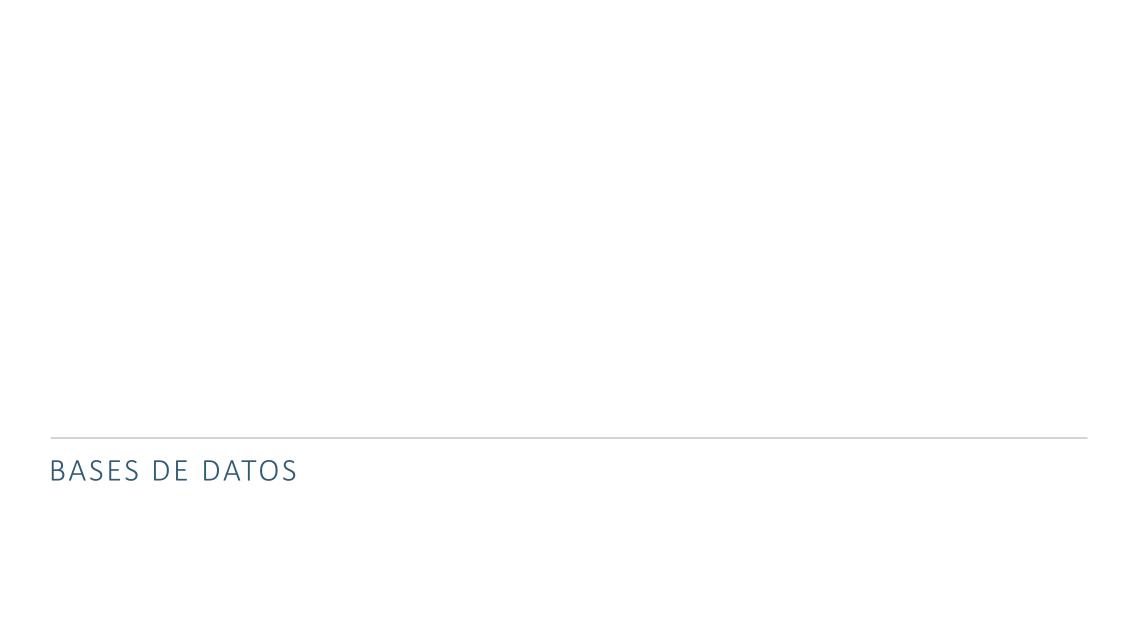
UD3. Desarrollo de Aplicaciones Web con PHP

DESARROLLO WEB EN ENTORNO SERVIDOR



Bases de datos relacionales

¿Cómo accedemos a una base de datos? Con una librería.

- La librería PHP Data Objects (PDO) permite manejar diferentes bases de datos con una interfaz común.
- Se ocupa de las diferencias entre SGBD.
- Si se cambia de base de datos, no hay que modificar el código.

bd_conect.php

PDO: conexión

```
<?php
$cadena conexion = 'mysql:dbname=empresa;host=127.0.0.1';
$usuario = 'root';
                                                             Cadena de conexión
$clave = '';
try {
  $bd = new PDO($cadena conexion, $usuario, $clave);
  $bd->close();
} catch (PDOException $e) {
  echo 'Error con la base de datos: ' . $e->getMessage();
```

PDO: consultas

bd_listar_usuario.php
bd_insert_update_delete.php

```
<?php
$cadena conexion = 'mysql:dbname=empresa;host=127.0.0.1';
$usuario = 'root';
$clave = '';
try {
 $bd = new PDO($cadena conexion, $usuario, $clave);
 echo "Conexión realizada con éxito";
 $sql = 'SELECT nombre, clave, rol FROM usuarios';
 $usuarios = $bd->query($sql); 
 echo $usuarios->rowCount() . "<br>";
 foreach ($usuarios as $row) {
   print $row['nombre'] . "\t";
   print $row['clave'] . "\t";
```

Devuelve false si hay error y un objeto PDOStatement en caso contrario

PDO: instrucciones preparadas

Se inicializan una sola vez con prepare ().

Se ejecutan después tantas veces como se necesite con execute (params).

Dos opciones para pasar parámetros:

- 1. Por posición. Se utiliza el símbolo ? para denotarlos en la query. Se van asociando en base a un array que se pasa a execute.
- 2. Por nombre. Se utiliza : nombre para denotarlos. En el array deben aparecer como claves.

PDO: transacciones

Las transacciones son un conjunto de operaciones que deben hacerse de manera atómica.

Si falla alguna operación de la transacción, hay que deshacer todas.

Para hacer una transacción:

- 1. Se señaliza el comienzo con bd->beginTransaction()
- 2. Se salva con bd->commit()
- 3. Se deshace con bd->rollback()