







Controlador

Filtros la entrada (*autenticación, validación de petición*)

Ejecución de acciones

Modelo

Conexiones a la base de datos
Modelo del negocio
Reglas y relaciones

Vista

Templates
HTML, CSS, JS, AJAX
Interfaz gráfica



`http://mihost.com/index.php?controlador=alumno&accion=listar`

Petición



Mandamos a llamar el archivo **index.php** el cual se encarga de realizar lo que se requiere.

Se define que **controlador** se va a mandar a llamar para que se puedan realizar las acciones.

Hemos definido la variable **accion** para indicar que parte de nuestro sistema queremos ejecutar.

El archivo **index.php**

```
switch($_REQUEST['controlador']){
    case 'alumno':
        //Carga el controlador
        require ( 'controlador/alumnoCtl.php' );
        $controlador = new AlumnoCtl();
        Break;
    case 'materia':
        //Carga el controlador
        require ( 'controlador/materiaCtl.php' );
        $controlador = new MateriaCtl();
        break;
}
//Ejecuta el controlador
$controlador → ejecutar();
```



Controlador

Filtrar la entrada (*autenticación, validación de petición*)

Ejecución de acciones

1. Archivo que **recibe las variables** que vienen por POST o GET.
2. Realiza validaciones de autenticación.
3. En base a las variables procede a **crear el modelo** necesario.
4. Hace que **el modelo realice las acciones** necesarias para obtener lo que se ha pedido.
5. **Obtiene los datos devueltos por el modelo** y en caso de ser necesario realiza validaciones.
6. **Manda a ejecutar la vista** con los datos que se han obtenido.

El archivo alumnoCtl.php

```
class AlumnoCtl{
    public $modelo;

    function __construct(){
        //Incluir el modelo
        include('modelo/alumnoBss.php');

        //Creo el objeto del modelo
        $this->modelo = new AlumnoBss();
    }

    function ejecutar(){
        switch($_REQUEST['accion']) {
            case 'listar':
                //Aquí se pueden hacer validaciones de
                //usuarios autenticados
                $this->listar();
                break;
        }
    }
}
```

El archivo alumnoCtl.php

```
case insertar:  
    //Se limpian las variables  
    $nombre = $this->sanitizaTexto($_REQUEST['nombre']);  
    $mail = $this->sanitizaCorreo($_REQUEST['mail']);  
    $pass = $this->sanitizaEncriptado($_REQUEST['pass']);  
    $privilegio = $this->sanitizaNumero($_REQUEST['privilegio']);  
  
    $alumno = $this->modelo->insertar($nombre,$mail,$pass,  
    $privilegio);  
  
    if ( is_object($alumno) ){  
        //Incluir la vista para mostrar el alumno insertado  
        include('vista/alumnoViewAlumno.php');  
    }  
    else{  
        //Mando a llamar la vista de errores  
    }  
  
default:  
    //Aquí se pueden hacer validaciones de  
    //usuarios autenticados  
    $this->perfil();  
}
```

Como todos estos métodos de sanitizado se repiten en todos los controladores puedes realizar un controlador estandar abstracto y que los demás controladores hereden de el.



Modelo

Conexiones a la base de datos
Modelo del negocio
Reglas y relaciones

1. Clase que **se instancia en el controlador**.
2. Realiza **consultas a la base de datos** y procesa la información para devolverla al controlador.
3. En los métodos de la clase se cumplen las **reglas de negocios**, por ejemplo, *las materias no pueden ser eliminadas, solo cambiadas de status*.

alumnoBss.php

El archivo

```
class AlumnoBss{  
  
    function insertar($nombre, $mail, $pass, $priv){  
        //Conectarse a la base de datos  
        require('dbdata.inc');  
        require('dbClass.php');  
  
        $conexion = new DB($hostdb, $userdb, $passdb, $db);  
        if(!$conexion -> conecta())  
            die('No se ha podido realizar la conexion a la bd');  
  
        //Crear el query  
        $consulta = "INSERT INTO  
                    alumno (nombre, mail, pass, privilegio)  
                VALUES  
                    ('$nombre',  
                     '$mail',  
                     '$pass',  
                     $privilegio)";  
  
        //Ejecutar el query  
        $resultado = $conexion -> ejecutarConsulta($consulta);  
    }  
}
```

La clase DB puede ser abstracta para que heredes de ella y así solo hagas la conexión durante la vida del modelo.

El archivo **alumnoBSS.php**

```
if($resultado == FALSE){
    echo 'FALLO la consulta';
    //Cerrar la conexion
    $conexion -> cerrar();
    return FALSE;
}
else{
    $id = $resultado;
    //Cerrar la conexion
    $conexion -> cerrar();

    //El objeto usuario
    require('alumnoClass.php');
    $alumno = new Alumno($resultado['id'],
$resultado['nombre'], $resultado['mail'], $resultado['pass'],
$resultado['privilegio']);

    return $alumno;
}
}
```



La carpeta modelo

1. Archivo con los **datos para la conexión** a la base de datos. (*es privado y no debe estar en tu repositorio en github*)
2. La clase utilizada para la **abstracción de la base de datos**.
3. La clase del **modelo**.
4. La clase del **elemento**.