

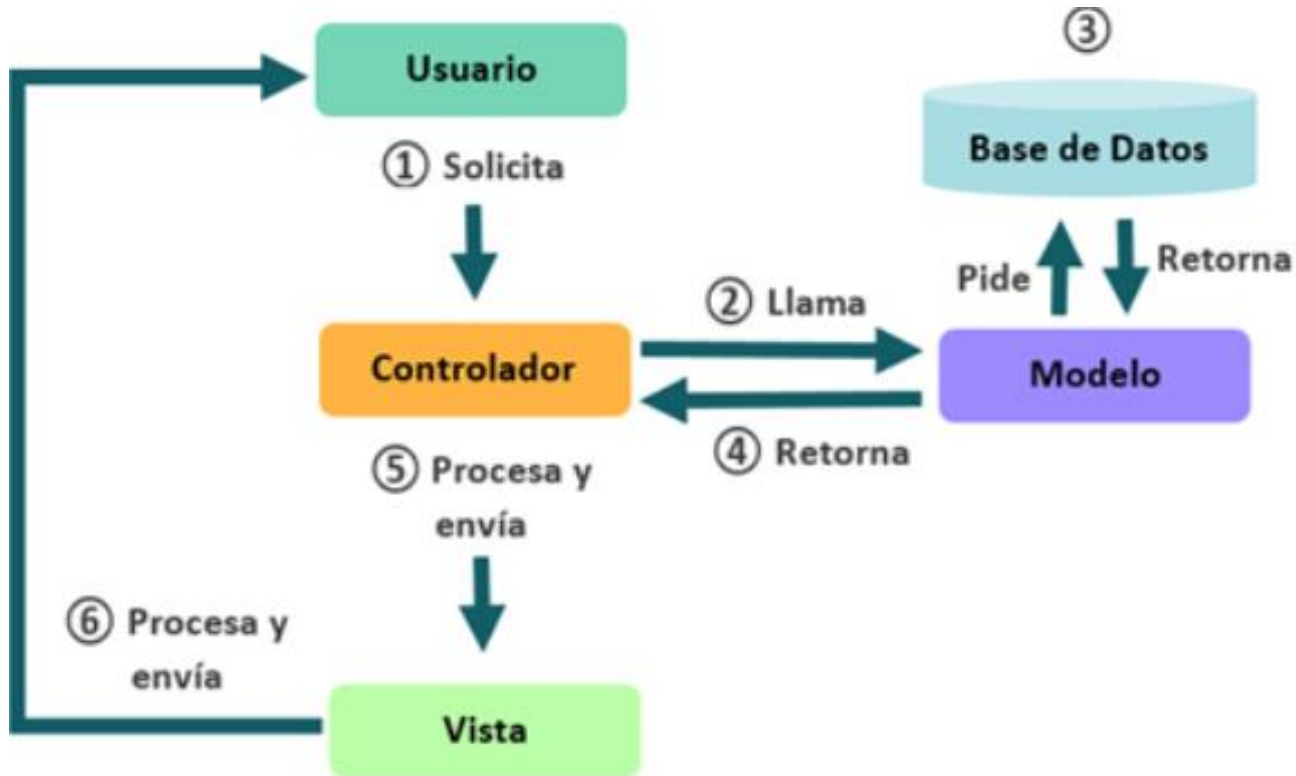
# UT 5: Modelo Vista Controlador en PHP



2ºDAW – Desarrollo Entorno Servidor

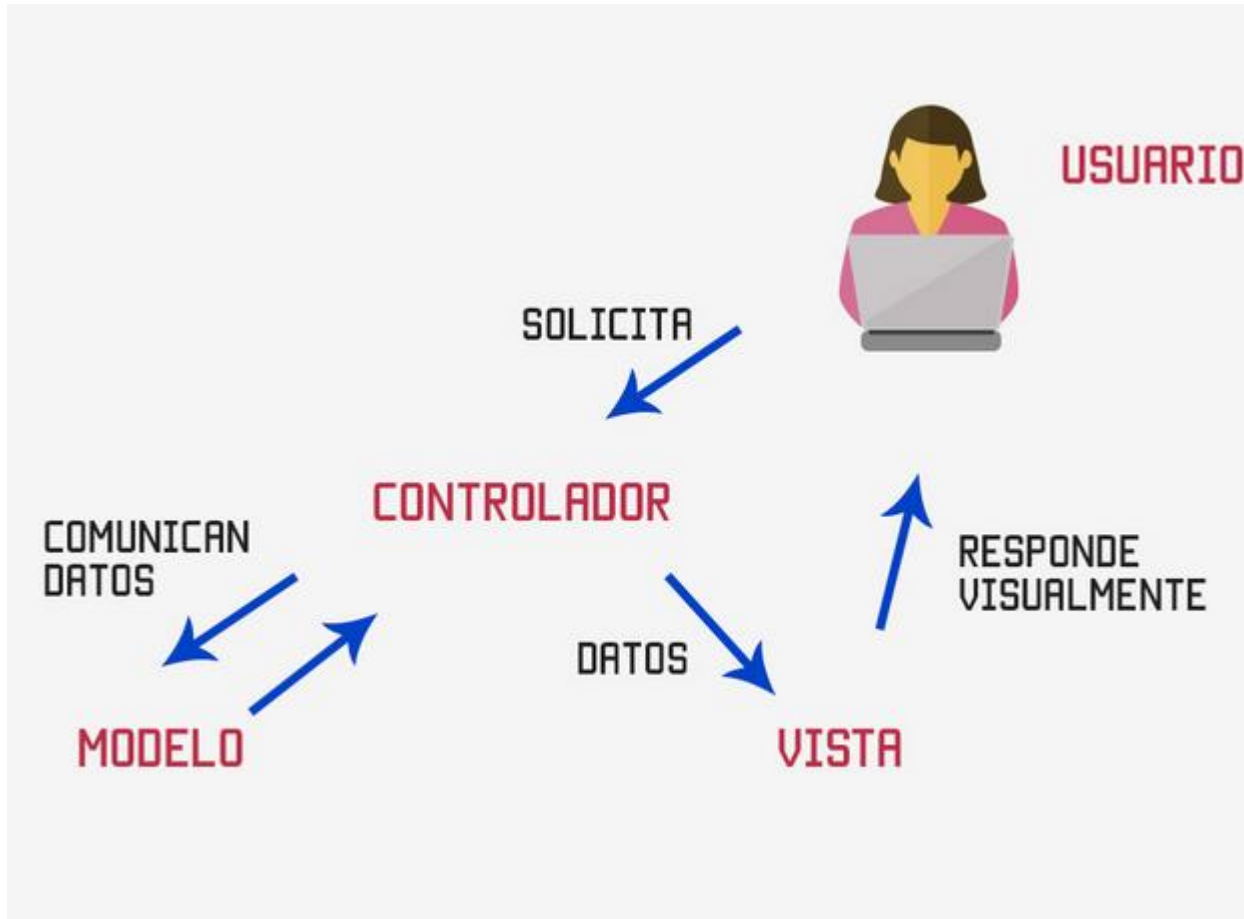
# Modelo Vista Controlador MVC

Patrón de Arquitectura para el desarrollo de aplicaciones web. Se fundamenta en la separación del código en tres capas:



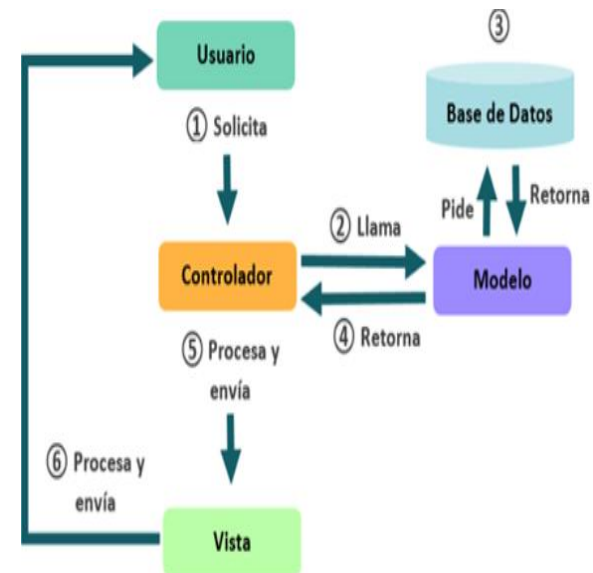
# Modelo Vista Controlador MVC

Patrón de Arquitectura para el desarrollo de aplicaciones web. Se fundamenta en la separación del código en tres capas:



# Modelo Vista Controlador MVC

- **Modelo:** contiene la lógica de la aplicación (clases y métodos que se comunican directamente con base datos). Todas las peticiones de acceso a los datos pasarán por esta capa.
- **Vista:** contiene el diseño de la interfaz gráfica, mostrará la información que verá el usuario (se mostrarán los datos pero no se accede a ellos)
- **Controlador:** intermediario entre la vista y el modelo. Se encarga de controlar las interacciones del usuario con la vista. El usuario solicitará información por medio de la vista y esta hará la petición al controlador. Posteriormente, este, realizará la petición al modelo.

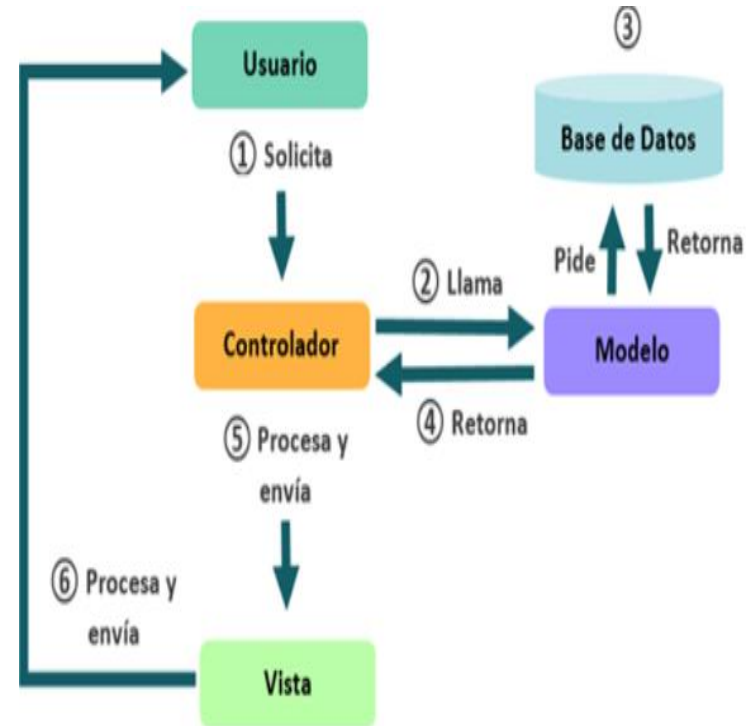


# Modelo Vista Controlador MVC

## ¿Cómo funciona el MVC?

El funcionamiento básico del patrón MVC, puede resumirse en:

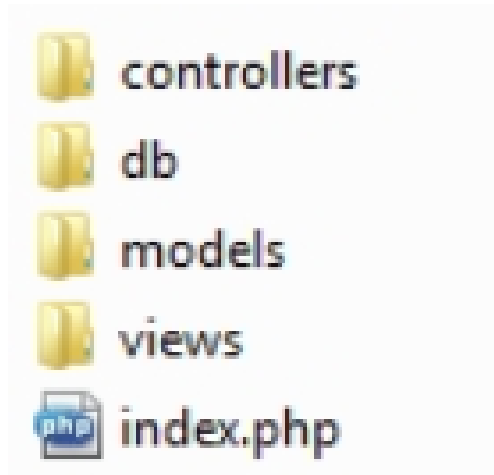
1. El usuario realiza una petición.
2. El controlador captura la petición y hace llamada al modelo correspondiente.
3. El modelo será el encargado de interactuar con la base de datos.
4. Una vez que el modelo tiene la información se la envía al controlador
5. El controlador recibe la información y la envía a la vista.
6. La vista muestra la información.



# Modelo Vista Controlador MVC

## ¿Cómo implementart MVC en PHP?

Para implementar el MVC es imprescindible crear una estructura de ficheros parecida a esta:



# Modelo Vista Controlador MVC

**Fichero** index.php

```
<?php
```

```
// Llamada al fichero que inicia la conexión a la Base de Datos  
require_once("db/db.php");
```

```
// Llamada al controlador  
require_once("controllers/provincias_controller.php");
```

```
?>
```

# Modelo Vista Controlador MVC

**Fichero** /db/db.php

```
<?php
// Creación de una clase que nos permita conectarnos a la BD
class Conectar{
    public static function conexion(){
        $conexion=new mysqli("localhost", "root", "rootroot", "test");
        return $conexion;
    }
}
?>
```



# Modelo Vista Controlador MVC

**Fichero** /models/provincias\_model.php

```
<?php

// Modelo contiene la lógica de la aplicación: clases y métodos que se
// comunican con la Base de Datos

//Creación de una clase para ejecutar la sentencia SQL
class provincias_model{
    private $db;
    private $provincias;

    public function __construct() {
        $this->db=Conectar::conexion();
        $this->provincias=array();
    }

    public function get_provincias() {
        $consulta=$this->db->query("select * from provincia;");
        while($filas=$consulta->fetch_assoc()){
            $this->provincias[]=$filas;
        }
        return $this->provincias;
    }
}
```

# Modelo Vista Controlador MVC

**Fichero** /controllers/provincias\_controller.php

```
<?php
//Llamada al modelo -- Intermediario entre vista y modelo !!!
require_once("models/provincias_model.php");

$provincia=new provincias_model();
$datos=$provincia->get_provincias();

//Llamada a la vista -- Intermediario entre vista y modelo !!!
require_once("views/provincias_view.phtml");
?>
```

# Modelo Vista Controlador MVC

**Fichero** /views/provincias\_view.php

```
<!DOCTYPE html>
<html lang="es">
  <head>
    <meta charset="UTF-8" />
    <title>Provincias</title>
  </head>
  <body>
    <?php
      // Solo muestra datos no accede a los mismos
      foreach ($datos as $dato) {
        echo $dato["provincia"]."<br/>";
      }
    ?>
  </body>
</html>
```