Práctica: sistema de toma de pedidos para restaurante

Servidor - Joaquín Lorca Nieto

****

C.E.S. Vega Media

Desarrollo de Aplicaciones Web – 2º D.A.W.

**Índice**

[**1.** **Modelo relacional de la BBDD.** 2](#_Toc188353072)

[**2.** **Manual de uso: camarero.** 6](#_Toc188353073)

[**3.** **Manual de uso: encargado.** 8](#_Toc188353074)

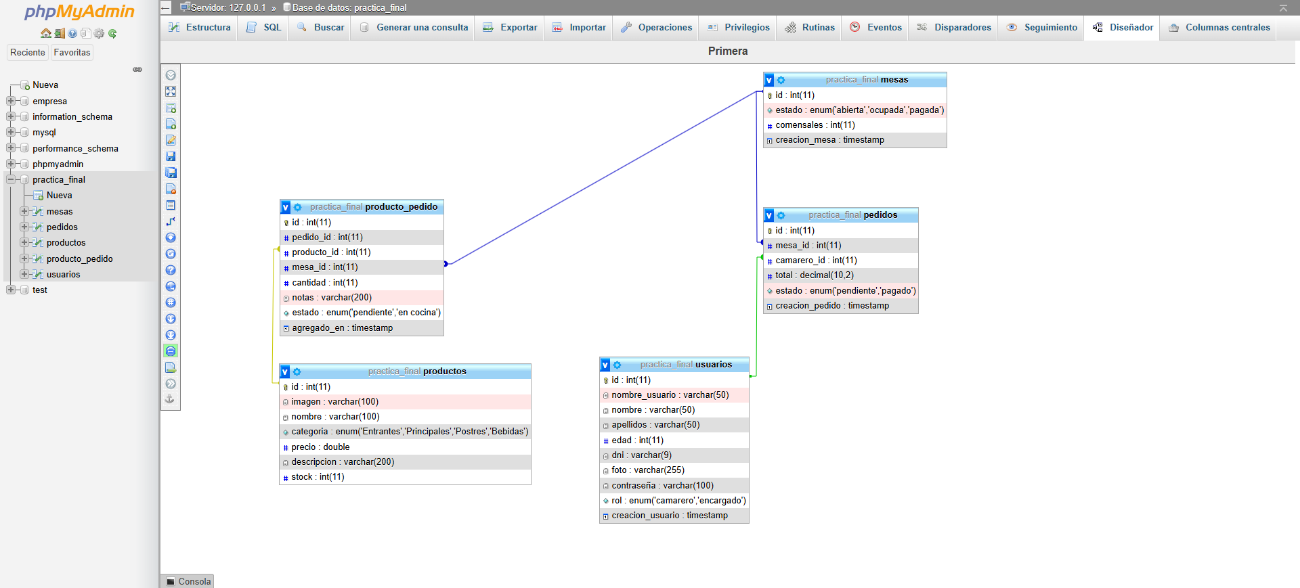
[**4.** **Problemas encontrados.** 11](#_Toc188353075)

[**5.** **Propuestas de mejora.** 13](#_Toc188353076)

[**6.** **Conclusiones finales.** 14](#_Toc188353077)

# **Modelo relacional de la BBDD.**

Esta es mi BBDD utilizada en la aplicacción del restaurante.



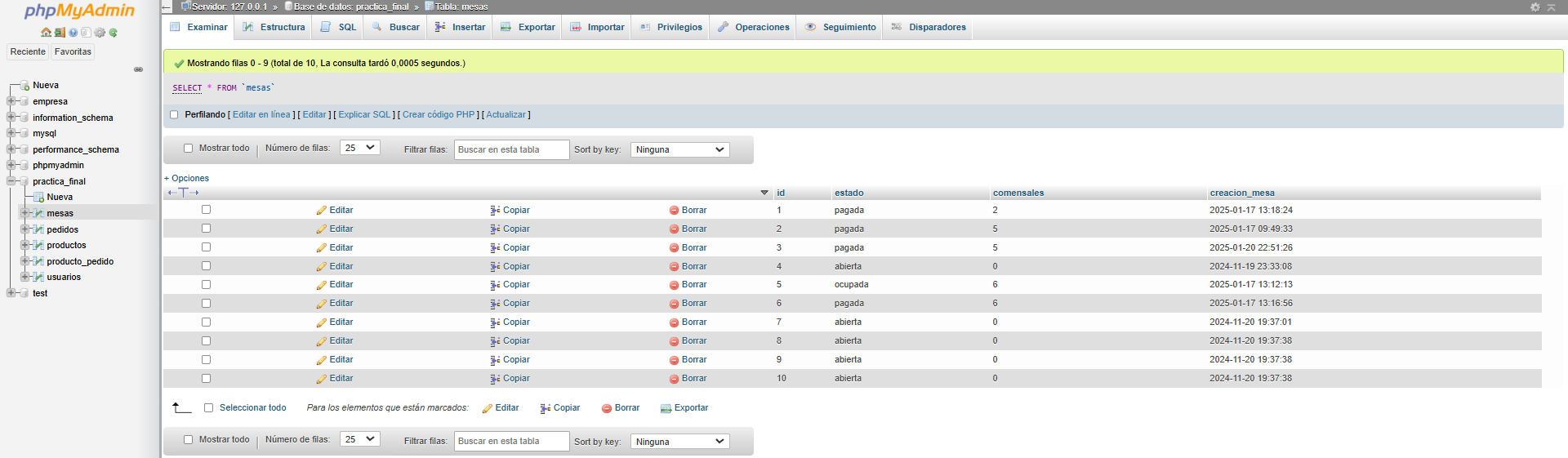
**Tablas y relaciones**



1. **Mesas**

Registra las mesas disponibles en nuestro restaurante, son fijas, una cantidad de 10 mesas. De cada mesa se guarda su estado (abierta, ocupada, pagada), nº de comensales y fecha de creación.

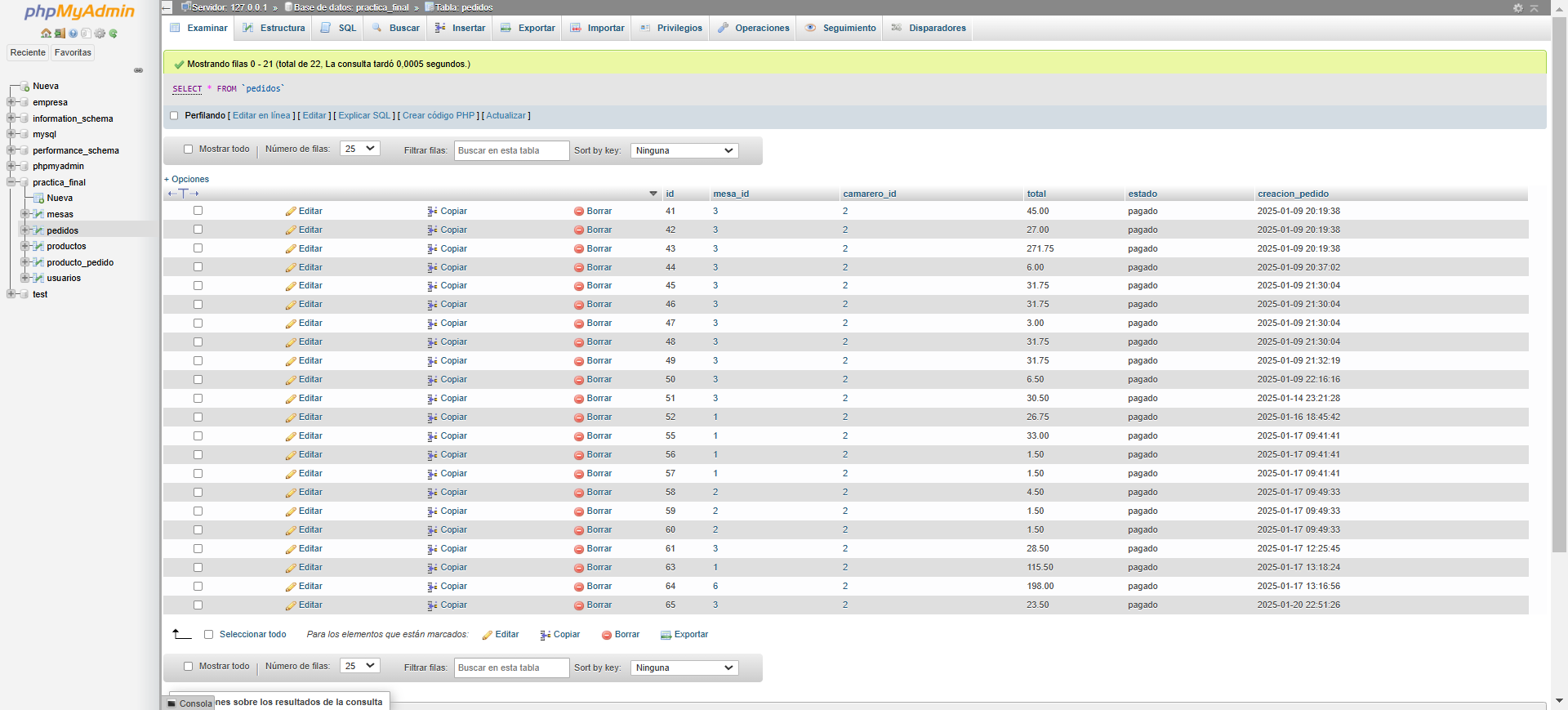
* + Columnas: id, estado, comensales, creacion\_mesa
  + Relaciones: Relacionada con la tabla pedidos y producto\_pedido.



1. **Pedidos**

Guarda los pedidos creados por los camareros al tomar nota a los clientes. Dentro de cada pedido se registra la mesa, el camarero, el total del pedido, su estado (pendiente o pagado), y su fecha de creación.

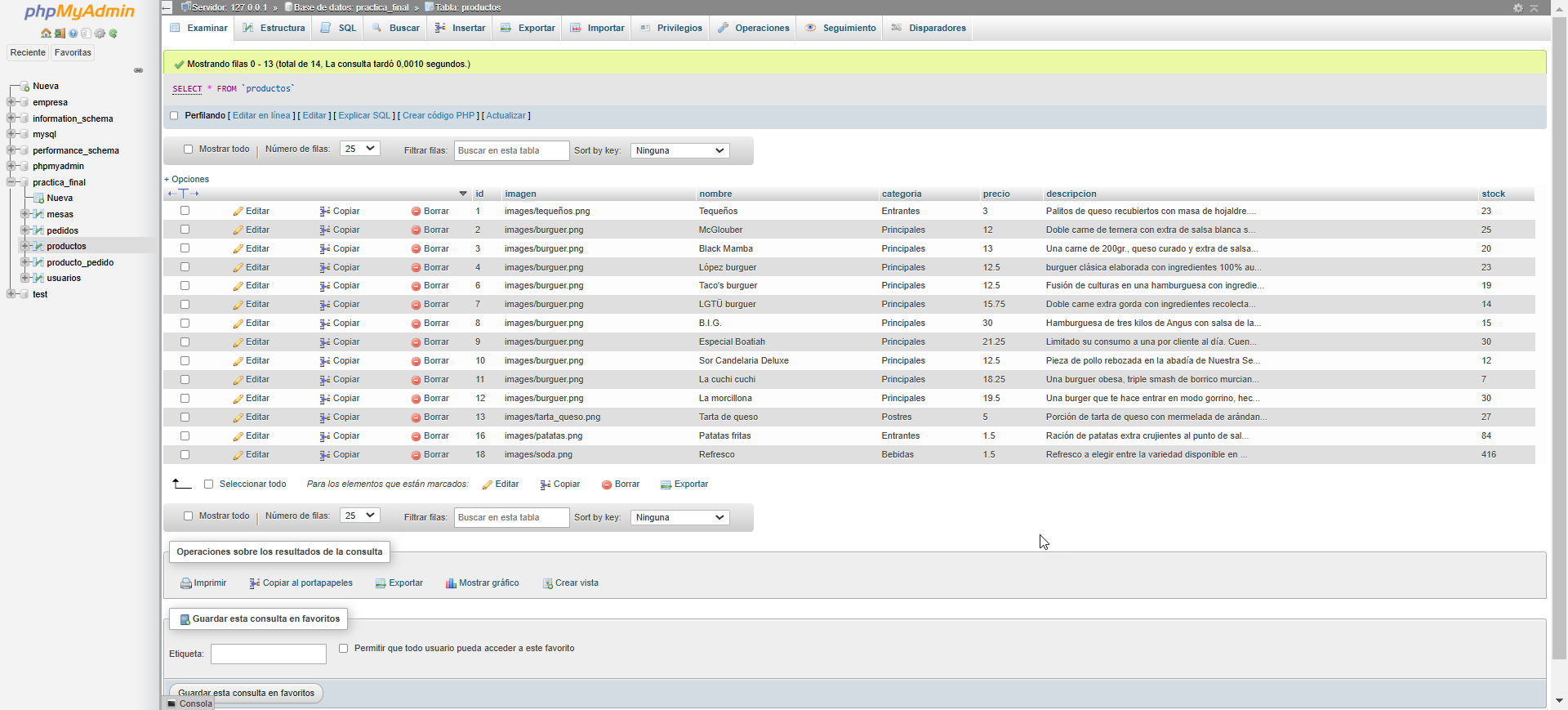
* + Columnas: id, mesa\_id, camarero\_id, total, estado, creacion\_pedido
  + Relaciones:
    - FK: mesa\_id → mesas(id) (ON DELETE CASCADE, ON UPDATE CASCADE)
    - FK: camarero\_id → usuarios(id) (ON DELETE CASCADE, ON UPDATE CASCADE)



1. **Productos**

Equivale al menú del restaurante. Se inserta de cada producto una imagen descriptiva, su nombre, precio por unidad o ración, categoría (entrante, principal, postre o bebida), descripción del producto, y el stock del producto.

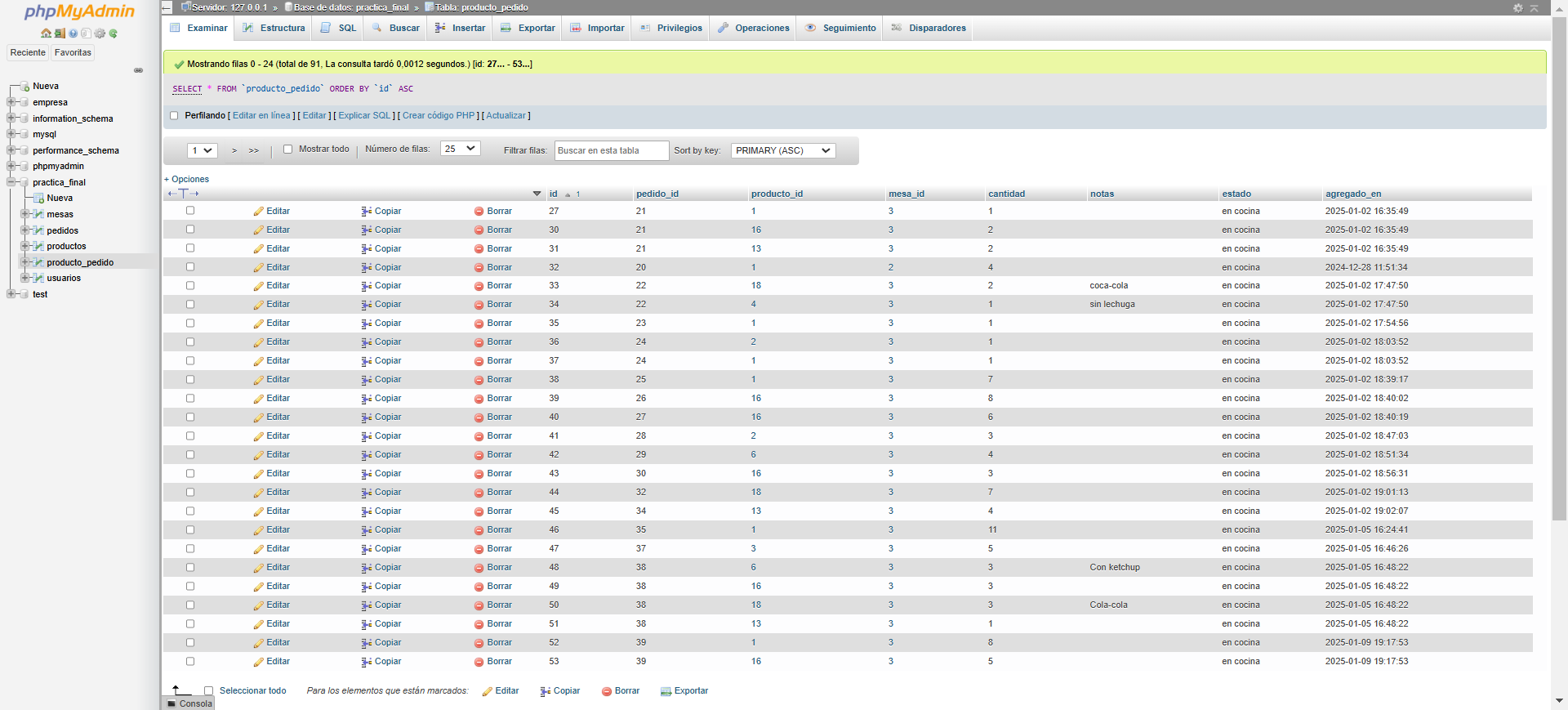
* + Columnas: id, imagen, nombre, categoria, precio, descripcion, stock
  + Relaciones: Relacionada con producto\_pedido.



1. **Producto\_Pedido**

Esta tabla hace referencia a la comanda del camarero, es decir, los productos pedidos por el cliente, que componen el pedido. De cada comanda o producto pedido se guarda la cantidad o unidades que quiere el cliente de dicho producto, notas de cocina, el estado (pendiente de envío a cocina o en cocina), y su fecha de agregación al pedido.

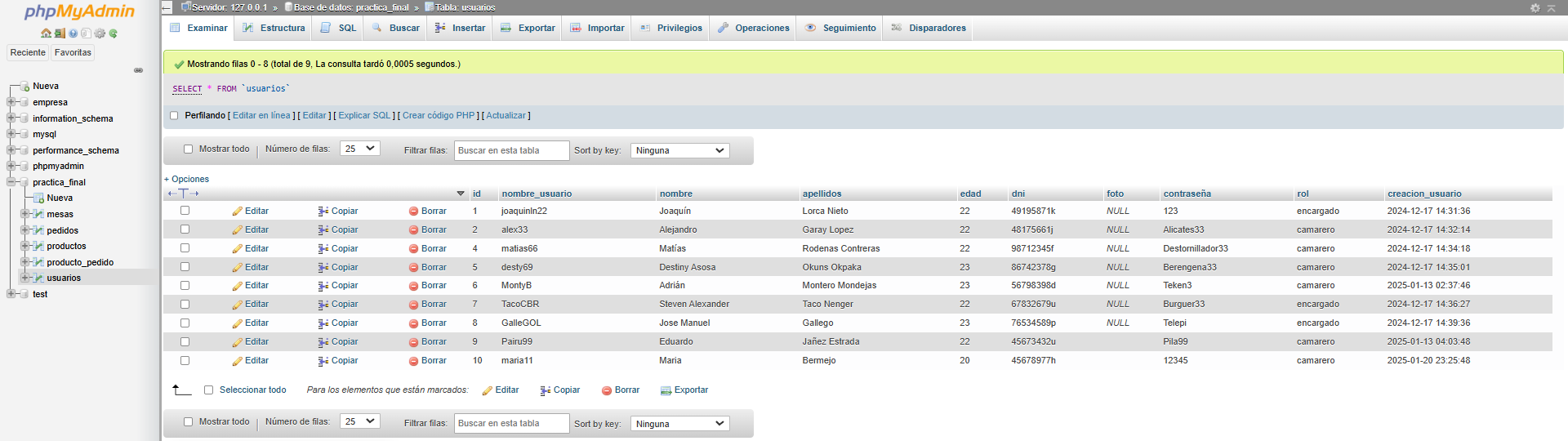
* + Columnas: id, pedido\_id, producto\_id, mesa\_id, cantidad, notas, estado, agregado\_en.
  + Relaciones:
    - FK: producto\_id → productos(id)
    - FK: mesa\_id → mesas(id) (ON DELETE CASCADE, ON UPDATE CASCADE)



1. **Usuarios**

El registro de todas las cuentas de usuario para camareros y encargados. Gracias al rol guardado cada cuenta puede tener acceso a una parte u otra de la aplicación.

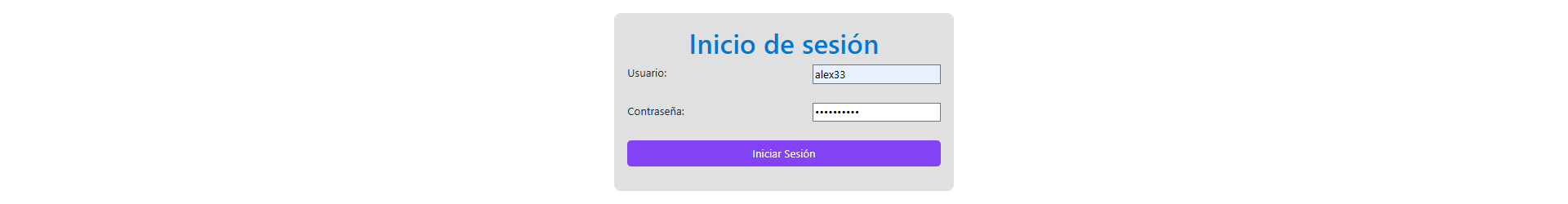
* + Columnas: id, nombre\_usuario, nombre, apellidos, edad, dni, foto, contraseña, rol, creacion\_usuario.
  + Relaciones: Relacionada con la tabla pedidos a través de camarero\_id.



**Relación General**

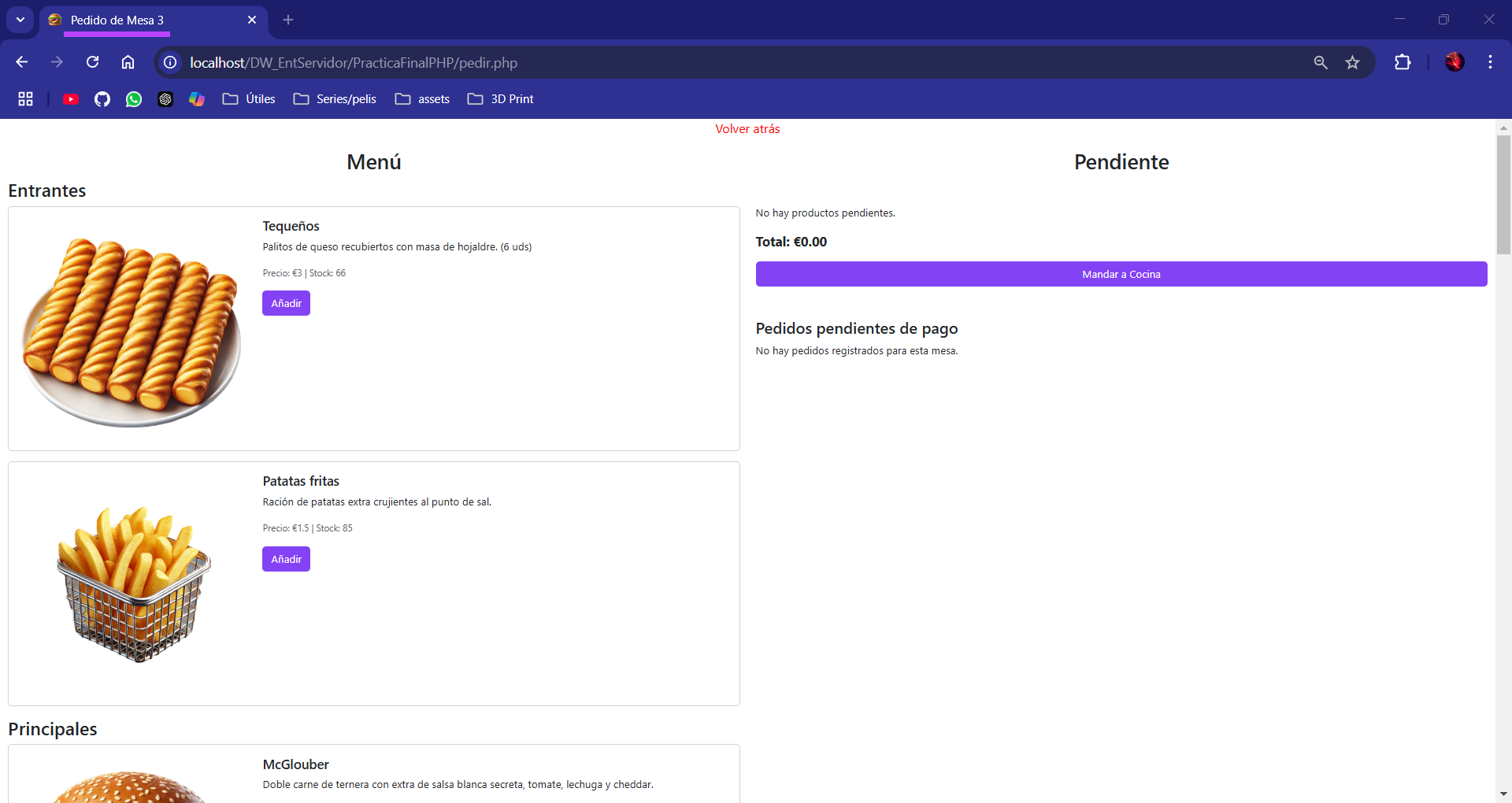
* **Mesas** es la tabla principal, vinculada con pedidos y producto\_pedido.
* **Pedidos** está vinculada tanto a las mesas como a los usuarios que actúan como camareros y encargados.
* **Productos** es una tabla independiente que registra detalles de los artículos disponibles en el menú.
* **Producto\_Pedido** sirve como tabla de intersección entre los productos, pedidos y mesas, permitiendo registrar cantidades y estados específicos.

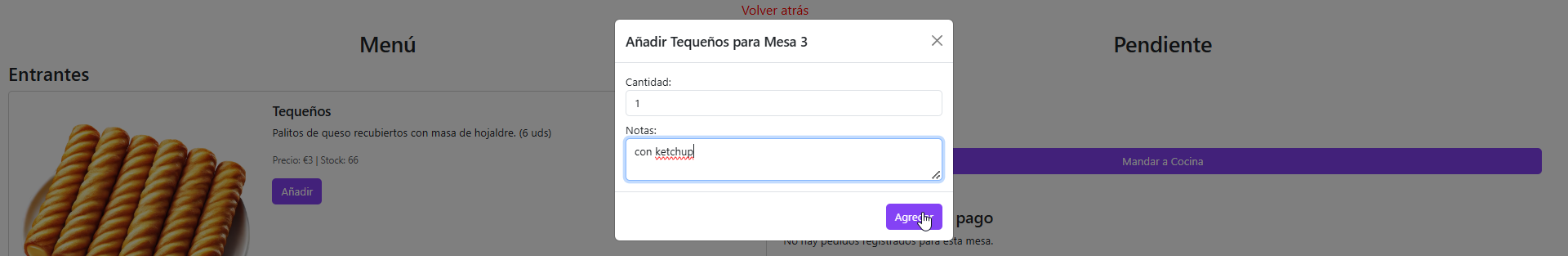
# **Manual de uso: camarero.**

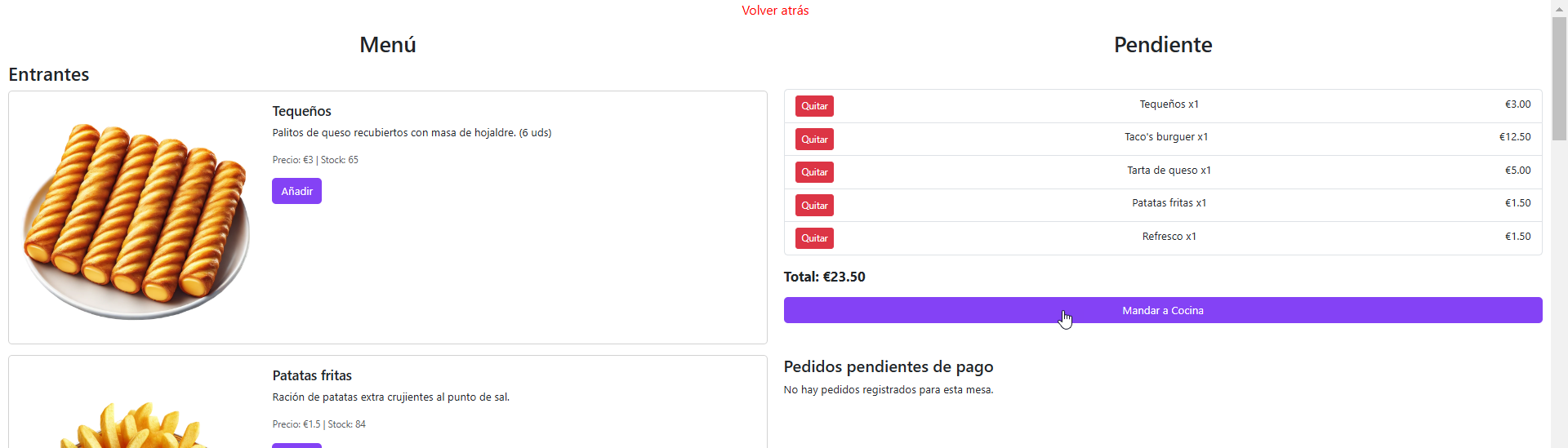
Primero pinchamos nos registramos con la cuenta de uno de los camareros. 

Después seleccionamos una de las mesas “abiertas” o “pagadas” para comenzar a tomar el pedido. Para ello rellenamos el formulario en la parte superior, donde indicaremos nº de mesa y nº de comensales, y clicamos en “crear”. 

Ahora la mesa 3 pasa a estar ocupada y ya podremos pinchar sobre ella para comenzar el pedido. 

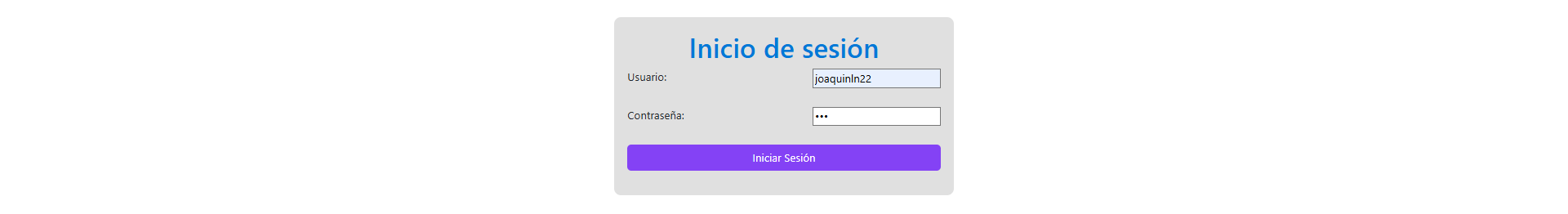
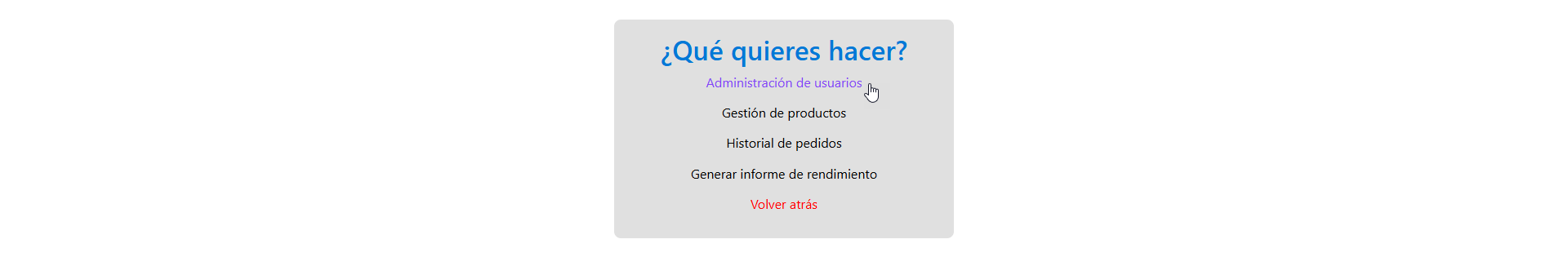


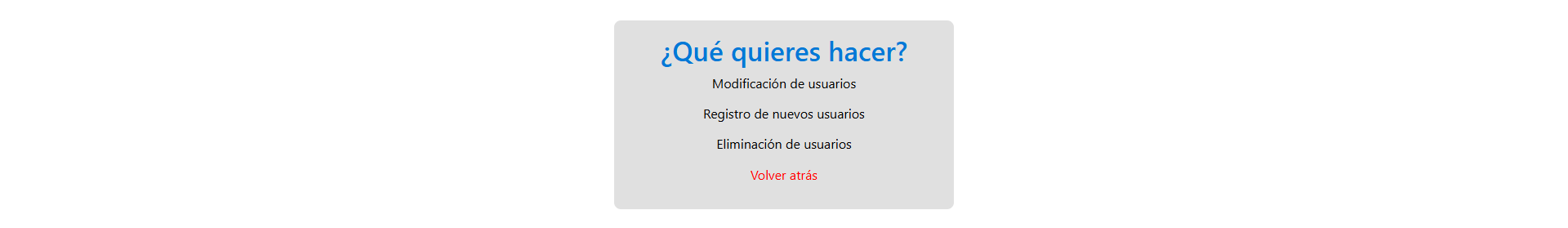
Ahora, solo resta agregar productos al listado de productos pendientes, para despues mandarlos a cocina y crear nuestro pedido. Pinchamos en el botón “añadir” del producto que queramos de nuestro menú y se desplegará una ventana modal con un formulario. Seleccionamos cantidad, escribimos notas de cocina, y pinchamos en agregar. 

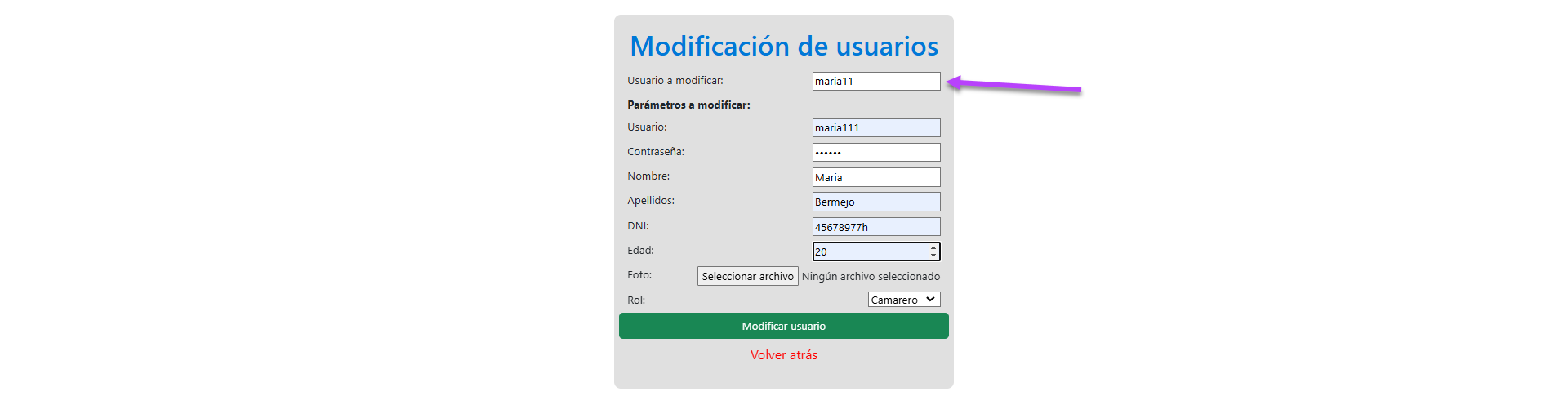
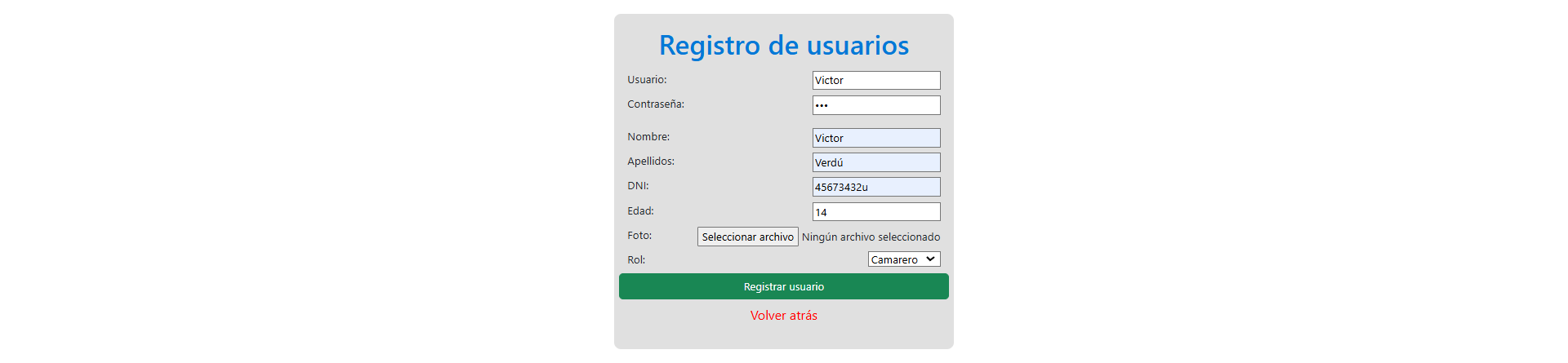
Una vez tenemos todos los productos deseados en el listado de “pendiente”, clicamos en “mandar a cocina”, para crear el pedido, cambiar el estado de los productos a “en cocina”, y generar el ticket de pedido. El estado del pedido se establece como “pendiente”.  

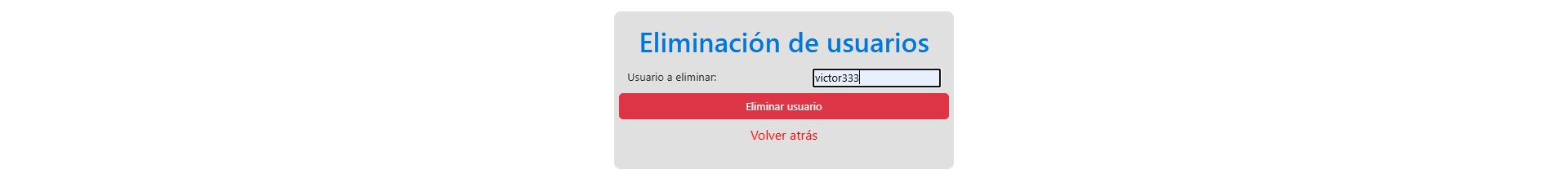
Para cerrar el pedido y marcarlo como “pagado”, simplemente clicamos en “pagar todos”, el pedido o pedidos pendientes de pago desaparecerán de este listado y la mesa pasará a estar “pagada”, con lo cual podrá volver a usarse para tomar otro pedido.

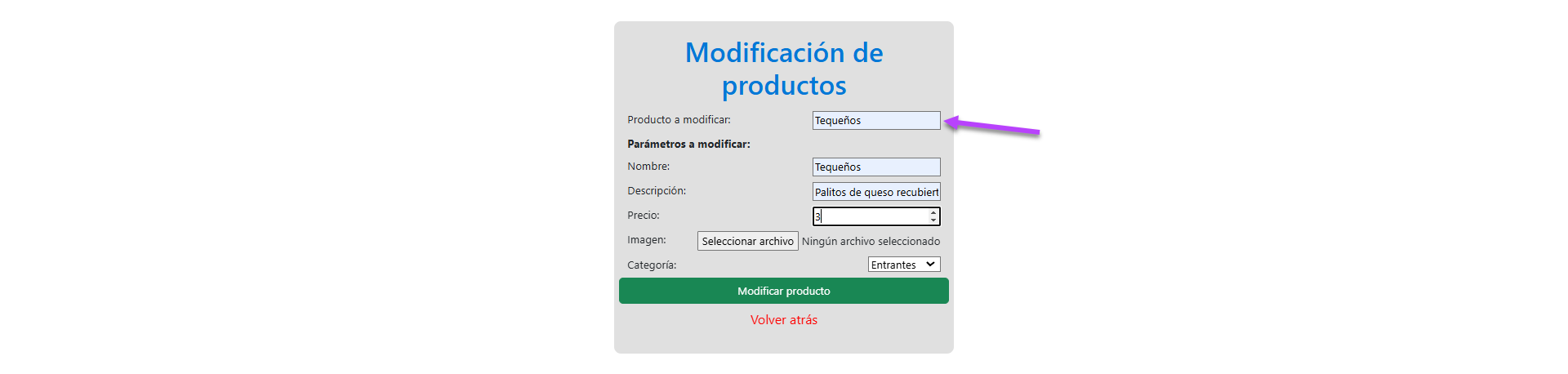
# **Manual de uso: encargado.**

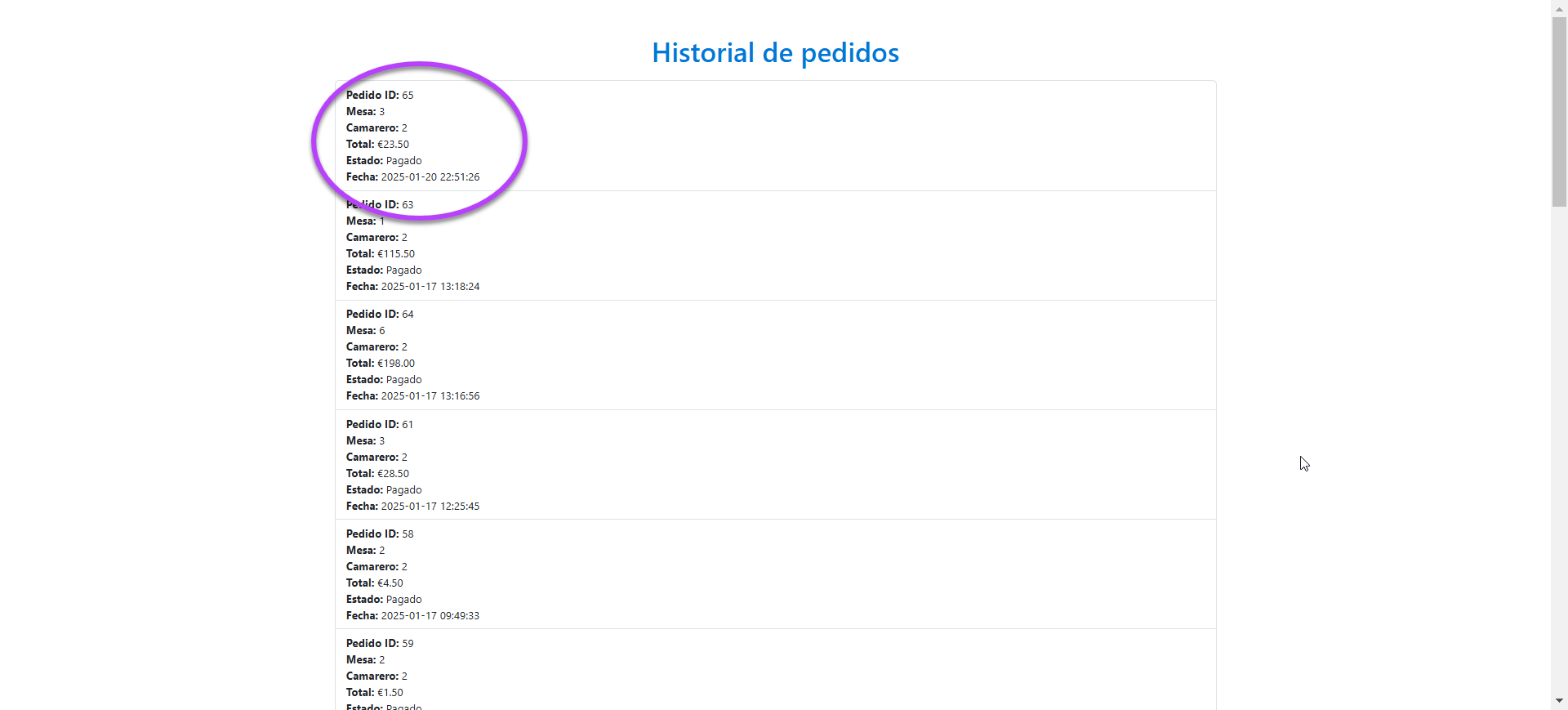
Como antes, rellenamos los datos del formulario con la cuenta de uno de los encargados y pinchamos en “iniciar sesión” para acceder al menú de encargados.  

En el **primer apartado** tenemos la “administración de usuarios” con tres opciones, modificación, registro y eliminación de usuarios. Cada opción nos abre una pagina nueva con un formulario para rellenar los datos necesarios para llevar acabo cada función.

En “modificación de usuarios” tenemos que seleccionar el usuario a modificar, y despues rellenar los campos que queremos modificar. Tenemos que rellenarlos todos, excepto si queremos dejar alguno como nulo. Después pinchamos en el boton “modificar usuario” y ya se habrá actualizado la base de datos.En cuanto al formulario de “registro de usuarios” simplemente rellenamos el formulario y clicamos en el boton verde de “registrar usuario”, y se registrará nuestro nuevo usuario en la base de datos.

Y para eliminar un usuario escribimos el usuario a eliminar y se ejecutará la consulta para llevar a cabo su eliminación de la base de datos. 

En el **segundo apartado** del menú de encargados tenemos la opción de “gestión de productos”, con cuatro opciones, modificación, agregado, eliminación y actualización. Para modificar un producto, simplemente introducimos el nombre de cuyo producto queremos modificar y rellenamos el resto del formulario con los datos que queremos cambiar, y pinchamos en “modificar producto”.Para el “registro de productos”, igual rellenamos los datos del formulario y al clicar en el botón se registrará un nuevo producto en la tabla productos.Ahora, para eliminar, al igual que para eliminar usuarios, escribimos el nombre del producto y pinchamos en el botón rojo.Y, por último, si queremos actualizar las existencias de un producto, introducimos el nombre de dicho producto y la cantidad de stock que vamos a ponerle. Clicamos en el botón “actualizar” y se integrará el nuevo stock en la base de datos.

Ahora, nos situamos en el **tercer apartado** del menú de opciones del encargado, en la opción de “historial de pedidos”. Está opción nos muestra un listado en tiempo real de todos los pedidos guardados en la base de datos.

Y, por último, en el **cuarto apartado** del menú, encontramos la opción de “generar informe de rendimiento”, la cual nos aporta una tabla a modo de informe sobre las estadísticas diárias de nuestro restaurante. Nos muestra, fecha, ingresos, nº de pedidos y nº de comensales.

# **Problemas encontrados.**

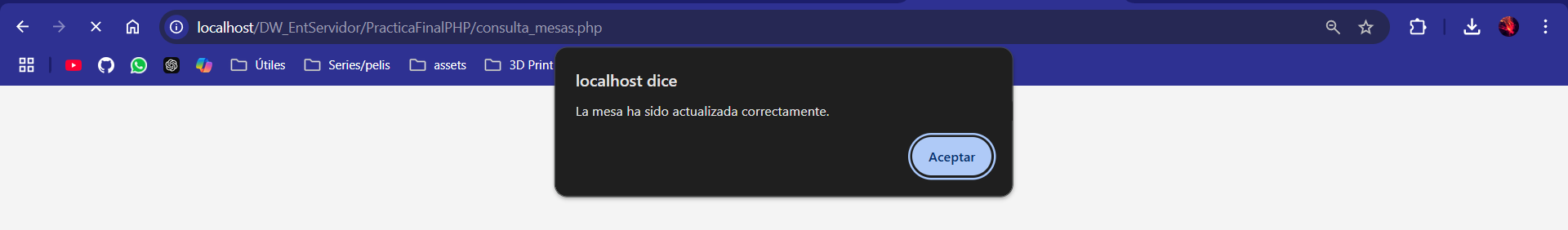
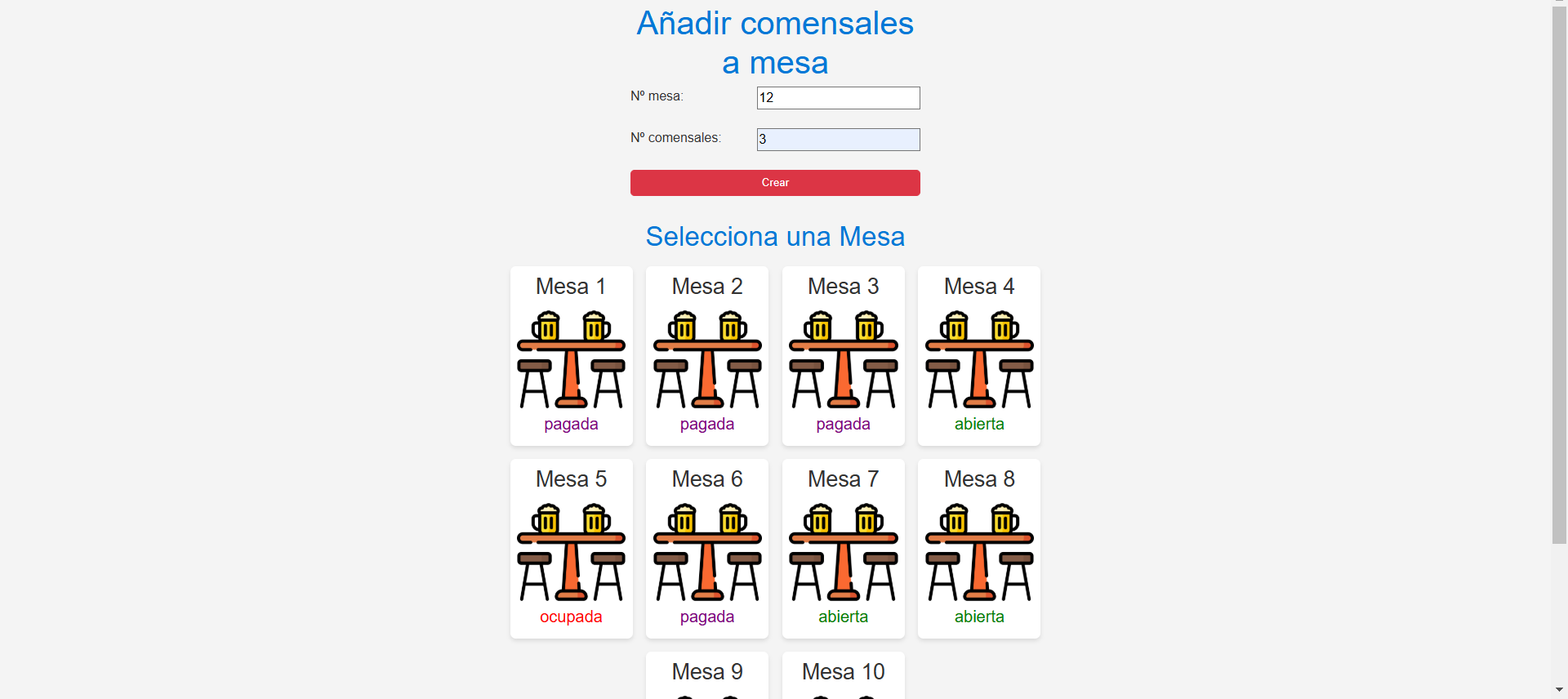
* Fallo en seguridad:

Me falto tener en cuenta que el navegador guarda los datos de inicio de sesión, es decir, al ingresar como camarero o encargado la sesión se guarda para acceder a las siguientes paginas, pero si cogemos y copiamos la URL e intentamos acceder desde otra pestaña del navegador en lugar de redirigirnos al login, la sesion permanece abierta.

* Error de relaciones en la BBDD:

Mientras modificaba la base de datos, hubo un momento en el que no me dejaba relacionar la tabla pedidos con la de usuarios porque supuestamente ya estaban relacionados de alguna manera los indices. Lo solucioné borrando los indices que generaban conflicto.

* Fallo de integridad de mesas:

En el apartado de mesas, al seleccionar una mesa para agregar comensales he iniciar una comanda, en lugar de solo permitir elegir de entre las 10 mesas existentes, en el campo de “Nº mesa”, erroneamente nos deja escoger un número inexistente de mesa. Esto no se ve reflejado en la BBDD pero no debería de permitirlo ni mostrar el alert de “Mesa agregada”**.** 

* Generación de ticket:

Al generar el ticket se generan ticket de las comandas indiviaduales, se me olvidó tener en cuenta que si en una mesa hay varios pedidos por un mismo cliente, la aplicación debería unificar los pedidos y generar un ticket global y no parcial. Este error viene dado por el hecho de que el pedido se genera despues que el “carrito”.

* Selección de cantidad de producto pedido:

Durante el desarrollo del menú para la toma de pedidos, al agregar un producto a la comanda, el input te permitía seleccionar una cantidad mayor del stock existente. Este problema fue solucionado y corregido modificando la siguiente línea.



* Eliminación productos:

Este problema surge a la hora de eliminar un producto por su nombre, al usar como identificador el nombre del producto, si en el menú hay dos productos con el mismo nombre, se borran todos los que tengan ese mismo nombre, en lugar de borrar solo el que nosotros queremos.

* Propagación del borrado:

Al eliminar productos o usuarios no tuve en cuenta que esto afectaría a otras tablas como “pedidos” o “producto\_pedido”, lo que produce un efecto en cadena al eliminar algún producto o usuario, y también borra en consecuencia los registros afectados de las tablas mencionadas.

# **Propuestas de mejora.**

* Arreglar el problema de seguridad para que no guarde la sesión activa.
* Solucionar el error de selección de mesas, para que si seleccionamos una mesa incorrecta nos salga un aviso adecuado.
* Mejorar el apartado de generación del ticket, para que cree el pedido nada más entrar en la parte de tomar la comanda y así no crear varios pedidos con varios tickets, sino que vaya modificando el existente.
* Controlar la subida de archivos en los campos de selección de imágenes para los formularios de registro y modificación de usuarios y productos. No tuve en cuenta que las imágenes se tienen que poder subir desde fuera de la carpeta de recursos de la aplicación, por lo que las imágenes no se pueden subir a la base de datos correctamente, al no pillar bien la ruta del archivo.
* Cambiar las consultas de las funciones de borrado de productos o usuarios, para que, en lugar de borrarlos, solo los desactive y así no afectar a las demás tablas y no se propague el borrado.

# **Conclusiones finales.**

A mi parecer el trabajo es un acierto, te ayuda mucho a aprender PHP, hace que tú mismo te busques las castañas y poco a poco vas aprendiendo cosas nuevas, aunque al principio todo se ve muy negro, ya que no tienes ni idea de por dónde empezar, pero con la ayuda de herramientas como chatGPT o W3Schools para ayudarte a aprender o solucionar errores, poco a poco se va esclareciendo el trabajo.

Además de que te hace acostumbrarte a la metodología de trabajo de los sprint, la cual es de gran ayuda porque así no dejas nada para el final y te mantienes al día.

También mencionar que es una practica a mi parecer divertida o entretenida, porque cuando vas viendo los resultados es muy satisfactorio, además de que en el caso del restaurante en una aplicación que resulta familiar y fácil de pensar en la idea del proyecto y su funcionamiento.

Ahora bien, esos son los pros, pero también hay contras. Por ejemplo, quizás es un poco extensa teniendo en cuenta del poco tiempo que disponemos en el segundo curso. Si, fuese menos extensa, quizás podríamos asentar mejor las bases de PHP al inicio del curso, y así no ir tan perdidos al principio. Y, en mi caso, también hubiese ayudado a que fuese mejor compaginable con otros trabajos o tareas, ya que es una práctica que solo con el trabajo en clase no se puede llevar a cabo, y te ocupa muchas tardes, que si no tienes libres te hace imposible llevarlo al día.