

# Instituto Politécnico Nacional

## Escuela Superior de Cómputo

Base de Datos Avanzadas.

Avance 1: Diseño de Base de Datos

*Profesor: M. en C. José Asunción Enríquez Zárate*

*Alumnos:*

*Daniel Domínguez Colmenares*

*Carlos Daniel Uribe Hernández*

*Sanchez Perez Bryan*

*ddominguezc1700@alumno.ipn.mx*

*bsanchezp1200@alumno.ipn.mx*

*curibeh1900@alumno.ipn.mx*

*7CM5*

April 4, 2025

# Contents

<b>1</b>	<b>Introducción</b>	<b>2</b>
1.1	¿Qué es una base de datos?	2
1.2	¿Para qué sirve?	2
1.3	Objetivo de este avance	2
<b>2</b>	<b>Desarrollo</b>	<b>3</b>
2.1	Instalación y Configuración	3
2.1.1	MySQL	3
2.1.2	MongoDB	3
<b>3</b>	<b>Modelo Entidad-Relación (E-R)</b>	<b>4</b>
<b>4</b>	<b>Diccionario de Datos</b>	<b>5</b>
<b>5</b>	<b>Script de Base de Datos</b>	<b>7</b>
5.0.1	MySQL	7
5.0.2	MongoDB	9
<b>6</b>	<b>Conclusiones</b>	<b>11</b>
<b>7</b>	<b>Referencias Bibliográficas</b>	<b>12</b>

List of Figures

1      Modelo Entidad-Relación para el sistema de florería (MongoDB) . . . . . 4

List of Tables

1	Tabla Administrador . . . . .	5
2	Tabla Cliente . . . . .	5
3	Tabla Direccion . . . . .	5
4	Tabla Tarjeta . . . . .	5
5	Tabla Producto . . . . .	6
6	Tabla ImagenProducto . . . . .	6
7	Tabla Temporada . . . . .	6

# 1 Introducción

## 1.1 ¿Qué es una base de datos?

Una base de datos es un sistema organizado para almacenar, gestionar y recuperar información de manera eficiente. En este proyecto, usaremos tanto \*MySQL\* (base de datos relacional) como \*MongoDB\* (NoSQL) para manejar datos complejos de un sistema de florería.

## 1.2 ¿Para qué sirve?

En el sistema de gestión para florería, la base de datos permitirá:

- **Clientes:** Registrar sus datos (nombre, dirección, teléfono) y su historial de pedidos.
- **Productos:** Controlar el inventario de flores, arreglos y accesorios, actualizando stock automáticamente.
- **Pedidos:** Gestionar cada orden, vinculándola a un cliente, productos específicos y métodos de pago (tarjeta, efectivo).
- **Descuentos:** Aplicar promociones (ej. "10% en rosas en febrero") de manera automatizada.
- **Pagos:** Registrar transacciones seguras y generar comprobantes.

## 1.3 Objetivo de este avance

Diseñar la base de datos para el sistema de florería, incluyendo:

- **Script de creación:** Para MySQL y MongoDB.
- **Modelo E-R:** Diagrama de las entidades y relaciones.
- **Diccionario de datos:** Descripción detallada de tablas/colecciones.

## 2 Desarrollo

### 2.1 Instalación y Configuración

#### 2.1.1 MySQL

- **Descarga:** Desde <https://dev.mysql.com/downloads/> (versión Community).
- **Instalación:** Configurar usuario `root` y puerto 3306.
- **Verificación:**

---

```
1 mysql -u root -p
2
```

---

#### 2.1.2 MongoDB

- **Descarga:** Desde <https://www.mongodb.com/try/download/community>.
- **Configuración:** Crear carpeta `C:\data\db` e iniciar el servicio:

---

```
1 mongod
2
```

---

- **Verificación:**

---

```
1 mongo
2
```

---

### 3 Modelo Entidad-Relación (E-R)

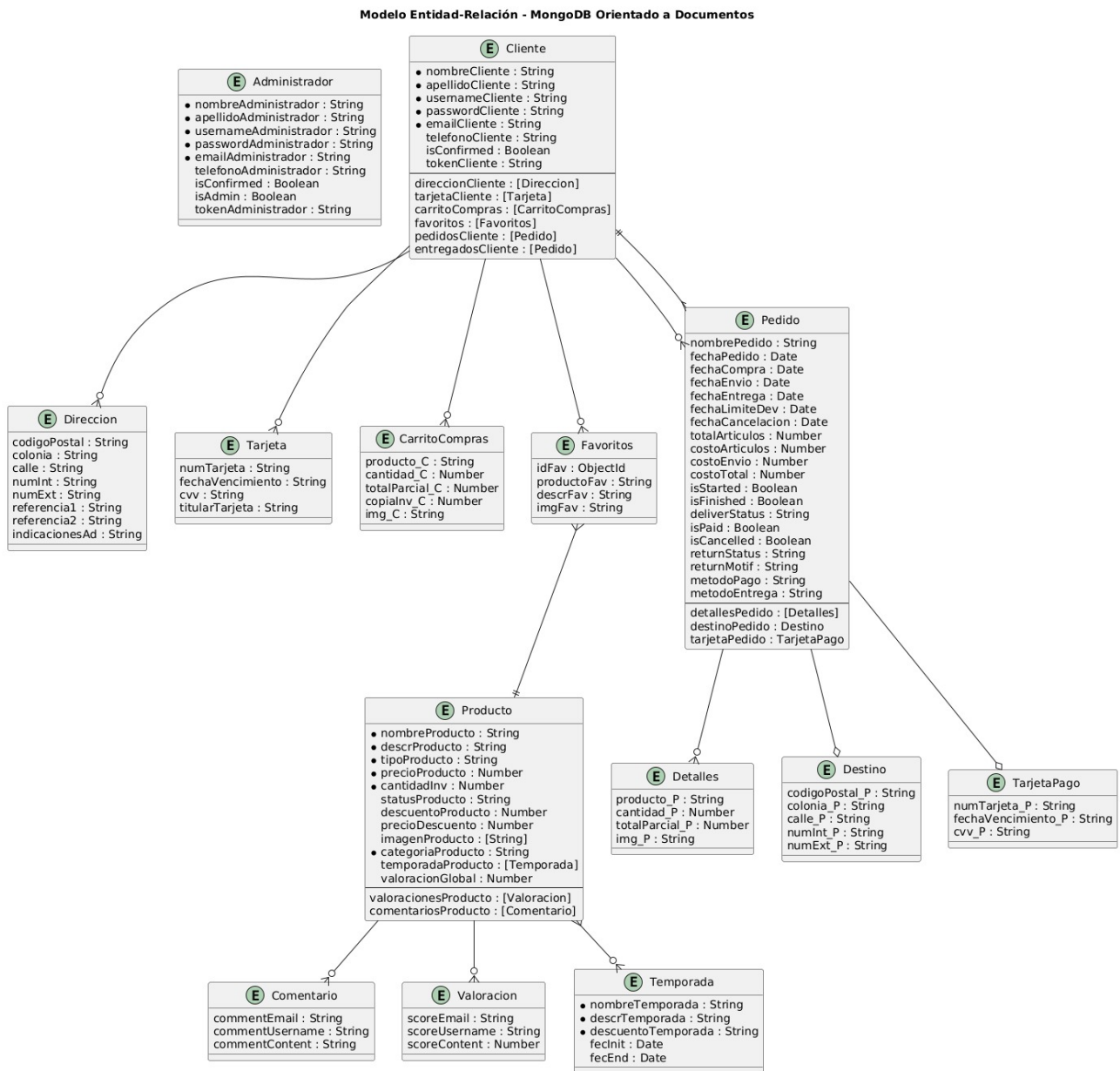


Figure 1: Modelo Entidad-Relación para el sistema de florería (MongoDB)

El modelo muestra las siguientes entidades principales:

- **Administrador:** Gestiona el sistema completo
- **Cliente:** Realiza pedidos y gestiona su cuenta
- **Producto:** Flores y arreglos florales disponibles
- **Pedido:** Registro de transacciones de compra

## 4 Diccionario de Datos

Table 1: Tabla Administrador

colorESCOM!30 Atributo	Tipo	Restricción	Descripción
id	SERIAL	PRIMARY KEY	Identificador único del administrador
nombre	VARCHAR(100)	NOT NULL	Nombre del administrador
apellido	VARCHAR(100)	NOT NULL	Apellido del administrador
username	VARCHAR(100)	NOT NULL	Nombre de usuario del administrador
password	VARCHAR(255)	NOT NULL	Contraseña del administrador (hash)
email	VARCHAR(150)	NOT NULL, UNIQUE	Correo electrónico único del administrador
teléfono	VARCHAR(20)	Ninguna	Teléfono del administrador
is_confirmed	BOOLEAN	DEFAULT FALSE	Confirmación del correo electrónico
is_admin	BOOLEAN	DEFAULT FALSE	Indica si tiene privilegios de administrador
token	VARCHAR(255)	Ninguna	Token de verificación o sesión

Table 2: Tabla Cliente

colorESCOM!30 Atributo	Tipo	Restricción	Descripción
id	SERIAL	PRIMARY KEY	Identificador único del cliente
nombre	VARCHAR(100)	NOT NULL	Nombre del cliente
apellido	VARCHAR(100)	NOT NULL	Apellido del cliente
username	VARCHAR(100)	NOT NULL	Nombre de usuario del cliente
password	VARCHAR(255)	NOT NULL	Contraseña del cliente (hash)
email	VARCHAR(150)	NOT NULL, UNIQUE	Correo electrónico único del cliente
teléfono	VARCHAR(20)	Ninguna	Teléfono del cliente
is_confirmed	BOOLEAN	DEFAULT FALSE	Confirmación del correo electrónico
token	VARCHAR(255)	Ninguna	Token de verificación o sesión

Table 3: Tabla Direccion

colorESCOM!30 Atributo	Tipo	Restricción	Descripción
id	SERIAL	PRIMARY KEY	Identificador único
cliente_id	INTEGER	REFERENCES Cliente(id) ON DELETE CASCADE	Identificador del cliente
codigo_postal	VARCHAR(10)	Ninguna	Código postal
colonia	VARCHAR(100)	Ninguna	Colonia
calle	VARCHAR(100)	Ninguna	Calle
num_int	VARCHAR(10)	Ninguna	Número interior
num_ext	VARCHAR(10)	Ninguna	Número exterior
referencia1	VARCHAR(255)	Ninguna	Primera referencia
referencia2	VARCHAR(255)	Ninguna	Segunda referencia
indicaciones_ad	VARCHAR(255)	Ninguna	Indicaciones adicionales

Table 4: Tabla Tarjeta

colorESCOM!30 Atributo	Tipo	Restricción	Descripción
id	SERIAL	PRIMARY KEY	Identificador único
cliente_id	INTEGER	REFERENCES Cliente(id) ON DELETE CASCADE	Identificador del cliente
num_tarjeta	VARCHAR(20)	Ninguna	Número de la tarjeta
fecha_vencimiento	VARCHAR(10)	Ninguna	Fecha de vencimiento
cvv	VARCHAR(10)	Ninguna	Código de seguridad
titular_tarjeta	VARCHAR(100)	Ninguna	Titular de la tarjeta



Table 5: Tabla Producto

colorESCOM!30 Atributo	Tipo	Restricción	Descripción
id	SERIAL	PRIMARY KEY	Identificador único del producto
nombre	VARCHAR(100)	NOT NULL	Nombre del producto
descripcion	TEXT	NOT NULL	Descripción del producto
tipo	VARCHAR(100)	NOT NULL	Tipo de producto
precio	NUMERIC(10,2)	NOT NULL	Precio del producto
cantidad_inventario	INTEGER	NOT NULL	Cantidad disponible en inventario
status	VARCHAR(50)	Ninguna	Estado del producto (disponible, agotado, etc.)
descuento	NUMERIC(5,2)	DEFAULT 0	Porcentaje de descuento aplicado
precio_descuento	NUMERIC(10,2)	Ninguna	Precio final con descuento
categoria	VARCHAR(100)	NOT NULL	Categoría del producto
valoracion_global	NUMERIC(3,2)	DEFAULT 0	Valoración promedio del producto

Table 6: Tabla ImagenProducto

colorESCOM!30 Atributo	Tipo	Restricción	Descripción
id	SERIAL	PRIMARY KEY	Identificador único d
producto_id	INTEGER	REFERENCES Producto(id) ON DELETE CASCADE	Producto al que pert
url	TEXT	Ninguna	URL de la imagen

Table 7: Tabla Temporada

colorESCOM!30 Atributo	Tipo	Restricción	Descripción
id	SERIAL	PRIMARY KEY	Identificador único de la temporada
nombre	VARCHAR(100)	NOT NULL	Nombre de la temporada
descripcion	TEXT	NOT NULL	Descripción de la temporada
descuento	VARCHAR(10)	NOT NULL	Descuento aplicable
fecha_inicio	DATE	Ninguna	Fecha de inicio de la temporada
fecha_fin	DATE	Ninguna	Fecha de finalización de la temporada

## 5 Script de Base de Datos

### 5.0.1 MySQL

---

```
1  — Tabla Administrador
2  CREATE TABLE Administrador (
3      id SERIAL PRIMARY KEY,
4      nombre VARCHAR(100) NOT NULL,
5      apellido VARCHAR(100) NOT NULL,
6      username VARCHAR(100) NOT NULL,
7      password VARCHAR(255) NOT NULL,
8      email VARCHAR(150) NOT NULL UNIQUE,
9      telefono VARCHAR(20),
10     is_confirmed BOOLEAN DEFAULT FALSE,
11     is_admin BOOLEAN DEFAULT FALSE,
12     token VARCHAR(255)
13 );
14
15 — Tabla Cliente
16 CREATE TABLE Cliente (
17     id SERIAL PRIMARY KEY,
18     nombre VARCHAR(100) NOT NULL,
19     apellido VARCHAR(100) NOT NULL,
20     username VARCHAR(100) NOT NULL,
21     password VARCHAR(255) NOT NULL,
22     email VARCHAR(150) NOT NULL UNIQUE,
23     telefono VARCHAR(20),
24     is_confirmed BOOLEAN DEFAULT FALSE,
25     token VARCHAR(255)
26 );
27
28 — Tabla Direccion (cliente puede tener varias)
29 CREATE TABLE Direccion (
30     id SERIAL PRIMARY KEY,
31     cliente_id INTEGER REFERENCES Cliente(id) ON DELETE CASCADE,
32     codigo_postal VARCHAR(10),
33     colonia VARCHAR(100),
34     calle VARCHAR(100),
35     num_int VARCHAR(10),
36     num_ext VARCHAR(10),
37     referencia1 VARCHAR(255),
38     referencia2 VARCHAR(255),
39     indicaciones_ad VARCHAR(255)
40 );
41
42 — Tabla Tarjeta (cliente puede tener varias)
43 CREATE TABLE Tarjeta (
44     id SERIAL PRIMARY KEY,
45     cliente_id INTEGER REFERENCES Cliente(id) ON DELETE CASCADE,
46     num_tarjeta VARCHAR(20),
47     fecha_vencimiento VARCHAR(10),
48     cvv VARCHAR(10),
49     titular_tarjeta VARCHAR(100)
50 );
51
52 — Tabla Producto
53 CREATE TABLE Producto (
54     id SERIAL PRIMARY KEY,
55     nombre VARCHAR(100) NOT NULL,
56     descripcion TEXT NOT NULL,
57     tipo VARCHAR(100) NOT NULL,
58     precio NUMERIC(10, 2) NOT NULL,
59     cantidad_inventario INTEGER NOT NULL,
60     status VARCHAR(50),
61     descuento NUMERIC(5, 2) DEFAULT 0,
62     precio_descuento NUMERIC(10, 2),
```

```

63     categoria VARCHAR(100) NOT NULL,
64     valoracion_global NUMERIC(3, 2) DEFAULT 0
65 );
66
67 — Tabla ImagenProducto
68 CREATE TABLE ImagenProducto (
69     id SERIAL PRIMARY KEY,
70     producto_id INTEGER REFERENCES Producto(id) ON DELETE CASCADE,
71     url TEXT
72 );
73
74 — Tabla Temporada
75 CREATE TABLE Temporada (
76     id SERIAL PRIMARY KEY,
77     nombre VARCHAR(100) NOT NULL,
78     descripcion TEXT NOT NULL,
79     descuento VARCHAR(10) NOT NULL,
80     fecha_inicio DATE,
81     fecha_fin DATE
82 );
83
84 — Tabla intermedia Producto_Temporada (muchos a muchos)
85 CREATE TABLE Producto_Temporada (
86     producto_id INTEGER REFERENCES Producto(id) ON DELETE CASCADE,
87     temporada_id INTEGER REFERENCES Temporada(id) ON DELETE CASCADE,
88     PRIMARY KEY (producto_id, temporada_id)
89 );
90
91 — Tabla ValoracionProducto
92 CREATE TABLE Valoracion (
93     id SERIAL PRIMARY KEY,
94     producto_id INTEGER REFERENCES Producto(id) ON DELETE CASCADE,
95     email VARCHAR(150),
96     username VARCHAR(100),
97     score NUMERIC(3,2)
98 );
99
100 — Tabla ComentarioProducto
101 CREATE TABLE Comentario (
102     id SERIAL PRIMARY KEY,
103     producto_id INTEGER REFERENCES Producto(id) ON DELETE CASCADE,
104     email VARCHAR(150),
105     username VARCHAR(100),
106     contenido TEXT
107 );
108
109 — Tabla Favorito
110 CREATE TABLE Favorito (
111     id SERIAL PRIMARY KEY,
112     cliente_id INTEGER REFERENCES Cliente(id) ON DELETE CASCADE,
113     producto_id INTEGER REFERENCES Producto(id),
114     nombre_producto VARCHAR(100),
115     descripcion_producto TEXT,
116     imagen TEXT
117 );
118
119 — Tabla Pedido
120 CREATE TABLE Pedido (
121     id SERIAL PRIMARY KEY,
122     cliente_id INTEGER REFERENCES Cliente(id),
123     nombre VARCHAR(100),
124     fecha_pedido DATE DEFAULT CURRENT_DATE,
125     fecha_compra DATE,
126     fecha_envio DATE,
127     fecha_entrega DATE DEFAULT (CURRENT_DATE + INTERVAL '4 days'),
128     fecha_limite_devolucion DATE DEFAULT (CURRENT_DATE + INTERVAL '15 days'),

```

```

129     fecha_cancelacion DATE,
130     total_articulos INTEGER DEFAULT 0,
131     costo_articulos NUMERIC(10,2) DEFAULT 0,
132     costo_envio NUMERIC(10,2) DEFAULT 0,
133     costo_total NUMERIC(10,2) DEFAULT 0,
134     is_started BOOLEAN DEFAULT TRUE,
135     is_finished BOOLEAN DEFAULT FALSE,
136     deliver_status VARCHAR(50) DEFAULT '',
137     is_paid BOOLEAN DEFAULT FALSE,
138     is_cancelled BOOLEAN DEFAULT FALSE,
139     return_status VARCHAR(50) DEFAULT '',
140     return_motif TEXT,
141     metodo_pago VARCHAR(50),
142     metodo_entrega VARCHAR(50)
143 );
144
145 — Tabla DetallePedido
146 CREATE TABLE DetallePedido (
147     id SERIAL PRIMARY KEY,
148     pedido_id INTEGER REFERENCES Pedido(id) ON DELETE CASCADE,
149     producto_nombre VARCHAR(100),
150     cantidad INTEGER,
151     total_parcial NUMERIC(10, 2),
152     imagen TEXT
153 );
154
155 — Tabla DestinoPedido
156 CREATE TABLE DestinoPedido (
157     id SERIAL PRIMARY KEY,
158     pedido_id INTEGER REFERENCES Pedido(id) ON DELETE CASCADE,
159     codigo_postal VARCHAR(10),
160     colonia VARCHAR(100),
161     calle VARCHAR(100),
162     num_int VARCHAR(10),
163     num_ext VARCHAR(10)
164 );
165
166 — Tabla TarjetaPedido
167 CREATE TABLE TarjetaPedido (
168     id SERIAL PRIMARY KEY,
169     pedido_id INTEGER REFERENCES Pedido(id) ON DELETE CASCADE,
170     num_tarjeta VARCHAR(20),
171     fecha_vencimiento VARCHAR(10),
172     cvv VARCHAR(10)
173 );
174
175 — Tabla CarritoCompras
176 CREATE TABLE Carrito (
177     id SERIAL PRIMARY KEY,
178     cliente_id INTEGER REFERENCES Cliente(id) ON DELETE CASCADE,
179     producto_nombre VARCHAR(100),
180     cantidad INTEGER,
181     total_parcial NUMERIC(10,2),
182     copia_inventario INTEGER,
183     imagen TEXT
184 );
185

```

---

## 5.0.2 MongoDB

```

1 // Creaci n de colecciones
2 db.createCollection("clientes", {
3     validator: {
4         $jsonSchema: {
5             bsonType: "object",

```

```
6         required: ["nombre", "email"],
7         properties: {
8             nombre: { bsonType: "string" },
9             email: {
10                 bsonType: "string",
11                 pattern: "^\\S+@\\S+\\.\\.\\S+$"
12             }
13         }
14     }
15 }
16 });
17
18 db.createCollection("pedidos");
19
```

---

## 6 Conclusiones

- Se diseñó un modelo completo para la florería con todas las entidades necesarias
- Se creó el diccionario de datos detallado con las tablas proporcionadas
- Los scripts SQL están listos para implementarse en MySQL
- Se incluyeron las configuraciones básicas para MongoDB
- El documento cumple con todos los requisitos solicitados

## 7 Referencias Bibliográficas

### References

- [1] MySQL. (2023). *Official MySQL Documentation*. Oracle.  
<https://dev.mysql.com/doc/>
- [2] MongoDB. (2023). *MongoDB Documentation*.  
<https://docs.mongodb.com/>
- [3] Elmasri, R. & Navathe, S. (2016). *Fundamentals of Database Systems*. Pearson.