# BASES DE DATOS

## ACCESO A LA TERMINAL

Para acceder a su bash : docker exec -ti mysql /bin/bash

Para entrar en mysql mysql -u root -p

Para ver las bbdd existentes : show databases;

Para usar una bbdd : use nombrebbdd;

Crear una bbdd : CREATE DATABASE IF NOT EXISTS nombrebbdd

Crear tabla : use nombrebbdd ; CREATE TABLE IF NO EXISTS nTabla

## CONEXIÓN mediante archivo config.php.

```
require clases/RepositorioMYSQL;
    $host = "mysql"; // El nombre del servicio en el archivo YML
    $usuario = "root"; // Usuario por defecto
    $contraseña = "1234"; // Contraseña definida en .yml
    $base_datos = "emergencia"; // según la bbdd que uses
    $repositorio = new RepositorioMYSQL($host, $usuario,
    $contraseña, $base_datos); // repo que utilizaremos en este ejercicio
```

Este archivo lo configuraremos y guardaremos dentro del proyecto asi evitamos configurar la conexión en cada clase y proteger los datos.

La configuracion de la conn en la clase Repositorio sigue siendo la misma

CONEXIÓN mediante instancia en clase y controlador.

**En la clase Repositorio lo creamos:**

```
    private $conn;
// su constructor
    public function __construct($host, $usuario, $pass, $db) {
// Configura la conexión a la base de datos
        $this->conn = new PDO("mysql:host=$host;dbname=$db",
        $usuario,   $pass);
// Configurar el manejo de errores
        $this->conn->setAttribute(PDO::ATTR_ERRMODE,
        PDO::ERRMODE_EXCEPTION);
    }
```

**En el controlador lo instanciamos con sus credenciales:**

```
    require 'RepositorioMYSQL.php';
// Configuración de la base de datos
    $repositorio = new RepositorioMYSQL("mysql", "root", "1234",
    "emergencia");
```

Ahora podremos hacer uso de todo el repo de RepositorioMYSQL

CONSULTAS

Ver :

```
"SELECT fecha_reporte
 FROM estado_municipios
 WHERE nombre_poblacion = :nombre_poblacion
 ORDER BY fecha_reporte
 DESC LIMIT 1";
```

**Actualizar:**

```
"UPDATE estado_municipios
 SET personas_afectadas = :personas_afectadas,
        comunicaciones_cortadas = :comunicaciones_cortadas,
 WHERE nombre_poblacion = :nombre_poblacion";
```

**Insertar:**

```
"INSERT INTO estado_municipios (nombre_poblacion,
personas_afectadas, comunicaciones_cortadas, agua, productos_limpieza,
viveres, medicinas, otros)
 VALUES(:nombre_poblacion, :personas_afectadas,
:comunicaciones_cortadas, :agua, :productos_limpieza, :viveres,
 :medicinas, :otros)";
```

**Eliminar:**

```
"DELETE FROM estado_municipios
 WHERE nombre_poblacion = :nombre_poblacion";
```

**Operaciones en consultas**

COUNT (Contar registros)

```
"SELECT COUNT(*)  AS total
 FROM estado_municipios
 WHERE nombre_poblacion = :nombre_poblacion";
```

SUM (Suma de valores numéricos)

```
"SELECT SUM(personas_afectadas) AS suma
 FROM estado_municipios “;
```

AVG (Promedio de valores numéricos)

```
"SELECT AVG(personas_afectadas) AS media
 FROM estado_municipios “;
```

MIN y MAX (Valor mínimo y máximo)

```
"SELECT MIN(fecha_reporte) AS minimo,
        MAX(fecha_reporte) AS maximo
 FROM estado_municipios “;
```

**GROUP BY** (Agrupamiento de resultados)

Se utiliza cuando necesitas agrupar los resultados por una columna, como contar el número de registros por población o sumar los afectados por cada población.

```
"SELECT nombre_poblacion, COUNT(*) FROM estado_municipios
GROUP BY nombre_poblacion";
```

**HAVING** (Filtrar después de GROUP BY) filtrar después GROUP BY

```
"SELECT nombre_poblacion, SUM(personas_afectadas)
 FROM estado_municipios
 GROUP BY nombre_poblacion
 HAVING SUM(personas_afectadas) > 1000";
```

**ORDER BY** (Ordenar resultados) ASC (ascendente) o DESC (descendente).

```
"SELECT nombre_poblacion, fecha_reporte
 FROM estado_municipios
 WHERE nombre_poblacion = :nombre_poblacion
 ORDER BY fecha_reporte DESC";
```

**LIMIT** (Limitar el número de resultados)

```
"SELECT *
 FROM estado_municipios
 ORDER BY fecha_reporte DESC
 LIMIT 10";
```

FUNCION CON CONSULTA

Primero realizaremos la funcion en el repositorio y luego la instanciaremos

**Con argumentos:**

```
public function obtenerHabitantesMunicipio($nombreMunicipio){
        $consulta = "SELECT habitantes
                        FROM estado_municipio
                        WHERE nombre_poblacion =:nombreMunicipio ";


// una vez parametrizada la consulta la preparamos
        $rC = $this→conn→prepare($consulta);
// ahora vinculamos el parametro con su igual en la bbdd
        $rC → bindParam(':nombre_poblacion' , $nombreMunicipio);
// ejecutamos la consulta para poder trabajar con su devolucion
        $rC→execute();
// recuperamos el resultado y lo asignamos
        $resultado = $rC→fetchAll(PDO::FETCH_ASSOC);
// asignamos valor de retorno a la funcion (por si acaso indexa siempre)
        return $resultado['habitantes'];
        // en este caso podemos manejar tipo de return según que devuelva
                if(!empty($resultado['habitantes'])){
                        return $resultado['habitantes'];
                }else{
                        return null ;
                }
// instanciarla desde controlador
$habitantes=$repositorio → obtenerHabitantesM...($nombreMnicipio);
```

**Sin argumentos procesando su resultado:**

```
    public function obtenerInfoMunicipios(){
        $consulta = " SELECT *
                    FROM estado_municipio ";
        $rC = $this→conn→prepare($consulta);
        $rC→execute();
        $resultado = $rC→fetchAll(PDO::FETCH_ASSOC);
        if(!empty($resultado)){
            return $resultado;// array entero que trataremos en control
        }else{
            return null;
        }
// vamos a instanciarla y procesar su resultado
    $municipios=$repositorio →obtenerInfoMunicipios();
// como devuelve una colección de objetos tenemos que iterar
    foreach($municipios as $municipio){
// sacamos los valores de sus indices por si los necesitamos para algo
        $nombre = $municipio['nombre_poblacion']?? "desconocido";
        $habitantes = $municipio['poblacion'];
        echo "Informacion del Municipio : " .$nombre;
// podemos poner algun if o directamente imprimir
        foreach($municipio as $campo => $valor){
        echo $campo. " : " .$valor."<br>";
        }
        echo "<hr>";
    }
```