

Crear urls amigables.

RewriteEngine on

RewriteCond %{REQUEST\_FILENAME} !-d

RewriteCond %{REQUEST\_FILENAME} !-f

RewriteRule ^(.\*)\$ index.php?url=\$1 [QSA,L]

Crear el archivo de configuración en la raíz.

```
<?php
require_once 'Config/Config.php';
$ruta = !empty($_GET['url']) ? $_GET['url'] : "home/index";
$array = explode("/", $ruta);
$controller = ucfirst($array[0]);
$metodo = "index";
$parametro = "";
if (!empty($array[1])) {
    if (!empty($array[1] != "")) {
        $metodo = $array[1];
    }
}
if (!empty($array[2])) {
    if (!empty($array[2] != "")) {
        for ($i = 2; $i < count($array); $i++) {
            $parametro .= $array[$i] . ",";
        }
        $parametro = trim($parametro, ",");
    }
}
require_once 'Config/App/Autoload.php';
require_once 'Config/Helpers.php';
$dirControllers = "Controllers/" . $controller . ".php";
if (file_exists($dirControllers)) {
    require_once $dirControllers;
    $controller = new $controller();
    if (method_exists($controller, $metodo)) {
        $controller->$metodo($parametro);
    } else {
        header('Location: '.BASE_URL.'errors');
    }
} else {
    header('Location: ' . BASE_URL . 'errors');
}
?>
```

En la raíz crearemos un directorio Config.

```
<?php
```

```

const BASE_URL = "http://localhost/ecommerce/";
const HOST = "localhost";
const USER = "root";
const PASS = "";
const DB = "carrito";
const CHARSET = "charset=utf8";
const TITLE = "SHOP";
const MONEDA = "USD";
const CLIENT_ID = "";
?>

```

Dentro del directorio Config una nueva subcarpeta APP

```

<?php
class Controller{
    //esta linea no lo veras en el video es necesario para solucionar las advertencias
    protected $views, $model;
    //
    public function __construct()
    {
        $this->views = new Views();
        $this->cargarModel();
    }
    public function cargarModel()
    {
        $model = get_class($this)."Model";
        $ruta = "Models/".$model.".php";
        if (file_exists($ruta)) {
            require_once $ruta;
            $this->model = new $model();
        }
    }
}
?>

```

Crear el archivo Autoload.php

```

<?php
spl_autoload_register(function($class){
    if (file_exists("Config/App/".$class.".php")) {
        require_once "Config/App/" . $class . ".php";
    }
})
?>

```

Crear el archivo Views.php

```

<?php
class Views{

    public function getView($ruta, $vista, $data="")
    {
        if ($ruta == "home") {
            $vista = "Views/".$vista.".php";
        }else{
            $vista = "Views/".$ruta."/".$vista.".php";
        }
        require $vista;
    }
}
?>

```

Crear consultas preparadas.

```

<?php
class Query extends Conexion{
    private $pdo, $con, $sql, $datos;
    public function __construct() {
        $this->pdo = new Conexion();
        $this->con = $this->pdo->conect();
    }
    public function select(string $sql)
    {
        $this->sql = $sql;
        $resul = $this->con->prepare($this->sql);
        $resul->execute();
        $data = $resul->fetch(PDO::FETCH_ASSOC);
        return $data;
    }
    public function selectAll(string $sql)
    {
        $this->sql = $sql;
        $resul = $this->con->prepare($this->sql);
        $resul->execute();
        $data = $resul->fetchAll(PDO::FETCH_ASSOC);
        return $data;
    }
    public function save(string $sql, array $datos)
    {
        $this->sql = $sql;
        $this->datos = $datos;
        $insert = $this->con->prepare($this->sql);
        $data = $insert->execute($this->datos);
        if ($data) {
            $res = 1;

```

```

    }else{
        $res = 0;
    }
    return $res;
}
public function insertar(string $sql, array $datos)
{
    $this->sql = $sql;
    $this->datos = $datos;
    $insert = $this->con->prepare($this->sql);
    $data = $insert->execute($this->datos);
    if ($data) {
        $res = $this->con->lastInsertId();
    } else {
        $res = 0;
    }
    return $res;
}
}
?>

```

Crear la conexion.

```

<?php
class Conexion{
    private $conect;
    public function __construct()
    {
        $pdo = "mysql:host=".HOST.";dbname=".DB.";".CHARSET;
        try {
            $this->conect = new PDO($pdo, USER, PASS);
            $this->conect->setAttribute(PDO::ATTR_ERRMODE,
PDO::ERRMODE_EXCEPTION);
        } catch (PDOException $e) {
            echo "Error en la conexion".$e->getMessage();
        }
    }
    public function conect()
    {
        return $this->conect;
    }
}
?>

```

Crear directorio Controllers en la raíz.

```

<?php

```

```

class Home extends Controller
{
    public function __construct() {
        parent::__construct();
        session_start();
    }
    public function index()
    {
        $data['title'] = 'Pagina Principal';
        $this->views->getView('home', "index", $data);
    }
}

```

Crear Directorio Models en la raíz.

```

<?php
class HomeModel extends Query{

    public function __construct()
    {
        parent::__construct();
    }

}

?>

```

Crear controlador Errors.php

```

<?php
class Errors extends Controller
{
    public function __construct()
    {
        parent::__construct();
    }
    public function index()
    {
        $this->views->getView('errors', "index");
    }
}

```

Crear directorio Views en la raíz.

```

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

<head>

```

```
<meta charset="UTF-8">
<title><?php echo TITLE . ' - Pagina no encontrada'; ?></title>
<link rel="stylesheet" href="<?php echo BASE_URL; ?>assets/css/bootstrap.min.css">
</head>

<body class="d-flex h-100 text-center text-white bg-dark">

    <div class="cover-container d-flex w-100 h-100 p-3 mx-auto flex-column">

        <main class="px-3">
            <h1>Not Found.</h1>
            <p class="lead">La página que buscas no existe</p>
            <p class="lead">
                <a href="<?php echo BASE_URL; ?>" class="btn btn-lg btn-secondary fw-bold
border-white bg-white text-dark"><?php echo TITLE; ?></a>
            </p>
        </main>
    </div>

</body>

</html>
```