



Benemérita Universidad
Autónoma de Puebla



Facultad de Ciencias de la
Computación

Implementación de la pestaña “Películas rentadas” para los clientes

Actividad 9

Materia:

Aplicaciones web

Integrantes del equipo:

Rosa Suarez Calderon

Ana Laura Torres Marin

Alfredo de Jesus Ramos Benavides

Profesor:

Pedro Bello López

Periodo:

Primavera 2022

Fecha de entrega:

29 de marzo de 2022

Desarrollo del trabajo

Para el trabajo es necesario que se agregue una nueva pestaña en la barra de navegación para que el usuario pueda observar las películas que ha rentado, para ello, tenemos que acceder a la base de datos, a la tabla de rentas y extraer los datos de esa misma. Por lo menos, se debe de mostrar el título de la película, la fecha de renta y la fecha de entrega.

Implementación de la sentencia SQL.

Para saber que películas a rentado cierto usuario debemos de entrar a la base de datos y acceder a la tabla de “*rentas*” que hemos creado.

```
mysql> select * from rentas;
```

fecha_inicio	fecha_fin	id_cliente	id_pelicula
2022-02-15	2022-02-19	3	6
2022-02-20	2022-02-22	1	5
2022-02-20	2022-02-22	1	3
2022-02-20	2022-02-22	1	4
2022-02-20	2022-02-22	2	4
2022-03-24	2022-03-26	3	3
2022-03-26	2022-03-28	3	1
2022-03-26	2022-03-28	8	5

Figura 1. Tabla “*rentas*” de la base de datos “*videoteca*”.

Como observamos en la figura 1, solo vemos datos numéricos y lo que deseamos es que aparezca el nombre del usuario y el nombre de la película que ha rentado, en la figura vemos que solo tenemos el número del cliente y el numero de la película.

Para poder tener la información que deseamos utilizamos la cláusula **INNER JOIN**. Esta cláusula busca coincidencias entre 2 tablas, en función a una columna que tienen en común. De tal modo que, sólo la intersección se mostrara en los resultados [1].

En nuestro caso el campo *id_cliente* pertenece a la tabla clientes, mientras que el campo *id_pelicula* pertenece a la tabla película, por lo que, debemos de trabajar con dos **INNER JOIN** para poder acceder a ambas tablas desde los campos antes mencionados.

Ahora que sabemos que necesitamos usar dos **INNER JOIN**, debemos de tener claro los campos que debemos de utilizar y las tablas donde se encuentra la información. A continuación, se muestra la sentencia que se construyó para obtener los datos:

```
SELECT      rentas.id_cliente,      rentas.fecha_inicio,      rentas.fecha_fin,
clientes.cliente, pelicula.titulo FROM rentas
```

```
INNER JOIN clientes ON rentas.id_cliente =clientes.id_cliente
```

```
INNER JOIN pelicula ON rentas.id_pelicula = pelicula.id_pelicula;
```

```
mysql> SELECT rentas.id_cliente, rentas.fecha_inicio, rentas.fecha_fin, clientes.cliente, pelicula.titulo FROM rentas
-> INNER JOIN clientes ON rentas.id_cliente = clientes.id_cliente
-> INNER JOIN pelicula ON rentas.id_pelicula = pelicula.id_pelicula;
```

id_cliente	fecha_inicio	fecha_fin	cliente	titulo
3	2022-02-15	2022-02-19	Jose Luis Lopez	La guerra de las galaxias
1	2022-02-20	2022-02-22	Jorge Perez	Tron
1	2022-02-20	2022-02-22	Jorge Perez	Doce monos
1	2022-02-20	2022-02-22	Jorge Perez	Contact
2	2022-02-20	2022-02-22	Juan Dominguez	Contact
3	2022-03-24	2022-03-26	Jose Luis Lopez	Doce monos
3	2022-03-26	2022-03-28	Jose Luis Lopez	Blade Runner
8	2022-03-26	2022-03-28	Laura Torres	Tron

Figura 2. Sentencia SQL con INNER JOIN

La figura 2 muestra el resultado obtenido de usar la sentencia SQL antes mencionada, como observamos, ya no nos muestra el *id_cliente* y el *id_pelicula*, sino que ahora nos aparece el nombre del cliente y el título de la película y era justamente esto lo que requeríamos. Si queremos agregar más campos por mostrar solo debemos de agregarlos.

Una vez teniendo la sentencia, se procedió a implementarlo en la página web y ver si es que realmente funciona.

Implementación en la página web.

Para implementarlo en la página web, primero creamos una vista llamada “**Películas rentadas**”, como se mencionó antes, se mostrarán las películas que el usuario a rentado.

Como ya se vio el tema de sesiones activas en la clase, debemos de tener en cuenta lo siguiente: “Cada vez que un usuario ingrese a la plataforma solo debe de ver su propia información y no la de alguien más y, mucho menos debe de escoger que información ver”. Por ello, una vez que un usuario inicie sesión solo vera su información y también solo vera las películas que ha rentado. A continuación, se muestra la implementación:

Antes de implementar la sentencia SQL, se definió una variable (Figura 3) de sesión, con esta variable, se obtiene la información del cliente.

```
$id = $_SESSION['kuser'];
```

Figura 3. Variable de sesión

Como mostramos en la 2, cuando usamos la sentencia SQL, se observa que se obtiene toda la información almacenada en la tabla, pero no deseamos que esto ocurra. Lo que queremos es que solo se obtenga la información del usuario que ha iniciado sesión, para ello agregamos la siguiente clausula: `WHERE clientes.cliente = '$id'`, con esta cláusula indicamos que solo muestra la información del usuario que ha iniciado sesión, a continuación, se muestra la sentencia actualizada:

```
"SELECT rentas.id_cliente, rentas.fecha_inicio, rentas.fecha_fin, clientes.cliente, pelicula.titulo, pelicula.imagen FROM rentas
INNER JOIN clientes ON rentas.id_cliente = clientes.id_cliente
INNER JOIN pelicula ON rentas.id_pelicula = pelicula.id_pelicula
WHERE clientes.cliente = '$id';";
```

Figura 4. Sentencia actualizada

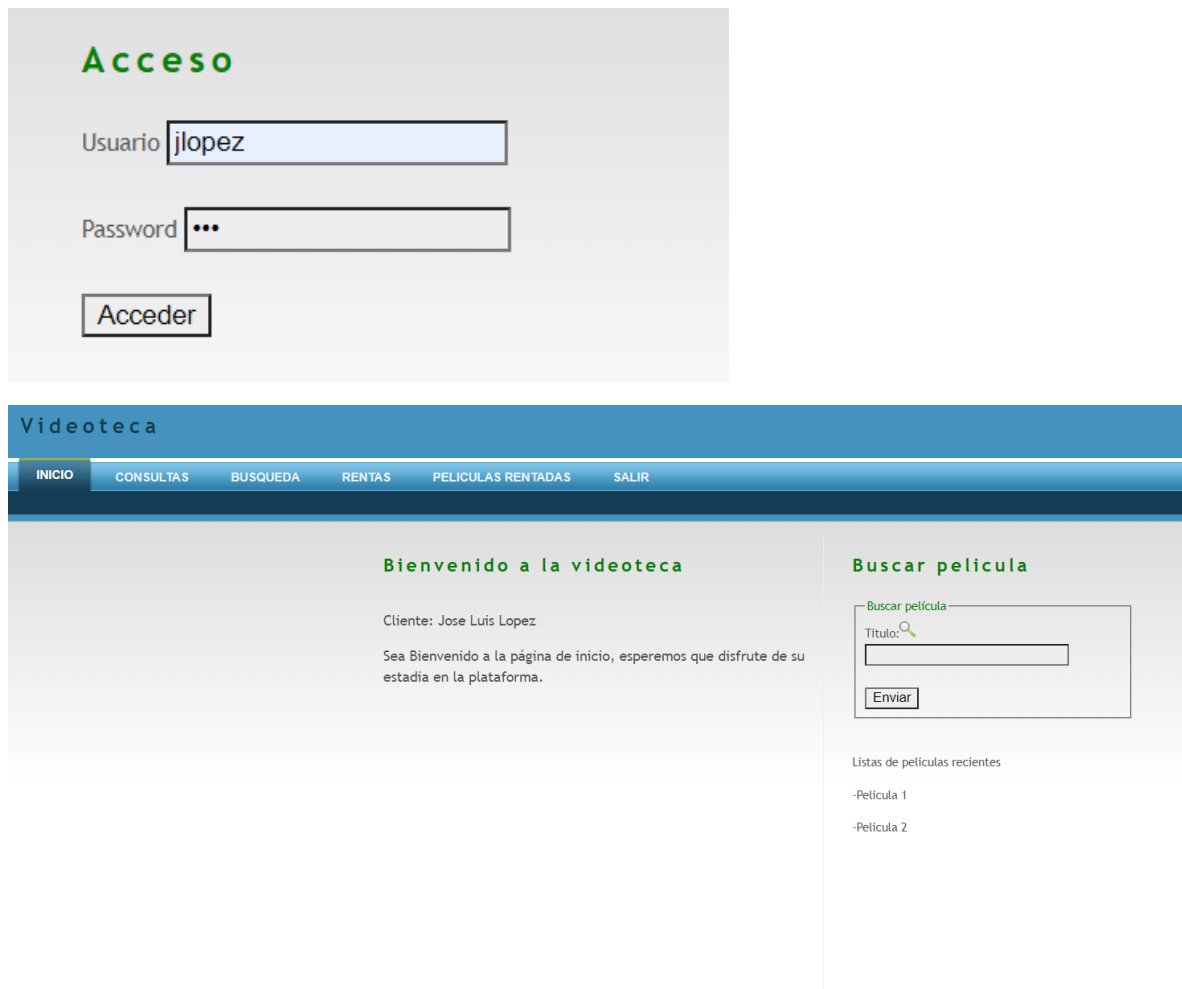
Una vez hecho esto, ahora solo queda mostrar la información en la página web, a continuación, se muestra su implementación:

```
echo "<table border='1'>";
echo "<TR><TD> Cliente</TD> <TD> Pelicula </TD> <TD> Fecha de renta </TD> <TD> Entrega de pelicula </TD> <TD> Imagen </TD></TR>";
while ($row=mysqli_fetch_array($result))
{
    $nom=$row['cliente'];
    $nom_peli= $row['titulo'];
    $fecha_i=$row['fecha_inicio'];
    $fecha_f=$row['fecha_fin'];
    $im=$row['imagen'];

    echo"<TR><TD>$nom</TD> <TD>$nom_peli</TD> <TD>$fecha_i</TD> <TD>$fecha_f</TD>
        <TD><img src='img/$im' width='80' height='80'/></TD></TR>";
}
mysqli_free_result($result);
mysqli_close($link);
echo"</table>";
```

Figura 5. Implementación para la observación de los datos

Capturas de pantalla del funcionamiento del programa.



Videoteca

INICIO

CONSULTAS

BUSQUEDA

RENTAS

PELICULAS RENTADAS

SALIR

Películas rentadas

Cliente	Pelicula	Fecha de renta	Entrega de pelicula	Imagen
Jose Luis Lopez	Blade Runner	2022-03-26	2022-03-28	
Jose Luis Lopez	Doce monos	2022-03-24	2022-03-26	
Jose Luis Lopez	La guerra de las galaxias	2022-02-15	2022-02-19	

Buscar pelicula

Buscar pelicula

Titulo:

Enviar

Listas de peliculas recientes

-Pelicula 1

-Pelicula 2

Agregamos una nueva renta al cliente Jose Luis Lopez

Renta de pelicula

Página donde el cliente puede rentar una pelicula

SpiderMan

▼

Enviar

Renta de pelicula

Pelicula rentada con exito

Id del cliente: 3

Id de la pelicula: 7

La fecha de entrega es: 2022-03-29 a partir del 2022-03-27

Videoteca

INICIO
CONSULTAS
BUSQUEDA
RENTAS
PELICULAS RENTADAS
SALIR

Películas rentadas

Cliente	Pelicula	Fecha de renta	Entrega de pelicula	Imagen
Jose Luis Lopez	Blade Runner	2022-03-26	2022-03-28	
Jose Luis Lopez	Doce monos	2022-03-24	2022-03-26	
Jose Luis Lopez	La guerra de las galaxias	2022-02-15	2022-02-19	
Jose Luis Lopez	SpiderMan	2022-03-26	2022-03-28	

Buscar pelicula

Buscar película
Titulo:

Listas de películas recientes

- Pelicula 1
- Pelicula 2

Como se observa, en las fechas de las últimas dos imágenes vemos que las fechas difieren, esto es debido a que toma la zona horaria de América, y cuando se muestra la tabla de las películas rentadas, se ajusta a la zona horaria de México.

Ingresamos con otro usuario para visualizar que tiene otras películas rentadas a las que tiene el cliente Jose Luis Lopez. Accedemos con el usuario laurat que le pertenece a Laura Torres:

Acceso

Usuario

Password

Videoteca

INICIO

CONSULTAS

BUSQUEDA

RENTAS

PELICULAS RENTADAS

SALIR

Películas rentadas

Cliente	Petícula	Fecha de renta	Entrega de película	Imagen
Laura Torres	Tron	2022-03-26	2022-03-28	

Buscar película

Buscar película

Titulo:

Enviar

Listas de películas recientes

-Película 1

-Película 2

Referencias

[1] ¿Cómo funciona INNER JOIN, LEFT JOIN, RIGHT JOIN y FULL JOIN? (2020, 9 de septiembre). Programación y más | Aprende desarrollo web y móvil. <https://programacionymas.com/blog/como-functiona-inner-left-right-full-join>

<https://code.tutsplus.com/es/tutorials/how-to-use-sessions-and-session-variables-in-php--cms-31839>

https://www.youtube.com/watch?v=aUcD_HlkNw0