablas
- Delet.sql: Elimina las tablas
- Init.sql: Inicializa las tablas
- Reset.sql: Hace todas las acciones anteriores
- Selec.sql: Realiza las funciones de enseñar las tablas

2) Creación de una base de datos para almacenar los datos de las estaciones meteorológicas de nuestra página web:

```
CREATE TABLE estacionbase (
    country          VARCHAR (20)          NOT NULL,
    city             VARCHAR (20)          NOT NULL,
    location         VARCHAR (20)          ,
    temperatura      int                   NOT NULL,
    humedad          int                   NOT NULL,
    dir_viento       int                   NOT NULL,
    vel_viento       int                   NOT NULL,
    presion          int                   NOT NULL,
    lluvia           int                   NOT NULL,
    PRIMARY KEY (city)
);
```

3) Creación de un Script para inicializar la tabla con los datos de las estaciones que tenemos guardadas en formato XML de la entrega anterior.

- Ya que tenemos los datos de 20 estaciones meteorológicas en formato XML que pedimos mediante Ajax, hemos creado un script que en vez de transformarlas en tablas, lo que hace es exportarlas al formato de SQL para poder importarlas a la tabla.

El script se encuentra dentro del fichero **tablas2.js**

```
x = xmlhttp.responseXML.getElementsByTagName("tabla");
xx=x[0].getElementsByTagName("ciudad");

txt = "";
for (i=0; i < 1;i++){
    for ( j = 0; j < xx.length; j++){
        txt += "\n"
        txt = txt + "Spain" + "\t";
        xx=x[i].getElementsByTagName("ciudad");
        txt = txt+xx[j].firstChild.nodeValue + "\t";
        txt = txt + "NULL\t";
        //alert(xx[j].firstChild.nodeValue);

        xx=x[i].getElementsByTagName("temperatura");
        txt=txt + xx[j].firstChild.nodeValue+ "\t";

        xx=x[i].getElementsByTagName("humedad");
        txt=txt + xx[j].firstChild.nodeValue+ "\t";
```

```
xx=x[i].getElementsByTagName("dir_viento");
txt=txt +xx[j].firstChild.nodeValue+ "\t";

xx=x[i].getElementsByTagName("vel_viento");
txt=txt + xx[j].firstChild.nodeValue+ "\t";

xx=x[i].getElementsByTagName("lluvia");
txt=txt + xx[j].firstChild.nodeValue;

}
}
alert(txt);
```

Después las cargamos a la base de datos mediante LOAD.

4) Creación de un formulario para la búsqueda de una determinada estación base en la BBDD a partir de su localización y/o condiciones atmosféricas.

El formulario se encuentra en la misma página que las tablas.

El código para interconexión entre la BBDD y el formulario se deja para la siguiente práctica.