



**Escuela Superior
de Ingeniería y Tecnología**
Universidad de La Laguna

Administración y Diseño de Bases de Datos:

Desarrollo de Aplicaciones: MySQL y PHP

Alejandro Martín de León
(alu0101015941@ull.edu.es)



Índice:

1. Introducción.	2
2. Estructura de la Base de Datos	2
2.1. Scripts	3
2.2. Triggers	4
3. Detalles sobre el código	5
3.1. Creación de las tablas	5
3.2. Operaciones sobre las tablas	6
3.3. Ordenación de atributos	8
3.4. Búsqueda en las tablas	10
4. Mejora en la visualización de las tablas	11
5. Conclusión	12



1. Introducción.

En el presente informe se pretende desarrollar los pasos que han sido llevados a cabo para abordar el problema propuesto en el enunciado, así como, de detallar el código en PHP que ha sido elaborado.

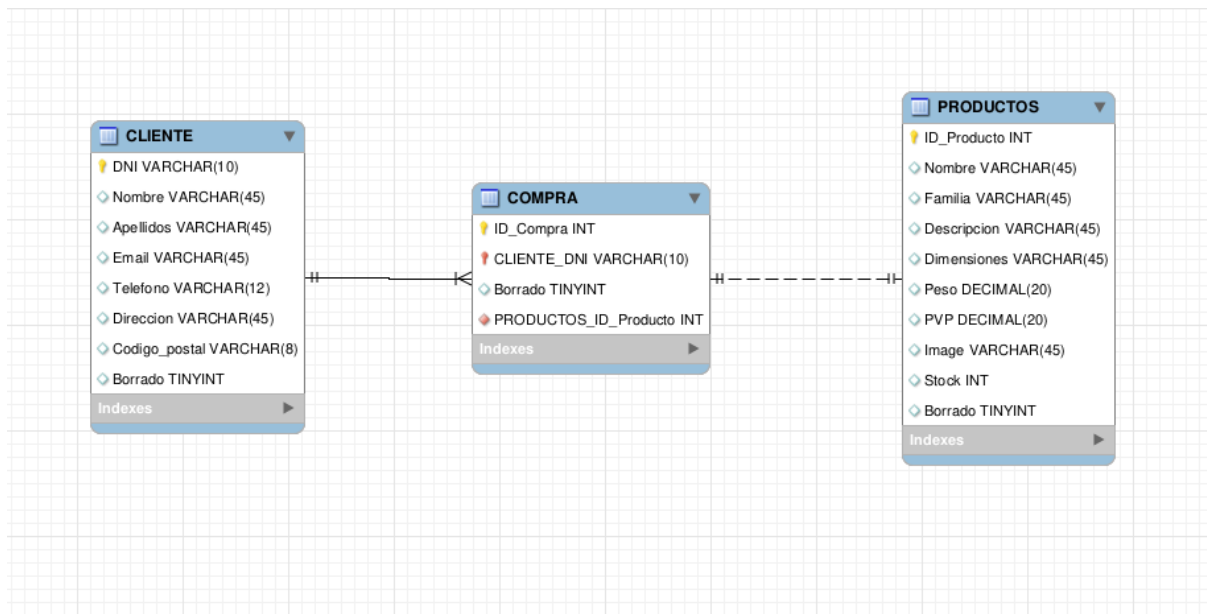
Antes de comenzar con el desarrollo del informe, se exponen los siguientes enlaces:

- Enlace de la web: <http://amarleo.atwebpages.com/>
- Enlace del repositorio Github:
https://github.com/amarleo/ADBD_MySQL_PHP

2. Estructura de la Base de Datos

Respecto a la base de datos, se ha planteado desde un principio almacenar las principales características y datos propios de los **clientes** y **productos** de una tienda online. Además, como forma opcional, se ha añadido el concepto de **compra**, donde por cada fila se relaciona un producto a un cliente.

Para la creación de las tablas en este caso se ha hecho uso del Software MYSQL Workbench, donde se ha elaborado el esquema principal de la base de datos propuesta. A continuación se muestra una captura de pantalla de cómo ha quedado las tablas:



Tal y como se puede observar, la tabla Compra hace de intermediario entre las tablas Cliente y Producto. De esta manera, se puede establecer que tenga como claves ajenas las claves primarias de ambas tablas. Con esto se consigue que si o si, sólomente se puede asociar a una compra un producto y un cliente si ellos mismos ya existen previamente. Otro detalle importante en el modelo planteado, son las elecciones de claves primarias para cada una de las tablas. Se han elegido los elementos únicos que representan a cada una de las tablas.

Por último, comentar que además se ha añadido la columna Borrado a cada tabla, de carácter booleano. Con este atributo se va a poder representar el borrado lógico, donde se van a ocultar algunos registros de las tablas, en vez de realizar un borrado como tal.

2.1. Scripts

Una vez terminado el planteamiento del modelo, se procede a exportar el script, sobre el que se ha editado el formato. El fichero en cuestión se puede ver a través del siguiente [enlace](#).

De esta forma, desde la aplicación MySQL descargada para el stack LAMP, y después de crear un usuario, se ha de poder ejecutar dicho script, siendo capaz de crear la base de datos junto con sus tablas y entradas.



Por otro lado, para poder establecer la base de datos en la web de phpMyAdmin y debido a la versión de SQL que se utiliza, se han tenido que realizar ciertas modificaciones. El fichero en cuestión se encuentra en el siguiente [enlace](#).

2.2. Triggers

Con el objetivo de mejorar en la funcionalidad de la web, se han creado una serie de triggers que serán activados según diferentes supuestos de la práctica. Por ejemplo, será necesario controlar que el email introducido tenga un formato correcto.

En concreto, los triggers que han sido creados son los siguientes:

- **insert_email:** A la hora de insertar un nuevo Cliente, se ha de comprobar que el email tiene un formato acertado. Para ello, se ha hecho uso de expresión regular:

```
CREATE TRIGGER TRIGGER_insert_email BEFORE INSERT ON CLIENTE
FOR EACH ROW BEGIN
  IF NEW.Email NOT REGEXP '^[A-Z0-9._%~]+@[A-Z0-9.-]+\.[A-Z]{2,63}$' THEN
    SIGNAL SQLSTATE VALUE '45000'
    SET MESSAGE_TEXT = '[table:CLIENTE] - Email is not valid';
  END IF;
END;
```

- **insert_DNI:** En este trigger se comprueba que el DNI se componga de 8 dígitos seguido de una letra de las posibles.
- **insert_TLF:** En este caso, se comprueba que el número de teléfono insertado se compone de nueve dígitos.
- **insert_CodPostal:** Para este ejemplo, se comprueba que el código postal introducido se compone de cinco dígitos
- **update_stock:** Este trigger es activado tras insertar una fila en compra. Con esto se consigue que, justo después de confirmar una compra, se disminuya el Stock del producto en la cantidad que ha sido comprada.
- **edit_stock:** Adicionalmente se ha incluido este trigger con el objetivo de evitar el posible error de editar el stock al editar la tabla compra.



3. Detalles sobre el código

Después de comentar los detalles del modelo de datos de la base de datos, se procede a explicar todo lo que ha sido desarrollado en PHP. El primer paso que se ha llevado a cabo ha sido el crear el fichero de conexión a la base de datos de Awardspace, denominado como **bbdd.php**, con el que cada uno de los ficheros serán capaces de realizar una primera conexión a la misma.

Por otro lado, se ha creado el fichero **index.php** que se encarga de ser la página principal de la web, donde se propondrá la elección de acceso a las tablas.

3.1. Creación de las tablas

En primer lugar, se han creado principalmente tres ficheros, siendo cada uno de ellos referentes a cada una de las tablas del modelo. Situando la sobre la tabla Clientes, se ha creado el fichero **cliente.php**. Para poder mostrar los registros de la tabla, se realiza la consulta apropiada y se muestran a través de un bucle todos los registros que contiene la tabla en la base de datos. Se muestra a continuación, tanto donde se recoge la consulta como el bucle:

```
if ($result = mysqli_query($db, "SELECT * FROM CLIENTE WHERE Borrado = 0 ORDER BY $column $sort_order")) {
```

```
while($data = mysqli_fetch_assoc($result)) {

    ?>
    <tr>
        <td <?php echo $column == 'DNI' ? $add_class : ''; ?><?php echo $data['DNI'];?></td>
        <td <?php echo $column == 'Nombre' ? $add_class : ''; ?><?php echo
$data['Nombre'];?></td>
        <td <?php echo $column == 'Apellidos' ? $add_class : ''; ?><?php echo
$data['Apellidos'];?></td>
        <td <?php echo $column == 'Email' ? $add_class : ''; ?><?php echo
$data['Email'];?></td>
        <td <?php echo $column == 'Telefono' ? $add_class : ''; ?><?php echo
$data['Telefono'];?></td>
        <td <?php echo $column == 'Direccion' ? $add_class : ''; ?><?php echo
$data['Direccion'];?></td>
```



```
<td <?php echo $column == 'Codigo_postal' ? $add_class : ''; ?><?php echo  
$data['Codigo_postal']; ?></td>  
<td ><a class = "edit" href="edit.php?id=<?php echo $data['DNI']; ?>">Edit</a></td>  
<td ><a class = "delete" href="delete.php?id=<?php echo $data['DNI'];  
?>">Delete</a></td>  
</tr>  
  
<?php  
}
```

Con el array **\$data** se recogen todos los registros de la tabla y se muestran a través de una tabla gracias a las etiquetas HTML. En esta parte aparecen más elementos de los necesarios, que serán detallados más adelante. Un ejemplo de lo que muestra la pantalla es lo siguiente:

DNI	Nombre	Apellidos	Email	Teléfono	Dirección	Código Postal
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
00000000A	Juan	López Gutierrez	jlogu@gmail.com	600600600	Calle La Pardela	38000
11111111A	Elizabeth	Estévez García	esga@gmail.com	611611611	Calle El Cernicalo	38111
22222222A	Armando	Urdaneta Muñoz	arumu@gmail.com	622622622	Calle El Vencejo	38222
33333333A	Agustín	Marrero Vera	agumave@gmail.com	633633633	Calle La Tórtola	38333
44444444A	Rosario	Villalba Sánchez	rovisa@gmail.com	644644644	Calle El Cuco	38444
55555555A	Lucía	Santana Flores	lusafo@gmail.com	655655655	Calle El Paíño	38555
66666666A	Dario	Andrade Ibáñez	dandi@gmail.com	666666666	Calle El Alcatraz	38666
77777777A	Lily	Martínez Murillo	limarmu@gmail.com	677677677	Calle La Garza	38777
88888888A	Olivia	Espinoza Martín	oliesma@gmail.com	688688688	Calle El Avetorillo	38888

Este mismo procedimiento se utiliza para representar las tablas compra y producto en la web.

3.2. Operaciones sobre las tablas

El objetivo ahora es conseguir que para cada una de las tablas se pueda editar, insertar y borrar registros. Cada una de estas operaciones serán detalladas a continuación.

- **Editar:** Se ha creado un fichero **edit.php** en el que se van a realizar las operaciones para cada una de las tablas. A través de una variable booleana se comprueba desde qué fichero se ha llegado a edit.php y se realizan las operaciones pertinentes en cada caso. Además, con la siguiente sentencia se permite pasar la referencia del DNI del cliente que va a ser editado en cuestión:

```
<td ><a class = "edit" href="edit.php?id=<?php echo $data['DNI']; ?>">Edit</a></td>
```



Siguiendo con el ejemplo de Clientes, se crea un nuevo formulario sobre el que se va a poder editar cualquiera de las columnas de la entrada seleccionada. Una vez pulsado el botón, se realiza un update sobre la tabla con los registros, tanto los editados como los que no fueron editados.

```
if(isset($_POST['update'])) {
    $newDNI = $_POST['DNI'];
    $newNombre = $_POST['Nombre'];
    $newApellidos = $_POST['Apellidos'];
    $newEmail = $_POST['Email'];
    $newTelefono = $_POST['Telefono'];
    $newDireccion = $_POST['Direccion'];
    $newCodigo_postal = $_POST['Codigo_postal'];

    $editarCliente = "UPDATE CLIENTE SET Nombre = '$newNombre', Apellidos = '$newApellidos', Telefono = '$newTelefono', Email = '$newEmail', Telefono = '$newTelefono', Direccion = '$newDireccion', Codigo_postal = '$newCodigo_postal' WHERE DNI = '$dni'";

    if(mysqli_query($db, $editarCliente)){
        echo "Updated Successfully!";
        ?>
        <a href="cliente.php?">return</a>
        <?php
    } else{
        echo "ERROR: Could not able to execute $editarCliente. " . mysqli_error($db);
    }
}
} else{
    echo "ERROR: Could not able to execute $mostrarCliente. " . mysqli_error($db);
}
}
```

Finalmente se comprueba si la edición se ha realizado correctamente.

- **Insertar:** El caso de insertar es similar al anterior. Se ha creado un fichero aparte denominado **insert.php** sobre el que se comprueba desde qué tabla se accede a este fichero. Posteriormente, se recogen todos los nuevos registros a insertar y se realiza la consulta:

```
$sql = "INSERT INTO PRODUCTOS (Nombre, Familia, Descripcion, Dimensiones, Peso, PVP, Image, Stock, Borrado) VALUES
```




```
(' $enterNombre', '$enterFamilia', '$enterDescripcion', '$enterDimensiones', '$enterPeso', '$enterPVP', '$enterImage', '$enterStock', 0);
```

- **Eliminar:** Al igual que los dos casos anteriores, se procede a comprobar desde qué fichero se ha llegado a **delete.php**. Una vez hecho esto, simplemente se realiza una actualización del atributo “Borrado” de la tabla.

```
if (isset($display_cliente) && $display_cliente == TRUE) {  
    $dni = $_GET['id'];  
    $borrarCliente = "UPDATE CLIENTE SET Borrado = 1 WHERE DNI = '$dni'";  
    if(mysqli_query($db, $borrarCliente)){  
        echo "Deleted successfully DNI = '$dni'";  
        ?>  
        <a href="cliente.php?">return</a>  
        <?php  
    } else{  
        echo "ERROR: Could not able to execute $borrarCliente. " . mysqli_error($db);  
        ?>  
        <a href="cliente.php?">return</a>  
        <?php  
    }  
}
```

3.3. Ordenación de atributos

Para el ordenamiento de atributos de las tablas, se han creado como links la cabecera de las tablas. Al ser pulsadas, ordena de forma ascendente o descendente.

Entrando en los detalles del código, se han creado una serie de variables que contienen datos necesarios a la hora de realizar la consulta.

```
$columns = array('DNI','Nombre','Apellidos', 'Email', 'Telefono', 'Direccion',  
'Codigo_postal');  
$column = isset($_GET['column']) && in_array($_GET['column'], $columns) ? $_GET['column'] :  
$columns[0];  
$sort_order = isset($_GET['order']) && strtolower($_GET['order']) == 'desc' ? 'DESC' :  
'ASC';
```

Como se puede ver con anterioridad, se ha creado un array con todas las columnas posibles de entre las que se compone la tabla de clientes. En la última línea se comprueba cuál ha sido el orden elegido.



Con estas variables se podrán sustituir sobre la consulta de búsqueda sql, de forma que:

```
"SELECT * FROM CLIENTE WHERE Borrado = 0 ORDER BY $column $sort_order"
```

Posteriormente, sobre cada una de las cabeceras de la tabla se han creado links personalizados dependiendo del atributo elegido, mostrando además el orden. Por ejemplo, para la columna DNI:

```
<th><a class = "table-title" href="cliente.php?column=DNI&order=<?php echo $asc_or_desc;
?>">DNI<i class = "<?php echo $column == 'DNI' ? '-' : '' . $up_or_down : '' ?>"></i></a></th>
```

A continuación, se muestra un ejemplo de cómo se ordenan de forma ascendente.

Búsqueda

DNI	Nombre	Apellidos	Email	Teléfono	Direccion	Código Postal
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>

Tabla Cliente

DNI	Nombre	Apellidos	Email	Teléfono	Direccion	Código Postal		
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	Insert	
00000000A	Juan	López Gutierrez	jlogu@gmail.com	600600600	Calle La Pardela	38000	Edit	Delete
11111111A	Elizabeth	Estévez García	esga@gmail.com	611611611	Calle El Cernicalo	38111	Edit	Delete
22222222A	Armando	Urdaneta Muñoz	arumu@gmail.com	622622622	Calle El Vencejo	38222	Edit	Delete
33333333A	Agustín	Marrero Vera	agumave@gmail.com	633633633	Calle La Tórtola	38333	Edit	Delete
44444444A	Rosario	Villalba Sánchez	rovisa@gmail.com	644644644	Calle El Cuco	38444	Edit	Delete
55555555A	Lucía	Santana Flores	lusafo@gmail.com	655655655	Calle El Paño	38555	Edit	Delete
66666666A	Darío	Andrade Ibáñez	dandi@gmail.com	666666666	Calle El Alcatraz	38666	Edit	Delete
77777777A	Lily	Martínez Murillo	limarmu@gmail.com	677677677	Calle La Garza	38777	Edit	Delete
88888888A	Olivia	Espinoza Martín	oliesma@gmail.com	688688688	Calle El Avetorillo	38888	Edit	Delete

[Return](#)

Se puede comprobar que toda la cabecera de la tabla Cliente ahora es reactiva al ser pulsada. También se observa cómo cambia el link en función del orden. Ahora, para el caso descendiente:



← → No es seguro | amario.atwebpages.com/cliente.php?column=DNI&orden=desc

Aplicaciones Gmail YouTube Maps Mi unidad - Go... How to Sort T... Free Web Hos... triggers - MYS... Lista de lectura

Búsqueda

DNI	Nombre	Apellidos	Email	Teléfono	Dirección	Código Postal	
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="button" value="Search"/>

Tabla Cliente

DNI	Nombre	Apellidos	Email	Teléfono	Dirección	Código Postal	
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="button" value="Insert"/>
88888888A	Olivia	Espinoza Martin	oliesma@gmail.com	688688688	Calle El Avetorillo	38888	<input type="button" value="Edit"/> <input type="button" value="Delete"/>
77777777A	Lily	Martinez Murillo	limarmu@gmail.com	677677677	Calle La Garza	38777	<input type="button" value="Edit"/> <input type="button" value="Delete"/>
66666666A	Dario	Andrade Ibáñez	dandi@gmail.com	666666666	Calle El Alcatraz	38666	<input type="button" value="Edit"/> <input type="button" value="Delete"/>
55555555A	Lucia	Santana Flores	lusafo@gmail.com	655655655	Calle El Paillo	38555	<input type="button" value="Edit"/> <input type="button" value="Delete"/>
44444444A	Rosario	Villalba Sánchez	rovisa@gmail.com	644644644	Calle El Cuco	38444	<input type="button" value="Edit"/> <input type="button" value="Delete"/>
33333333A	Agustin	Marrero Vera	agumave@gmail.com	633633633	Calle La Tórtola	38333	<input type="button" value="Edit"/> <input type="button" value="Delete"/>
22222222A	Armando	Urdaneta Muñoz	arurmu@gmail.com	622622622	Calle El Vencejo	38222	<input type="button" value="Edit"/> <input type="button" value="Delete"/>
11111111A	Elizabeth	Estévez García	esga@gmail.com	611611611	Calle El Cernícalo	38111	<input type="button" value="Edit"/> <input type="button" value="Delete"/>
00000000A	Juan	López Gutiérrez	jlogu@gmail.com	600600600	Calle La Pardela	38000	<input type="button" value="Edit"/> <input type="button" value="Delete"/>

3.4. Búsqueda en las tablas

Por cada una de las tablas se ha creado un pequeño formulario donde se podrá introducir uno o más campos de campos a buscar sobre la misma tabla. De esta manera, se podrá filtrar el contenido de la tabla en cuestión dependiendo de si ha encontrado alguna fila que incluya lo introducido en el formulario.

El procedimiento que ha sido llevado a cabo en cada una de las tablas es similar. Creado el nuevo formulario html, se procede a recoger todos los elementos dependiendo de la columna en la que se encuentra. Estos se pasan por una consulta sql y se imprimen todas las filas resultantes. Esto es posible gracias a:

```
if(isset($_GET['search'])) {  
    $fields = array('DNI', 'Nombre', 'Apellidos', 'Email', 'Telefono', 'Direccion',  
'Codigo_Postal');  
    $conditions = array();  
  
    foreach($fields as $field) {  
        //if the field is set and not empty  
        if(isset($_POST[$field]) && $_POST[$field] != '') {  
            $conditions[] = " $field = '" .  
mysqli_real_escape_string($db,$_POST[$field]) . "'";  
        }  
    }  
  
    $query = "SELECT * FROM CLIENTE";  
    if(count($conditions) > 0) {
```



```
$query .= " WHERE " . implode(" AND ", $conditions);  
}  
$result = mysqli_query($db, $query);  
}
```

Se adjunta una imagen de ejemplo:

DNI	Nombre	Apellidos	Email	Teléfono	Dirección	Código Postal	
<input type="text"/>	<input type="text" value="Juan"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="button" value="Search"/>

DNI	Nombre	Apellidos	Email	Teléfono	Dirección	Código Postal	
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="button" value="Insert"/>
00000000A	Juan	López Gutiérrez	jlogu@gmail.com	600600600	Calle La Pardela	38000	<input type="button" value="Edit"/> <input type="button" value="Delete"/>
21212121C	Juan	Toledo García	jutoga@gmail.com	621621621	Calle El Zarapito	38212	<input type="button" value="Edit"/> <input type="button" value="Delete"/>
21212121D	Juan	Sierra Mesa	jusime@gmail.com	612613614	Calle La Tórtola	38121	<input type="button" value="Edit"/> <input type="button" value="Delete"/>

Otro ejemplo, buscando por más de un campo:

Búsqueda

DNI	Nombre	Apellidos	Email	Teléfono	Dirección	Código Postal	
<input type="text"/>	<input type="text" value="Juan"/>	<input type="text" value="Toledo García"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="button" value="Search"/>

Tabla Cliente

DNI	Nombre	Apellidos	Email	Teléfono	Dirección	Código Postal	
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="button" value="Insert"/>
21212121C	Juan	Toledo García	jutoga@gmail.com	621621621	Calle El Zarapito	38212	<input type="button" value="Edit"/> <input type="button" value="Delete"/>

4. Mejora en la visualización de las tablas

Adicionalmente, con el objetivo de adornar un poco el resultado de la web, se ha procedido a añadir CSS sobre la aplicación. Sin entrar demasiado en detalles del código, comentar que el repositorio se ha seccionado de manera que se encuentre el código puro php y por otro lado, el que tiene decoración con css. Este último se encuentra en el siguiente [enlace](#).



5. Conclusión

PHP es un lenguaje del lado del servidor que tiene miles de propósitos, donde principalmente se utiliza para generar contenido dinámico en una web.

Durante el desarrollo de este proyecto, no sólo he aprendido a trabajar con este lenguaje, sino que también he comprendido en qué entornos puede funcionar, qué herramientas son las más frecuentemente utilizadas, así como su propia metodología de trabajo y sintaxis. Por otro lado, también ha sido curioso aprender más sobre el stack LAMP, que aunque ha sido complicado de instalar en local, si es verdad que me ha aportado gran velocidad en lo que desarrollar se refiere.