

Entrega 1

**Sistema de Gestión Integral
para Restaurantes**

Diseño de la Base de Datos Relacional

Coderhouse

Curso: SQL



Alumna: García, Valeria Micol

Profesor: Santiago Luis Acosta Rapoani

- Parte Funcional

Introducción

Este proyecto tiene como propósito el desarrollo de una base de datos relacional orientada a la gestión integral de restaurantes. La base de datos servirá como núcleo para una plataforma que permitirá almacenar y administrar información clave sobre los restaurantes, incluyendo sus platos, empleados, mesas, reservas y clientes. Además, busca proporcionar herramientas para la organización eficiente de los recursos y el análisis de datos relacionados con las operaciones diarias.

La base de datos será diseñada para ser escalable y adaptable, con la capacidad de gestionar múltiples restaurantes en una única solución centralizada. De esta manera, se garantizará un sistema confiable y robusto para el manejo de datos esenciales.

Objetivo

El principal objetivo del proyecto es crear un sistema de gestión de datos que facilite la operación diaria de restaurantes, incluyendo:

- Organización de información operativa (restaurantes, empleados, clientes y mesas).
- Registro de pedidos, platos y promociones de forma centralizada.
- Administración eficiente de reservas y gestión de insumos.
- Generación de reportes analíticos que proporcionen insights sobre ventas, preferencias de clientes y desempeño de los empleados.

El proyecto pretende ofrecer una herramienta integral que conecte aspectos operativos, de gestión y análisis, facilitando la logística, el control contable y la generación de datos clave para apoyar la toma de decisiones estratégicas

Situación Problemática

En la actualidad, muchos restaurantes enfrentan dificultades para gestionar de manera eficiente la información clave de su operación diaria, como el manejo de reservas, administración de inventarios, seguimiento de pedidos y análisis del comportamiento de los clientes. Estas brechas generan ineficiencias operativas, limitan la capacidad de responder a las demandas de los clientes y dificultan la toma de decisiones estratégicas.

Además, la ausencia de una base de datos centralizada que integre la información de múltiples restaurantes agrava el problema al dificultar la organización, el análisis y la optimización de los recursos. La falta de acceso a reportes consolidados limita la capacidad de los administradores para identificar tendencias, mejorar la logística y ofrecer un mejor servicio al cliente.

Este proyecto busca solucionar estas brechas mediante la implementación de una base de datos relacional que permita centralizar, organizar y optimizar la información de los restaurantes. Con esta herramienta, se logrará una mejor gestión de las operaciones, una mayor satisfacción del cliente y un soporte más sólido para la toma de decisiones basadas en datos.

Modelo de Negocio

El modelo de negocio para esta solución se centra en proporcionar una herramienta tecnológica que permita a los restaurantes gestionar y optimizar sus operaciones diarias de manera eficiente. La base de datos centralizada actúa como el núcleo del sistema, integrando múltiples áreas clave del negocio, como la gestión de reservas, control de inventarios, administración de empleados, seguimiento de pedidos y análisis del comportamiento del cliente.

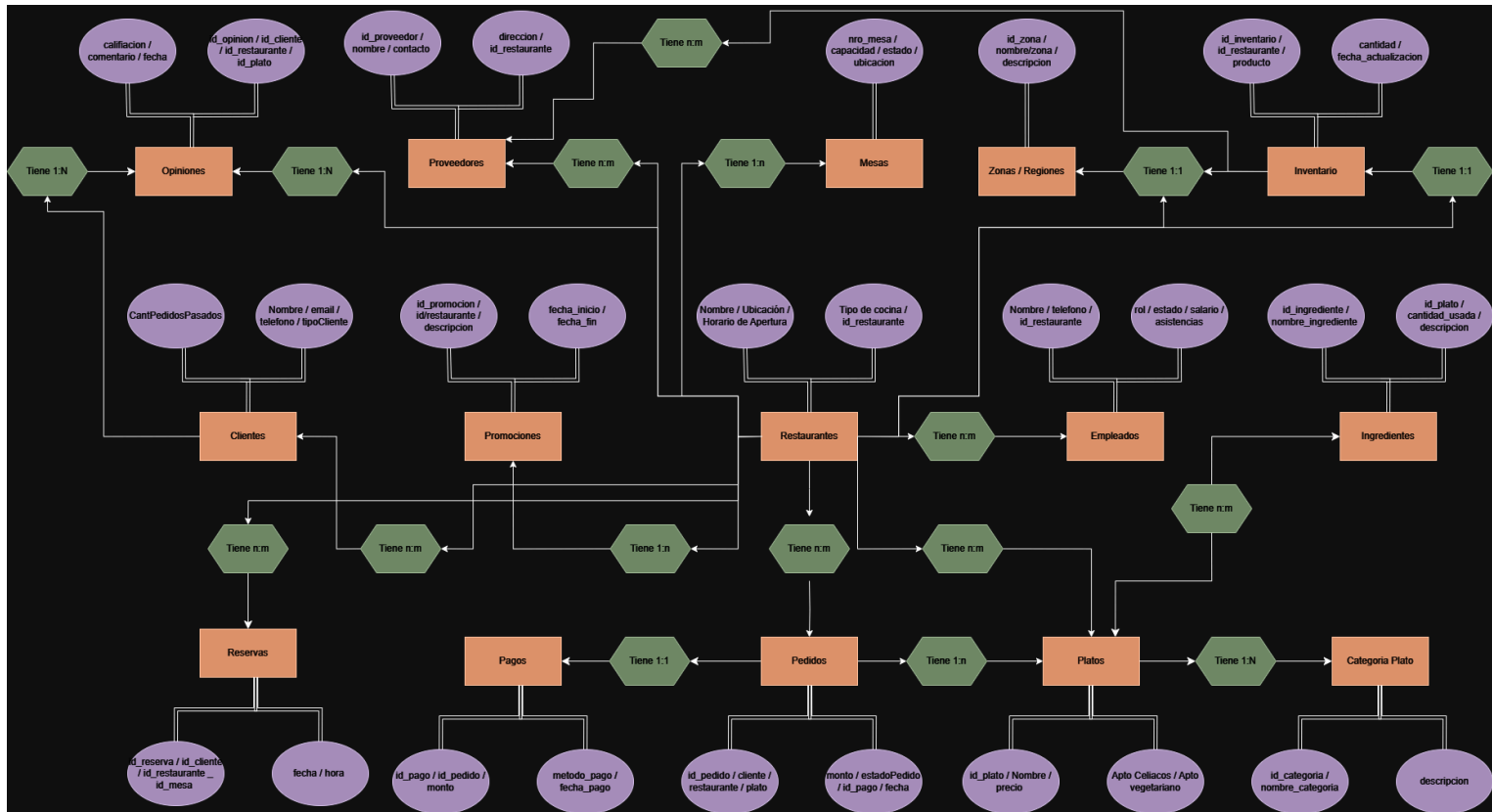
Cada restaurante que utiliza la plataforma tiene acceso a un espacio dedicado en la base de datos, lo que garantiza la organización y seguridad de su información. Además, el sistema permite a los administradores visualizar métricas clave, identificar tendencias y tomar decisiones informadas para mejorar su rentabilidad y la experiencia del cliente.

El diseño compartido de la solución permite que varios restaurantes utilicen el mismo sistema de manera independiente, lo que optimiza los costos de implementación y mantenimiento. Esta arquitectura también fomenta la escalabilidad, permitiendo a los restaurantes incorporar nuevas funcionalidades a medida que sus necesidades evolucionan. A través de esta base de datos, los restaurantes pueden fortalecer su competitividad, optimizar procesos y ofrecer un servicio más personalizado y eficiente.

Solución

Para abordar la problemática de la gestión integral de los restaurantes, se propone centralizar toda la información en una base de datos relacional diseñada en MySQL. Esta solución permitirá organizar, gestionar y analizar de manera eficiente aspectos clave como los pedidos, reservas, inventarios, promociones y relaciones con clientes, empleados y proveedores. Con un diseño intuitivo y robusto, la base de datos busca optimizar las operaciones del restaurante y proporcionar herramientas analíticas que faciliten la toma de decisiones estratégicas.

Diagrama E-R (Entidad-Relación)



- Se recomienda hacer zoom para una mejor visualización.

Listado de Tablas

1. **Restaurante:** Contiene los datos básicos de los restaurantes, como su nombre, ubicación, horario de apertura, tipo de cocina y la zona a la que pertenecen.

Tabla Restaurante

Nombre de Columna	Tipo de dato	Null	Key	Default	Extra
RestauranteID	INT	NO	PRIMARY	NULL	AUTO_INCREMENT
NombreRestaurante	VARCHAR(100)	NO		NULL	
UbicacionRestaurante	VARCHAR(200)	NO		NULL	
HorarioApertura	TIME	NO		NULL	
TipoDeCocina	VARCHAR(50)	NO		NULL	
ZonaID	VARCHAR(100)	NO	FOREIGN	NULL	

2. **Platos:** Detalla los platos ofrecidos por los restaurantes, con información como nombre, precio, categoría, y si son aptos para vegetarianos o celíacos.

Tabla Platos

Nombre de Columna	Tipo de dato	Null	Key	Default	Extra
PlatoID	INT	NO	PRIMARY	NULL	AUTO_INCREMENT
RestauranteID	INT	NO	FOREIGN	NULL	
NombrePlato	VARCHAR(100)	NO		NULL	
PrecioPlato	DECIMAL(10,2)	NO		NULL	
AptoCeliacos	BOOLEAN	YES		FALSE	
AptoVegetariano	BOOLEAN	YES		FALSE	
CategoriaID	INT	NO	FOREIGN	NULL	

3. **Cientes:** Registra la información de los clientes, como nombre, email, teléfono, tipo de cliente, y la cantidad de pedidos pasados.

Tabla Cientes

Nombre de Columna	Tipo de dato	Null	Key	Default	Extra
ClienteID	INT	NO	PRIMARY	NULL	AUTO_INCREMENT
Nombre	VARCHAR(100)	NO		NULL	
Email	VARCHAR(100)	NO	UNIQUE	NULL	
Telefono	VARCHAR(20)	NO		NULL	
TipoCliente	VARCHAR(50)	NO		NULL	
CantPedidosPasados	INT	YES		0	

4. **Mesas:** Administra la información de las mesas en los restaurantes, incluyendo su número, capacidad y estado.

Tabla Mesas

Nombre de Columna	Tipo de dato	Null	Key	Default	Extra
MesaID	INT	NO	PRIMARY	NULL	AUTO_INCREMENT
RestauranteID	INT	NO	FOREIGN	NULL	
NumeroMesa	INT	NO		NULL	
Capacidad	INT	NO		NULL	
Estado	VARCHAR(50)	YES		NULL	
Ubicacion	VARCHAR(255)	YES		NULL	

5. **Reservas:** Almacena las reservas realizadas por los clientes, incluyendo la mesa reservada, la fecha y la hora.

Tabla Reservas

Nombre de Columna	Tipo de dato	Null	Key	Default	Extra
ReservaID	INT	NO	PRIMARY	NULL	AUTO_INCREMENT
ClienteID	INT	NO	FOREIGN	NULL	
RestauranteID	INT	NO	FOREIGN	NULL	
MesaID	INT	NO	FOREIGN	NULL	
Fecha	DATE	NO		NULL	
Hora	TIME	NO		NULL	

6. **Promociones:** Registra las promociones activas en los restaurantes, incluyendo su descripción y duración.

Tabla Promociones

Nombre de Columna	Tipo de dato	Null	Key	Default	Extra
PromocionID	INT	NO	PRIMARY	NULL	AUTO_INCREMENT
RestauranteID	INT	NO	FOREIGN	NULL	
Descripcion	VARCHAR(250)	NO		NULL	
FechaInicio	DATE	NO		NULL	
FechaFin	DATE	NO		NULL	

7. **Opiniones:** Guarda las calificaciones y comentarios de los clientes sobre restaurantes o platos específicos.

Tabla Opiniones

Nombre de Columna	Tipo de dato	Null	Key	Default	Extra
OpinionID	INT	NO	PRIMARY	NULL	AUTO_INCREMENT
ClienteID	INT	NO	FOREIGN	NULL	
RestauranteID	INT	NO	FOREIGN	NULL	
PlatoID	INT	YES	FOREIGN	NULL	
Calificacion	INT	NO		NULL	
Comentario	VARCHAR(250)	YES		NULL	
Fecha	DATE	NO		NULL	

8. **Proveedores:** Contiene los datos de los proveedores asociados con los restaurantes, como nombre, contacto y dirección.

Tabla Proveedores

Nombre de Columna	Tipo de dato	Null	Key	Default	Extra
Proveedor_ID	INT	NO	PRIMARY	NULL	AUTO_INCREMENT
NombreProveedor	VARCHAR(100)	NO		NULL	
Contacto	VARCHAR(100)	YES		NULL	
Direccion	VARCHAR(200)	YES		NULL	
RestauranteID	INT	YES	FOREIGN	NULL	

9. **Zonas:** Define las zonas o regiones donde se encuentran los restaurantes, incluyendo su nombre y descripción.

Tabla Zonas

Nombre de Columna	Tipo de dato	Null	Key	Default	Extra
ZonaID	INT	NO	PRIMARY	NULL	AUTO_INCREMENT
NombreZona	VARCHAR(100)	NO		NULL	
Descripcion	VARCHAR(200)	YES		NULL	

10. **Inventario:** Lleva el control del inventario de productos en cada restaurante, con cantidad disponible y fecha de actualización.

Tabla Inventario

Nombre de Columna	Tipo de dato	Null	Key	Default	Extra
InventarioID	INT	NO	PRIMARY	NULL	AUTO_INCREMENT
RestauranteID	INT	NO	FOREIGN	NULL	
Producto	VARCHAR(100)	NO		NULL	
Cantidad	INT	NO		0	
FechaActualizacion	DATE	NO		NULL	

11. **Empleados:** Registra a los empleados de los restaurantes, incluyendo su rol, estado, salario y asistencias.

Tabla Empleados

Nombre de Columna	Tipo de dato	Null	Key	Default	Extra
EmpleadoID	INT	NO	PRIMARY	NULL	AUTO_INCREMENT
RestauranteID	INT	NO	FOREIGN	NULL	
NombreEmpleado	VARCHAR(100)	NO		NULL	
Telefono	VARCHAR(100)	YES		NULL	
Rol	VARCHAR(50)	NO		NULL	
Estado	VARCHAR(50)	YES		NULL	
Salario	DECIMAL(10,2)	NO		0	
Asistencias	INT	YES		0	

12. **Ingredientes:** Administra los ingredientes utilizados en los platos, detallando el nombre, cantidad usada y descripción.

Tabla Ingredientes

Nombre de Columna	Tipo de dato	Null	Key	Default	Extra
IngredienteID	INT	NO	PRIMARY	NULL	AUTO_INCREMENT
Nombre Ingrediente	VARCHAR(100)	NO		NULL	
Descripcion	VARCHAR(50)	YES		NULL	
Cantidad Usada	INT	NO		0	
PlatoID	INT	NO	FOREIGN	NULL	

13. **Pagos:** Registra los pagos de los pedidos, detallando el método de pago, monto y fecha.

Tabla Pagos

Nombre de Columna	Tipo de dato	Null	Key	Default	Extra
PagoID	INT	NO	PRIMARY	NULL	AUTO_INCREMENT
PedidoID	INT	NO	FOREIGN	NULL	
Monto	DECIMAL(10,2)	NO		0	
MetodoPago	VARCHAR(50)	NO		NULL	
FechaPago	DATE	NO		NULL	

14. **Pedidos:** Almacena los pedidos realizados por los clientes, incluyendo el plato solicitado, estado del pedido y monto.

Tabla Pedidos

Nombre de Columna	Tipo de dato	Null	Key	Default	Extra
PedidoID	INT	NO	PRIMARY	NULL	AUTO_INCREMENT
ClienteID	INT	NO	FOREIGN	NULL	
RestauranteID	INT	NO	FOREIGN	NULL	
PlatoID	INT	NO	FOREIGN	NULL	
Monto	DECIMAL(10,2)	NO		0	
EstadoPedido	VARCHAR(50)	YES		NULL	
FechaPedido	DATE	NO		NULL	

15. **CategoriaPlato:** Almacena las categorías de platos, como entradas, principales o postres, junto con una descripción.

Tabla CategoriaPlato

Nombre de Columna	Tipo de dato	Null	Key	Default	Extra
CategoriaID	INT	NO	PRIMARY	NULL	AUTO_INCREMENT
NombreCategoria	VARCHAR(100)	NO		NULL	
Descripcion	VARCHAR(255)	YES		NULL	

- Parte técnica

Script con la creación de las tablas

1. Ejecución del script

Output				
Action Output				
#	Time	Action	Message	Duration / Fetch
9	23:09:07	CREATE DATABASE DBResto;	1 row(s) affected	0.000 sec
10	23:09:07	USE DBResto;	0 row(s) affected	0.000 sec
11	23:09:07	CREATE TABLE Zonas (ZonaID INT PRIMARY KEY AUTO_INCREMENT, NombreZona VARCHAR(100) NOT NULL, Descripcion VARCHAR(200));	0 row(s) affected	0.047 sec
12	23:09:07	CREATE TABLE CategoriaPlato (CategoriaID INT PRIMARY KEY AUTO_INCREMENT, NombreCategoria VARCHAR(100) NOT NULL, Descripcion VARCHAR(255));	0 row(s) affected	0.015 sec
13	23:09:07	CREATE TABLE Clientes (ClienteID INT PRIMARY KEY AUTO_INCREMENT, Nombre VARCHAR(100) NOT NULL, Email VARCHAR(100) UNIQUE NOT NULL, Telefono VARCHAR(20) NOT NULL, TipoCliente VARCHAR(50) NOT NULL, CantPedidosPasados INT DEFAULT 0);	0 row(s) affected	0.016 sec
14	23:09:07	CREATE TABLE Restaurante (RestauranteID INT PRIMARY KEY AUTO_INCREMENT, NombreRestaurante VARCHAR(100) NOT NULL, UbicacionRestaurante VARCHAR(200) NOT NULL, HorarioApertura TIME NOT NULL, TipoDeCocina VARCHAR(50) NOT NULL, ZonaID INT, FOREIGN KEY (ZonaID) REFERENCES Zonas(ZonaID));	0 row(s) affected	0.015 sec
15	23:09:07	CREATE TABLE Mesas (MesaID INT PRIMARY KEY AUTO_INCREMENT, RestauranteID INT NOT NULL, NumeroMesa INT NOT NULL, Capacidad INT NOT NULL, Estado VARCHAR(50), Ubicacion VARCHAR(255), FOREIGN KEY (RestauranteID) REFERENCES Restaurante(RestauranteID));	0 row(s) affected	0.032 sec
16	23:09:07	CREATE TABLE Platos (PlatoID INT PRIMARY KEY AUTO_INCREMENT, RestauranteID INT NOT NULL, NombrePlato VARCHAR(100) NOT NULL, PrecioPlato DECIMAL(10,2) NOT NULL, AptoCeliacos BOOLEAN DEFAULT FALSE, AptoVegetariano BOOLEAN DEFAULT FALSE, CategoriaID INT NOT NULL, FOREIGN KEY (RestauranteID) REFERENCES Restaurante(RestauranteID), FOREIGN KEY (CategoriaID) REFERENCES CategoriaPlato(CategoriaID));	0 row(s) affected	0.031 sec
17	23:09:07	CREATE TABLE Promociones (PromocionID INT PRIMARY KEY AUTO_INCREMENT, RestauranteID INT NOT NULL, NombrePromocion VARCHAR(100) NOT NULL, Descripcion VARCHAR(255));	0 row(s) affected	0.031 sec
18	23:09:07	CREATE TABLE Proveedores (ProveedorID INT PRIMARY KEY AUTO_INCREMENT, NombreProveedor VARCHAR(100) NOT NULL, Descripcion VARCHAR(255));	0 row(s) affected	0.016 sec
19	23:09:07	CREATE TABLE Reservas (ReservaID INT PRIMARY KEY AUTO_INCREMENT, ClienteID INT NOT NULL, RestauranteID INT NOT NULL, FechaReserva DATETIME NOT NULL, HoraReserva TIME NOT NULL, Estado VARCHAR(50), FOREIGN KEY (ClienteID) REFERENCES Clientes(ClienteID), FOREIGN KEY (RestauranteID) REFERENCES Restaurante(RestauranteID));	0 row(s) affected	0.047 sec
20	23:09:07	CREATE TABLE Opiniones (OpinionID INT PRIMARY KEY AUTO_INCREMENT, ClienteID INT NOT NULL, RestauranteID INT NOT NULL, TextoOpinion VARCHAR(255) NOT NULL, FechaOpinion DATETIME NOT NULL, FOREIGN KEY (ClienteID) REFERENCES Clientes(ClienteID), FOREIGN KEY (RestauranteID) REFERENCES Restaurante(RestauranteID));	0 row(s) affected	0.031 sec
21	23:09:07	CREATE TABLE Inventario (InventarioID INT PRIMARY KEY AUTO_INCREMENT, RestauranteID INT NOT NULL, IngredienteID INT NOT NULL, Cantidad INT NOT NULL, FOREIGN KEY (RestauranteID) REFERENCES Restaurante(RestauranteID), FOREIGN KEY (IngredienteID) REFERENCES Ingredientes(IngredienteID));	0 row(s) affected	0.031 sec
22	23:09:07	CREATE TABLE Empleados (EmpleadoID INT PRIMARY KEY AUTO_INCREMENT, RestauranteID INT NOT NULL, NombreEmpleado VARCHAR(100) NOT NULL, EmailEmpleado VARCHAR(100) UNIQUE NOT NULL, TelefonoEmpleado VARCHAR(20) NOT NULL, TipoEmpleado VARCHAR(50) NOT NULL, CantPedidosPasados INT DEFAULT 0);	0 row(s) affected	0.063 sec
23	23:09:07	CREATE TABLE Ingredientes (IngredienteID INT PRIMARY KEY AUTO_INCREMENT, NombreIngrediente VARCHAR(100) NOT NULL, Descripcion VARCHAR(255));	0 row(s) affected	0.046 sec
24	23:09:07	CREATE TABLE Pedidos (PedidoID INT PRIMARY KEY AUTO_INCREMENT, ClienteID INT NOT NULL, RestauranteID INT NOT NULL, FechaPedido DATETIME NOT NULL, HoraPedido TIME NOT NULL, Estado VARCHAR(50), FOREIGN KEY (ClienteID) REFERENCES Clientes(ClienteID), FOREIGN KEY (RestauranteID) REFERENCES Restaurante(RestauranteID));	0 row(s) affected	0.063 sec
25	23:09:08	CREATE TABLE Pagos (PagoID INT PRIMARY KEY AUTO_INCREMENT, PedidoID INT NOT NULL, MontoPago DECIMAL(10,2) NOT NULL, FechaPago DATETIME NOT NULL, FOREIGN KEY (PedidoID) REFERENCES Pedidos(PedidoID));	0 row(s) affected	0.031 sec

2. Desarrollo de las tablas (Primero se creó la base de datos y luego se crearon las tablas). A continuación se muestran capturas del script. Sin embargo, el archivo se adjunta a la entrega.

```

1 • CREATE DATABASE DBResto;
2 • USE DBResto;
3
4
5 • CREATE TABLE Zonas (
6     ZonaID INT PRIMARY KEY AUTO_INCREMENT,
7     NombreZona VARCHAR(100) NOT NULL,
8     Descripcion VARCHAR(200)
9 );
10
11 • CREATE TABLE CategoriaPlato (
12     CategoriaID INT PRIMARY KEY AUTO_INCREMENT,
13     NombreCategoria VARCHAR(100) NOT NULL,
14     Descripcion VARCHAR(255)
15 );
16
17 • CREATE TABLE Clientes (
18     ClienteID INT PRIMARY KEY AUTO_INCREMENT,
19     Nombre VARCHAR(100) NOT NULL,
20     Email VARCHAR(100) UNIQUE NOT NULL,
21     Telefono VARCHAR(20) NOT NULL,
22     TipoCliente VARCHAR(50) NOT NULL,
23     CantPedidosPasados INT DEFAULT 0
24 );
25
28 • CREATE TABLE Restaurante (
29     RestauranteID INT PRIMARY KEY AUTO_INCREMENT,
30     NombreRestaurante VARCHAR(100) NOT NULL,
31     UbicacionRestaurante VARCHAR(200) NOT NULL,
32     HorarioApertura TIME NOT NULL,
33     TipoDeCocina VARCHAR(50) NOT NULL,
34     ZonaID INT,
35     FOREIGN KEY (ZonaID) REFERENCES Zonas(ZonaID)
36 );
37
38 • CREATE TABLE Mesas (
39     MesaID INT PRIMARY KEY AUTO_INCREMENT,
40     RestauranteID INT NOT NULL,
41     NumeroMesa INT NOT NULL,
42     Capacidad INT NOT NULL,
43     Estado VARCHAR(50),
44     Ubicacion VARCHAR(255),
45     FOREIGN KEY (RestauranteID) REFERENCES Restaurante(RestauranteID)
46 );
47
48 • CREATE TABLE Platos (
49     PlatoID INT PRIMARY KEY AUTO_INCREMENT,
50     RestauranteID INT NOT NULL,
51     NombrePlato VARCHAR(100) NOT NULL,
52     PrecioPlato DECIMAL(10,2) NOT NULL,
53     AptoCeliacos BOOLEAN DEFAULT FALSE,
54     AptoVegetariano BOOLEAN DEFAULT FALSE,
55     CategoriaID INT NOT NULL,
56     FOREIGN KEY (RestauranteID) REFERENCES Restaurante(RestauranteID),
57     FOREIGN KEY (CategoriaID) REFERENCES CategoriaPlato(CategoriaID)
58 );
59

```

```

61
62 • CREATE TABLE Promociones (
63     PromocionID INT PRIMARY KEY AUTO_INCREMENT,
64     RestauranteID INT NOT NULL,
65     Descripcion VARCHAR(250) NOT NULL,
66     FechaInicio DATE NOT NULL,
67     FechaFin DATE NOT NULL,
68     FOREIGN KEY (RestauranteID) REFERENCES Restaurante(RestauranteID)
69 );
70
71 • CREATE TABLE Proveedores (
72     ProveedorID INT PRIMARY KEY AUTO_INCREMENT,
73     NombreProveedor VARCHAR(100) NOT NULL,
74     Contacto VARCHAR(100),
75     Direccion VARCHAR(200),
76     RestauranteID INT,
77     FOREIGN KEY (RestauranteID) REFERENCES Restaurante(RestauranteID)
78 );
79
80 • CREATE TABLE Reservas (
81     ReservaID INT PRIMARY KEY AUTO_INCREMENT,
82     ClienteID INT NOT NULL,
83     RestauranteID INT NOT NULL,
84     MesaID INT NOT NULL,
85     Fecha DATE NOT NULL,
86     Hora TIME NOT NULL,
87     FOREIGN KEY (ClienteID) REFERENCES Clientes(ClienteID),
88     FOREIGN KEY (RestauranteID) REFERENCES Restaurante(RestauranteID),
89     FOREIGN KEY (MesaID) REFERENCES Mesas(MesaID)
90 );
91
92 • CREATE TABLE Opiniones (
93     OpinionID INT PRIMARY KEY AUTO_INCREMENT,
94     ClienteID INT NOT NULL,
95     RestauranteID INT NOT NULL,
96     PlatoID INT,
97     Calificacion INT NOT NULL,
98     Comentario VARCHAR(250),
99     Fecha DATE NOT NULL,
100     FOREIGN KEY (ClienteID) REFERENCES Clientes(ClienteID),
101     FOREIGN KEY (RestauranteID) REFERENCES Restaurante(RestauranteID),
102     FOREIGN KEY (PlatoID) REFERENCES Platos(PlatoID)
103 );
104
105 • CREATE TABLE Inventario (
106     InventarioID INT PRIMARY KEY AUTO_INCREMENT,
107     RestauranteID INT NOT NULL,
108     Producto VARCHAR(100) NOT NULL,
109     Cantidad INT DEFAULT 0 NOT NULL,
110     FechaActualizacion DATE DEFAULT NULL,
111     FOREIGN KEY (RestauranteID) REFERENCES Restaurante(RestauranteID)
112 );
113
114 • CREATE TABLE Empleados (
115     EmpleadoID INT PRIMARY KEY AUTO_INCREMENT,
116     RestauranteID INT NOT NULL,
117     NombreEmpleado VARCHAR(100) NOT NULL,
118     Telefono VARCHAR(100),
119     Rol VARCHAR(50) NOT NULL,
120     Estado VARCHAR(50),
121     Salario DECIMAL(10,2) DEFAULT 0 NOT NULL,
122     Asistencias INT DEFAULT 0,
123     FOREIGN KEY (RestauranteID) REFERENCES Restaurante(RestauranteID)
124 );
125
126 • CREATE TABLE Ingredientes (
127     IngredienteID INT PRIMARY KEY AUTO_INCREMENT,
128     NombreIngrediente VARCHAR(100) NOT NULL,
129     Descripcion VARCHAR(50),
130     CantidadUsada INT DEFAULT 0 NOT NULL,
131     PlatoID INT NOT NULL,
132     FOREIGN KEY (PlatoID) REFERENCES Platos(PlatoID)
133 );
134
135 • CREATE TABLE Pedidos (
136     PedidoID INT PRIMARY KEY AUTO_INCREMENT,
137     ClienteID INT NOT NULL,
138     RestauranteID INT NOT NULL,
139     PlatoID INT NOT NULL,
140     Monto DECIMAL(10,2) DEFAULT 0 NOT NULL,
141     EstadoPedido VARCHAR(50),
142     FechaPedido DATE NOT NULL,
143     FOREIGN KEY (ClienteID) REFERENCES Clientes(ClienteID),
144     FOREIGN KEY (RestauranteID) REFERENCES Restaurante(RestauranteID),
145     FOREIGN KEY (PlatoID) REFERENCES Platos(PlatoID)
146 );
147
148 • CREATE TABLE Pagos (
149     PagoID INT PRIMARY KEY AUTO_INCREMENT,
150     PedidoID INT NOT NULL,
151     Monto DECIMAL(10,2) DEFAULT 0 NOT NULL,
152     MetodoPago VARCHAR(50) NOT NULL,
153     FechaPago DATE NOT NULL,
154     FOREIGN KEY (PedidoID) REFERENCES Pedidos(PedidoID)
155 );
156
157

```