MEmoria Práctica php

Web Restaurante **Antonio J**esús Ibáñez López

****

Contenido

[Memoria del Proyecto Restaurante 4](#_Toc183019356)

[Sprint 1 - Fundamentos del Sistema y Roles 4](#_Toc183019357)

[1. Introducción 4](#_Toc183019358)

[2. Objetivos 4](#_Toc183019359)

[3. Estructura del Proyecto 4](#_Toc183019360)

[4. Proceso de Desarrollo 5](#_Toc183019361)

[Sprint 2 - Módulo Camarero 8](#_Toc183019362)

[1. Introducción 8](#_Toc183019363)

[2. Estructura del Módulo 8](#_Toc183019364)

[3. Funcionalidades Implementadas 9](#_Toc183019365)

[4. Mejoras en la Interfaz de Usuario 10](#_Toc183019366)

[5. Seguridad Implementada 11](#_Toc183019367)

[6. Funcionalidades de Gestión de Pedidos 12](#_Toc183019368)

[7. Mejoras en la Experiencia de Usuario 12](#_Toc183019369)

[8. Próximas Mejoras 13](#_Toc183019370)

[9. Conclusiones 13](#_Toc183019371)

[Sprint 3 - Gestión de Cuentas y Pagos 14](#_Toc183019372)

[1. Base de Datos 14](#_Toc183019373)

[2. Nuevas Funcionalidades 14](#_Toc183019374)

[3. Mejoras en la Interfaz 16](#_Toc183019375)

[4. Optimizaciones 17](#_Toc183019376)

[5. Archivos Modificados 17](#_Toc183019377)

[Próximos Pasos 17](#_Toc183019378)

[6. Conclusiones: 18](#_Toc183019379)

[**Sprint 4: Mejora del Sistema de Tickets y Manejo de Errores** 18](#_Toc183019380)

[**1. Sistema de Generación de Tickets Mejorado** 18](#_Toc183019381)

[**2. Integración con Impresora Térmica** 18](#_Toc183019382)

[**3. Sistema de Notificaciones** 18](#_Toc183019383)

[Sprint 4.5 - Mejoras visuales y funcionalidad PDF 23](#_Toc183019384)

[1. Unificación del diseño 23](#_Toc183019385)

[2. Mejoras en la interfaz de usuario 23](#_Toc183019386)

[3. Implementación de exportación PDF 24](#_Toc183019387)

[4. Mejoras en la gestión de cuentas 25](#_Toc183019388)

[5. Optimizaciones generales 26](#_Toc183019389)

[6. Próximos pasos 26](#_Toc183019390)

[7. Conclusiones 26](#_Toc183019391)

[Sprint 5: Mejoras en la Interfaz del Panel de Encargado 27](#_Toc183019392)

[1. Adaptación Responsive del Panel Principal 27](#_Toc183019393)

[2. Mejora en el Listado de Camareros 27](#_Toc183019394)

[3. Optimización del Formulario de Registro 28](#_Toc183019395)

[4. Características Técnicas Implementadas 28](#_Toc183019396)

[5. Aspectos de Seguridad 28](#_Toc183019397)

[6. Mejoras en la Experiencia de Usuario 28](#_Toc183019398)

[7. Conclusiones 29](#_Toc183019399)

[Sprint 6: Añadir mejoras y gestión de productos en (encargado) 29](#_Toc183019400)

[Introducción 29](#_Toc183019401)

[1. Registro de Usuarios 30](#_Toc183019402)

[2. Listado y Gestión de Usuarios 30](#_Toc183019403)

[3. Gestión de Productos 31](#_Toc183019404)

[4. Interfaz Responsiva 32](#_Toc183019405)

[Conclusiones 34](#_Toc183019406)

[BASE DE DATOS FINAL: 34](#_Toc183019407)

[GUIA DE USO: 35](#_Toc183019408)

[Introducción 35](#_Toc183019409)

[Pantalla de inicio del portal: 35](#_Toc183019410)

[SECCION DE ENCARGADO: 36](#_Toc183019411)

[Index.php(encargado) 36](#_Toc183019412)

[Registrar Usuario: 37](#_Toc183019413)

[Listar Usuarios: 37](#_Toc183019414)

[Acceder como Camarero: 38](#_Toc183019415)

[Gestionar Productos: 39](#_Toc183019416)

[Generar Informe: 40](#_Toc183019417)

[Informe de ventas: 40](#_Toc183019418)

[40](#_Toc183019419)

[SECCION DE CAMARERO: 41](#_Toc183019420)

[Index.php (camarero): 41](#_Toc183019421)

[Gestionar mesas: 42](#_Toc183019422)

[Gestionar pedido: 43](#_Toc183019423)

[Cuenta: 44](#_Toc183019424)

[Historial de cuentas pagadas: 46](#_Toc183019425)

# Memoria del Proyecto Restaurante

## Sprint 1 - Fundamentos del Sistema y Roles

### 1. Introducción

Este proyecto se centra en el desarrollo de un sistema de gestión integral para un restaurante. El sistema abarca funcionalidades esenciales como la gestión de usuarios (con roles diferenciados), la administración de mesas, el procesamiento de pedidos y el control de productos. La modularidad del sistema permite adaptarlo a las necesidades específicas del restaurante, ofreciendo interfaces personalizadas para camareros y encargados.

### 2. Objetivos

* **Automatizar la gestión de pedidos:** Agilizar la toma de pedidos, la comunicación con la cocina y la generación de cuentas.
* **Controlar el inventario:** Mantener un registro actualizado de los productos disponibles, facilitando la gestión de stock y la previsión de compras.
* **Mejorar la atención al cliente:** Optimizar los tiempos de espera y reducir errores en las comandas.
* **Facilitar la administración del restaurante:** Proporcionar al encargado herramientas para la gestión de usuarios, el análisis de ventas y la generación de informes.

### 3. Estructura del Proyecto

La estructura de archivos y directorios del proyecto se organiza de la siguiente manera:

├── encargado  
│ └── index.php  
├── camarero  
│ ├── pedidos.php  
│ └── index.php  
├── BBDD  
│ ├── restaurante (con datos).sql  
│ └── restaurante.sql  
├── logout.php  
├── sesion.php  
├── login.php  
├── conexion.php  
├── index.php  
└── Instrucciones  
 └── leer.txt

#### 3.1. Descripción de Archivos

* **BBDD/restaurante.sql:** Define la estructura de la base de datos, incluyendo tablas, campos y relaciones.
* **BBDD/restaurante (con datos).sql:** Contiene la estructura de la base de datos con datos de ejemplo para facilitar las pruebas iniciales.
* **camarero/index.php:** Panel principal para camareros, con acceso a la gestión de mesas y pedidos.
* **camarero/pedidos.php:** Permite a los camareros tomar nuevos pedidos, modificar existentes y enviarlos a la cocina.
* **encargado/index.php:** Interfaz de administración para el encargado, con funcionalidades para la gestión de usuarios, productos, informes, etc.
* **conexion.php:** Centraliza la configuración de la conexión a la base de datos.
* **index.php:** Página principal del sistema, redirige a los usuarios según su rol.
* **login.php:** Gestiona la autenticación de usuarios, controlando el acceso al sistema.
* **logout.php:** Cierra la sesión del usuario.
* **sesion.php:** Implementa la gestión de sesiones para mantener el estado del usuario durante su interacción con el sistema.

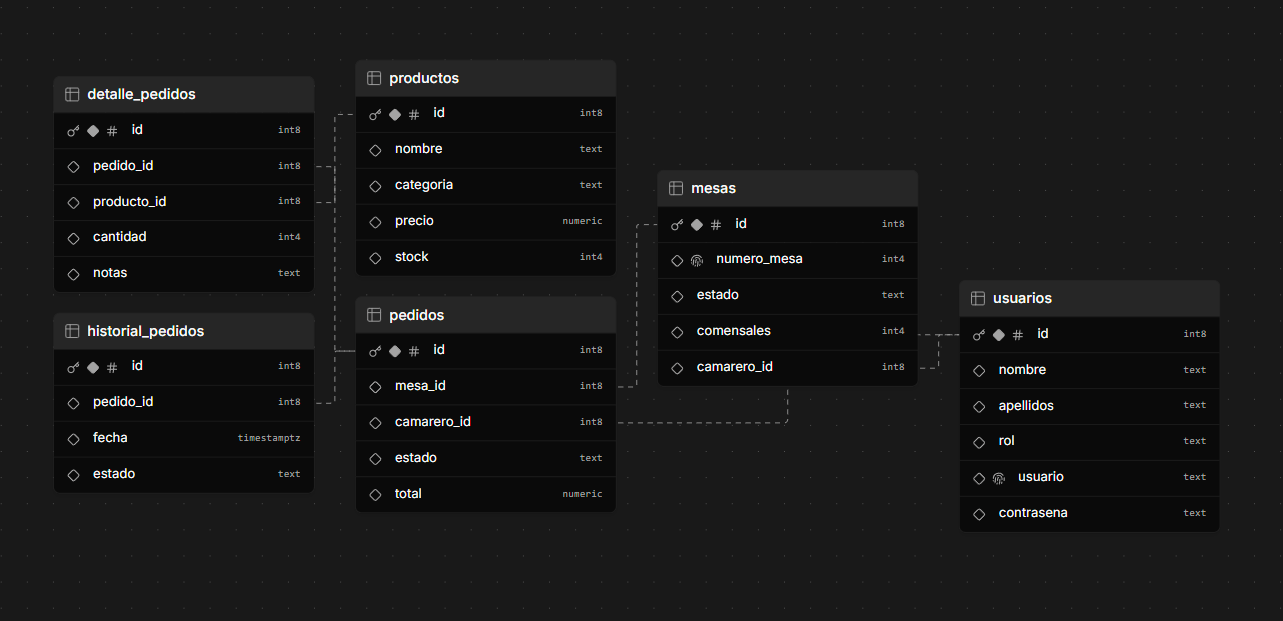
### 4. Proceso de Desarrollo

El desarrollo del proyecto se llevó a cabo siguiendo las siguientes etapas:

#### 4.1. Diseño de la Base de Datos

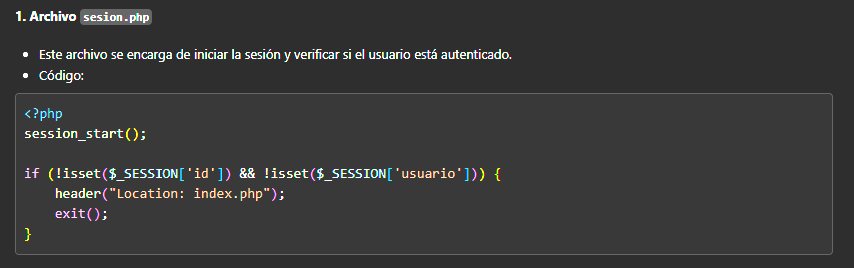
Se definieron las entidades necesarias para el sistema (usuarios, mesas, pedidos, productos) y se establecieron las relaciones entre ellas. Se utilizaron claves primarias y foráneas para asegurar la integridad referencial de los datos.

Primer esquema relacional:

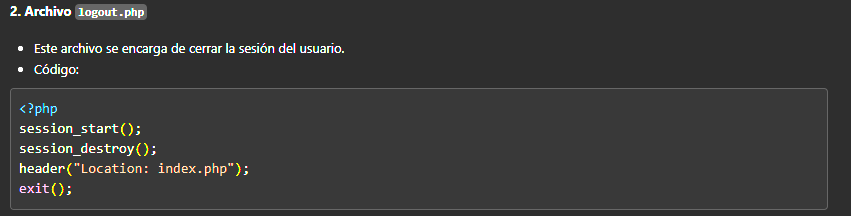


#### 4.2. Implementación del Sistema de Autenticación

Se implementó un sistema de autenticación de usuarios para restringir el acceso a las funcionalidades del sistema.

Ejemplo de código PHP (sesion.php):

Ejemplo de código PHP (logout.php):



Ejemplo de código PHP (login.php):

#### 4.3. Desarrollo de la Interfaz de Usuario

Se diseñaron interfaces intuitivas y fáciles de usar para cada tipo de usuario (camarero y encargado). Se utilizaron tecnologías web como HTML, CSS y JavaScript para crear las interfaces.

#### 4.4. Implementación de la Lógica de Negocio

Se programaron las funcionalidades del sistema, como la toma de pedidos, la gestión de mesas, el control de stock, etc. Se utilizó PHP y consultas SQL para interactuar con la base de datos.

#### 4.5. Pruebas y Validación

Se realizaron pruebas exhaustivas para asegurar el correcto funcionamiento del sistema y la integridad de los datos. Se probaron diferentes escenarios y casos de uso para identificar y corregir posibles errores.

## Sprint 2 - Módulo Camarero

### 1. Introducción

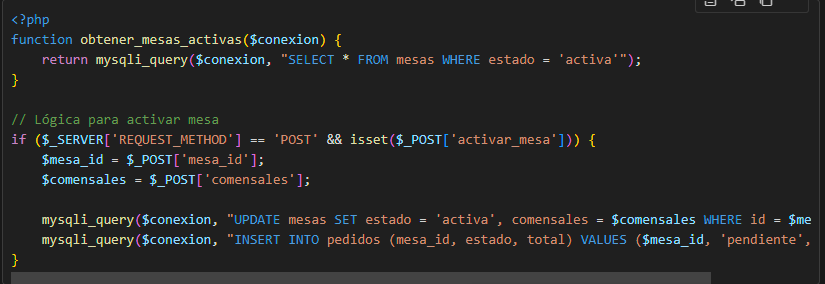
Durante el Sprint 2, se ha desarrollado y mejorado el módulo de camarero del sistema de gestión de restaurante. Este módulo permite la gestión eficiente de mesas, pedidos y productos, con una interfaz responsive y fácil de usar.

### 2. Estructura del Módulo

#### 2.1 Organización de Archivos

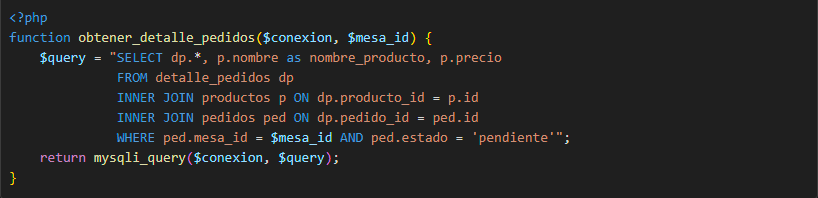
### 3. Funcionalidades Implementadas

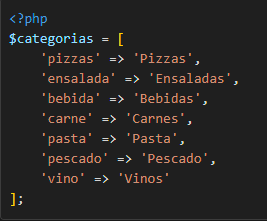
#### 3.1 Gestión de Mesas

****Ejemplo de código para la gestión de mesas activas:

#### 3.2 Gestión de Pedidos

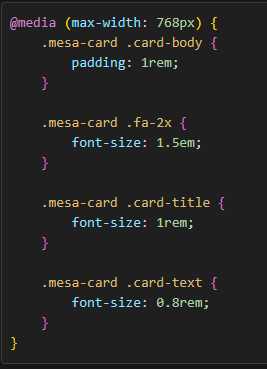
Se implementó un sistema completo de gestión de pedidos con las siguientes características:

**Código de productos añadidos:**

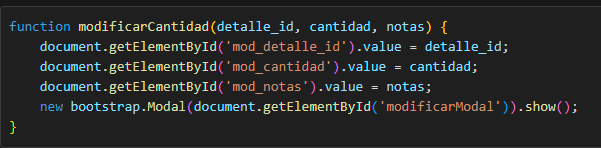
**Sistema de categorías de productos:**

### 4. Mejoras en la Interfaz de Usuario

#### 4.1 Diseño Responsive

****

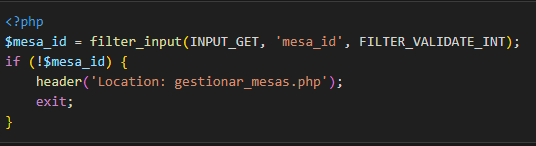
#### 4.2 Interactividad con JavaScript

****

### 5. Seguridad Implementada

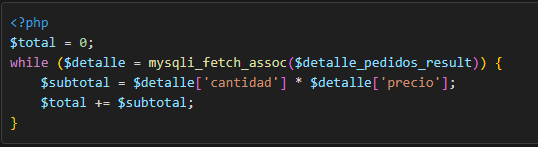
#### 5.1 Validación de Sesiones

#### 5.2 Sanitización de Datos

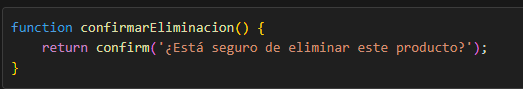


### 6. Funcionalidades de Gestión de Pedidos

#### 6.1 Cálculo de Totales



#### 6.2 Sistema de Confirmación



### 7. Mejoras en la Experiencia de Usuario

#### 7.1 Notificaciones y Feedback

* Mensajes de confirmación para acciones importantes
* Indicadores visuales de estado
* Transiciones suaves en la interfaz

#### 7.2 Optimización de Rendimiento

* Consultas SQL optimizadas
* Carga asíncrona de datos
* Minimización de recargas de página

### 8. Próximas Mejoras

* Implementación de sistema de notificaciones en tiempo real
* Mejora en la gestión de inventario
* Implementación de estadísticas de ventas
* Sistema de reserva

### 9. Conclusiones

**El Sprint 2 ha logrado implementar un sistema robusto y fácil de usar para la gestión de mesas y pedidos, con especial énfasis en la experiencia de usuario y la adaptabilidad a dispositivos móviles. Las mejoras en la interfaz y la implementación de funcionalidades clave han resultado en un sistema más eficiente y fácil de usar para los camareros. He tenido que ajustar la base de datos de nuevo y tener que quitar el “id” de las mesas ya que me hacía tener que comprobar el camarero con su “id” y me creaba demasiados errores, por lo demás la base de datos creo que hacerte con ella y estoy teniendo muy buenos resultados.**

## Sprint 3 - Gestión de Cuentas y Pagos

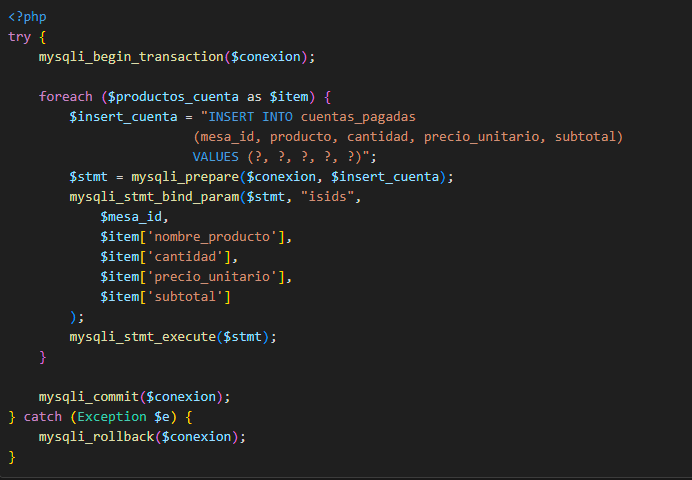
### 1. Base de Datos

* Creación de nueva tabla cuentas\_pagadas para almacenar el historial de pagos
* Eliminación de tablas innecesarias (temp\_ticket, historial\_pedidos)
* Optimización de la estructura de la tabla cuenta actual

### 2. Nuevas Funcionalidades

#### 2.1 Gestión de Cuentas

* Implementación de vista detallada de cuenta por mesa
* Cálculo automático de subtotales y total
* Interfaz responsive para visualización de productos en la cuenta
* Funcionalidad para procesar pagos

Ejemplo de procesamiento de pago:

#### 2.2 Historial de Cuentas

* Nueva sección de historial de cuentas pagadas
* Visualización de cuentas agrupadas por mesa y fecha
* Modal para ver detalles de cuentas históricas
* Sistema de consulta mediante AJAX para cargar detalles
* Ejemplo de consulta para obtener historial:



#### 2.3 Dashboard del Camarero

* Actualización del panel principal con últimas cuentas pagadas
* Widget de resumen de actividad reciente
* Mejora en la navegación entre secciones

### 3. Mejoras en la Interfaz

#### 3.1 Diseño Responsive

* Optimización para dispositivos móviles
* Implementación de tablas responsive
* Mejora en la visualización de datos en pantallas pequeñas

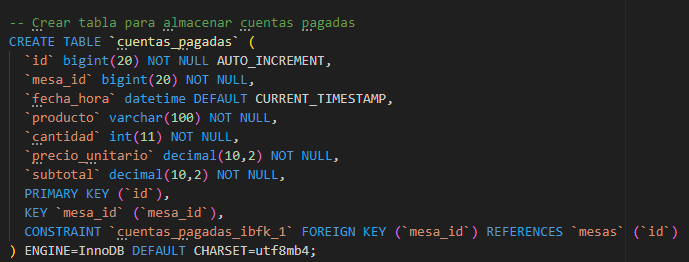
#### 3.2 Experiencia de Usuario

* Nuevo sistema de navegación más intuitivo
* Botones de acción contextuales
* Feedback visual mejorado para acciones importante

### 4. Optimizaciones

* Implementación de prepared statements para consultas seguras
* Mejora en el manejo de transacciones para pagos
* Optimización de consultas SQL para mejor rendimiento

### 5. Archivos Modificados

1. camarero/cuenta.php - Nueva implementación
2. camarero/cuentas\_pagadas.php - Nueva implementación
3. camarero/obtener\_detalle\_cuenta.php - Nuevo archivo
4. camarero/index.php - Actualizado con historial
5. BBDD/restaurante.sql - Actualización de estructura

### Próximos Pasos

1. Implementar sistema de tickets PDF
2. Añadir filtros en el historial de cuentas
3. Implementar composer y añadido para el corte de los tickets
4. Mejorar la gestión de errores
5. Añadir funcionalidad de búsqueda en el historial

### Conclusiones:

**Este sprint se ha centrado en mejorar la gestión financiera y el seguimiento de pagos, proporcionando herramientas más robustas para los camareros y una mejor experiencia de usuario.**

## **Sprint 4: Mejora del Sistema de Tickets y Manejo de Errores**

Objetivo

Mejorar el sistema de generación de tickets y el manejo de errores en la impresión de tickets y envío a cocina, implementando un sistema robusto de redirección y notificaciones de errores.

Funcionalidades Implementadas

### **1. Sistema de Generación de Tickets Mejorado**

* Implementación de validaciones robustas
* Manejo de errores centralizado
* Redirección inteligente en caso de fallos

### **2. Integración con Impresora Térmica**

* Timeout en la conexión para evitar bloqueos
* Verificación previa de conexión
* Manejo de errores de impresión

### **3. Sistema de Notificaciones**

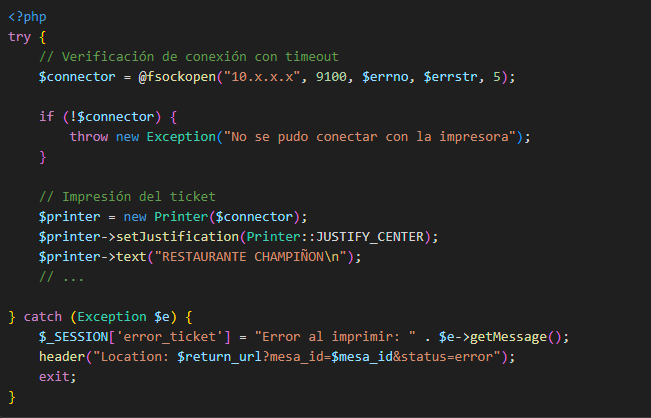
* Mensajes de error específicos
* Alertas visuales en la interfaz
* Persistencia de mensajes entre redirecciones

Código Relevante

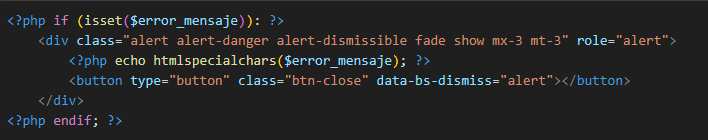
**Generación de Ticket (generar\_ticket.php)**

****

**Impresión de Ticket (ethernet.php)**



**Visualización de Errores (cuenta.php)**



Mejoras Técnicas Implementadas

1. **Manejo de Errores**
   * Try-catch en operaciones críticas
   * Mensajes de error específicos
   * Rollback en transacciones fallidas
2. **Seguridad**
   * Validación de datos de entrada
   * Escapado de caracteres especiales
   * Preparación de consultas SQL
3. **Experiencia de Usuario**
   * Mensajes claros de error
   * Redirecciones automáticas
   * Alertas visuales informativas

Conclusiones y Beneficios

1. **Mayor Robustez**
   * El sistema no se bloquea por errores de impresora
   * Manejo gracioso de fallos
   * Recuperación automática de errores
2. **Mejor Experiencia de Usuario**
   * Feedback inmediato de errores
   * Flujo continuo de trabajo
   * Mensajes claros y específicos
3. **Mantenibilidad**
   * Código centralizado de manejo de errores
   * Estructura clara de logs y mensajes
   * Facilidad para debugging

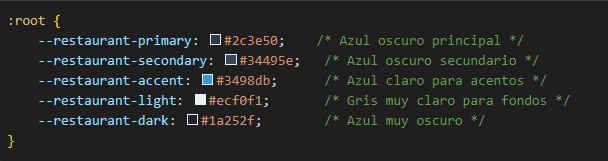
Próximos Pasos

* **Desarrollar la funcionalidad para que los encargados puedan agregar, modificar, y eliminar productos en el menú y ajustar el stock**
* **Implementar la capacidad para que el encargado registre, suspenda o elimine perfiles de camareros.**
* **Desarrollar el sistema de generación de informes de rendimiento del restaurante, con opciones de consulta sobre ingresos, pedidos y número de comensales**
* **Proveer acceso al historial de pedidos para auditoría o gestión de incidencias.**

## Sprint 4.5 - Mejoras visuales y funcionalidad PDF

### 1. Unificación del diseño

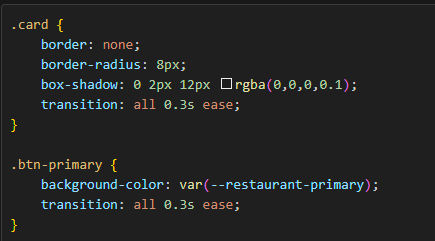
Se ha implementado una paleta de colores común basada en tonos azules oscuros para mantener consistencia visual en toda la aplicación:



### 2. Mejoras en la interfaz de usuario

#### 2.1. Navegación mejorada

* Implementación de sombras y efectos hover
* Mejora en la visibilidad de elementos activos
* Transiciones suaves para mejor feedback visual



#### 2.2. Responsive Design

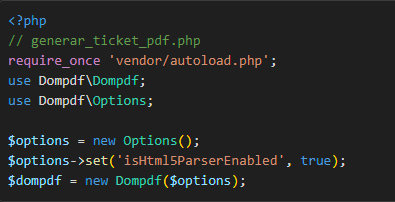
* Optimización para dispositivos móviles
* Mejoras en la visualización de tablas en pantallas pequeñas
* Ajustes de padding y márgenes para mejor usabilidad

### 3. Implementación de exportación PDF

#### 3.1. Instalación de dependencias

**composer require dompdf/dompdf**

#### 3.2. Configuración del generador PDF

****

#### 3.3. Botón de descarga PDF

Se agregó el botón en las vistas de cuenta y cuentas pagadas:

#### 3.4. Generación del ticket PDF

****

### 4. Mejoras en la gestión de cuentas

#### 4.1. Vista detallada de cuentas

* **Implementación de modal para ver detalles**
* **Mejora en la presentación de datos**
* **Funcionalidad de reimpresión de tickets**

#### 4.2. Historial de cuentas

* **Visualización mejorada del historial**
* **Filtros y ordenación de datos**
* **Acceso rápido a tickets anteriores**

### 5. Optimizaciones generales

#### 5.1. Rendimiento

* **Optimización de consultas SQL**
* **Mejora en el tiempo de carga de páginas**
* **Reducción de recursos utilizados**

#### 5.2. Usabilidad

* **Mensajes de confirmación más claros**
* **Mejor feedback para acciones del usuario**
* **Navegación más intuitiva**

### 6. Próximos pasos

* **Implementar más opciones de exportación**
* **Mejorar la gestión de errores**
* **Añadir estadísticas y reportes**
* **Optimizar más la experiencia móvil**

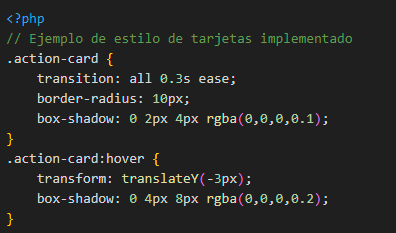
### 7. Conclusiones

**Las mejoras implementadas en este sprint han resultado en una interfaz más profesional y coherente, además de añadir funcionalidad valiosa como la exportación a PDF. La usabilidad general del sistema ha mejorado significativamente, especialmente en dispositivos móviles.**

## Sprint 5: Mejoras en la Interfaz del Panel de Encargado

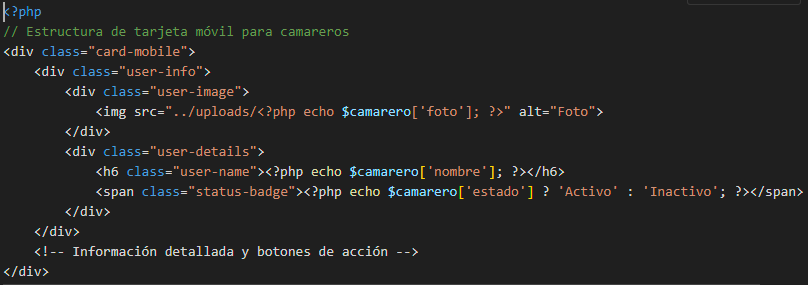
### 1. Adaptación Responsive del Panel Principal

Se mejoró la interfaz del panel de encargado para garantizar una experiencia de usuario óptima en todos los dispositivos. Se implementaron tarjetas de acción con efectos visuales y transiciones suaves.



### 2. Mejora en el Listado de Camareros

Se desarrolló una vista dual para el listado de camareros:

* Vista de escritorio: Tabla completa con todos los datos
* Vista móvil: Tarjetas individuales con información detallada y acciones accesibles

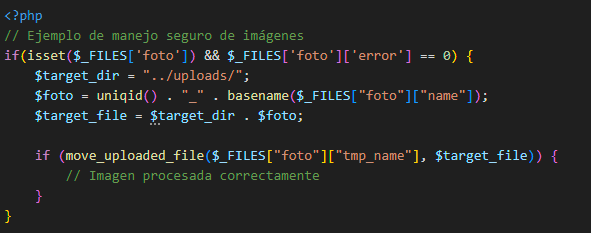
### 3. Optimización del Formulario de Registro

Se implementó un diseño más intuitivo para el formulario de registro de camareros, con validación de campos y mejor manejo de la carga de imágenes.

### 4. Características Técnicas Implementadas

* **Diseño Responsive**: Uso de Media Queries para adaptación a diferentes dispositivos
* **Efectos Visuales**: Transiciones y animaciones suaves para mejor feedback
* **Gestión de Imágenes**: Manejo optimizado de fotos de perfil
* **Validación de Formularios**: Implementación de validación del lado del cliente
* **Interfaz Intuitiva**: Diseño centrado en la experiencia de usuario

### 5. Aspectos de Seguridad



### 6. Mejoras en la Experiencia de Usuario

* Botones de acción más grandes en móvil
* Información claramente estructurada
* Feedback visual en interacciones
* Navegación simplificada
* Acciones críticas con confirmación

### 7. Conclusiones

**Las mejoras implementadas en este sprint han resultado en una interfaz más moderna, accesible y funcional para el panel de encargado, facilitando la gestión eficiente de camareros tanto en dispositivos de escritorio como móviles. Se ha modificado el registro para poder subir una imagen y visualizarla junto con todos los datos de los camareros.**

## Sprint 6: Añadir mejoras y gestión de productos en (encargado)

### Introducción

En el Sprint 6 del proyecto “Restaurante”, se implementaron diversas mejoras y nuevas funcionalidades para optimizar la gestión de usuarios y productos, así como para mejorar la interfaz de usuario. A continuación, se detallan los cambios realizados en los diferentes archivos del sistema.

#### Objetivos del Sprint 6

* **Mejorar la gestión de usuarios** añadiendo roles y funcionalidades de administración.
* **Implementar la carga y gestión de imágenes** para usuarios.
* **Optimizar la interfaz de usuario** para ser más responsiva y amigable.
* **Mejorar la gestión de productos** con funcionalidades completas de CRUD (Crear, Leer, Actualizar, Eliminar).
* **Aumentar la seguridad** en las operaciones de base de datos mediante el uso de consultas preparadas.

#### Cambios Realizados

### 1. Registro de Usuarios

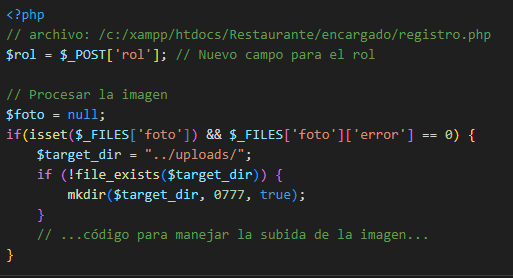
#### Archivos Modificados:

* encargado/registro.php
* encargado/registrar\_usuario.php

#### Descripción de Cambios:

* **Añadido Campo 'Rol':** Se incorporó un nuevo campo rol en el formulario de registro para asignar roles específicos a los usuarios (camarero, encargado).
* **Carga de Fotos de Usuario:** Se implementó la funcionalidad para subir imágenes de perfil de los usuarios, almacenándolas en la carpeta uploads.

##### Código Relevante:

****

### 2. Listado y Gestión de Usuarios

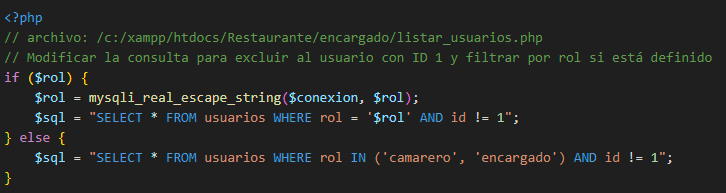
#### Archivos Modificados:

* encargado/listar\_usuarios.php
* encargado/gestionar\_usuario.php

#### Descripción de Cambios:

* **Filtrado por Rol:** Se añadió la posibilidad de filtrar usuarios por rol, permitiendo una gestión más específica.
* **Exclusión del Usuario Principal:** Se excluye al usuario con id = 1 de las operaciones de modificación y eliminación para proteger el usuario administrador principal.
* **Confirmaciones Modales:** Implementación de modales de confirmación para acciones críticas como suspender, activar o eliminar usuarios.

##### Código Relevante:

****

### 3. Gestión de Productos

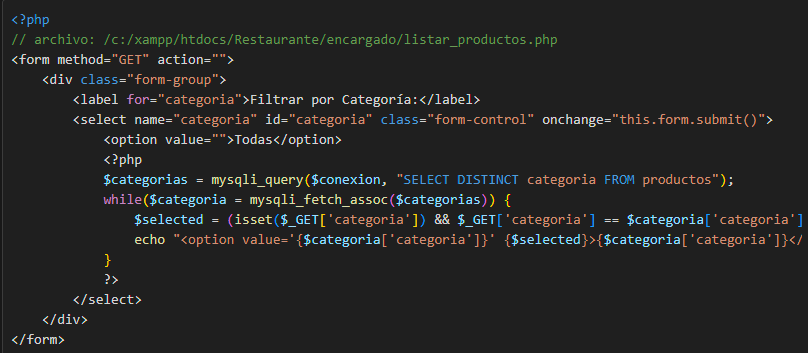
#### Archivos Modificados:

* **encargado/listar\_productos.php**
* **encargado/agregar\_producto.php**
* **encargado/modificar\_producto.php**
* **encargado/eliminar\_producto.php**
* **encargado/gestionar\_producto.php**

#### Descripción de Cambios:

* **CRUD Completo: Se implementaron todas las operaciones de CRUD para la gestión de productos, permitiendo añadir, listar, modificar y eliminar productos.**
* **Filtrado por Categoría: Se añadió la funcionalidad de filtrar los productos por categoría para facilitar la búsqueda y organización.**
* **Interfaz Responsiva: Se mejoró la visualización de las tablas de productos en dispositivos móviles mediante estilos CSS específicos.**

##### Código Relevante:



### 4. Interfaz Responsiva

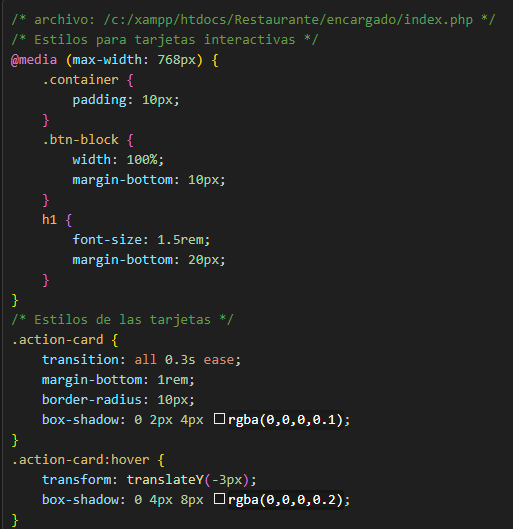
#### Archivos Modificados:

* **encargado/registrar\_usuario.php**
* **encargado/listar\_usuarios.php**
* **encargado/listar\_productos.php**
* **encargado/index.php**

#### Descripción de Cambios:

* **Medias Queries en CSS: Se añadieron reglas CSS para mejorar la visualización en dispositivos móviles, ajustando tamaños, márgenes y disposición de elementos.**
* **Tarjetas Interactivas: Se implementaron tarjetas con efectos hover para una mejor experiencia de usuario en la página principal del encargado.**

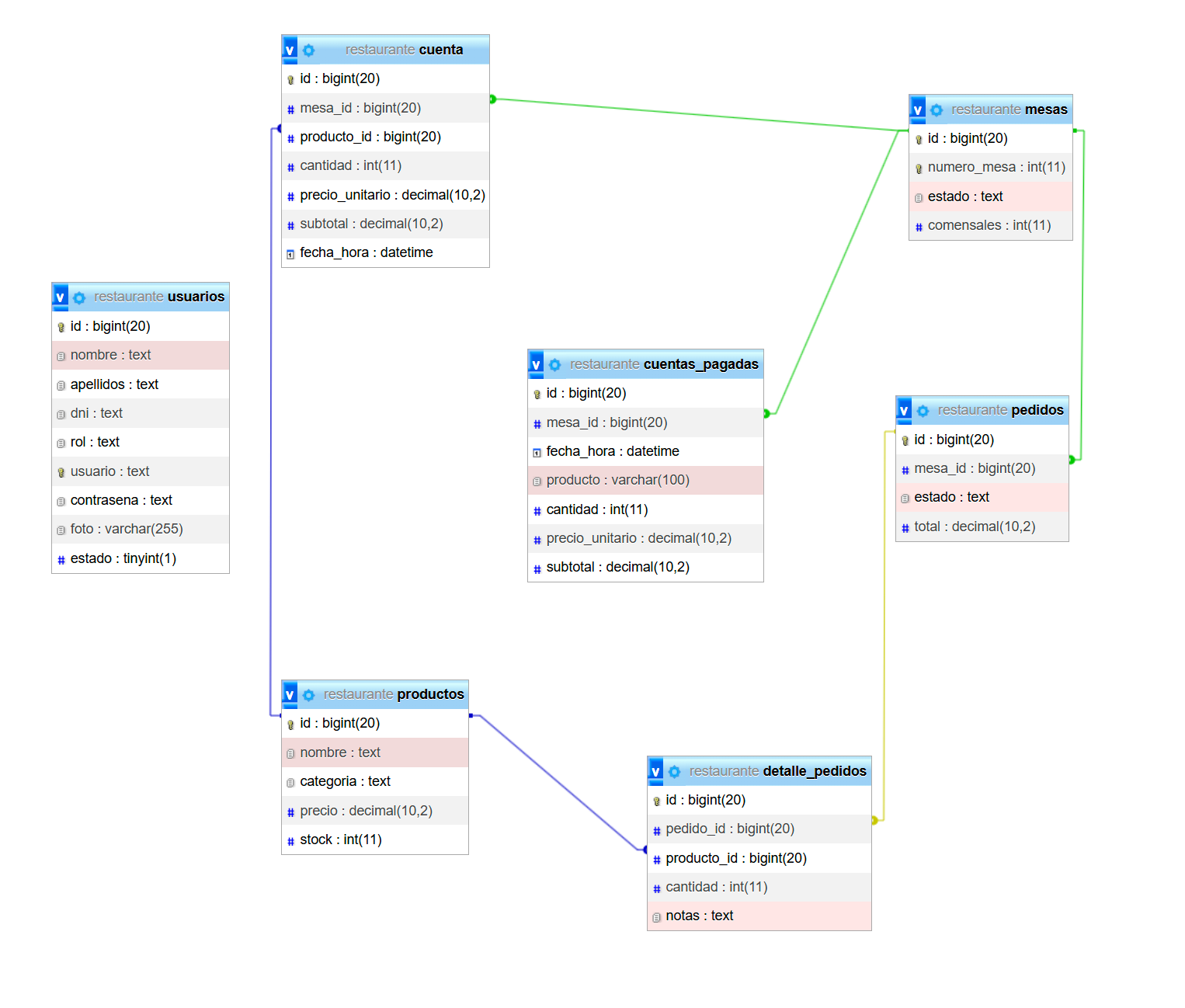
##### Código Relevante:

****

### Conclusiones

**Durante el Sprint 6, se lograron significativas mejoras en la gestión de usuarios y productos, así como en la experiencia de usuario a través de una interfaz más responsiva y funcional. La implementación de roles y la seguridad en las operaciones garantizan un control adecuado y protegido del sistema, mientras que las nuevas funcionalidades de gestión de productos permiten una administración más eficiente del inventario del restaurante.**

## BASE DE DATOS FINAL:

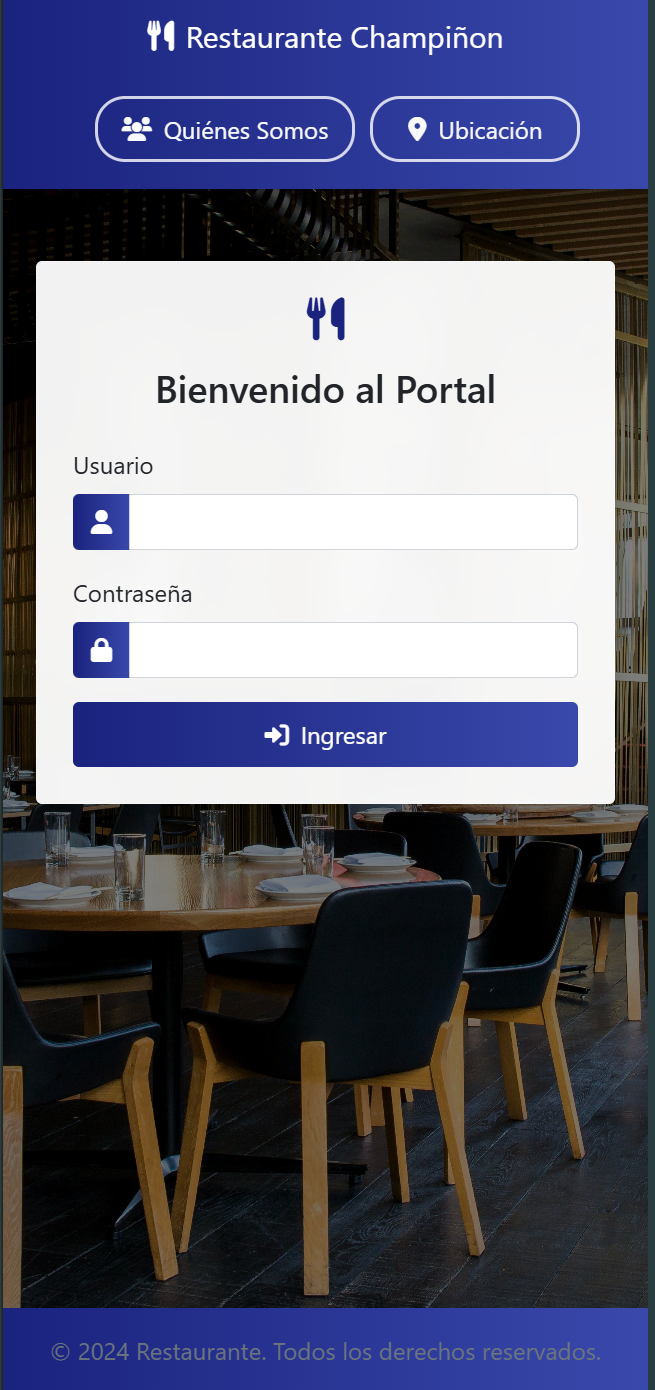


# GUIA DE USO:

### Introducción

Bienvenido a la guía de uso de la página web del restaurante. Aquí aprenderás a gestionar mesas, pedidos y productos de manera sencilla, con herramientas diseñadas tanto para camareros como para encargados.

**¿Qué incluye esta guía?**

* Organización de mesas y asignación de productos
* Gestión de pedidos y actualizaciones en tiempo real.
* Control de productos e inventario.
* Funciones específicas para camareros y encargados.

## Pantalla de inicio del portal:

Desde aquí podemos acceder a la web con nuestros usuarios, tenemos 2 preestablecidos:

1-Encargado:

Usuario: master

Contraseña: pizza

2-Camarero:

Usuario: periko\_elmaki

Contraseña: 1234

## SECCION DE ENCARGADO:

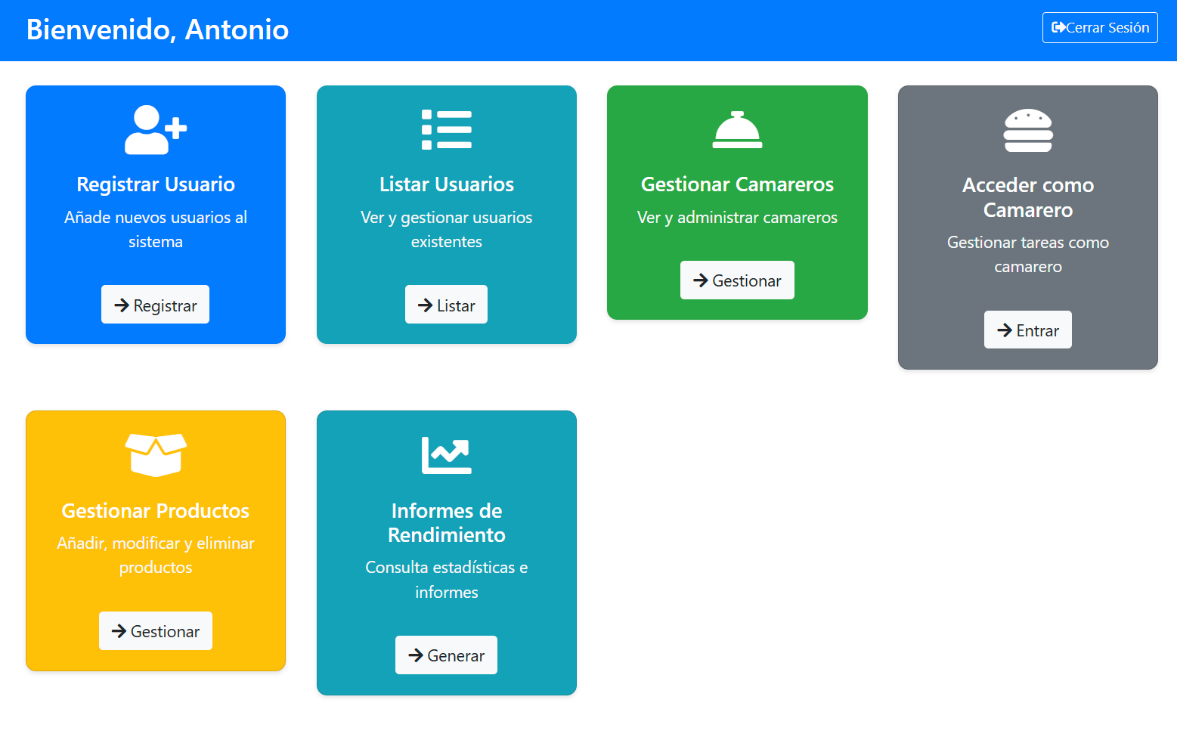
### Index.php(encargado)

Es el panel de control de los encargados (solo pueden acceder con un usuario encargado).

Tenemos varias formas de visualizar siempre la web, con tamaño adaptado para móvil responsive, en una Tablet o monitor.

Encontramos 5 secciones en el panel:

1. Registrar usuario
2. Listar Usuarios
3. Gestionar camareros
4. Acceder como Camarero
5. Gestionar Productos
6. Informe de Rendimiento

A continuación, detallamos las secciones:

### Registrar Usuario:

Página para registrar un nuevo usuario y asignarle un

Rol: encargado/camarero

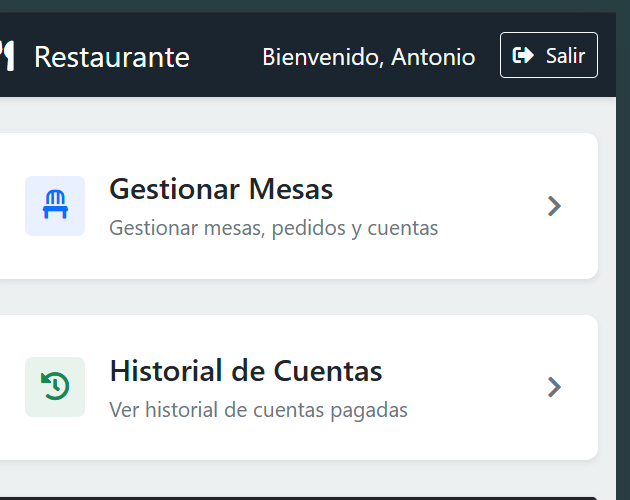
### Listar Usuarios:

Forma de visualizar los usuarios de la base de datos de restaurante.

### Acceder como Camarero:

**Desde encargado también podemos acceder a la parte de camarero y poder utilizar todas sus secciones.**

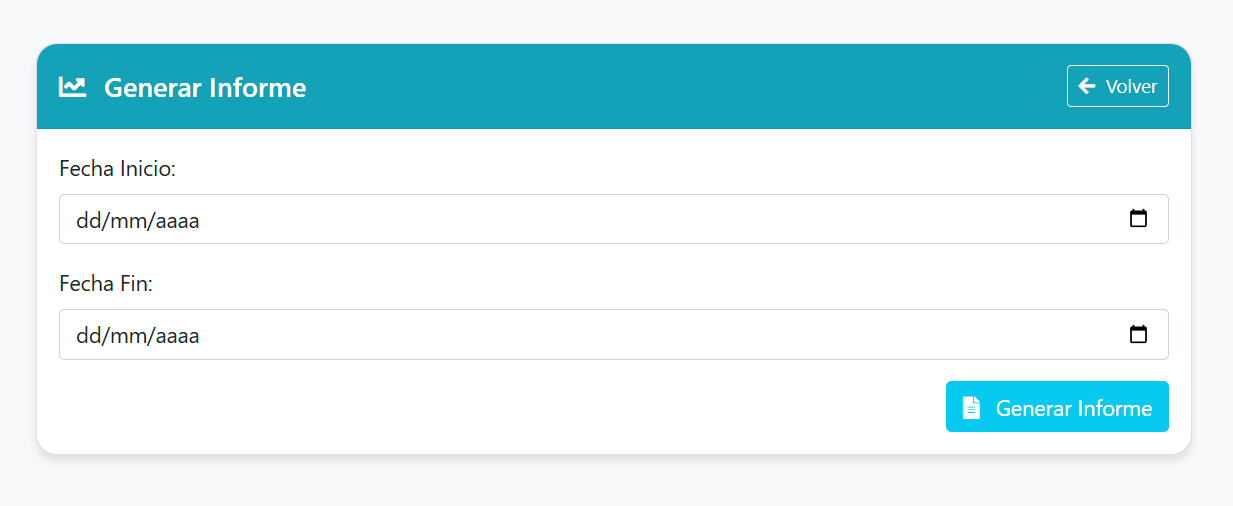
**Tu nombre aparecerá en el panel inicial de camarero**.



### Gestionar Productos:

**Apartado en el cuál puedes añadir, modificar o eliminar productos de la base de datos.**

### Generar Informe:

**Puedes seleccionar la fecha entre los días que quieras conocer la cantidad total de ingresos y la cantidad de pedidos.**

### Informe de ventas:

**Aquí se visualiza el informe con la fecha, el número de ventas y el total de ingreso ese día.**

### 

## SECCION DE CAMARERO:

### Index.php (camarero):

Panel de control de la página de camarero.

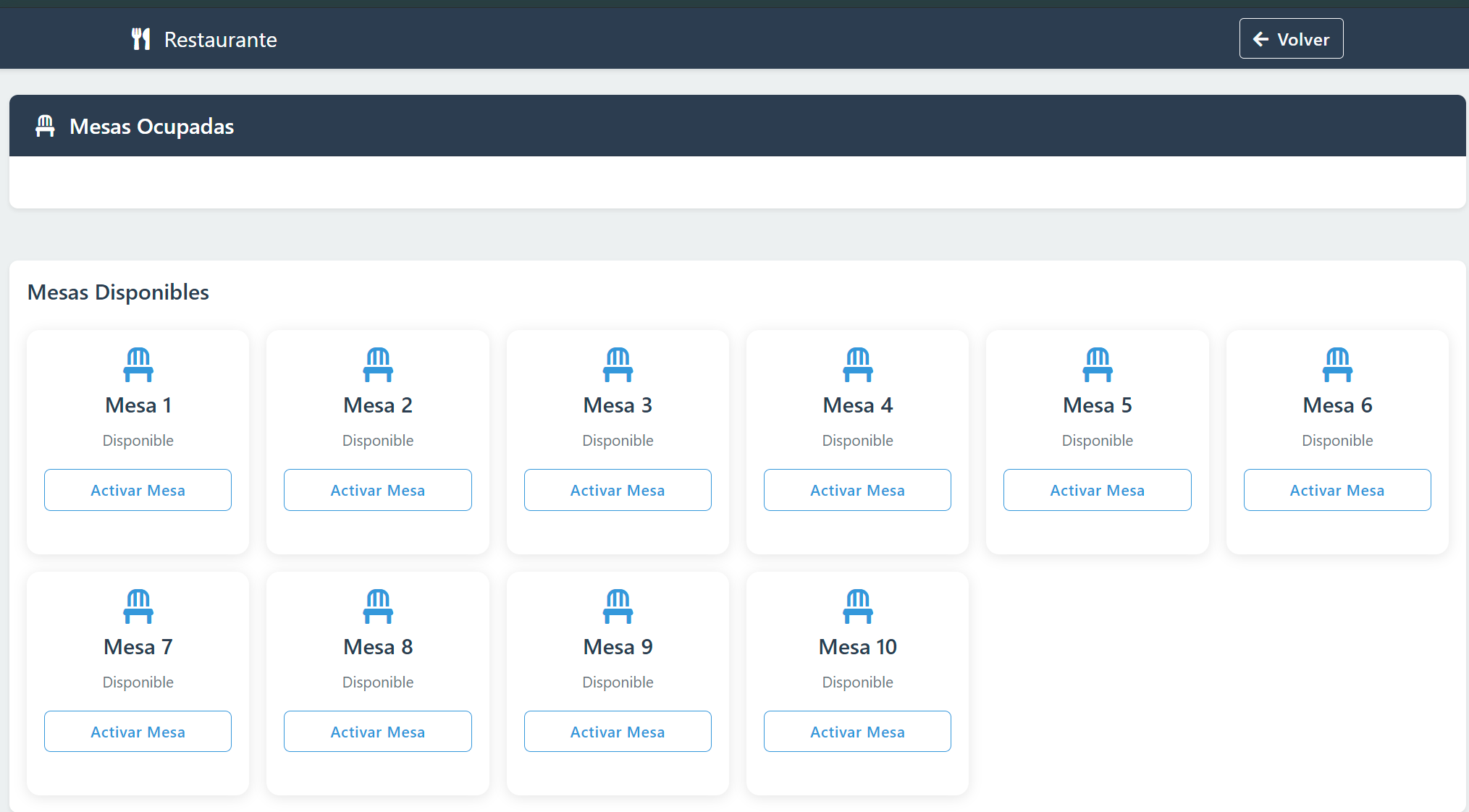
#### Gestionar Mesas:

* Activar mesas
* Gestionar pedidos
* Cuentas

#### Historial de Cuentas:

**Página donde poder acceder a las cuentas pagadas anteriores ordenadas por fechas.**

### Gestionar mesas:

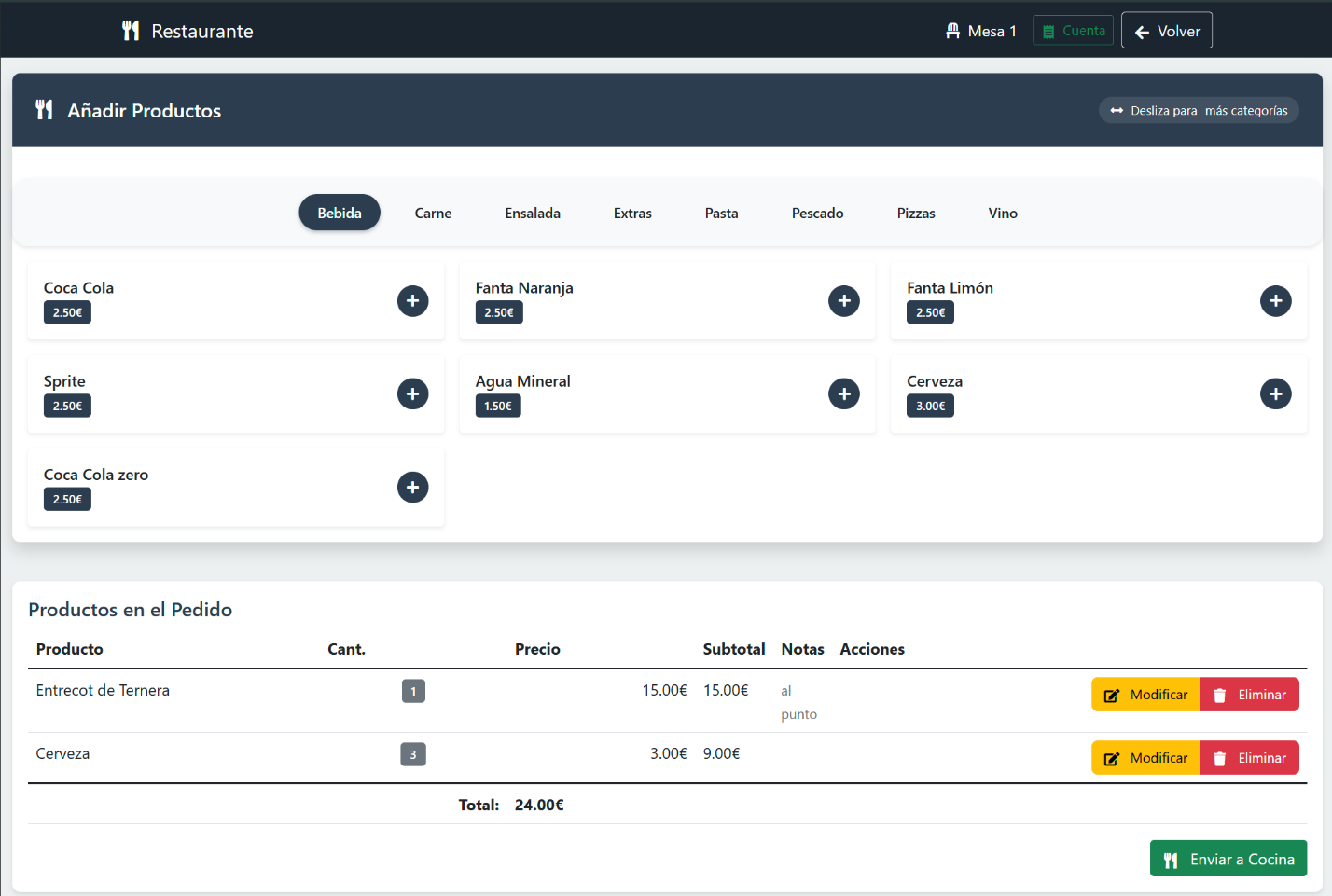
**Sección para visualizar las mesas y activarlas para poder gestionar el pedido.**



**Al activar una mesa, esa mesa pasará a ser “ocupada” y desde el menú de las mesas se podrá entrar en varias secciones como:**

* Gestionar (para agregar productos)
* Cuenta (acceder a la cuenta de la mesa)
* Cerrar (posibilidad de cerrar la mesa, si no hay productos en ella)

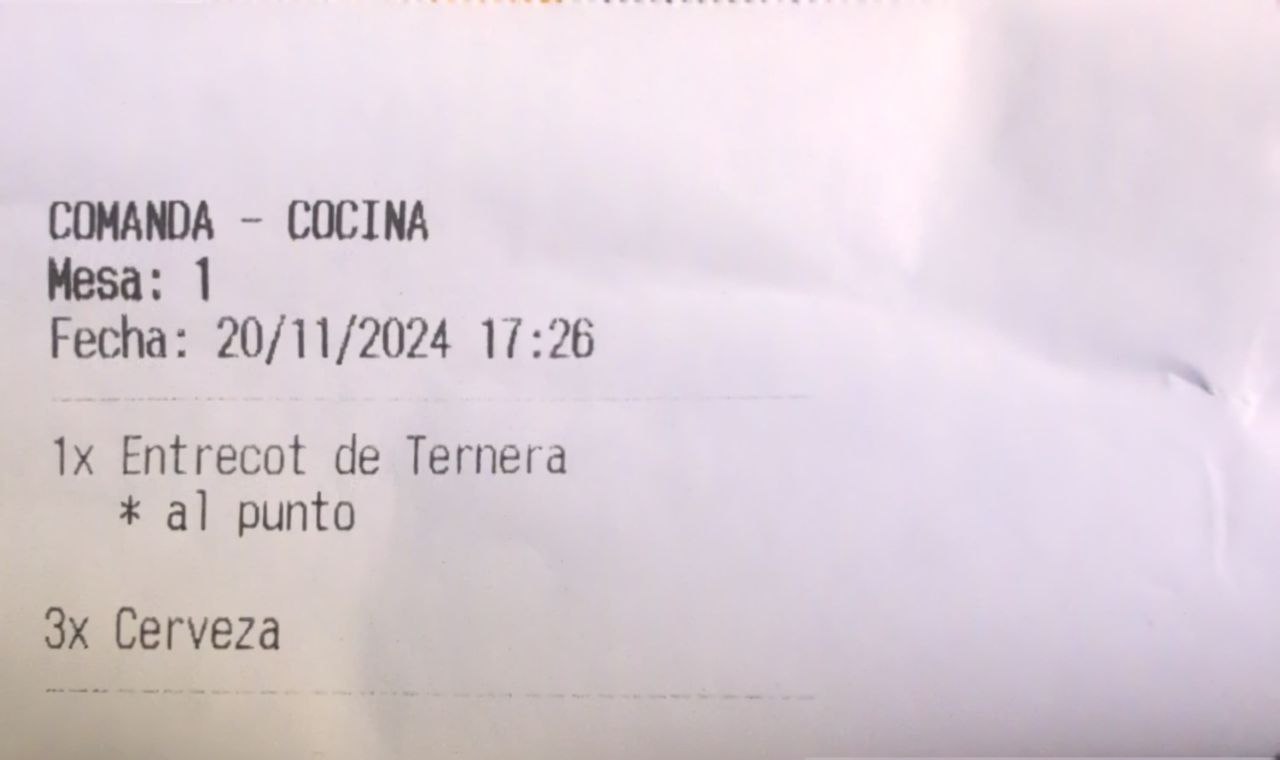
### Gestionar pedido:



**Aquí podrás seleccionar los productos a la mesa y que posteriormente se envían a la cocina para su preparación.**

**Los productos están organizados por categorías según su tipo y puede seleccionar la cantidad y añadir una nota cuando seleccione el producto.**

**Se puede modificar y eliminar los productos en el pedido si hiciera falta y visualizar lo que hay en ese momento.**

**Al pulsar** “Enviar a Cocina” **los productos pasarán a la** cuenta, **pero antes se generará un** ticket **a modo de comanda para la cocina con la cantidad y notas si las hubiera.**

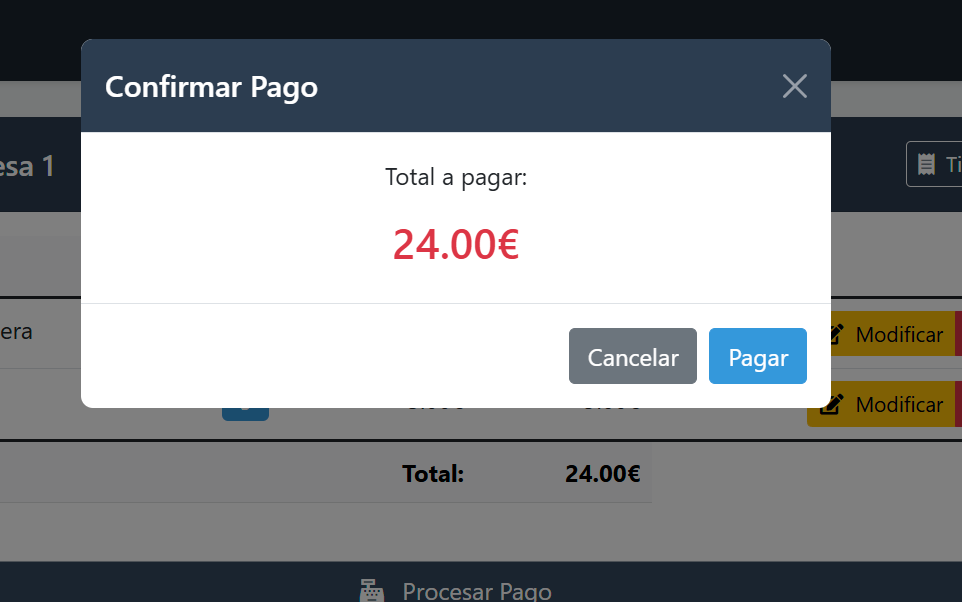
### Cuenta:

**Tras pulsar en** “Enviar a Cocina” **llegamos a** la cuenta **donde visualizamos los productos que llevamos.**

**Posibilidad de modificar y eliminar productos de la cuenta.**

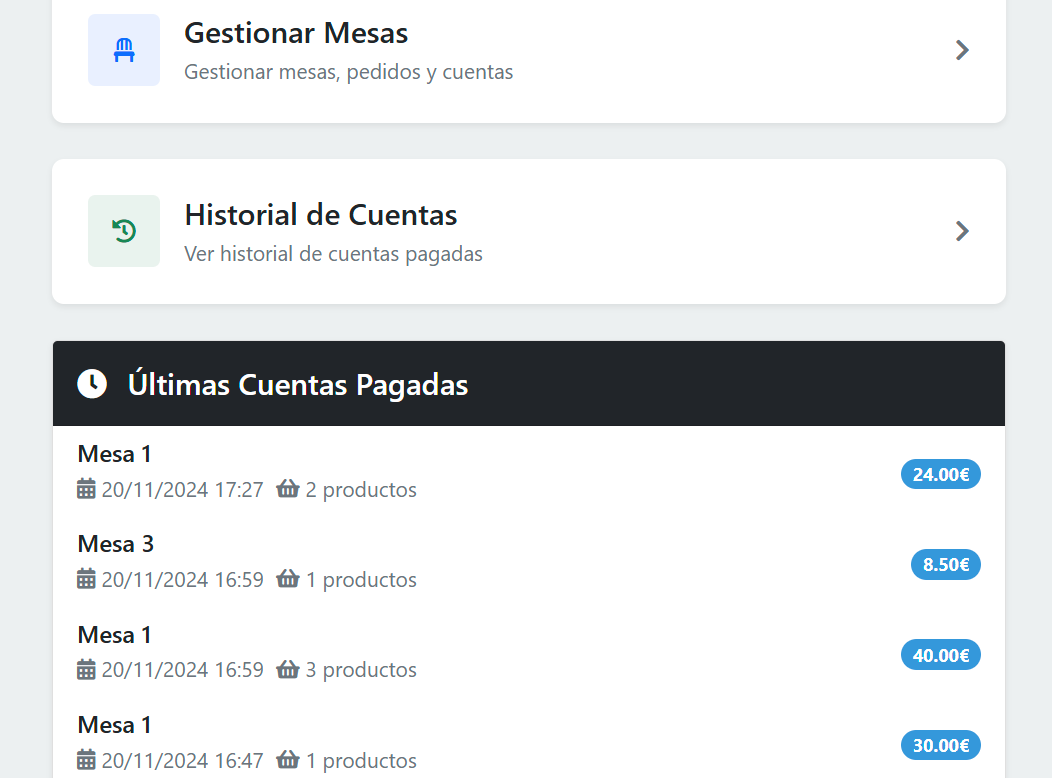
**Aquí se puede imprimir un ticket con la cuenta que hay en ese momento, aunque una vez que pulsemos en** “Procesar pago”**, también nos genera el ticket de la cuenta con todos sus detalles.**



**Antes de procesar el pago, se pide confirmación:**

**Se genera** un ticket de la cuenta **con todo lo que se encontraba en ella en el momento de procesar el pago.**

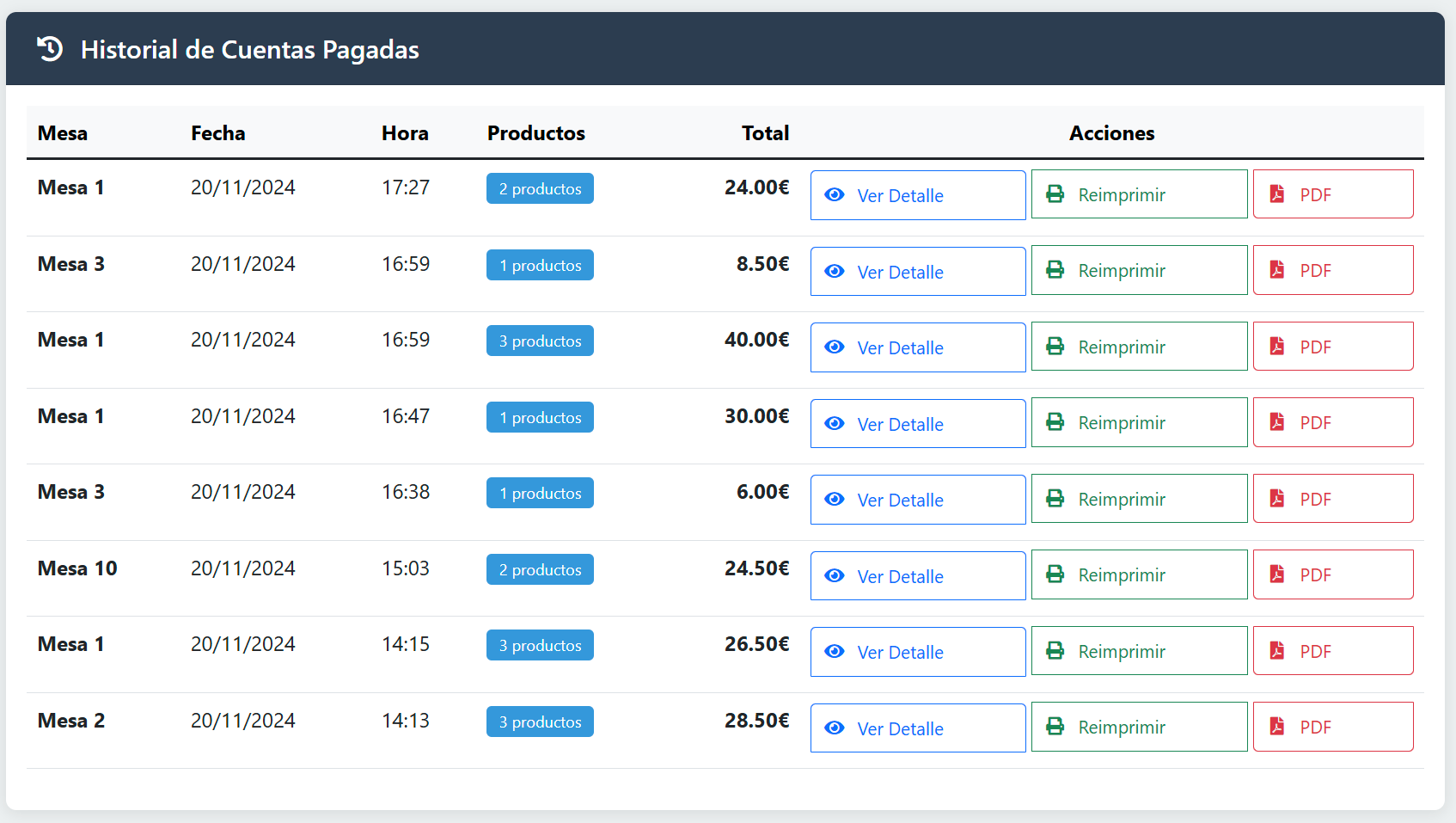
**También se libera la mesa para poder activarla de nuevo con un pedido nuevo.**

**Una que ya hemos completado todo el proceso con la mesa y la cuenta, solo queda visualizar el apartado de** historial de cuentas **para ver todas las cuentas anteriores con sus fechas correspondientes.**

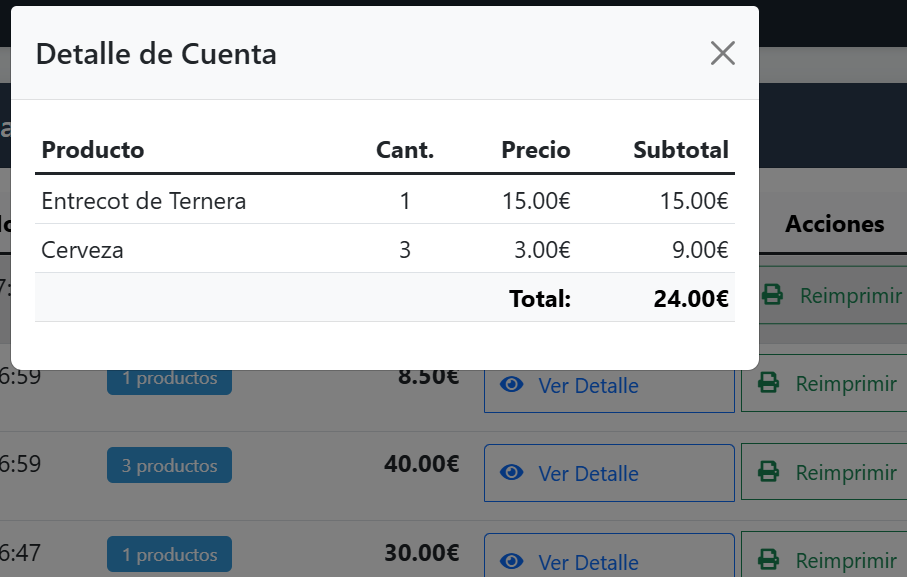
**Entramos en la sección en el panel principal del camarero en** Historial de Cuentas

### Historial de cuentas pagadas:

**Sección en la que se visualizan todas las cuentas por fechas y número de mesa y en la que se pueden realizar ciertas acciones:**

****

* Ver Detalle:

**Aquí surgirá una ventana (modal) para ver en detalle todo lo que contiene esa cuenta.**

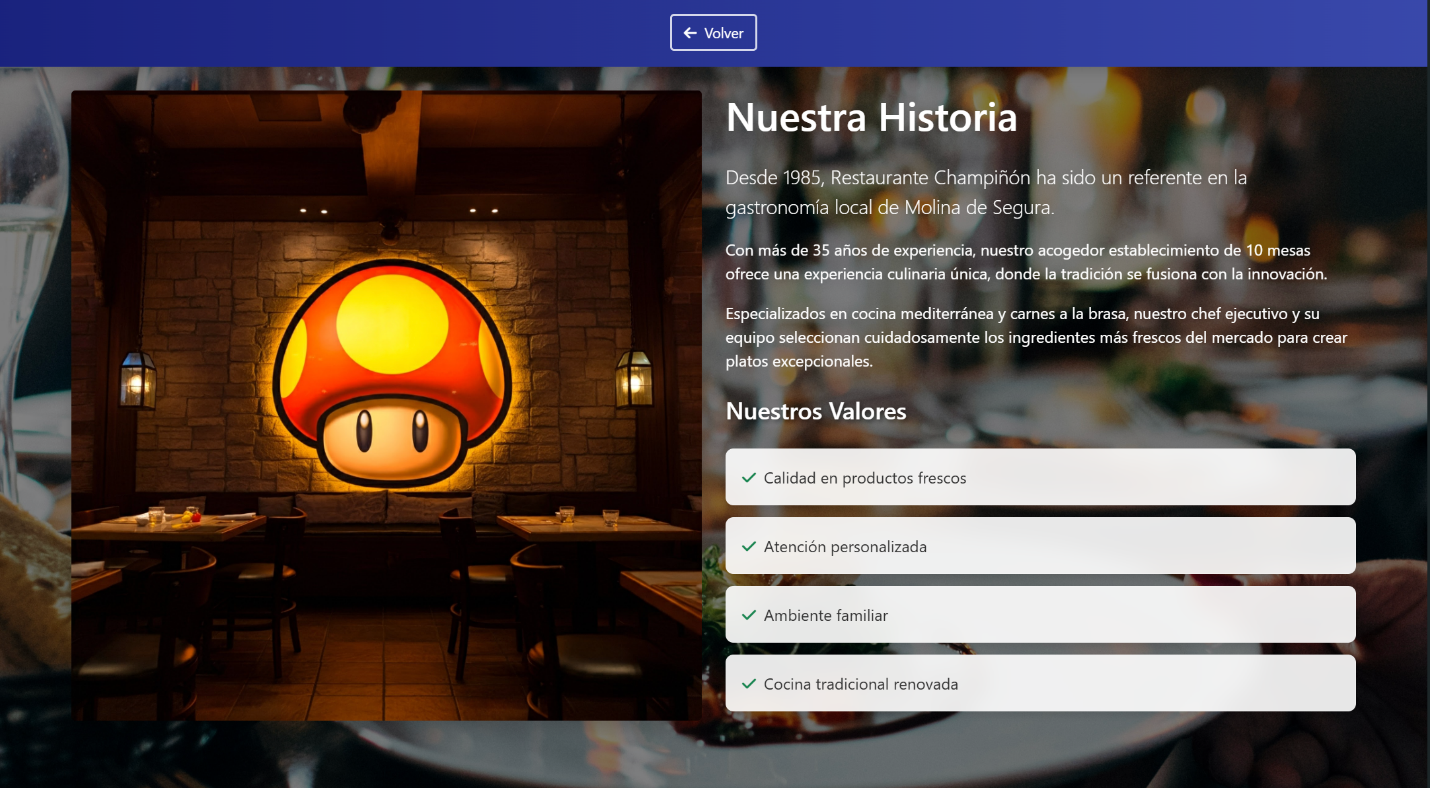
* ****Reimprimir:

**Opción para volver a generar un ticket de cuenta en formato de COPIA**

* PDF:

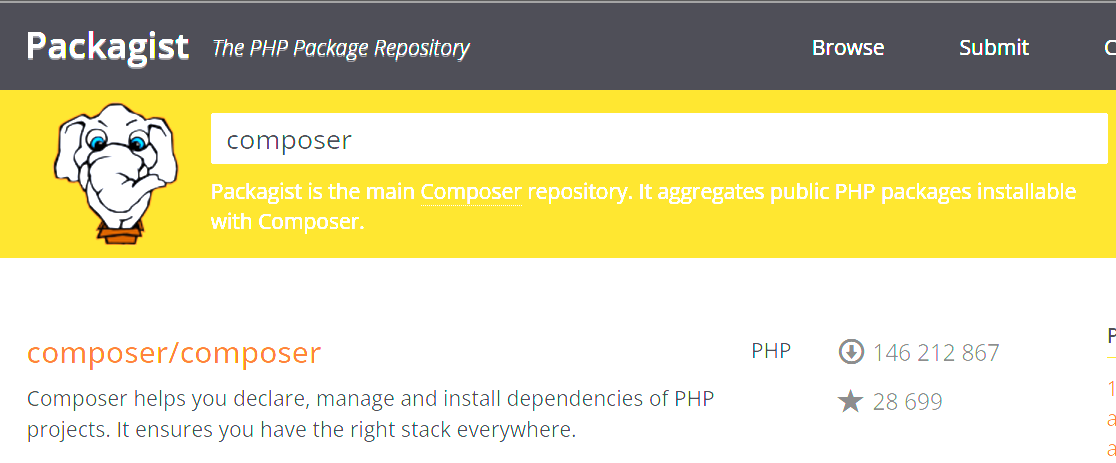
**Generación de un PDF con los datos de la cuenta en tablas para su visualización o posterior impresión.**

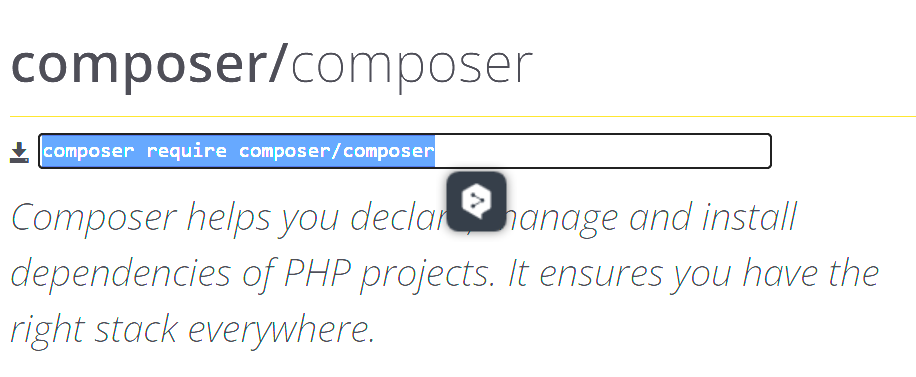
Guía terminada sobre el uso de la página web de Restaurante.



## Instalación de impresoras de tickets y PrintNode (para su uso a través de web)

### Instalación local y conexión con composer y librerías mike42:

Descargar composer en nuestro proyecto.

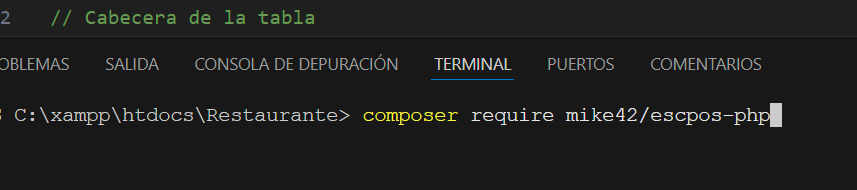
Copiar la ruta resaltada.

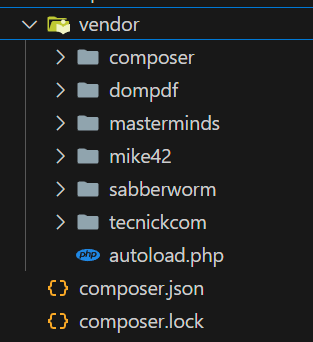
Una de las formas más rápidas de hacer con visual studio code sería accediento a nuestro terminal (control-ñ) y pegando la ruta anterior.

Buscar en la página oficial de composer la librería de mike42/escpos-php



Mismo proceso que con la librería de composer:

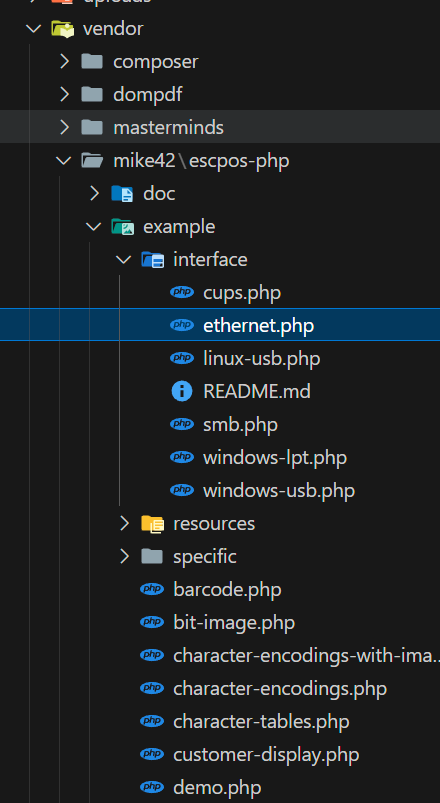




Se nos creará en la raíz de nuestro proyecto 1 carpeta (Vendor) y dos archivos de composer:

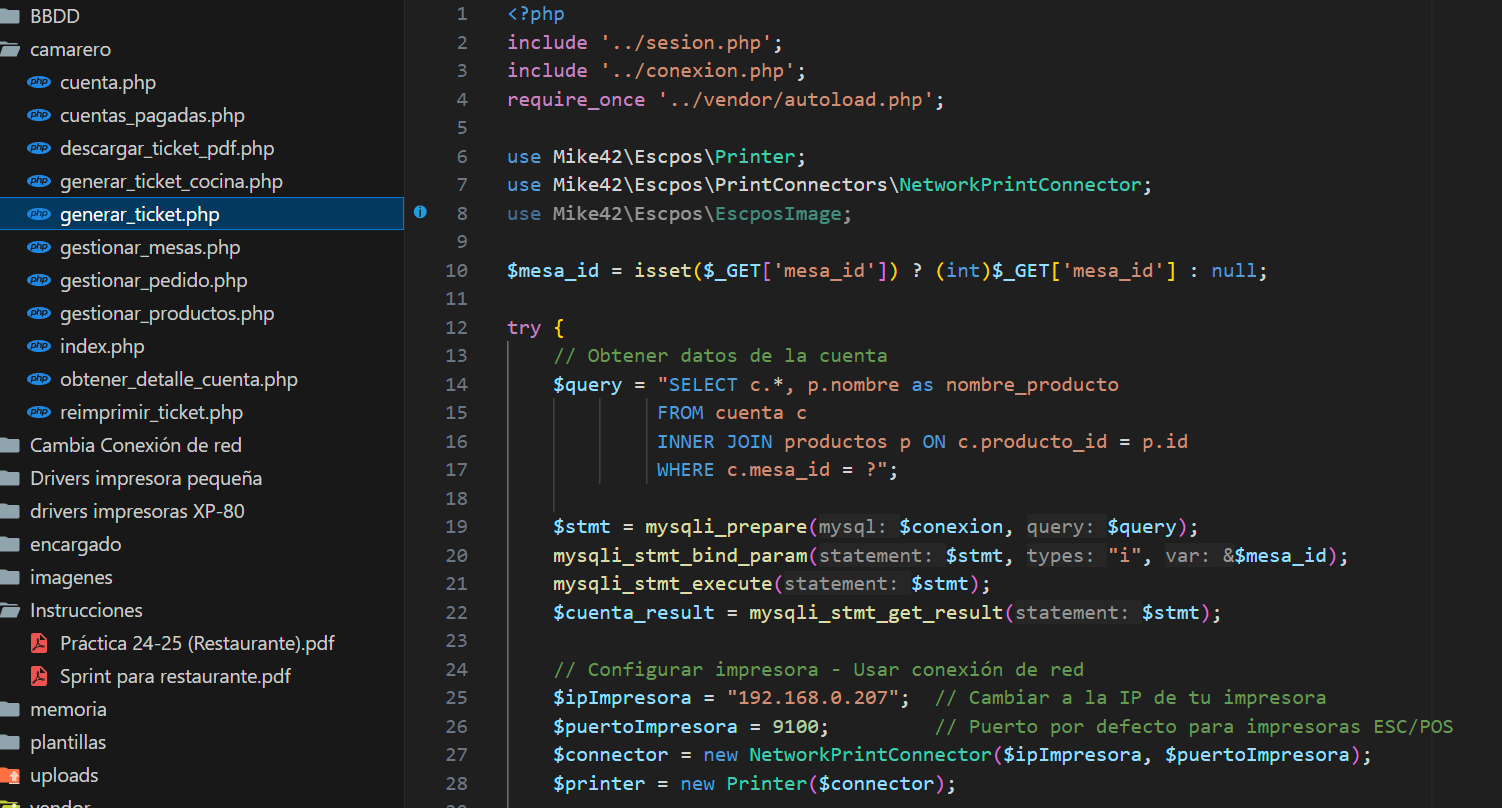
Dentro de la carpeta de vendor se encuentra la carpeta de mike42 con todo lo necesario para conectar la impresora de ticket.

Accedemos a ella y podemos encontrar ejemplos de varias formas de conectar la impresora:

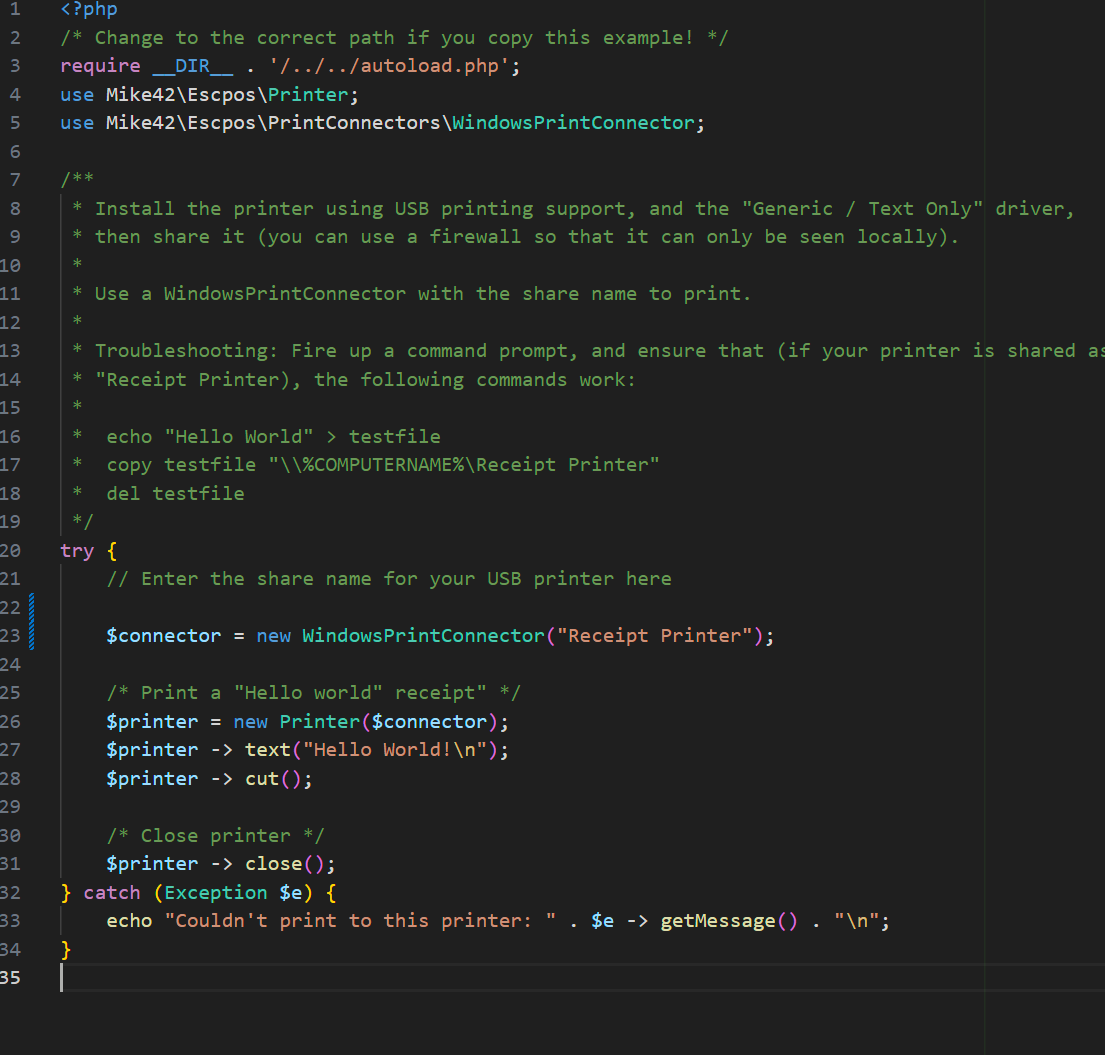


En mi caso yo he utilizado la conexión por IP , y he creado un php especifico para la impresión de ticket.

Así seria uno de los formato para la conexión con la impresora y estructura del ticket.



También se puede conectar mediante USB añadiendo estos cambios:



Aquí se debe introducir el puerto de la impresora que generalmente es “USB001” si no existe otra impresora conecta.

### Instalación web con PrintNode:

<https://www.printnode.com/es>