

DOCUMENTACIÓN EXAMEN FINAL

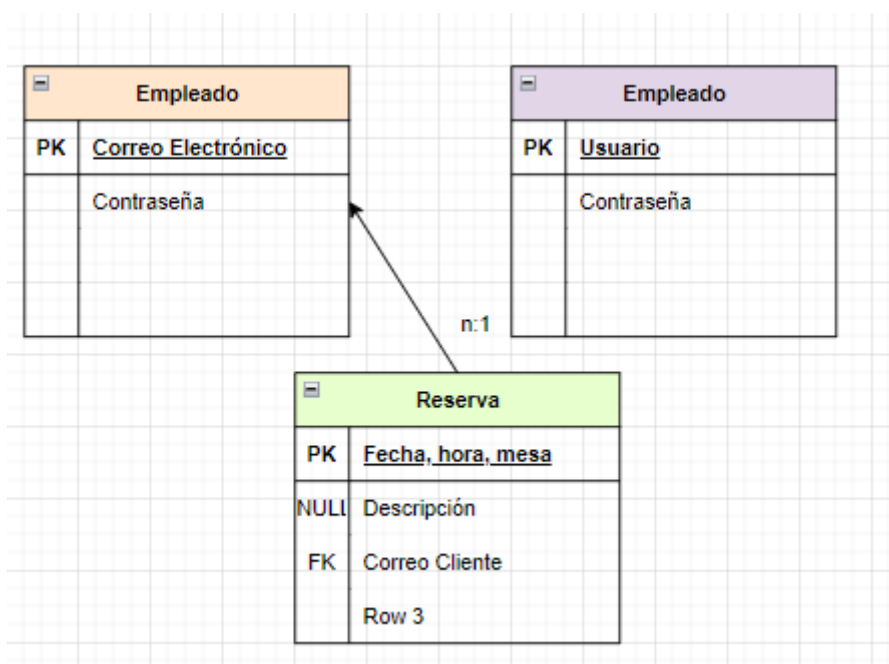
DWES

Bienvenido a la aplicación de gestión de reservas para un restaurante de sushi. Esta aplicación ha sido diseñada siguiendo el patrón Modelo Vista Controlador (MVC) y utiliza SQL como base de datos para almacenar y gestionar los datos. El enfoque MVC permite una separación clara de la lógica de negocio, la presentación de la interfaz de usuario y la manipulación de datos, lo que facilita el mantenimiento y la escalabilidad del sistema.

Modelo Entidad-Relación:

La aplicación sigue un modelo entidad-relación que consta de tres entidades principales:

- **Cliente:** Representa a los usuarios que realizan reservas en el restaurante. Cada cliente tiene un nombre de usuario (o correo electrónico) y una contraseña para acceder a la aplicación.
- **Empleado:** Representa al personal del restaurante encargado de gestionar las reservas. Al igual que los clientes, los empleados tienen un nombre de usuario y una contraseña.
- **Reservas:** Esta entidad almacena la información relacionada con las reservas realizadas. Cada reserva está asociada a un cliente y contiene detalles como la hora, mesa y fecha de la reserva.



Funcionalidades:

- **Clientes:** Los clientes pueden crear nuevas reservas, eliminar las reservas que hayan realizado y visualizar sus reservas antiguas. Para acceder a estas funcionalidades, deben iniciar sesión con su nombre de usuario (o correo electrónico) y contraseña.
- **Empleados:** Los empleados tienen privilegios adicionales, como la capacidad de eliminar cualquier reserva activa realizada por un cliente. Esto les permite gestionar eficientemente las reservas y optimizar la capacidad del restaurante. También pueden crear un nuevo usuario empleado.

Estructura de Carpetas:

Controlador:

- **Controlador.php:** Este archivo recibe todas las peticiones de la vista y dirige el flujo de la aplicación. Gestiona si el usuario es un empleado o un cliente, mostrando el inicio de sesión correspondiente y redireccionando al menú apropiado entre otras funciones:

```
////////// OPCIONES CLIENTE ////////////

//Registro Cliente:
if(isset($_POST['registrar'])){
    $correo_cliente = $_POST['correo'];
    $contrasena_cliente = $_POST['contrasena'];
    echo "correo: " . $correo_cliente;
    echo "contra: " . $contrasena_cliente;

    if(correo_ya_registrado($correo_cliente) || $correo_cliente == "" || $contrasena_cliente == ""){
        $data['body'] = 'VISTA/vistaRegistroCliente.php';
    }else{
        registrar_cliente($correo_cliente, $contrasena_cliente);
        $data['body'] = 'VISTA/vistaMenuCliente.php';
    }
}

//Inicio sesión cliente:
if(isset($_POST['iniciarSesion'])){
    $data['body'] = 'VISTA/vistaInicioCliente.php';
}
if(isset($_POST['iniciar'])){
    $correo_cliente = $_POST['correo'];
    $contrasena_cliente = $_POST['contrasena'];
    if(inicio_correcto($correo_cliente, $contrasena_cliente)){
        $data['body'] = 'VISTA/vistaMenuCliente.php';
    }else{
        $data['body'] = 'VISTA/vistaInicioCliente.php';
        //controlar aquí los errores de inicio de sesión.
    }
}
```

Cada petición viene con el comentario asociado de lo que gestiona y hacia que vista lleva y qué métodos del modelo usa.

Modelo:

- **bbdd.php**: Contiene los métodos y el código necesario para conectar a la base de datos y realizar consultas.

```
<?php

function crear_conexion($servidor, $usuario, $contrasena, $base_datos){
    return new mysqli($servidor, $usuario, $contrasena, $base_datos);
}

function cerrar_conexion($conexion){
    $conexion->close();
}

function consulta_bbdd($consulta, $conexion){
    return $conexion->query($consulta);
}

function obtener_resultados($resultado){
    return $resultado->fetch_assoc();
}

?>
```

- **config.php**: Archivo con datos constantes sobre la base de datos, como el nombre de usuario, contraseña, localhost, etc.

```
<?php
define('DB_HOST', 'localhost');
define('DB_USER', 'RESTAURANTE');
define('DB_PASSWORD', 'userUSER2');
define('DB_NAME', 'RESTAURANTE');

?>
```

- **modelo.php**: Contiene todos los métodos necesarios para gestionar los datos que llegan o salen de la base de datos. Cada método realiza una función específica:
 - **correo_ya_registrado(\$correo)**: Verifica si el correo ya está registrado en la base de datos.
 - **registrar_cliente(\$correo, \$contrasena)**: Registra un nuevo cliente en la base de datos.
 - **inicio_correcto(\$correo, \$contrasena)**: Verifica si las credenciales de inicio de sesión de un cliente son correctas.
 - **existe_reserva(\$fecha, \$hora, \$mesa)**: Verifica si ya existe una reserva para la fecha, hora y mesa especificadas.
 - **nueva_reserva(\$fecha, \$hora, \$mesa, \$descripcion, \$correoCliente)**: Registra una nueva reserva en la base de datos.
 - **reservas_activas(\$correoCliente)**: Obtiene todas las reservas activas de un cliente.

- **eliminar_reserva(\$fecha, \$hora, \$mesa):** Elimina una reserva específica de la base de datos.
- **reservas_antiguas(\$correoCliente):** Obtiene las reservas antiguas de un cliente.
- **inicio_correcto_empleado(\$usuario, \$contrasena):** Verifica si las credenciales de inicio de sesión de un empleado son correctas.
- **existe_usuario(\$usuario):** Verifica si ya existe un usuario (empleado) en la base de datos.
- **crear_usuario(\$usuario, \$contrasena):** Crea un nuevo usuario (empleado) en la base de datos.
- **todas_reservas_activas():** Obtiene todas las reservas activas en la base de datos.
- **reservas_filtradas(\$fecha):** Obtiene las reservas filtradas por fecha.

Vista:

Contiene todas las vistas de cada menú desplegable:

- **vista.php:** puedes elegir si entrar siendo cliente o entrar siendo empleado:
 - EMPLEADO:
 - vistaInicioEmpleado: para iniciar sesión
 - vistaMenuEmpleado: opción de crear un usuario o gestionar reservas
 - vistaNuevoUsuario: para crear un nuevo empleado
 - vistaReservasEmpleado: para gestionar las reservas de los clientes.
 - CLIENTE:
 - vistaRegistroCliente: para registrarse como cliente
 - vistaInicioCliente: para iniciar como cliente
 - VistaMenuCliente: con las opciones de ver, gestionar o crear reserva
 - vistaReservasCliente: ver reservas activas del cliente
 - vistaHistorialReservasCliente: ver reservas antiguas del cliente
 - vistaNuevaReserva: crear una nueva reserva.

Layout: *(carpeta dentro de VISTA):*

- **layout.php:** Esqueleto de la aplicación que incluye el encabezado y pie de página.
- **layout.css:** Hoja de estilos CSS para la aplicación.