

RA5 ORD

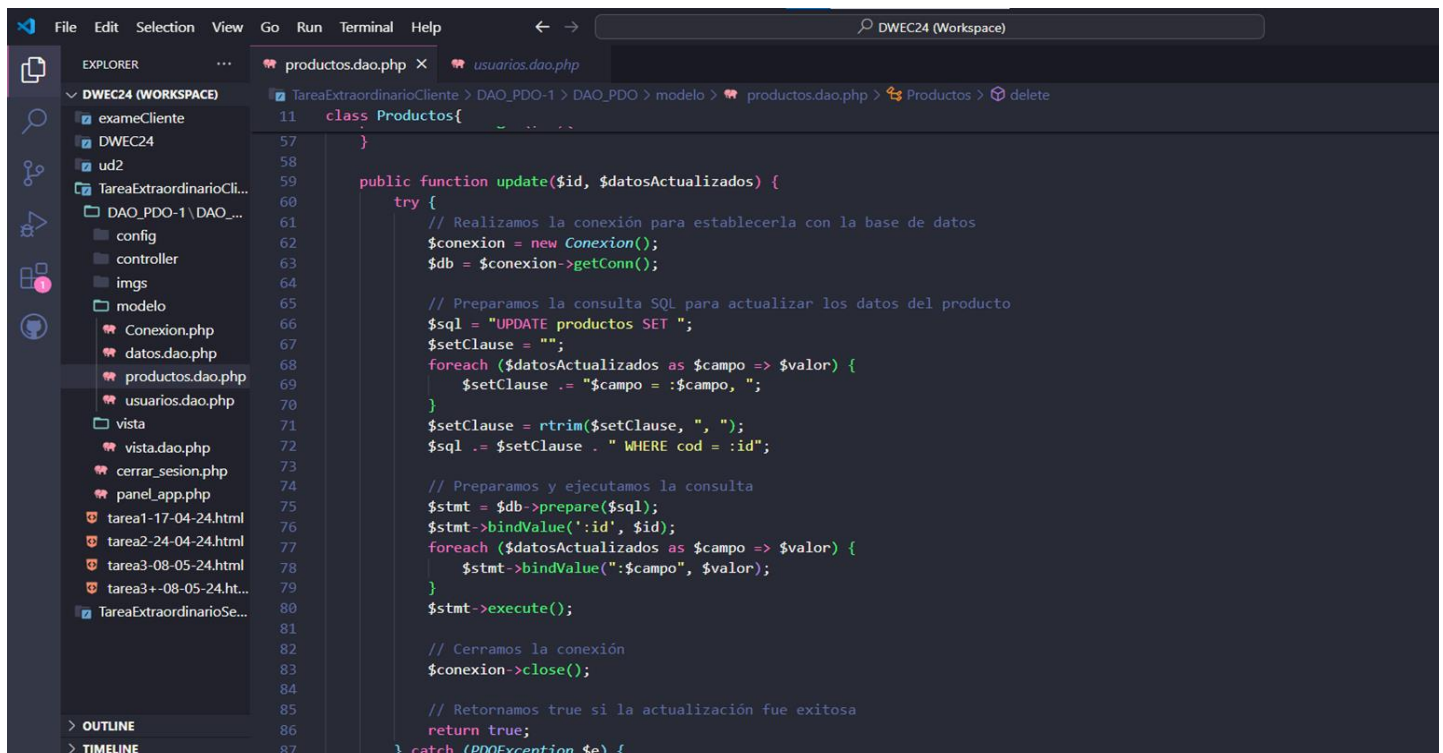
PREGUNTA 1

En relación con la aplicación de carrito de compra desarrollada en clase (UD3)..

1.Explica brevemente en el área de texto cómo se genera el formulario para editar o borrar productos del carrito en el proyecto (3 puntos)

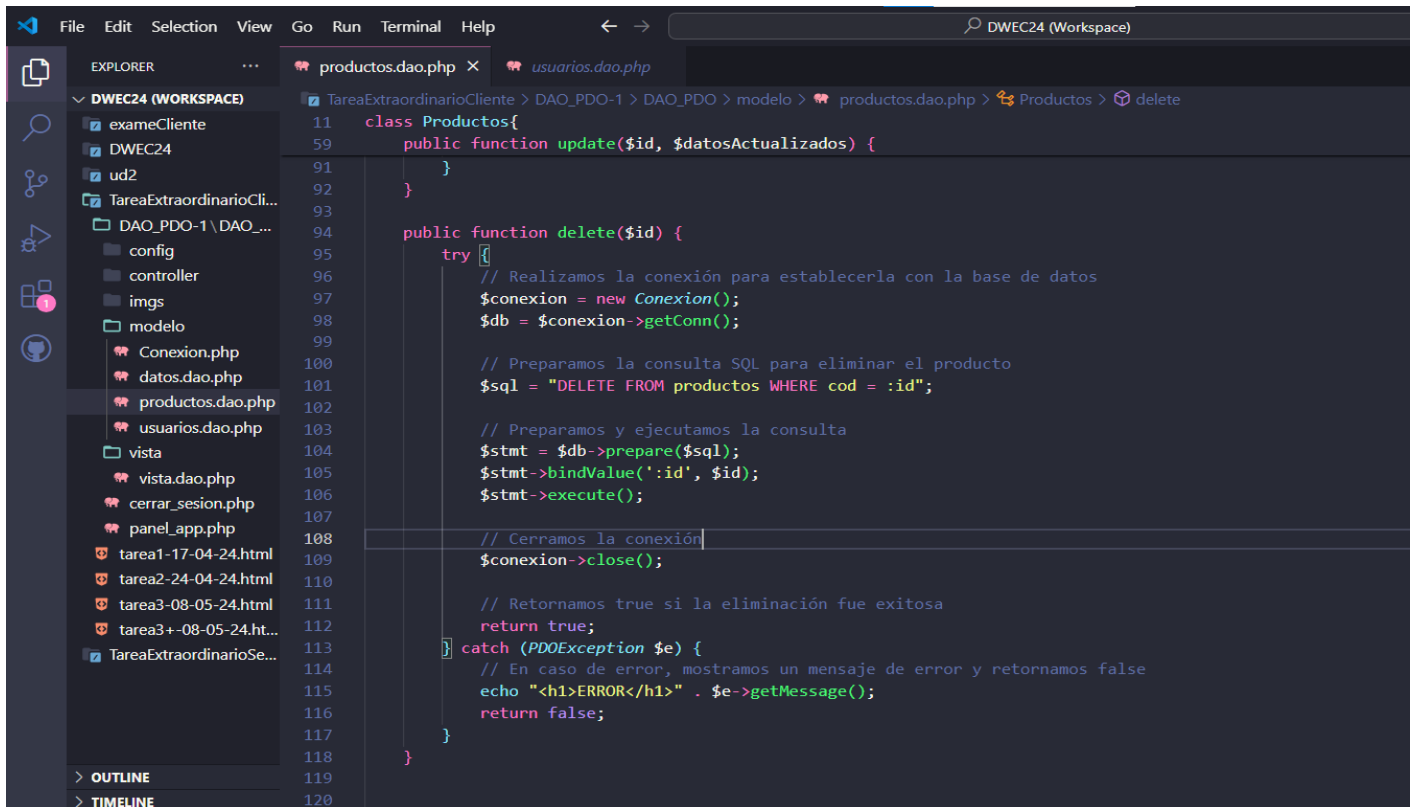
El formulario se genera dinámicamente mediante la vista con HTML que permite al usuario editar o borrar cualquier producto mediante el identificador correspondiente donde dos funciones situadas en el modelo llamadas update y delete que es donde se realiza una conexión con la base de datos y donde se hace la consulta SQL.

2.Sube una captura (sin explicaciones, solo la captura) donde se vea lo explicado en el apartado anterior. (2 puntos)



The screenshot shows a code editor with a file explorer on the left and a code editor on the right. The file explorer shows a project structure with folders like 'examenCliente', 'DWEC24', 'ud2', 'TareaExtraordinarioCli...', 'DAO_PDO-1 \ DAO...', 'config', 'controller', 'imgs', 'modelo', 'Conexion.php', 'datos.dao.php', 'productos.dao.php', 'usuarios.dao.php', 'vista', 'vista.dao.php', 'cerrar_sesion.php', 'panel_app.php', and 'tarea1-17-04-24.html'. The code editor shows the 'update' function in 'productos.dao.php'.

```
11 class Productos{
57 }
58
59 public function update($id, $datosActualizados) {
60     try {
61         // Realizamos la conexión para establecerla con la base de datos
62         $conexion = new Conexion();
63         $db = $conexion->getConn();
64
65         // Preparamos la consulta SQL para actualizar los datos del producto
66         $sql = "UPDATE productos SET ";
67         $setClause = "";
68         foreach ($datosActualizados as $campo => $valor) {
69             $setClause .= "$campo = :$campo, ";
70         }
71         $setClause = rtrim($setClause, ", ");
72         $sql .= $setClause . " WHERE cod = :id";
73
74         // Preparamos y ejecutamos la consulta
75         $stmt = $db->prepare($sql);
76         $stmt->bindValue(':id', $id);
77         foreach ($datosActualizados as $campo => $valor) {
78             $stmt->bindValue(":$campo", $valor);
79         }
80         $stmt->execute();
81
82         // Cerramos la conexión
83         $conexion->close();
84
85         // Retornamos true si la actualización fue exitosa
86         return true;
87     } catch (PDOException $e) {
```



```
11 class Productos{
59     public function update($id, $datosActualizados) {
91     }
92 }
93
94     public function delete($id) {
95         try {
96             // Realizamos la conexión para establecerla con la base de datos
97             $conexion = new Conexion();
98             $db = $conexion->getConn();
99
100             // Preparamos la consulta SQL para eliminar el producto
101             $sql = "DELETE FROM productos WHERE cod = :id";
102
103             // Preparamos y ejecutamos la consulta
104             $stmt = $db->prepare($sql);
105             $stmt->bindValue(':id', $id);
106             $stmt->execute();
107
108             // Cerramos la conexión
109             $conexion->close();
110
111             // Retornamos true si la eliminación fue exitosa
112             return true;
113         } catch (PDOException $e) {
114             // En caso de error, mostramos un mensaje de error y retornamos false
115             echo "<h1>ERROR</h1>" . $e->getMessage();
116             return false;
117         }
118     }
119 }
120
```

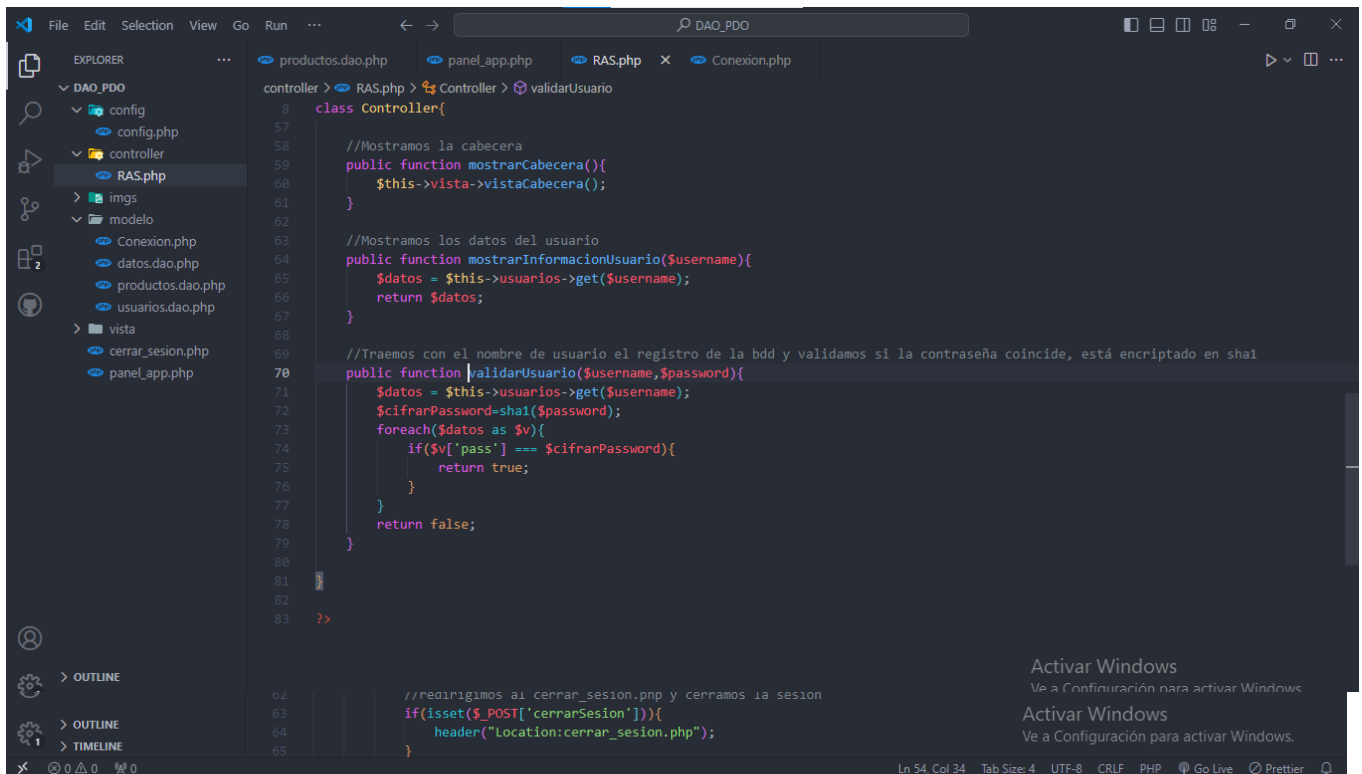
PREGUNTA 2

En relación con la aplicación de carrito de compra desarrollada en clase (UD3)..

1.Explica brevemente en el área de texto cómo se controla el nivel de privilegio del usuario logueado en el proyecto (3 puntos)

Se controla haciendo una validacion al momento de pulsar el boton de iniciar sesion donde con una funcion llamada validarUsuario() valida el username y el password y se asocia al rol que sea correspondiente

2.Sube una captura (sin explicaciones, solo la captura) donde se vea lo explicado en el apartado anterior. (2 puntos)



PREGUNTA 3

El patron MVC...(señala la opción más correcta)

- a) Propone generar vistas estáticas independientes del estado de la aplicación.
- b) Todas las demás opciones son incorrectas.
- c) Propone generar vistas dinámicas que se adaptan al estado de la aplicación.
- d) Propone generar vistas estáticas que se rellenan según el estado de la aplicación.

La respuesta correcta es la c) porque el patrón MVC separa la lógica de la aplicación en tres componentes modelo,vista,controlador por ello en este patron donde se generan vistas dinámicas y se adaptan perfectamente a la aplicación

PREGUNTA 5

El patron MVC...(señala la opción más correcta)

- a) Mejora la mantenibilidad de las aplicaciones pero el desarrollo es sencillo.
- b) Todas las demás opciones son incorrectas.
- c) No mejora la mantenibilidad de las aplicaciones pero es una convención de la comunidad de desarrolladores.
- d) Mejora la mantenibilidad de las aplicaciones pero el desarrollo es complejo.

La respuesta correcta es d) porque precisamente el patrón MVC se utiliza la mejorar la mantenibilidad de las aplicaciones pero es mas complejo porque implica dividir la lógica de la aplicaciones en tres partes distintas

RA5 TRI

PREGUNTA 1 REPETIDA ARRIBA(PREGUNTA 2)

En relación con la aplicación de carrito de compra desarrollada en clase (UD3)..

- 1.Explica brevemente en el área de texto cómo se controla el nivel de privilegio del usuario logueado en el proyecto (3 puntos)
- 2.Sube una captura (sin explicaciones, solo la captura) donde se vea lo explicado en el apartado anterior. (2 puntos)

PREGUNTA 2

En una aplicación desarrollada según el patrón MVC... (señala la opción más correcta)

- a) La capa de modelo se encarga de mantener el estado de la aplicación.
- b) La capa de control se encarga de mantener el estado de la aplicación.
- c) Todas las capas contribuyen al mantenimiento del estado.
- d) Las demás opciones son incorrectas.

La respuesta correcta es la c) debido a que el :

Modelo proporciona los datos y su estado a la vista y el controlador

Vista muestra el estado actual del modelo al usuario y permite al usuario interactuar con la aplicación lo que puede modificar el estado del modelo

Controlador recibe las interacciones del usuario y actualiza el estado del modelo

PREGUNTA 3

En una aplicación desarrollada según el patrón MVC... (señala la opción más correcta)

- a) Todas las capas gestionan los mecanismos de persistencia.
- b) El controlador gestiona la conexión con los mecanismos de persistencia.
- c) Las demás opciones son incorrectas.
- d) El modelo gestiona la conexión con los mecanismos de persistencia.

La respuesta correcta es la d) porque el modelo es responsable de manejar los datos y la lógica de negocio de la aplicación, lo que incluye la interacción con los mecanismos de persistencia,

PREGUNTA 4 REPETIDA ARRIBA(PREGUNTA 3)

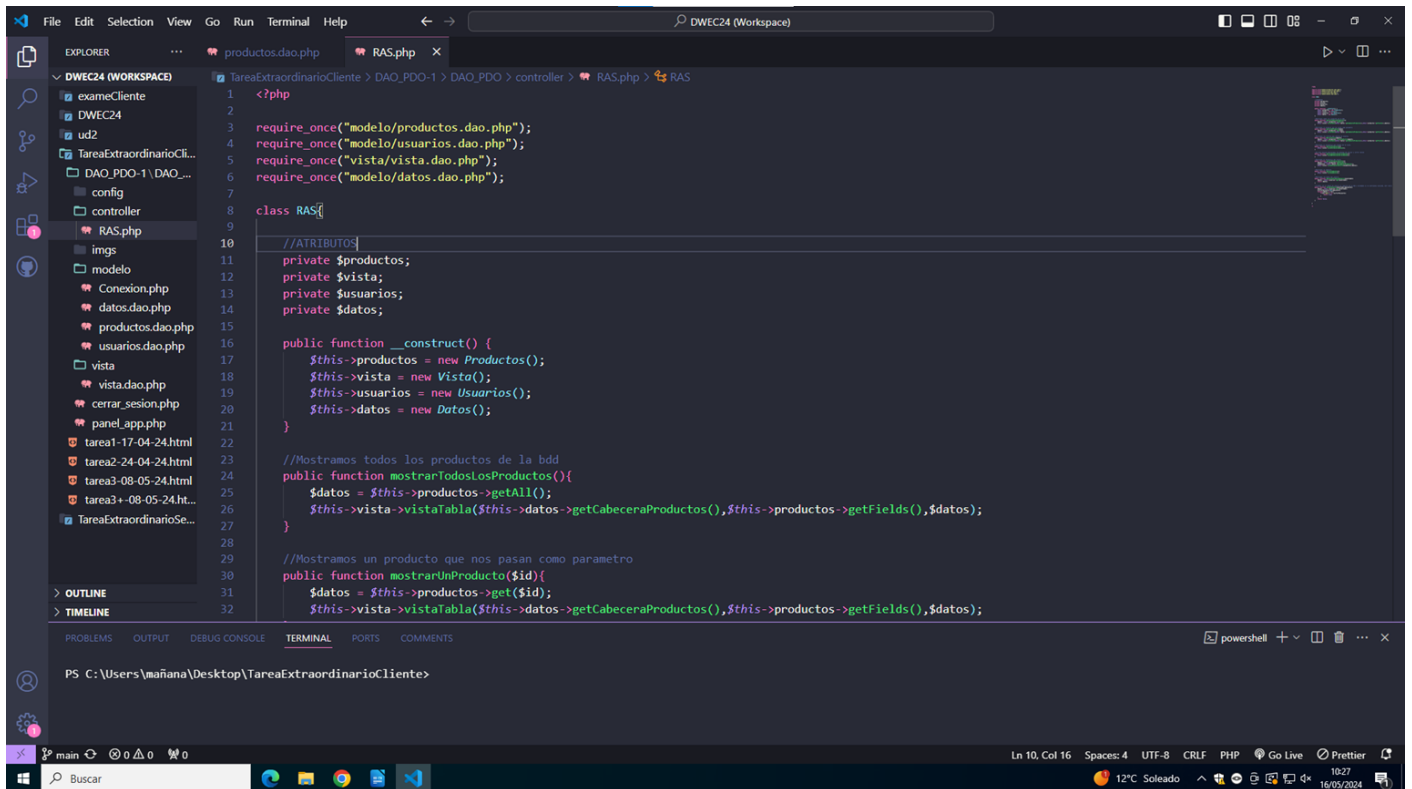
PREGUNTA 5

En relación con la aplicación de carrito de compra desarrollada en clase (UD3)..

1.Explica brevemente en el área de texto cómo se implementa el patrón MVC en el proyecto (3 puntos)

La manera de como se implementa es mediante el controlador que es el que llama a las funciones del modelo para actualizar los datos y también actualiza las vista con la información actualizada del modelo

2.Añade tu identificador de examen al nombre del controlador y sube una captura (sin explicaciones, solo la captura) donde se vea lo explicado en el apartado anterior. (2 puntos)



PREGUNTA 6 REPETIDA ARIIBA(PREGUNTA 5)

PREGUNTA 7

El patron MVC...(señala la opción más correcta)

- a) Todas las demás opciones son correctas.
- b) Exige generar páginas web dinámicas donde los datos y la presentación son independientes.
- c) Exige separar la lógica de negocio y la presentación de una aplicación.
- d) No permite realizar páginas web estáticas.

La respuesta correcta es la c) ya que como se ha comentado antes el patrón MVC consiste en separar las lógica con la aplicación

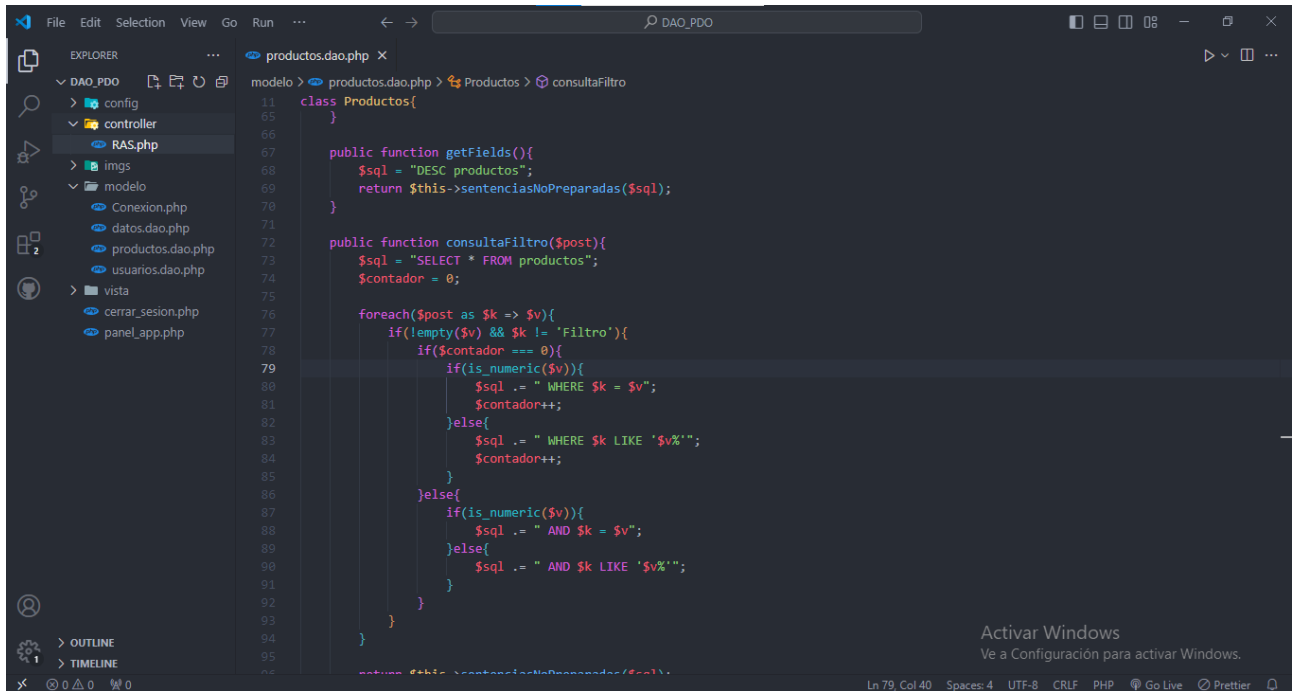
PREGUNTA 8

En relación con la aplicación de carrito de compra desarrollada en clase (UD3).

1.Explica brevemente en el área de texto cómo se genera el formulario para filtrar productos en el proyecto (3 puntos)

Funciona de tal manera que con un \$post como argumento de la funcion consultaFiltro() creamos una sentencia SQL y se va iterando y poniendo condicionales para asi poder crear ese filtro de busqueda

2.Sube una captura (sin explicaciones, solo la captura) donde se vea lo explicado en el apartado anterior. (2 puntos)



```
11 class Productos{
65 }
66
67 public function getFields(){
68     $sql = "DESC productos";
69     return $this->sentenciasNoPreparadas($sql);
70 }
71
72 public function consultaFiltro($post){
73     $sql = "SELECT * FROM productos";
74     $contador = 0;
75
76     foreach($post as $k => $v){
77         if(!empty($v) && $k != 'Filtro'){
78             if($contador === 0){
79                 if(is_numeric($v)){
80                     $sql .= " WHERE $k = $v";
81                     $contador++;
82                 }else{
83                     $sql .= " WHERE $k LIKE '$v%'";
84                     $contador++;
85                 }
86             }else{
87                 if(is_numeric($v)){
88                     $sql .= " AND $k = $v";
89                 }else{
90                     $sql .= " AND $k LIKE '$v%'";
91                 }
92             }
93         }
94     }
95     return $this->sentenciasNoPreparadas($sql);
96 }
```