

“PAGINA WEB PARA ATENCION AL CLIENTE EN UNA DULCERIA”

PROYECTO DE TALLER DE BASE DE DATOS

**INGENIERIA EN SISTEMAS COMPUTACIONALES**

PRESENTA:

**ISIDRO ARREDONDO CAMARENA**

**EDGAR CORTÉS RESÉNDIZ**

**MARTINEZ GONZALEZ LEONARDO**

JIQUILPAN, MICHOACÁN, SEPTIEMBRE DE 2024

Índice

[Link de GitHub con las colecciones y la inserción de datos 3](#_Toc181992034)

[Validación de Datos 3](#_Toc181992035)

[Compras 3](#_Toc181992036)

[Dulces 4](#_Toc181992037)

[Elaboración 5](#_Toc181992038)

[Fabricas 6](#_Toc181992039)

[Materias Primas 8](#_Toc181992040)

[Pedidos 9](#_Toc181992041)

[Proveedores 10](#_Toc181992042)

[Registros 11](#_Toc181992043)

[Repartos 12](#_Toc181992044)

[Sucursales 13](#_Toc181992045)

[Trabajadores 15](#_Toc181992046)

[Usuarios 17](#_Toc181992047)

[Ventas 19](#_Toc181992048)

[Modelado de datos 21](#_Toc181992049)

[Proveedores-Compras 21](#_Toc181992050)

[Materias Primas-Elaboración-Fabricas-Dulces-Trabajadores 23](#_Toc181992051)

[Sucursales-trabajadores-registros-pedidos-repartos 25](#_Toc181992052)

[Pedidos-dulces-usuarios 27](#_Toc181992053)

[Información en colecciones 29](#_Toc181992054)

[Compras 29](#_Toc181992055)

[Dulces 29](#_Toc181992056)

[Elaboración 30](#_Toc181992057)

[Fabricas 31](#_Toc181992058)

[Materias Primas 32](#_Toc181992059)

[Pedidos 33](#_Toc181992060)

[Proveedores 34](#_Toc181992061)

[Registros 35](#_Toc181992062)

[Repartos 35](#_Toc181992063)

[Sucursales 36](#_Toc181992064)

[Trabajadores 38](#_Toc181992065)

[Usuarios 39](#_Toc181992066)

[Ventas 40](#_Toc181992067)

[Consultas generadoras de JSON inserción de datos 42](#_Toc181992068)

[Compras 42](#_Toc181992069)

[Dulces 42](#_Toc181992070)

[Elaboración 43](#_Toc181992071)

[Fabricas 43](#_Toc181992072)

[Materias Primas 44](#_Toc181992073)

[Pedidos 44](#_Toc181992074)

[Proveedores 45](#_Toc181992075)

[Registros 45](#_Toc181992076)

[Repartos 45](#_Toc181992077)

[Sucursales 46](#_Toc181992078)

[Trabajadores 46](#_Toc181992079)

[Usuarios 47](#_Toc181992080)

[Ventas 47](#_Toc181992081)

# Link de GitHub con las colecciones y la inserción de datos

<https://github.com/sidral24/Dulceria>

# Validación de Datos

## Compras

//Usar la base de datos

use("Dulceria")

//Crear la conexion de usuarios

db.createCollection("compras", {

    validator: {

        $jsonSchema: {

            bsonType: 'object',

            title: 'Validacion del objeto Compras',

            required: ['id\_mp', 'id\_compra', 'id\_proveedor', 'lote', 'cantidad', 'precio'],

            properties: {

                id\_compra: { bsonType: 'int' },

                id\_proveedor: { bsonType: 'int' },

                id\_mp: { bsonType: 'int' },

                fecha\_compra: { bsonType: 'date' },

                lote: { bsonType: 'string' },

                cantidad: { bsonType: 'string' },

                precio: { bsonType: 'int' },

                descuento: { bsonType: 'int' }

            }

        }

    }

})

db.compras.createIndex(

    { id\_compra: 1 },

    { unique: true }

)

db.compras.drop()

## Dulces

//Usar la base de datos

use("Dulceria")

//Crear la conexion de usuarios

db.createCollection("dulces", {

    validator: {

        $jsonSchema: {

            bsonType: 'object',

            title: 'Validacion del objeto Dulces',

            required: ['id\_dulce', 'nombre', 'precio\_uni', 'lote'],

            properties: {

                id\_dulce: { bsonType: 'int' },

                nombre: { bsonType: 'string' },

                precio\_uni: { bsonType: 'int' },

                stock: { bsonType: 'int' },

                lote: { bsonType: 'string' },

                fecha\_cad: { bsonType: 'date' },

                reorden: { bsonType: 'int' },

                tipo: { enum: ['CON AZUCAR', 'SIN AZUCAR'] }

            }

        }

    }

})

db.dulces.createIndex(

    { id\_dulce: 1 },

    { unique: true }

)

db.dulces.drop()

## Elaboración

//Usar la base de datos

use("Dulceria")

//Crear la conexion de usuarios

db.createCollection("elaboracion", {

    validator: {

        $jsonSchema: {

            bsonType: 'object',

            title: 'Validacion del objeto Dulces',

            required: ['id\_elaboracion', 'id\_fabrica', 'id\_trabajdor', 'id\_mp', 'id\_dulce', 'cantidad'],

            properties: {

                id\_elaboracion: { bsonType: 'int' },

                id\_fabrica: { bsonType: 'int' },

                id\_trabajador: { bsonType: 'int' },

                id\_mp: { bsonType: 'int' },

                id\_dulce: { bsonType: 'int' },

                cantidad: { bsonType: 'int' }

            }

        }

    }

})

db.elaboracion.createIndex(

    { id\_elaboracion: 1 },

    { unique: true }

)

db.elaboracion.drop()

## Fabricas

//Usar la base de datos

use("Dulceria")

//Crear la conexion de usuarios

db.createCollection("Fabricas", {

    validator: {

        $jsonSchema: {

            bsonType: 'object',

            title: 'Validacion del objeto Dulces',

            required: ['id\_fabrica', 'nom\_responsable', 'estado', 'municipio', 'domicilio', 'colonia'],

            properties: {

                id\_fabrica: { bsonType: 'int' },

                telefono: { bsonType: 'string' },

                nom\_responsable: {

                    bsonType: "object",

                    required: ['nombre', 'apellido\_parteno'],

                    properties: {

                        nombre: { bsonType: 'string' },

                        apellido\_paterno: { bsonType: 'string' },

                        apellido\_materno: { bsonType: 'string' }

                    }

                },

                Direccion: {

                    bsonType: "object",

                    required: ['Calle', 'Numero', 'Colonia', 'cp'],

                    properties: {

                        Calle: { bsonType: 'string' },

                        Numero: { bsonType: 'string' },

                        Colonia: { bsonType: 'string' },

                        Municipio: { bsonType: 'string' },

                        Estado: { bsonType: 'string' },

                        cp: { bsonType: 'int' }

                    }

                }

            }

        }

    }

})

db.dulces.createIndex(

    { nom\_responsable: 1 },

    { unique: true }

)

db.dulces.drop()

## Materias Primas

//Usar la base de datos

use("Dulceria")

//Crear la conexion de usuarios

db.createCollection("materias\_primas", {

    validator: {

        $jsonSchema: {

            bsonType: 'object',

            title: 'Validacion del objeto Materias Primas',

            required: ['id\_mp', 'nombre', 'lote', 'fecha\_cad', 'precio'],

            properties: {

                id\_mp: { bsonType: 'int' },

                nombre: { bsonType: 'string' },

                tipo: { bsonType: 'string' },

                lote: { bsonType: 'string' },

                existencia: { bsonType: 'string' },

                fecha\_cad: { bsonType: 'date' },

                precio: { bsonType: 'int' }

            }

        }

    }

})

db.materias\_primas.createIndex(

    { id\_mp: 1, nombre: 1 },

    { unique: true }

)

db.materias\_primas.drop()

## Pedidos

//Usar la base de datos

use("Dulceria")

//Crear la conexion de usuarios

db.createCollection("pedidos", {

    validator: {

        $jsonSchema: {

            bsonType: 'object',

            title: 'Validacion del objeto Compras',

            required: ['id\_pedido', 'id\_usuario', 'id\_sucursal', 'id\_dulce', 'cantidad'],

            properties: {

                id\_pedido: { bsonType: 'int' },

                id\_usuario: { bsonType: 'int' },

                id\_sucursal: { bsonType: 'int' },

                id\_dulce: { bsonType: 'int' },

                cantidad: { bsonType: 'int' },

                fecha\_pedido: { bsonType: 'date' },

            }

        }

    }

})

db.pedidos.createIndex(

    { id\_pedido: 1 },

    { unique: true }

)

db.compras.drop()

## Proveedores

use("Dulceria")

//crear la coleccion de ususarios

db.createCollection("Proveedores", {

    validator: {

        $jsonSchema: {

            bsonType: "object",

            title: 'valdacion del objeto usuarios',

            required: ['id\_proveedor', 'nombre', 'rfc', 'direccion'],

            properties: {

                id\_proveedor: { bsonType: 'int' },

                nombre: { bsonType: 'string' },

                rfc: { bsonType: 'string' },

                direccion: { bsonType: "string" },

                telefono: {

                    bsonType: 'array',

                    minItems: 1, //Especificar el minimo requerido

                    uniqueItems: true, //No telefonos repetidos

                    items: { bsonType: 'string' }

                }

            }

        }

    }

})

db.Proveedores.createIndex(

    { id\_proveedor: 1 },

    { unique: true }

)

## Registros

//Usar la base de datos

use("Dulceria")

//Crear la conexion de usuarios

db.createCollection("registros", {

    validator: {

        $jsonSchema: {

            bsonType: 'object',

            title: 'Validacion del objeto Registros',

            required: ['id\_registro', 'id\_sucursal', 'id\_dulce', 'precio\_uni', 'fecha\_recepcion'],

            properties: {

                id\_registro: { bsonType: 'int' },

                id\_sucursal: { bsonType: 'int' },

                id\_dulce: { bsonType: 'int' },

                cantidad: { bsonType: 'int' },

                fecha\_recepcion: { bsonType: 'date' },

                precio\_uni: { bsonType: 'int' }

            }

        }

    }

})

db.registros.createIndex(

    { id\_registro: 1 },

    { unique: true }

)

## Repartos

use("Dulceria")

//Crear la conexion de usuarios

db.createCollection("repartos", {

    validator: {

        $jsonSchema: {

            bsonType: 'object',

            title: 'Validacion del objeto repartos',

            required: ['id\_fabrica', 'id\_repartos', 'id\_sucursal', 'id\_trabajador', 'id\_dulce'],

            properties: {

                id\_repartos: { bsonType: 'int' },

                id\_fabrica: { bsonType: 'int' },

                id\_sucursal: { bsonType: 'int' },

                id\_trabajador: { bsonType: 'int' },

                id\_dulce: { bsonType: 'int' },

                cantidad: { bsonType: 'int' },

                fecha\_reparto: { bsonType: 'date' }

            }

        }

    }

})

db.repartos.createIndex(

    { id\_repartos: 1 },

    { unique: true }

)

db.repartos.drop()

## Sucursales

use("Dulceria")

//Crear la conexion de usuarios

db.createCollection("sucursales", {

    validator: {

        $jsonSchema: {

            bsonType: 'object',

            title: 'Validacion del objeto sucursales',

            required: ['id\_sucursal', 'telefono'],

            properties: {

                id\_sucursal: { bsonType: 'int' },

                nom\_responsable: {

                    bsonType: "object",

                    required: ['nombre', 'apellido\_parteno'],

                    properties: {

                        nombre: { bsonType: 'string' },

                        apellido\_paterno: { bsonType: 'string' },

                        apellido\_materno: { bsonType: 'string' }

                    }

                },

                telefono: {

                    bsonType: 'array',

                    minItems: 1, //Especificar el minimo requerido

                    uniqueItems: true, //No telefonos repetidos

                    items: { bsonType: 'string' }

                },

                direccion: {

                    bsonType: "object",

                    title: 'Validación del objeto Address',

                    required: ['estado', 'municipio', 'colonia', 'calle', 'numero', 'cp'],

                    properties: {

                        calle: { bsonType: 'string' },

                        numero: { bsonType: 'string' },

                        colonia: { bsonType: 'string' },

                        municipio: { bsonType: 'string' },

                        estado: { bsonType: 'string' },

                        cp: { bsonType: 'int' },

                    }

                },

            }

        }

    }

})

db.repartos.createIndex(

    { id\_sucursal: 1 },

    { unique: true }

)

db.repartos.drop()

## Trabajadores

use("Dulceria")

//crear la coleccion de ususarios

db.createCollection("trabajadores", {

    validator: {

        $jsonSchema: {

            bsonType: "object",

            title: 'valdacion del objeto usuarios',

            required: ['id\_trabajador', 'id\_sucursal', 'nombre', 'rfc'],

            properties: {

                id\_trabajador: { bsonType: 'int' },

                id\_sucursal: { bsonType: 'int' },

                nombre: {

                    bsonType: "object",

                    title: 'Validación del objeto Address',

                    required: ['apellido\_paterno', 'nombres'],

                    properties: {

                        apellido\_paterno: { bsonType: 'string' },

                        apellido\_materno: { bsonType: 'string' },

                        nombres: { bsonType: 'string' },

                    }

                },

                fechaNac: { bsonType: 'date' },

                fechaIngreso: { bsonType: 'date' },

                rfc: { bsonType: 'string' },

                tipo: { bsonType: 'string' },

                telefono: {

                    bsonType: 'array',

                    minItems: 1, //Especificar el minimo requerido

                    uniqueItems: true, //No telefonos repetidos

                    items: { bsonType: 'string' }

                }

            }

        }

    }

})

db.Trabajadores.createIndex(

    {

        id\_sucursal: 1,

        id\_trabajador: 1

    },

    { unique: true }

)

## Usuarios

//USAR LA BASE DE DATOS

use("Dulceria")

//crear la coleccion de ususarios

db.createCollection("usuarios", {

    validator: {

        $jsonSchema: {

            bsonType: "object",

            title: 'valdacion del objeto usuarios',

            required: ['email', 'id\_usuario'],

            properties: {

                id\_usuario: {

                    bsonType: 'int'

                },

                nombre: {

                    bsonType: "object",

                    title: 'Validación del objeto nombre',

                    required: ['apellido\_paterno', 'nombres'],

                    properties: {

                        apellido\_paterno: { bsonType: 'string' },

                        apellido\_materno: { bsonType: 'string' },

                        nombres: { bsonType: 'string' },

                    }

                },

                email: {

                    bsonType: 'string',

                    pattern: "^[A-Za-z\\d.-]{1,}@[A-Za-z\\d.-]{1,}[.][A-Za-z\\d.-]{1,}$"

                },

                telefono: {

                    bsonType: 'array',

                    minItems: 1, //Especificar el minimo requerido

                    uniqueItems: true, //No telefonos repetidos

                    items: { bsonType: 'string' }

                }

            }

        }

    }

})

db.usuarios.createIndex(

    {

        email: 1,

        idUsuario: 1

    },

    { unique: true }

)

## Ventas

use("Dulceria")

//Crear la conexion de usuarios

db.createCollection("ventas", {

    validator: {

        $jsonSchema: {

            bsonType: 'object',

            title: 'Validacion del objeto Ventas',

            required: ['id\_venta', 'id\_usuario', 'id\_sucursal', 'id\_dulce', 'cantidad'],

            properties: {

                id\_venta: { bsonType: 'int' },

                id\_usuario: { bsonType: 'int' },

                id\_sucursal: { bsonType: 'int' },

                id\_dulce: { bsonType: 'int' },

                cantidad: { bsonType: 'int' },

                fecha: { bsonType: 'date' },

                precio\_uni: { bsonType: 'int' },

                descuento: { bsonType: 'int' }

            }

        }

    }

})

db.ventas.createIndex(

    { id\_venta: 1 },

    { unique: true }

)

db.ventas.drop()

# Modelado de datos

## Proveedores-Compras

Se hizo una relación de N:N en cada una de estas tablas, esto con la finalidad de que como los datos son independientes y si existe un cambio constante en los registros de estos mismos, es necesario que las tablas sean referenciadas de esta manera para así poder mantener un flujo de datos mas eficiente sin alterar tanto las colecciones y que sea menos eficiente.

|  |
| --- |
| Proveedores |
| idProveedor <int> |
| nombre <string> |
| rfc <string> |
| |  | | --- | | Direccion | | Calle <string> | | Ciudad <string> | | Region <string> | | CodigoPostal <string> | | Pais <string> | |
| telefono <array> |

|  |
| --- |
| Compras |
| id\_compra <int> |
| idProveedor <int> |
| id\_mp <int> |
| fecha\_compra <int> |
| lote <string> |
| cantidad <string> |
| precio <int> |
| descuento <int> |

|  |
| --- |
| Materias\_Primas |
| id\_mp <int> |
| nombre <string> |
| tipo <string> |
| lote <string> |
| existencia <int> |
| fecha\_cad <date> |
| precio <int> |

Por la parte de la tabla de proveedores decidimos embeber la dirección, ya que en este la dirección de los proveedores es muy difícil de que cambie frecuentemente, así que de esta manera nos evitamos de crear una colección exclusivamente para esta sección, además de que en las consultas necesitamos encontrar la dirección de manera mas eficiente así que de esta manera podemos adquirir esta información sin tiempo de búsqueda tan extensos.

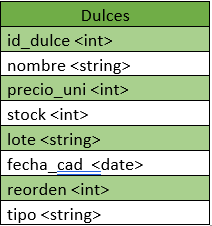
|  |
| --- |
| Proveedores |
| idProveedor <int> |
| nombre <string> |
| rfc <string> |
| |  | | --- | | Direccion | | Calle <string> | | Ciudad <string> | | Region <string> | | CodigoPostal <string> | | Pais <string> | |
| telefono <array> |

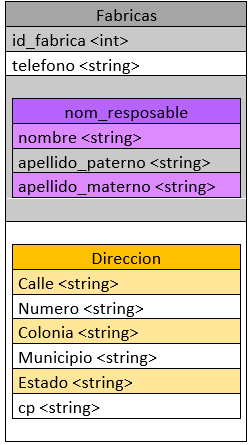
## Materias Primas-Elaboración-Fabricas-Dulces-Trabajadores

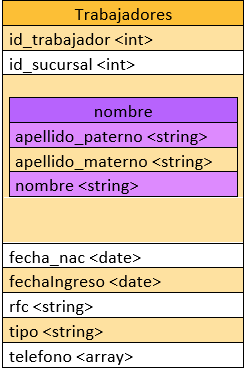
Para en esta sección de relaciones podemos observar que de la misma manera que la anterior podemos apreciar que las relaciones entre tablas se siguen manteniendo ya que en este apartado los datos son muy susceptibles a cambios, así que de esta manera nos evitamos tiempos de actualización de las colecciones que tienen otras colecciones embebidas, además de que por la parte de las relaciones podemos ver que la que lleva más carga de información y de cambios es la tabla de elaboración y esto demuestra que no podemos embeber aquí las tablas.

|  |
| --- |
| Materias\_Primas |
| id\_mp <int> |
| nombre <string> |
| tipo <string> |
| lote <string> |
| existencia <int> |
| fecha\_cad <date> |
| precio <int> |

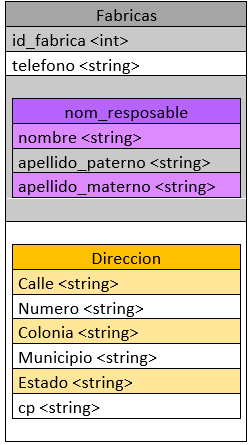
|  |
| --- |
| Elaboración |
| id\_elaboracion <int> |
| id\_fabrica <int> |
| id\_trabajador <int> |
| id\_mp <int> |
| id\_dulce <int> |
| cantidad <int> |

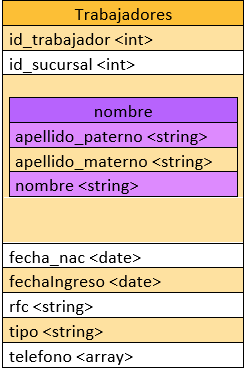






Además de que podemos ver que dos tablas tienen embebida unos campos, esto lo hicimos porque por ejemplo en los campos que son de nombre es necesario poder buscar por apellidos o por nombre y de esta manera podemos encontrarlos de una manera más eficiente, además de que es un campo que es necesario que este embebido ya que podemos obtener este bloque de información más rápido que si fuera por separado o en un solo campo. Por otro lado, podemos ver el dato embebido de “Dirección” que este nos ayuda a conocer donde tenemos las fábricas, este lo pusimos de igual manera embebido, porque consideramos que los datos de dirección en este aspecto de fabricas o de establecimientos no se consideran muy susceptibles a cambios y esto nos ayuda a una mejor integridad de los datos.





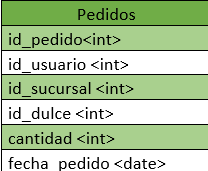
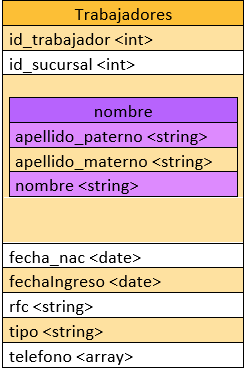
## Sucursales-trabajadores-registros-pedidos-repartos

En esta sección de la misma manera que en las anteriores podemos observar que las relaciones las hicimos de manera relacional, esto creando una mejor optimización y eficiencia en la actualización o incluso en el grabado de datos.

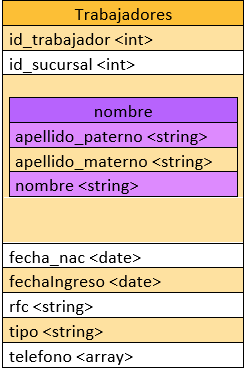


|  |
| --- |
| Sucursales |
| id\_sucursal <int> |
| |  | | --- | | Domicilio | | calle <string> | | numero <string> | | colonia <string> | | municipio <string> | | estado <string> | | cp <string> | |
| telefono <array> |
| |  | | --- | | nombre\_responsable | | apellido\_paterno <string> | | apellido\_materno <string> | | nombre <string> | |

|  |
| --- |
| Registros |
| id\_registro <int> |
| id\_sucursal <int> |
| id\_dulce <int> |
| cantidad <int> |
| fecha\_recepcion <date> |
| precio\_uni <int> |



Por otro lado, podemos ver que tenemos dos tablas que tienen datos embebidos esto lo hicimos de esta manera, porque consideramos que esos datos son difícilmente de que se modifiquen en un corto periodo de tiempo o que estén en constante cambio.

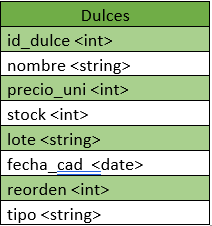


|  |
| --- |
| Sucursales |
| id\_sucursal <int> |
| |  | | --- | | Domicilio | | calle <string> | | numero <string> | | colonia <string> | | municipio <string> | | estado <string> | | cp <string> | |
| telefono <array> |
| |  | | --- | | nombre\_responsable | | apellido\_paterno <string> | | apellido\_materno <string> | | nombre <string> | |

## Pedidos-dulces-usuarios

Para estas tablas la hicimos relacional porque es mejor para los datos ya que estos pueden estar constante cambio, así que decidimos dejarlas relacionadas.

Además de que en la tabla usuarios embebimos los datos del nombre, esto lo realizamos así porque el nombre de uno de nuestros clientes no puede cambiar con frecuencia, aunque si puede cambiar por si hubo algún error en la inserción de sus datos, pero esto no puede ser muy común.

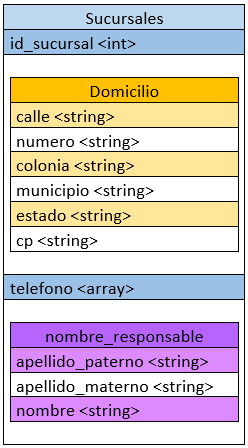


|  |
| --- |
| Pedidos |
| id\_pedido<int> |
| id\_usuario <int> |
| id\_sucursal <int> |
| id\_dulce <int> |
| cantidad <int> |
| fecha\_pedido <date> |

|  |
| --- |
| Usuarios |
| id\_usuario <int> |
| |  | | --- | | nombre | | apellido\_paterno <string> | | apellido\_materno <string> | | nombre <string> | |
| email <string> |
| telefono <array> |

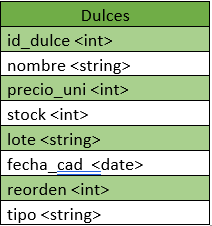
***Ventas-Dulces-Sucursales-Usuarios***

Para estas relaciones podemos observar que la tabla de ventas es un pivote o una tabla que guarda todas las claves de las demás tablas, esto ya que la tabla de ventas almacena todos los datos generados de una compra del usuario, esto generando una venta hacia nuestro sistema así que en esta tabla se prefirió guardar una estructura relacional ya que los datos están en constante movimiento y esto produce que si existe alguna corrección en los datos podamos modificarlos sin mayor problema.



|  |
| --- |
| Ventas |
| id\_venta <int> |
| id\_usuario <int> |
| id\_sucursal <int> |
| id\_dulce <int> |
| cantidad <int> |
| fecha <date> |
| precio\_uni <int> |
| descuento <int> |

|  |
| --- |
| Usuarios |
| id\_usuario <int> |
| |  | | --- | | nombre | | apellido\_paterno <string> | | apellido\_materno <string> | | nombre <string> | |
| email <string> |
| telefono <array> |



# Información en colecciones

## Compras

use("Dulceria")

//insertar datos

db.compras.insertMany(

    [

        {

            "id\_compra": 1,

            "id\_proveedor": 1,

            "id\_mp": 1,

            "fecha\_compra": new Date("2024-01-15"),

            "lote": "L1234",

            "cantidad": "500 kg",

            "precio": 10000,

            "descuento": 5

        },…..

## Dulces

use("Dulceria")

//insertar datos

db.dulces.insertMany(

    [

        {

            "id\_dulce": 1,

            "nombre": "Ate de Membrillo",

            "precio\_uni": 50.0000,

            "stock": 100,

            "lote": "Lote001",

            "fecha\_cad": new Date("2025-12-31"),

            "reorden": 20,

            "tipo": "CON AZUCAR"

        },…….

## Elaboración

use("Dulceria")

//insertar datos

db.elaboracion.insertMany(

    [

        {

            "id\_elaboracion": 1,

            "id\_fabrica": 1,

            "id\_trabajador": 1,

            "id\_mp": 1,

            "id\_dulce": 1,

            "cantidad": 200

        },

        {

            "id\_elaboracion": 2,

            "id\_fabrica": 1,

            "id\_trabajador": 2,

            "id\_mp": 2,

            "id\_dulce": 2,

            "cantidad": 150

        },…..

## Fabricas

use("Dulceria")

//insertar datos

db.fabricas.insertMany(

    [

        {

            "id\_fabrica": 1,

            "telefono": "5551234567",

            "nom\_responsable": "Juan Pérez",

            "domicilio": {

                "estado": "Ciudad de México",

                "municipio": "Iztapalapa",

                "colonia": "Col. Dulce",

                "domicilio": "Calle Caramelo 123",

                "cp": 9876

            }

        },

        {

            "id\_fabrica": 2,

            "telefono": "5557654321",

            "nom\_responsable": "María Gómez",

            "domicilio": {

                "estado": "Jalisco",

                "municipio": "Guadalajara",

                "colonia": "Col. Golosina",

                "domicilio": "Avenida Azúcar 456",

                "cp": 44100

            }

        },…..

## Materias Primas

use("Dulceria")

//insertar datos

db.materias\_primas.insertMany(

    [

        {

            "id\_mp": 1,

            "nombre": "Azúcar Refinada",

            "tipo": "Edulcorante",

            "lote": "L1234",

            "existencia": "1000 kg",

            "fecha\_cad": new Date("2025-12-31"),

            "precio": 20

        },

        {

            "id\_mp": 2,

            "nombre": "Cacao en Polvo",

            "tipo": "Saborizante",

            "lote": "L5678",

            "existencia": "500 kg",

            "fecha\_cad": new Date("2024-11-15"),

            "precio": 45

        },….

## Pedidos

use("Dulceria")

//insertar datos

db.pedidos.insertMany(

    [

        {

            "id\_pedido": 2,

            "id\_usuario": 1,

            "id\_sucursal": 1,

            "id\_dulce": 1,

            "cantidad": 5,

            "fecha\_pedido": new Date("2024-10-01")

        },

        {

            "id\_pedido": 3,

            "id\_usuario": 2,

            "id\_sucursal": 1,

            "id\_dulce": 2,

            "cantidad": 10,

            "fecha\_pedido": new Date("2024-10-02")

        },….

## Proveedores

use("Dulceria")

//insertar datos

db.Proveedores.insertMany(

    [

        {

            "id\_proveedor": 1,

            "nombre": "Azúcares del Valle",

            "rfc": "AVR123456789",

            "direccion": "Calle Caramelo 123, Ciudad Dulce",

            "telefono": [

                "555-123-4567"

            ]

        },

        {

            "id\_proveedor": 2,

            "nombre": "Esencias y Sabores S.A.",

            "rfc": "ESS234567890",

            "direccion": "Av. Vainilla 234, Colonia Sabrosa",

            "telefono": [

                "555-234-5678"

            ]

        },…..

## Registros

use("Dulceria")

//insertar datos

db.registros.insertMany(

    [

        {

            "id\_registro": 1,

            "id\_sucursal": 1,

            "id\_dulce": 1,

            "cantidad": 50,

            "fecha\_recepcion": new Date("2024-10-01"),

            "precio\_uni": 45.0000

        },

        {

            "id\_registro": 2,

            "id\_sucursal": 1,

            "id\_dulce": 2,

            "cantidad": 30,

            "fecha\_recepcion": new Date("2024-10-02"),

            "precio\_uni": 55.0000

        },……

## Repartos

use("Dulceria")

//insertar datos

db.repartos.insertMany(

    [

        {

            "id\_repartos": 2,

            "id\_fabrica": 1,

            "id\_sucursal": 1,

            "id\_trabajador": 1,

            "id\_dulce": 1,

            "cantidad": 100,

            "fecha\_reparto": new Date("2024-10-01")

        },

        {

            "id\_repartos": 3,

            "id\_fabrica": 1,

            "id\_sucursal": 2,

            "id\_trabajador": 2,

            "id\_dulce": 2,

            "cantidad": 150,

            "fecha\_reparto": new Date("2024-10-02")

        },……

## Sucursales

use("Dulceria")

//insertar datos

db.sucursales.insertMany(

    [

        {

            "id\_sucursal": 1,

            "nom\_responsable": "Pedro Sánchez",

            "telefono": [

                "5551234567"

            ],

            "direccion": {

                "estado": "Ciudad de México",

                "municipio": "Coyoacán",

                "colonia": "Col. Alegre",

                "domicilio": "Avenida Azucarera 101",

                "cp": 4360

            }

        },

        {

            "id\_sucursal": 2,

            "nom\_responsable": "María López",

            "telefono": [

                "5552345678"

            ],

            "direccion": {

                "estado": "Jalisco",

                "municipio": "Guadalajara",

                "colonia": "Col. Dulce",

                "domicilio": "Calle de la Chocolatería 202",

                "cp": 44100

            }

        },…..

## Trabajadores

use("Dulceria")

//insertar datos

db.trabajadores.insertMany(

    [

        {

            "id\_trabajador": 1,

            "id\_sucursal": 1,

            "nombre": {

                "apellido\_paterno": "Sánchez",

                "apellido\_materno": "Gómez",

                "nombres": "Pedro"

            },

            "fecha\_nacimiento": "1990-05-15",

            "fecha\_ingreso": "2022-01-10",

            "rfc": "SAGP900515HDFMPR01",

            "tipo": "Permanente",

            "telefono": [

                "5551234567"

            ]

        },

        {

            "id\_trabajador": 2,

            "id\_sucursal": 1,

            "nombre": {

                "apellido\_paterno": "López",

                "apellido\_materno": "Martínez",

                "nombres": "María"

            },

            "fecha\_nacimiento": "1985-08-22",

            "fecha\_ingreso": "2021-03-15",

            "rfc": "LOMM850822HDFMTR02",

            "tipo": "Permanente",

            "telefono": [

                "5552345678"

            ]

        },…..

## Usuarios

use("Dulceria")

//insertar datos

db.usuarios.insertMany(

    [

        {

            "id\_usuario": 1,

            "nombre": {

                "apellido\_paterno": "Sánchez",

                "apellido\_materno": "Gómez",

                "nombres": "Pedro"

            },

            "email": "pedro.sanchez@example.com",

            "telefono": [

                "5551234567"

            ]

        },

        {

            "id\_usuario": 2,

            "nombre": {

                "apellido\_paterno": "López",

                "apellido\_materno": "Martínez",

                "nombres": "María"

            },

            "email": "maria.lopez@example.com",

            "telefono": [

                "5552345678"

            ]

        },……

## Ventas

use("Dulceria")

//insertar datos

db.ventas.insertMany(

    [

        {

            "id\_venta": 1,

            "id\_usuario": 1,

            "id\_sucursal": 1,

            "id\_dulce": 1,

            "cantidad": 2,

            "fecha": new Date("2024-10-01"),

            "precio\_uni": 50.0000,

            "descuento": 5

        },

        {

            "id\_venta": 2,

            "id\_usuario": 2,

            "id\_sucursal": 1,

            "id\_dulce": 2,

            "cantidad": 5,

            "fecha": new Date("2024-10-02"),

            "precio\_uni": 60.0000,

            "descuento": 10

        },…….

# Consultas generadoras de JSON inserción de datos

## Compras

SELECT  id\_compra,

      id\_proveedor,

      id\_mp,

      fecha\_compra,

      lote,

      cantidad,

      precio,

      descuento

  FROM compras

  FOR JSON PATH

## Dulces

SELECT id\_dulce,

      nombre,

      precio\_uni,

      stock,

      lote,

      fecha\_cad,

      reorden,

      tipo

  FROM dulces

  FOR JSON PATH

## Elaboración

SELECT id\_elaboracion,

      id\_fabrica,

      id\_trabajador,

      id\_mp,

      id\_dulce,

      cantidad

  FROM elaboracion

  FOR JSON PATH

## Fabricas

SELECT id\_fabrica

      ,telefono

      ,nom\_responsable

      ,estado as [domicilio.estado]

      ,municipio as [domicilio.municipio]

      ,colonia as [domicilio.colonia]

      ,domicilio as [domicilio.domicilio]

      ,cp as [domicilio.cp]

  FROM fabricas

FOR JSON PATH

## Materias Primas

SELECT id\_mp

      ,nombre

      ,tipo

      ,lote

      ,existencia

      ,fecha\_cad

      ,precio

  FROM materias\_primas

  FOR JSON PATH

## Pedidos

SELECT id\_pedido

      ,id\_usuario

      ,id\_sucursal

      ,id\_dulce

      ,cantidad

      ,fecha\_pedido

  FROM pedidos

  FOR JSON PATH

## Proveedores

SELECT  id\_proveedor

      ,nombre

      ,rfc

      ,direccion

      ,JSON\_ARRAY(telefono) AS telefono

  FROM proveedores

  FOR JSON PATH

## Registros

SELECT id\_registro

      ,id\_sucursal

      ,id\_dulce

      ,cantidad

      ,fecha\_recepcion

      ,precio\_uni

  FROM registros

  FOR JSON PATH

## Repartos

SELECT  id\_repartos

      ,id\_fabrica

      ,id\_sucursal

      ,id\_trabajador

      ,id\_dulce

      ,cantidad

      ,fecha\_reparto

  FROM repartos

  FOR JSON PATH

## Sucursales

SELECT  id\_sucursal

      ,nom\_responsable

      ,JSON\_ARRAY(telefono) as telefono

      ,estado as [direccion.estado]

      ,municipio as [direccion.municipio]

      ,colonia as [direccion.colonia]

      ,domicilio as [direccion.domicilio]

      ,cp as [direccion.cp]

  FROM sucursales

  for JSON PATH

## Trabajadores

SELECT id\_trabajador

,id\_sucursal

,apellido\_paterno as [nombre.apellido\_paterno]

,apellido\_materno as [nombre.apellido\_materno]

,nombres as [nombre.nombres]

,fecha\_nacimiento

,fecha\_ingreso

,rfc

,tipo

,JSON\_ARRAY(telefono) as telefono

FROM trabajadores

for JSON PATH

## Usuarios

SELECT id\_usuario

      ,apellido\_paterno as [nombre.apellido\_paterno]

      ,apellido\_materno as [nombre.apellido\_materno]

      ,nombres as [nombre.nombres]

      ,email

      ,JSON\_ARRAY(telefono) as telefono

  FROM usuarios

  FOR JSON PATH

## Ventas

SELECT id\_venta

      ,id\_usuario

      ,id\_sucursal

      ,id\_dulce

      ,cantidad

      ,fecha

      ,precio\_uni

      ,descuento

  FROM ventas

  FOR JSON PATH