

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.

Ejemplo de código SQL (restaurante.sql):

SQL

```
CREATE TABLE usuarios (  
    id INT AUTO_INCREMENT PRIMARY KEY,  
    nombre VARCHAR(255) NOT NULL,  
    usuario VARCHAR(255) NOT NULL UNIQUE,  
    contraseña VARCHAR(255) NOT NULL,  
    rol ENUM('camarero', 'encargado') NOT NULL  
);
```

```
CREATE TABLE mesas (  
    id INT AUTO_INCREMENT PRIMARY KEY,
```

```

    numero INT NOT NULL UNIQUE,
    capacidad INT NOT NULL,
    estado ENUM('libre', 'ocupada') NOT NULL DEFAULT 'libre'
);

```

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):

PHP

```

<?php
session_start();

function verificar_sesion() {
    if (!isset($_SESSION['usuario'])) {
        header("Location: login.php");
        exit();
    }
}
?>

```

Ejemplo de código PHP (logout.php):

PHP

```

<?php
session_start();
session_destroy();
header("Location: login.php");
exit();
?>

```

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.

Ejemplo de código PHP (camarero/pedidos.php):

PHP

```
<?php
require_once '../conexion.php';
verificar_sesion();

// Obtener el ID de la mesa desde la URL
if (isset($_GET['mesa'])) {
    $id_mesa = $_GET['mesa'];

    // Consulta para obtener los productos
    $sql = "SELECT * FROM productos";
    $result = $conn->query($sql);
} else {
    // Redirigir si no se proporciona el ID de la mesa
    header("Location: index.php");
    exit();
}
?>
```

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.