NOMBRE: Quero Gómez, Juan Manuel

Documentación.

Crear base de datos y esqueleto del proyecto.

Crear base de datos.

```
MariaDB [(none)]> create database tienda;
Query OK, 1 row affected (0.001 sec)
```

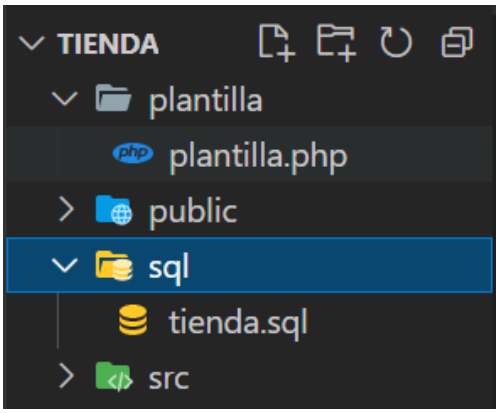
Crear tablas de la base de datos.

```
MariaDB [tienda]> drop table if exists articulos;
Query OK, 0 rows affected, 1 warning (0.000 sec)

MariaDB [tienda]> drop table if exists categorias;
Query OK, 0 rows affected, 1 warning (0.000 sec)

MariaDB [tienda]>
MariaDB [tienda]> -- tabla categorias
MariaDB [tienda]> create table categorias(
    ->     id int AUTO_INCREMENT primary key,
    ->     nombre VARCHAR(100) unique not null,
    ->     descripcion text not null
    -> );
Query OK, 0 rows affected (0.029 sec)

MariaDB [tienda]> -- tablas articulos
MariaDB [tienda]> create table articulos(
    ->     id int AUTO_INCREMENT primary key,
    ->     nombre varchar(100) not null,
    ->     precio float(5,2),
    ->     categoria_id int,
    ->     constraint fk_art_cat foreign key(categoria_id) references categorias(id) on delete cascade on update cascade
    -> );
Query OK, 0 rows affected (0.028 sec)
```

Dar permisos al usuario.

```
MariaDB [tienda]> grant all on tienda.* to usuario@'localhost';
Query OK, 0 rows affected (0.010 sec)
```

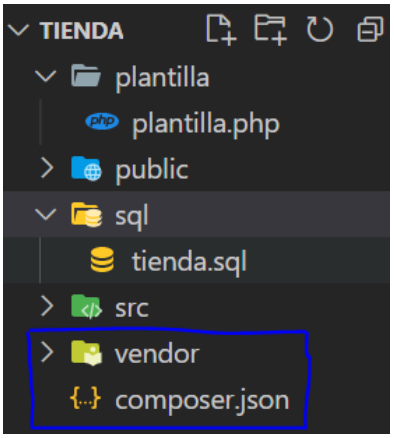
Crear esqueleto del proyecto.



Instalación de composer en el proyecto. Abrir terminal en Visual Studio Code.

```
PS C:\xampp\htdocs\proyectos\tema4\practicas\tienda> composer init

  Welcome to the Composer config generator



This command will guide you through creating your composer.json config.

Package name (<vendor>/<name>) [juanm/tienda]:
Description []: Practica PDO
Author [, n to skip]: juanma <juan@correo.es>
Minimum Stability []:
Package Type (e.g. library, project, metapackage, composer-plugin) []: project
License []: GNU
```

Se creará en el proyecto una carpeta con el nombre **“vendor”** y un archivo .json **“composer.json”.**



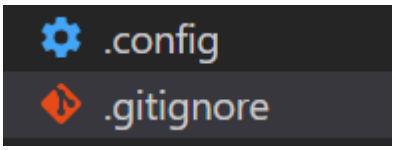
Instalación de la librería “Faker” mediante composer.

```
PS C:\xampp\htdocs\proyectos\tema4\practicas\tienda> composer require fakerphp/faker
Using version ^1.16 for fakerphp/faker
./composer.json has been updated
Running composer update fakerphp/faker
Loading composer repositories with package information
Updating dependencies
Lock file operations: 3 installs, 0 updates, 0 removals
  - Locking fakerphp/faker (v1.16.0)
  - Locking psr/container (2.0.1)
  - Locking symfony/deprecation-contracts (v2.4.0)
Writing lock file
```

En el archivo **“composer.json”** habrá que realizar un cambio para el uso de **“autoload”.**

```
"autoload": {
    "psr-4": {
        "Tienda\\": "src/"
    }
```

Una vez realizado lo anterior, toca crear un archivo con los parámetros para conectar con la base de datos con el nombre **“.config”** y otro archivo con el nombre **“.gitignore”** el cual permitirá el no subir los archivos que pongamos en él a GitHub.



También se puede crear otro archivo como guía para indicar a otros usuarios que tienen que poner en el archivo **“.config”**. A este nuevo archivo tendrá el nombre de **“config.example”.**

config.example

```
1   ;renombre este archivo como .config
2   [opciones]
3   user = tu_nombre_de_usuario
4   pass = tu_contraseña_para_el_usuario
5   host = tu_host
6   base = tu_base_de_datos
```

Crear la clase Conexión.

Una vez montado la base del proyecto, toca crear la clase Conexión que establecerá conexión la base de datos creada. Creamos un nuevo fichero en la carpeta **“src”**, con el nombre **“Conexión.php”.**

```php
<?php
namespace Tienda;

use PDO;
use PDOException;

class Conexion {
    protected static $conexion;

    //Creamos el constructor
    public function __construct()
```

```
    {
        if(self::$conexion == null) {
            self::crearConexion();
        }
    }

    //Creamos la conexion a la base de datos
    public static function crearConexion() {
        $fichero = dirname(__DIR__, 1)."/.config";
        $opciones = parse_ini_file($fichero);
        $usuario = $opciones['user'];
        $pass = $opciones['pass'];
        $base = $opciones['base'];
        $host = $opciones['host'];

        $dns = "mysql:host=$host;dbname=$base;charset=utf8mb4";

        try {
            self::$conexion = new PDO($dns,$usuario, $pass);
            self::$conexion->setAttribute(PDO::ATTR_ERRMODE,
PDO::ERRMODE_EXCEPTION);
        }catch(PDOException $ex) {
            die("Error al conectar a la base de datos:
".$ex->getMessage());
        }
    }
}
```

Una vez creada la clase **“Conexion.php”**, toca crear la clase **“Categoria.php”** la cual hereda de la clase **“Conexion.php”**.

```
<?php
namespace Tienda;

class Categorias extends Conexion {
    private $id;
    private $nombre;
    private $descripcion;

    //Creamos el constructor que heredara de la clase padre
    public function __construct()
    {
```

```
        parent::__construct();
    }

    //### CRUD ###
    public function create() {

    }
    public function read() {

    }
    public function update() {

    }
    public function delete() {

    }

    //### OTROS MÉTODOS ###

    //### GETTERS AND SETTER ###
    /**
     * Get the value of id
     */
    public function getId()
    {
        return $this->id;
    }

    /**
     * Set the value of id
     *
     * @return  self
     */
    public function setId($id)
    {
        $this->id = $id;

        return $this;
    }

    /**
     * Get the value of nombre
     */
```

```php
    public function getNombre()
    {
        return $this->nombre;
    }

    /**
     * Set the value of nombre
     *
     * @return  self
     */
    public function setNombre($nombre)
    {
        $this->nombre = $nombre;

        return $this;
    }

    /**
     * Get the value of descripcion
     */
    public function getDescripcion()
    {
        return $this->descripcion;
    }

    /**
     * Set the value of descripcion
     *
     * @return  self
     */
    public function setDescripcion($descripcion)
    {
        $this->descripcion = $descripcion;

        return $this;
    }
}
```

Creamos el método **create()** en la clase **“Categorias.php”** y un método nuevo al que llamaremos **generarCategorias()** para que nos genere categorías haciendo uso de la librería **Faker** y otro método más para comprobar si ya hay categorías creadas para que no nos genere otras nueva a la que llamaremos **hayCategorias().**

Método create()

```
public function create() {
        $q = "insert into categorias(nombre, descripcion) values(:n,
:d)";
        $stmt = parent::$conexion->prepare($q);
        try {
            $stmt->execute([
                ':n'=>$this->nombre,
                ':d'=>$this->descripcion
            ]);
        }catch(PDOException $ex) {
            die("Error al crear la categoria: ".$ex->getMessage());
        }
        //Cerramos la conexion a la base de datos
        parent::$conexion = null;
    }
```

Método hayCategorias()

```
public function hayCategorias() {
        $q = "select * from categorias";
        $stmt = parent::$conexion->prepare($q);
        try {
            $stmt->execute();
        }catch(PDOException $ex) {
            die("Error al comprobar si ya hay categorias:
".$ex->getMessage());
        }
        //Cerramos la conexion a la base de datos y devolvemos el total
de categorias
        parent::$conexion = null;
        return $stmt->rowCount();
    }
```

Método generarCategorias()

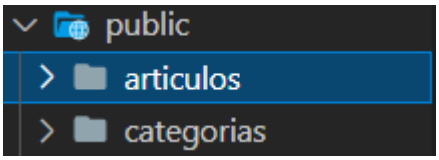
```
public function generarCategorias($cantidad) {
        if($this->hayCategorias() == 0) {
            $faker = Faker\Factory::create('es_ES');
            for($i=0; $i<$cantidad; $i++) {
                $nombre = $faker->word;
                $descripcion = $faker->text(50);
                (new Categorias)->setNombre($nombre)
                ->setDescripcion($descripcion)
                ->create();
            }
        }
    }
```

Creamos un archivo en la carpeta **“public”** con el nombre **“index.php”** que contendrá dos botones para dirigirnos a ambas tablas.

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width,
initial-scale=1.0">
    <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.m
in.css" rel="stylesheet"
integrity="sha384-1BmE4kWBq78iYhFldvKuhfTAU6auU8tT94WrHftjDbrCEXSU1oBoq
yl2QvZ6jIW3" crossorigin="anonymous">
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.0.0-beta2/c
ss/all.min.css"
integrity="sha512-YWzhKL2whUzgiheMoBFwW8CKV4qpHQAEuvilg9FAn5VJUDwKZZxkJ
NuGM4XkWuk94WCrrwslk8yWNGmY1EduTA==" crossorigin="anonymous"
referrerpolicy="no-referrer" />
    <title>Inicio</title>
</head>
<body>
    <h3 class="text-center mt-3">Inicio</h3>
    <div class="container text-center mt-4">
```

```
        <a href="categorias/index.php" class="btn btn-primary"><i
class="fas fa-tag"></i> Categorías</a>
        <a href="articulos/index.php" class="btn btn-danger"><i
class="fas fa-shopping-bag"></i> Artículos</a>
    </div>
</body>
</html>
```

Además, la carpeta **"public"** tendrá dos carpetas más, una con el nombre **"categorias"** y otra **"articulos".**



Dentro de la carpeta **"categorias"** creamos un fichero con el nombre **"index.php"** donde vamos a mostrar una tabla con todas las categorías creadas. Además, para poder mostrar las categorías que hemos generado hay que crear un método nuevo al que llamaremos **readAll()** en la clase **"Categorias.php".**

Método readAll()

```
public function readAll()
    {
        $q = "select * from categorias order by nombre";
        $stmt = parent::$conexion->prepare($q);
        try {
            $stmt->execute();
        } catch (PDOException $ex) {
            die("Error al mostrar los datos de las categorías: " .
$ex->getMessage());
        }
        //Cerramos la conexion a la base de datos y devolvemos los
datos que necesitamos
        parent::$conexion = null;
        return $stmt;
    }
```

index.php de categorías

```
<?php
session_start();
require dirname(__DIR__, 2) . "/vendor/autoload.php";
```

```php
use Tienda\Categorias;

(new Categorias)->generarCategorias(10);
$datos = (new Categorias)->readAll();
?>
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width,
initial-scale=1.0">
    <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.m
in.css" rel="stylesheet"
integrity="sha384-1BmE4kWBq78iYhFldvKuhfTAU6auU8tT94WrHftjDbrCEXSU1oBoq
yl2QvZ6jIW3" crossorigin="anonymous">
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.0.0-beta2/c
ss/all.min.css"
integrity="sha512-YWzhKL2whUzgiheMoBFwW8CKV4qpHQAEuvilg9FAn5VJUDwKZZxkJ
NuGM4XkWuk94WCrrwslk8yWNGmY1EduTA==" crossorigin="anonymous"
referrerpolicy="no-referrer" />
    <title>Categorías</title>
</head>

<body>
    <h3 class="text-center mt-3">Categorías</h3>
    <div class="container mt-4">
      <?php
        if (isset($_SESSION['mensaje'])) {
            echo <<<TXT
                <div class="alert alert-primary" role="alert">
                    {$_SESSION['mensaje']}
                </div>
            TXT;
            unset($_SESSION['mensaje']);
        }
        ?>
```

```
        <a href="addCat.php" class="btn btn-success"><i class="fas
fa-plus-circle"></i> Nueva categoría</a>
        <table class="table table-striped">
            <thead>
                <tr>
                    <th scope="col">ID</th>
                    <th scope="col">Nombre</th>
                    <th scope="col">Descripción</th>
                    <th scope="col">Acciones</th>
                </tr>
            </thead>
            <tbody>
                <?php
                while ($fila = $datos->fetch(PDO::FETCH_OBJ)) {
                    echo <<<TXT
                    <tr>
                        <th scope="row">{$fila->id}</th>
                        <td>{$fila->nombre}</td>
                        <td>{$fila->descripcion}</td>
                        <td>
                            <form name='a' action='deleteCat.php'
method='POST'>
                                <input type='hidden' name='id'
value='{$fila->id}'>
                                <a href='updateCat.php?id={$fila->id}'
class='btn btn-primary'><i class='fas fa-edit'></i> Editar</a>
                                <button type='submit' class='btn
btn-danger' onclick='return confirm("¿Desea borrar la categoría?")'><i
class='fas fa-trash'></i> Borrar </button>
                            </form>
                        </td>
                    </tr>
                    TXT;
                }
                ?>
            </tbody>
        </table>
    </div>
</body>

</html>
```

Creamos otro fichero nuevo con el nombre **“addCat.php”** el cual tendrá un formulario para crear una categoría nueva.

```php
<?php
session_start();
require dirname(__DIR__, 2) . "/vendor/autoload.php";
use Tienda\Categorias;

//Creamos una función para comprobar si los campos están vacios
function campoVacio($n, $d) {
    $error = false;
    if(strlen($n) == 0) {
        $_SESSION['vacio_nombre'] = "Rellene el campo nombre";
        $error = true;
    }

    if(strlen($d) == 0) {
        $_SESSION['vacio_descripcion'] = "Rellene el campo
descripción";
        $error = true;
    }
    return $error;
}

//Creamos otra función para comprobar que el campo nombre tiene que
tener al menos mas de 3 carácteres
function comprobarCampo($n) {
    $error = false;
    if(strlen($n) < 3) {
        $_SESSION['corto_nombre'] = "El campo nombre tiene que tener al
menos 3 carácteres";
        $error = true;
    }
    return $error;
}


if(isset($_POST['Crear'])) {
    $nombre = trim(ucfirst($_POST['nombre']));
    $descripcion = trim(ucfirst($_POST['descripcion']));
    if(!campoVacio($nombre, $descripcion) || !comprobarCampo($nombre))
{
        (new Categorias)->setNombre($nombre)
```

```
        ->setDescripcion($descripcion)
        ->create();
        $_SESSION['mensaje'] = "Categoría creada correctamente";
        header("Location:index.php");
        die();
    }
    header("Location:{$_SERVER['PHP_SELF']}");
} else {

?>
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width,
initial-scale=1.0">
    <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.m
in.css" rel="stylesheet"
integrity="sha384-1BmE4kWBq78iYhFldvKuhfTAU6auU8tT94WrHftjDbrCEXSU1oBoq
yl2QvZ6jIW3" crossorigin="anonymous">
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.0.0-beta2/c
ss/all.min.css"
integrity="sha512-YWzhKL2whUzgiheMoBFwW8CKV4qpHQAEuvilg9FAn5VJUDwKZZxkJ
NuGM4XkWuk94WCrrwslk8yWNGmY1EduTA==" crossorigin="anonymous"
referrerpolicy="no-referrer" />
    <title>Nuevo</title>
</head>

<body>
    <h3 class="text-center mt-3">Nueva Categoría</h3>
    <div class="container mt-4">
        <form action="<?php echo $_SERVER['PHP_SELF']; ?>" name="a"
method="POST">
            <div class="bg-info p-4 text-white rounded shadow-lg
m-auto" style="width: 40rem;">
                <div class="mb-3">
                    <label for="n" class="form-label">Nombre
Categoría</label>
```

```
                    <input type="text" class="form-control" id="n"
placeholder="Nombre" name="nombre" required>
                    <?php
                        if(isset($_SESSION['vacio_nombre'])) {
                            echo "<div class='text-danger
mt-2'>{$_SESSION['vacio_nombre']}</div>";
                            unset($_SESSION['vacio_nombre']);
                        }
                        if(isset($_SESSION['corto_nombre'])) {
                            echo "<div class='text-danger
mt-2'>{$_SESSION['corto_nombre']}</div>";
                            unset($_SESSION['corto_nombre']);
                        }
                    ?>
                </div>
                <div class="mb-3">
                    <label for="d" class="form-label">Descripción
Categoría</label>
                    <input type="text" class="form-control" id="d"
name="descripcion" placeholder="Descripción">
                    <?php
                        if(isset($_SESSION['vacio_descripcion'])) {
                            echo "<div class='text-danger
mt-2'>{$_SESSION['vacio_descripcion']}</div>";
                            unset($_SESSION['vacio_descripcion']);
                        }
                    ?>
                </div>
                <div class="mb-3">
                    <button type="submit" class="btn btn-primary"
name="Crear"><i class="fas fa-check-circle"></i> Crear</button>
                    <button type="reset" class="btn btn-danger"><i
class="fas fa-broom"></i> Limpiar</button>
                </div>
            </div>
        </form>
    </div>
</body>

</html>
<?php } ?>
```

Creamos el método **delete()** en la clase **“Categorias.php”** y un fichero en la carpeta **“categorias”** al que llamaremos **“deleteCat.php”.**

Método delete()

```php
public function delete($id)
    {
        $q = "delete from categorias where id=:i";
        $stmt = parent::$conexion->prepare($q);
        try {
            $stmt->execute([
                ':i'=>$id
            ]);
        }catch(PDOException $ex) {
            die("Error al borrar la categoría: ".$ex->getMessage());
        }
        //Cerramos la conexion con la base de datos
        parent::$conexion = null;
    }
```

deleteCat.php de categorías

```php
<?php
session_start();
require dirname(__DIR__, 2)."/vendor/autoload.php";
use Tienda\Categorias;

//Comprobar si existe el id que le pasamos
if(!isset($_POST['id'])) {
    header("Location_index.php");
    die();
}

(new Categorias)->delete($_POST['id']);
$_SESSION['mensaje'] = "Categoría borrada correctamente";
header("Location:index.php");
```

Por último, nos queda crear los métodos **read()** para mostrar los datos de la categoría que seleccionemos y el método **update()** para actualizar los datos de la categoría que vayamos actualizar y también crearemos el fichero **"updateCat.php"** en la carpeta **"categorias".**

Método read()

```
public function read($id)
    {
        $q = "select * from categorias where id=:i";
        $stmt = parent::$conexion->prepare($q);
        try {
            $stmt->execute([
                ':i'=>$id
            ]);
        }catch(PDOException $ex) {
            die("Error al devolver los datos de la categoría:
".$ex->getMessage());
        }
        //Cerramos la conexion a la base de datos y devolvemos los
datos que queremos mostrar
        parent::$conexion = null;
        return $stmt->fetch(PDO::FETCH_OBJ);
    }
```

Método update()

```
public function update()
    {
        $q = "update categorias set nombre=:n, descripcion=:d where
id=:i";
        $stmt = parent::$conexion->prepare($q);
        try {
            $stmt->execute([
                ':n'=>$this->nombre,
                ':d'=>$this->descripcion,
                ':i'=>$this->id
            ]);
        }catch(PDOException $ex) {
            die("Error al actualizar la categoría:
".$ex->getMessage());
        }
        //Cerramos la conexion a la base de datos
        parent::$conexion = null;
    }
```

updateCat.php de categorías

```php
<?php
if(!isset($_GET['id'])) {
    header("Location:index.php");
    die();
}
session_start();
require dirname(__DIR__, 2) . "/vendor/autoload.php";
use Tienda\Categorias;

$id = $_GET['id'];
$datosCategoria = (new Categorias)->read($id);
//Creamos una función para comprobar si los campos están vacios
function campoVacio($n, $d) {
    $error = false;
    if(strlen($n) == 0) {
        $_SESSION['vacio_nombre'] = "Rellene el campo nombre";
        $error = true;
    }

    if(strlen($d) == 0) {
        $_SESSION['vacio_descripcion'] = "Rellene el campo
descripción";
        $error = true;
    }
    return $error;
}

//Creamos otra función para comprobar que el campo nombre tiene que
tener al menos mas de 3 carácteres
function comprobarCampo($n) {
    $error = false;
    if(strlen($n) < 3) {
        $_SESSION['corto_nombre'] = "El campo nombre tiene que tener al
menos 3 carácteres";
        $error = true;
    }
    return $error;
}


if(isset($_POST['Actualizar'])) {
    $nombre = trim(ucfirst($_POST['nombre']));
```

```
    $descripcion = trim(ucfirst($_POST['descripcion']));
    if(!campoVacio($nombre, $descripcion) || !comprobarCampo($nombre))
{
        (new Categorias)->setNombre($nombre)
        ->setDescripcion($descripcion)
        ->setId($id)
        ->update();
        $_SESSION['mensaje'] = "Categoría actualizada correctamente";
        header("Location:index.php");
        die();
    }
    header("Location:{$_SERVER['PHP_SELF']}?id=$id");
} else {

?>
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width,
initial-scale=1.0">
    <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.m
in.css" rel="stylesheet"
integrity="sha384-1BmE4kWBq78iYhFldvKuhfTAU6auU8tT94WrHftjDbrCEXSU1oBoq
yl2QvZ6jIW3" crossorigin="anonymous">
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.0.0-beta2/c
ss/all.min.css"
integrity="sha512-YWzhKL2whUzgiheMoBFwW8CKV4qpHQAEuvilg9FAn5VJUDwKZZxkJ
NuGM4XkWuk94WCrrwslk8yWNGmY1EduTA==" crossorigin="anonymous"
referrerpolicy="no-referrer" />
    <title>Nuevo</title>
</head>

<body>
    <h3 class="text-center mt-3">Nueva Categoría</h3>
    <div class="container mt-4">
        <form action="<?php echo $_SERVER['PHP_SELF']."?id=$id"; ?>"
name="a" method="POST">
```

```
            <div class="bg-info p-4 text-white rounded shadow-lg
m-auto" style="width: 40rem;">
                <div class="mb-3">
                    <label for="n" class="form-label">Nombre
Categoría</label>
                    <input type="text" class="form-control" id="n"
value="<?php echo $datosCategoria->nombre; ?>" name="nombre" required>
                    <?php
                        if(isset($_SESSION['vacio_nombre'])) {
                            echo "<div class='text-danger
mt-2'>{$_SESSION['vacio_nombre']}</div>";
                            unset($_SESSION['vacio_nombre']);
                        }
                        if(isset($_SESSION['corto_nombre'])) {
                            echo "<div class='text-danger
mt-2'>{$_SESSION['corto_nombre']}</div>";
                            unset($_SESSION['corto_nombre']);
                        }
                    ?>
                </div>
                <div class="mb-3">
                    <label for="d" class="form-label">Descripción
Categoría</label>
                    <input type="text" class="form-control" id="d"
name="descripcion" value="<?php echo $datosCategoria->descripcion; ?>">
                    <?php
                        if(isset($_SESSION['vacio_descripcion'])) {
                            echo "<div class='text-danger
mt-2'>{$_SESSION['vacio_descripcion']}</div>";
                            unset($_SESSION['vacio_descripcion']);
                        }
                    ?>
                </div>
                <div class="mb-3">
                    <button type="submit" class="btn btn-primary"
name="Actualizar"><i class="fas fa-save"></i> Actualizar</button>
                    <a href="index.php" class="btn btn-dark"><i
class="fas fa-backward"></i> Volver</a>
                </div>
            </div>
        </form>
    </div>
</body>
```

```
</html>
<?php } ?>
```

Ahora crearemos la clase **“Articulos.php”,** en la carpeta **“src”,** que hereda de la clase **“Conexion.php”.**

```php
<?php
namespace Tienda;

class Articulos extends Conexion {
    private $id;
    private $nombre;
    private $precio;
    private $categoria_id;

    //Creamos el constructor que heredara de la clase padre
    public function __construct()
    {
        parent::__construct();
    }

    //### CRUD ###
    public function create() {

    }

    public function read() {

    }

    public function update() {

    }

    public function delete() {

    }


    //### OTROS MÉTODOS ###
```

```php
//### SETTERS AND GETTERS


/**
 * Get the value of id
 */
public function getId()
{
    return $this->id;
}

/**
 * Set the value of id
 *
 * @return  self
 */
public function setId($id)
{
    $this->id = $id;

    return $this;
}

/**
 * Get the value of nombre
 */
public function getNombre()
{
    return $this->nombre;
}

/**
 * Set the value of nombre
 *
 * @return  self
 */
public function setNombre($nombre)
{
    $this->nombre = $nombre;

    return $this;
```

```
    }

    /**
     * Get the value of precio
     */
    public function getPrecio()
    {
        return $this->precio;
    }

    /**
     * Set the value of precio
     *
     * @return  self
     */
    public function setPrecio($precio)
    {
        $this->precio = $precio;

        return $this;
    }

    /**
     * Get the value of categoria_id
     */
    public function getCategoria_id()
    {
        return $this->categoria_id;
    }

    /**
     * Set the value of categoria_id
     *
     * @return  self
     */
    public function setCategoria_id($categoria_id)
    {
        $this->categoria_id = $categoria_id;

        return $this;
    }
}
```

Creamos el método **create()** en la clase **"Articulos.php"** y un método nuevo al que llamaremos **generarArticulos()** para que nos genere artículos haciendo uso de la librería **Faker** y otro método más para comprobar si ya hay artículos creados para que no nos genere otros nuevos a la que llamaremos **hayArticulos().**

Método create()

```
public function create() {
        $q = "insert into articulos(nombre, precio, categoria_id)
values(:n, :p, :ci)";
        $stmt = parent::$conexion->prepare($q);
        try {
            $stmt->execute([
                ':n'=>$this->nombre,
                ':p'=>$this->precio,
                ':ci'=>$this->categoria_id
            ]);
        }catch(PDOException $ex) {
            die("Error al crear el artículo: ".$ex->getMessage());
        }
        //Cerramos la conexion a la base de datos
        parent::$conexion = null;
    }
```

Método hayArticulos()

```
public function hayArticulos() {
        $q = "select * from articulos";
        $stmt = parent::$conexion->prepare($q);
        try {
            $stmt->execute();
        }catch(PDOException $ex) {
            die("Error al comprobar si ya hay articulos:
".$ex->getMessage());
        }
        //Cerramos la conexion a la base de datos y devolvemos el total
de articulos
        parent::$conexion = null;
        return $stmt->rowCount();
    }
```

Método generarArticulos()

```
public function generarArticulos($cantidad) {
        if($this->hayArticulos() == 0) {
            $faker = Faker\Factory::create('es_ES');
            $categorias = (new Categorias)->devolverIdCategorias();
            for($i=0;$i<$cantidad;$i++) {
                $nombre = $faker->word();
                $precio = $faker->randomFloat($nbMaxDecimals = NULL,
$min = 0, $max = NULL);
                $categoria_id = $categorias[array_rand($categorias,
1)];
                (new Articulos)->setNombre($nombre)
                ->setPrecio($precio)
                ->setCategoria_id($categoria_id)
                ->create();
            }
        }
    }
```

Tendremos que crear otro método en la clase **"Categorias.php"** que nos devuelva los ids de las categorías.

Método devolverIdCategorias()

```
public function devolverIdCategorias() {
        $q = "select id from categorias order by id";
        $stmt = parent::$conexion->prepare($q);
        try {
            $stmt->execute();
        }catch(PDOException $ex) {
            die("Error al devolver el id de categorias:
".$ex->getMessage());
        }
        //Creamos una array y guardamos en el los ids de las categorias
        $id = [];
        while($fila = $stmt->fetch(PDO::FETCH_OBJ)) {
            $id[] = $fila->id;
        }
        //Cerramos la conexion a la base de datos y devolvermos el
array que hemos creado
        parent::$conexion = null;
        return $id;
    }
```

Dentro de la carpeta **“articulos”** creamos un fichero con el nombre **“index.php”** donde vamos a mostrar una tabla con todos los artículos creados. Además, para poder mostrar los artículos que hemos generado hay que crear un método nuevo al que llamaremos **readAll()** en la clase **“Articulos.php”.**

Método readAll()

```
public function readAll() {
        $q = "select * from articulos order by nombre";
        $stmt = parent::$conexion->prepare($q);
        try {
            $stmt->execute();
        }catch(PDOException $ex) {
            die("Error al devolver los artículos: ".$ex->getMessage());
        }
        //Cerramos la conexion a la base de datos y devolvemos todos
los datos
        parent::$conexion = null;
        return $stmt;
    }
```

index.php de artículos

```
<?php
session_start();
require dirname(__DIR__, 2) . "/vendor/autoload.php";

use Tienda\Articulos;

(new Articulos)->generarArticulos(50);
$datos = (new Articulos)->readAll();
?>
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width,
initial-scale=1.0">
    <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.m
in.css" rel="stylesheet"
integrity="sha384-1BmE4kWBq78iYhFldvKuhfTAU6auU8tT94WrHftjDbrCEXSU1oBoq
yl2QvZ6jIW3" crossorigin="anonymous">
```

```
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.0.0-beta2/c
ss/all.min.css"
integrity="sha512-YWzhKL2whUzgiheMoBFwW8CKV4qpHQAEuvilg9FAn5VJUDwKZZxkJ
NuGM4XkWuk94WCrrwslk8yWNGmY1EduTA==" crossorigin="anonymous"
referrerpolicy="no-referrer" />
    <title>Artículos</title>
</head>

<body>
    <h3 class="text-center mt-3">Artículos</h3>
    <div class="container mt-4">
        <?php
        if (isset($_SESSION['mensaje'])) {
            echo <<<TXT
                <div class="alert alert-primary" role="alert">
                    {$_SESSION['mensaje']}
                </div>
            TXT;
            unset($_SESSION['mensaje']);
        }
        ?>
        <a href="addArt.php" class="btn btn-success"><i class="fas
fa-plus-circle"></i> Nuevo Artículo</a>
        <table class="table table-striped">
            <thead>
                <tr>
                    <th scope="col">Detalles</th>
                    <th scope="col">Nombre</th>
                    <th scope="col">Precio</th>
                    <th scope="col">Acciones</th>
                </tr>
            </thead>
            <tbody>
                <?php
                while ($fila = $datos->fetch(PDO::FETCH_OBJ)) {
                    echo <<<TXT
                    <tr>
                        <th scope="row"><a
href='detalleArt.php?id={$fila->id}' class='btn btn-info'><i class='fas
fa-info-circle'></i> Detalles</a></th>

                        <td>{$fila->nombre}</td>
```

```
                        <td>{$fila->precio}</td>
                        <td>
                            <form name='a' action='deleteArt.php'
method='POST'>
                                <input type='hidden' name='id'
value='{$fila->id}'>
                                <a href='updateArt.php?id={$fila->id}'
class='btn btn-primary'><i class='fas fa-edit'></i> Editar</a>
                                <button type='submit' class='btn
btn-danger' onclick='return confirm("¿Desea borrar el artículo?")'><i
class='fas fa-trash'></i> Borrar </button>
                            </form>
                        </td>
                    </tr>
                    TXT;
                }
                ?>
            </tbody>
        </table>
    </div>
</body>

</html>
```

Ahora crearemos el fichero **“addArticulo.php”** para poder crear un nuevo artículo.

```
<?php
session_start();
require dirname(__DIR__, 2) . "/vendor/autoload.php";

use Tienda\Articulos;
use Tienda\Categorias;

$categorias = (new Categorias)->devolverCategorias();
//Creamos una función para comprobar si los campos están vacios
function campoVacio($n, $p)
{
    $error = false;
    if (strlen($n) == 0) {
        $_SESSION['vacio_nombre'] = "Rellene el campo nombre";
        $error = true;
    }
```

```
    if (strlen($p) == 0) {
        $_SESSION['vacio_precio'] = "Rellene el campo precio";
        $error = true;
    }
    return $error;
}

if (isset($_POST['Crear'])) {
    $nombre = trim(ucfirst($_POST['nombre']));
    $precio = trim($_POST['precio']);
    $categoria_id = $_POST['categoria_id'];
    if(!campoVacio($nombre, $precio)) {
        (new Articulos)->setNombre($nombre)
        ->setPrecio($precio)
        ->setCategoria_id($categoria_id)
        ->create();
        $_SESSION['mensaje'] = "Artículo creado correctamente";
        header("Location:index.php");
    } else {
        header("Location:{$_SERVER['PHP_SELF']}");
    }
} else {
?>
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width,
initial-scale=1.0">
    <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.m
in.css" rel="stylesheet"
integrity="sha384-1BmE4kWBq78iYhFldvKuhfTAU6auU8tT94WrHftjDbrCEXSU1oBoq
yl2QvZ6jIW3" crossorigin="anonymous">
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.0.0-beta2/c
ss/all.min.css"
integrity="sha512-YWzhKL2whUzgiheMoBFwW8CKV4qpHQAEuvilg9FAn5VJUDwKZZxkJ
NuGM4XkWuk94WCrrwslk8yWNGmY1EduTA==" crossorigin="anonymous"
referrerpolicy="no-referrer" />
    <title>Nuevo artículo</title>
```

```
</head>

<body>
    <h3 class="text-center mt-3">Nuevo Artículo</h3>
    <div class="container mt-4">
        <form action="<?php echo $_SERVER['PHP_SELF']; ?>" name="a"
method="POST">
            <div class="bg-info p-4 text-white rounded shadow-lg
m-auto" style="width: 40rem;">
                <div class="mb-3">
                    <label for="n" class="form-label">Nombre
Artículo</label>
                    <input type="text" class="form-control" id="n"
placeholder="Nombre" name="nombre" required>
                    <?php
                    if (isset($_SESSION['vacio_nombre'])) {
                        echo "<div class='text-danger
mt-2'>{$_SESSION['vacio_nombre']}</div>";
                        unset($_SESSION['vacio_nombre']);
                    }
                    ?>
                </div>
                <div class="mb-3">
                    <label for="p" class="form-label">Precio
Artículo</label>
                    <input type="text" class="form-control" id="p"
name="precio" placeholder="Precio">
                    <?php
                    if (isset($_SESSION['vacio_descripcion'])) {
                        echo "<div class='text-danger
mt-2'>{$_SESSION['vacio_descripcion']}</div>";
                        unset($_SESSION['vacio_descripcion']);
                    }
                    ?>
                </div>
                <div class="mb-3">
                    <label for="cid" class="form-label">Categoría del
Artículo</label>
                    <select name="categoria_id" id="cid"
class="form-select">
                        <?php
                            foreach($categorias as $item) {
```

```
                                echo "<option
value='{$item->id}'>{$item->nombre}</option>";
                            }
                        ?>
                    </select>
                </div>
                <div class="mb-3">
                    <button type="submit" class="btn btn-primary"
name="Crear"><i class="fas fa-check-circle"></i> Crear</button>
                    <button type="reset" class="btn btn-danger"><i
class="fas fa-broom"></i> Limpiar</button>
                </div>
            </div>
        </form>
    </div>
</body>

</html>
<?php } ?>
```

Hemos tenido que crear un método nuevo al que hemos llamado **“devolverCategorias()”** dentro de la clase **“Categorias.php”,** que como su nombre indica nos devuelve todas las categorías que tenemos.

Método devolverCategorias()

```
public function devolverCategorias() {
        $q = "select nombre, descripcion, id from categorias order by
nombre";
        $stmt = parent::$conexion->prepare($q);
        try {
            $stmt->execute();
        }catch(PDOException $ex) {
            die("Error al devolver las categorías:
".$ex->getMessage());
        }
        //Cerramos la conexion a la base de datos y devolvemos los
datos
        parent::$conexion = null;
        return $stmt->fetchAll(PDO::FETCH_OBJ);
    }
```

Creamos el fichero **“deleteArt.php”** dentro de la carpeta **“articulos”** y el método **“delete()”** en la clase **“Articulos.php”.**

Método delete()

```
public function delete($id)
    {
        $q = "delete from articulos where id=:i";
        $stmt = parent::$conexion->prepare($q);
        try {
            $stmt->execute([
                ':i' => $id
            ]);
        } catch (PDOException $ex) {
            die("Error al borrar el artículo: " . $ex->getMessage());
        }
    }
```

deleteArt.php de artículos

```
<?php
session_start();
require dirname(__DIR__, 2)."/vendor/autoload.php";
use Tienda\Articulos;

//Comprobar si existe el id que le pasamos
if(!isset($_POST['id'])) {
    header("Location:index.php");
    die();
}

(new Articulos)->delete($_POST['id']);
$_SESSION['mensaje'] = "Artículo borrado correctamente";
header("Location:index.php");
```

Ahora crearemos el fichero **“updateArt.php”** dentro de la carpeta **“articulos”** y los métodos **update()** para que actualice el artículo y **read()** para que nos muestre los datos del artículo que vamos a actualizar.

Método update()

```
public function update($id)
    {
```

```
        $q = "update articulos set nombre=:n, precio=:p,
categoria_id=:cid where id=:i";
        $stmt = parent::$conexion->prepare($q);
        try {
            $stmt->execute([
                ':n'=>$this->nombre,
                ':p'=>$this->precio,
                ':cid'=>$this->categoria_id,
                ':i'=>$id
            ]);
        }catch(PDOException $ex) {
            die("Error al actualizar el artículo: ".$ex->getMessage());
        }
        //Cerramos la conexion a la base de datos
        parent::$conexion = null;
    }
```

Método read()

```
public function read($id)
    {
        $q = "select articulos.*, articulos.nombre, descripcion from
articulos, categorias where
        categoria_id=categorias.id AND articulos.id=:i";
        $stmt = parent::$conexion->prepare($q);
        try {
            $stmt->execute([
                ':i' => $id
            ]);
        } catch (PDOException $ex) {
            die("Error al devolver los datos del artículo: " .
$ex->getMessage());
        }
        //Cerramos la conexion a la base de datos y devolvemos los
datos del articulo
        parent::$conexion = null;
        return $stmt->fetch(PDO::FETCH_OBJ);
    }
```

updateArt.php

```php
<?php
if (!isset($_GET['id'])) {
    header("Location:index.php");
    die();
}

session_start();
require dirname(__DIR__, 2) . "/vendor/autoload.php";

use Tienda\Articulos;
use Tienda\Categorias;

$id = $_GET['id'];
$articulo = (new Articulos)->read($id);
$categorias = (new Categorias)->devolverCategorias();
//Creamos una función para comprobar si los campos están vacios
function campoVacio($n, $p)
{
    $error = false;
    if (strlen($n) == 0) {
        $_SESSION['vacio_nombre'] = "Rellene el campo nombre";
        $error = true;
    }

    if (strlen($p) == 0) {
        $_SESSION['vacio_precio'] = "Rellene el campo precio";
        $error = true;
    }
    return $error;
}

if (isset($_POST['Actualizar'])) {
    $nombre = trim(ucfirst($_POST['nombre']));
    $precio = trim($_POST['precio']);
    $categoria_id = $_POST['categoria_id'];
    if (!campoVacio($nombre, $precio)) {
        (new Articulos)->setNombre($nombre)
            ->setPrecio($precio)
            ->setCategoria_id($categoria_id)
            ->update($id);
        $_SESSION['mensaje'] = "Artículo actualizado correctamente";
```

```
        header("Location:index.php");
    } else {
        header("Location:{$_SERVER['PHP_SELF']}");
    }
} else {
?>
    <!DOCTYPE html>
    <html lang="en">

    <head>
        <meta charset="UTF-8">
        <meta http-equiv="X-UA-Compatible" content="IE=edge">
        <meta name="viewport" content="width=device-width,
initial-scale=1.0">
        <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.m
in.css" rel="stylesheet"
integrity="sha384-1BmE4kWBq78iYhFldvKuhfTAU6auU8tT94WrHftjDbrCEXSU1oBoq
yl2QvZ6jIW3" crossorigin="anonymous">
        <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.0.0-beta2/c
ss/all.min.css"
integrity="sha512-YWzhKL2whUzgiheMoBFwW8CKV4qpHQAEuvilg9FAn5VJUDwKZZxkJ
NuGM4XkWuk94WCrrwslk8yWNGmY1EduTA==" crossorigin="anonymous"
referrerpolicy="no-referrer" />
        <title>Actualiar Artículo</title>
    </head>

    <body>
        <h3 class="text-center mt-3">Actualizar Artículo</h3>
        <div class="container mt-4">
            <form action="<?php echo $_SERVER['PHP_SELF']. "?id=$id";
?>" name="a" method="POST">
                <div class="bg-info p-4 text-white rounded shadow-lg
m-auto" style="width: 40rem;">
                    <div class="mb-3">
                        <label for="n" class="form-label">Nombre
Artículo</label>
                        <input type="text" class="form-control" id="n"
name="nombre" value="<?php echo $articulo->nombre; ?>" required>
                        <?php
                        if (isset($_SESSION['vacio_nombre'])) {
```

```
                            echo "<div class='text-danger
mt-2'>{$_SESSION['vacio_nombre']}</div>";
                            unset($_SESSION['vacio_nombre']);
                        }
                        ?>
                    </div>
                    <div class="mb-3">
                        <label for="p" class="form-label">Precio
Artículo</label>
                        <input type="text" class="form-control" id="p"
value="<?php echo $articulo->precio; ?>" name="precio">
                        <?php
                        if (isset($_SESSION['vacio_precio'])) {
                            echo "<div class='text-danger
mt-2'>{$_SESSION['vacio_precio']}</div>";
                            unset($_SESSION['vacio_precio']);
                        }
                        ?>
                    </div>
                    <div class="mb-3">
                        <label for="cid" class="form-label">Categoría
del Artículo</label>
                        <select name="categoria_id" id="cid"
class="form-select">
                            <?php
                            foreach ($categorias as $item) {
                                if ($item->id ==
$articulo->categoria_id) {
                                    echo "<option value='{$item->id}'
selected>{$item->nombre}</option>";
                                } else {
                                    echo "<option
value='{$item->id}'>{$item->nombre}</option>";
                                }
                            }
                            ?>
                        </select>
                    </div>
                    <div class="mb-3">
                        <button type="submit" class="btn btn-primary"
name="Actualizar"><i class="fas fa-save"></i> Actualizar</button>
                        <a href="index.php" class="btn btn-dark"><i
class="fas fa-backward"></i> Volver</a>
```

```
                    </div>
                </div>
            </form>
        </div>
    </body>

    </html>
<?php } ?>
```

Por último, haremos la página detalles donde se mostrará los detalles de cada artículo.

detalleArt.php

```
<?php
if (!isset($_GET['id'])) {
    header("Location:index.php");
    die();
}
require dirname(__DIR__, 2) . "/vendor/autoload.php";

use Tienda\Articulos;
use Tienda\Categorias;

$id = $_GET['id'];
$datosArticulo = (new Articulos)->read($id);

?>
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width,
initial-scale=1.0">
    <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.m
in.css" rel="stylesheet"
integrity="sha384-1BmE4kWBq78iYhFldvKuhfTAU6auU8tT94WrHftjDbrCEXSU1oBoq
yl2QvZ6jIW3" crossorigin="anonymous">
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.0.0-beta2/c
ss/all.min.css"
```

```
integrity="sha512-YWzhKL2whUzgiheMoBFwW8CKV4qpHQAEuvilg9FAn5VJUDwKZZxkJ
NuGM4XkWuk94WCrrwslk8yWNGmY1EduTA==" crossorigin="anonymous"
referrerpolicy="no-referrer" />
    <title>Detalle Artículo</title>
</head>

<body>
    <h3 class="text-center mt-2">Detalle Artículo (<?php echo
$datosArticulo->id; ?>)</h3>
    <div class="container mt-2">
        <div class="shadow-lg mx-auto bg-info text-white rounded-3 p-4"
style="width:40rem">
            <h5 class="text-center"><?php echo $datosArticulo->nombre;
?></h5>
            <p class="mt-3">
                <b>Precio: </b>
                <?php echo $datosArticulo->precio ?>

            </p>
            <p class="mt-3">
                <b>Categoria: </b>
                <?php echo $datosArticulo->nombre . " (" .
$datosArticulo->categoria_id . ")"; ?>
            </p>
            <div class="mt-2">
                <a href="index.php" class="btn btn-dark">
                    <i class="fas fa-backward"></i> Volver</a>
            </div>
            <div>

            </div>
</body>

</html>
```