

“**ALMA DE CAFÉ**”

PROYECTO DE TALLER DE BASE DE DATOS

**INGENIERÍA EN SISTEMAS COMPUTACIONALES**

PRESENTA:

**ALONDRA ALEJANDRA LÓPEZ MEDELLIN**

**DANIEL ORLANDO CEJA OLLOQUI**

**MARTÍNEZ GONZÁLEZ LEONARDO**

JIQUILPAN, MICHOACÁN, 5 DE NOVIEMBRE DE 2024

Contenido

[Validación de Datos 3](#_Toc181721164)

[Bebidas 3](#_Toc181721165)

[Empleados 3](#_Toc181721166)

[Ingredientes 3](#_Toc181721167)

[Postres 3](#_Toc181721168)

[Recursos 4](#_Toc181721169)

[Pedidos 5](#_Toc181721170)

[Puestos 6](#_Toc181721171)

[Entregas 6](#_Toc181721172)

[Proveedores 7](#_Toc181721173)

[Mesas 8](#_Toc181721174)

[Accesos 8](#_Toc181721175)

[Tipos\_De\_Accesos 9](#_Toc181721176)

[Ventas 9](#_Toc181721177)

[Clientes 10](#_Toc181721178)

[Modelo de Datos 10](#_Toc181721179)

[Informacion en Colecciones 10](#_Toc181721180)

[Script de Validación y Script para el llenado de las colecciones 10](#_Toc181721181)

# Validación de Datos

## Bebidas

|  |  |  |
| --- | --- | --- |
| **Dato** | **Tipo** | **Descripción** |
| id\_bebida | int | ID de la bebida |
| nombre\_bebida | string | Nombre de la bebida |
| tipo\_bebida | string | Tipo de bebida (CALIENTE o FRÍA) |
| precio\_bebida | double | Precio de la bebida |
| descripcion\_bebida | string | Descripción de la bebida |
| forma\_de\_preparacion | string | Forma de preparación de la bebida |
| **ingredientes** | **array** | **Lista de ingredientes** |
| id\_ingrediente | int | ID del ingrediente dentro de 'ingredientes' |
| nombre | string | Nombre del ingrediente |
| sabor | string | Sabor del ingrediente |
| cantidad | int | Cantidad de ingrediente requerido |
| especificaciones | string | Especificaciones del ingrediente para la bebida |

## Empleados

|  |  |  |
| --- | --- | --- |
| **Dato** | **Tipo** | **Descripción** |
| id\_empleado | int | ID único del empleado |
| nombre\_empleado | string | Nombre del empleado |
| apellido\_paterno\_emp | string | Apellido paterno del empleado |
| apellido\_materno\_emp | string | Apellido materno del empleado |
| puesto | object | Puesto del empleado |
| id\_puesto | int | ID único del puesto dentro de 'puesto' |
| nombre\_puesto | string | Nombre del puesto |

## Ingredientes

|  |  |  |
| --- | --- | --- |
| **Dato** | **Tipo** | **Descripción** |
| id\_ingrediente | int | ID del ingrediente |
| recurso | object | Datos del recurso asociado al ingrediente |
| id\_recurso | int | ID del recurso dentro de 'recurso' |
| cantidad\_existente | int | Cantidad existente del recurso |
| fecha\_caducidad | date | Fecha de caducidad del recurso |
| tipo\_ingrediente | string | Tipo de ingrediente (LIQUIDO, POLVOS, SOLIDO) |
| sabor | string | Sabor del ingrediente (AMARGO, DULCE, SALADO, AGRIO) |

## Postres

|  |  |  |
| --- | --- | --- |
| **Dato** | **Tipo** | **Descripción** |
| id\_postre | int | ID del postre |
| precio\_postre | decimal | Precio del postre |
| descripcion\_postre | string | Descripción del postre |
| recurso | object | Datos del recurso asociado al postre |
| id\_recurso | int | ID del recurso dentro de 'recurso' |
| cantidad\_existente | int | Cantidad existente del recurso |
| fecha\_caducidad | date | Fecha de caducidad del recurso |
| margen\_de\_ganancia | decimal | Margen de ganancia del postre |

## Recursos

|  |  |  |
| --- | --- | --- |
| **Dato** | **Tipo** | **Descripción** |
| id\_recurso | int | ID del recurso |
| nombre\_recurso | string | Nombre del recurso |
| tipo\_recurso | string | Tipo de recurso (POSTRE o INGREDIENTE) |
| cantidad\_existente | int | Cantidad existente del recurso, debe ser mayor a 0 |
| descripcion\_recurso | string | Descripción del recurso |
| costo\_recurso | string | Costo del recurso |
| fecha\_caducidad | date | Fecha de caducidad del recurso |
| lugar\_se\_encuentra | string | Lugar donde se encuentra el recurso (REFRIGERADOR, BODEGA, REPISA, CAJONES) |

## Pedidos

|  |  |  |
| --- | --- | --- |
| **Dato** | **Tipo** | **Descripción** |
| id\_pedido | int | ID del pedido |
| total | decimal | Total del pedido |
| subtotal | decimal | Subtotal del pedido |
| dia\_pedido | int | Día del pedido (1-31) |
| mes\_pedido | int | Mes del pedido (1-12) |
| anio\_pedido | int | Año del pedido, entre 2022 y el año actual |
| estatus | string | Estatus del pedido (ACTIVO, PROCESO, TERMINADO) |
| empleado | object | Información del empleado |
| id\_empleado | int | ID del empleado |
| nombre | string | Nombre del empleado |
| puesto | string | Puesto del empleado |
| mesa | object | Información de la mesa |
| id\_mesa | int | ID de la mesa |
| capacidad | int | Capacidad de la mesa |
| **postres** | **array** | **Lista de postres en el pedido** |
| id\_postre | int | ID del postre |
| nombre\_postre | string | Nombre del postre |
| cantidad | int | Cantidad de postres |
| precio | decimal | Precio del postre |
| **bebidas** | **array** | **Lista de bebidas en el pedido** |
| id\_bebida | int | ID de la bebida |
| nombre\_bebida | string | Nombre de la bebida |
| cantidad | int | Cantidad de bebidas |
| precio | decimal | Precio de la bebida |

## Puestos

|  |  |  |
| --- | --- | --- |
| **Dato** | **Tipo** | **Descripción** |
| id\_puesto | int | ID único del puesto |
| nombre\_puesto | string | Nombre del puesto |
| descripcion | string | Descripción del puesto |
| requisitos | string | Requisitos del puesto |
| jornada\_laboral | string | Horario laboral del puesto en formato de 12 horas (ej. 09:00AM - 05:00PM) |
| Salario | decimal | Salario del puesto |

## Entregas

|  |  |  |
| --- | --- | --- |
| **Campo** | **Tipo** | **Descripción** |
| id\_entrega | int | ID único de la entrega |
| dia\_entrega | int | Día de la entrega (1-31) |
| mes\_entrega | int | Mes de la entrega (1-12) |
| anio\_entrega | int | Año de la entrega (2022-2100) |
| fecha\_caducidad | date | Fecha de caducidad |
| Hora | string | Hora de la entrega |
| cantidad\_entregada | int | Cantidad entregada, debe ser mayor a 0 |
| Costo | decimal | Costo de la entrega, con un valor mínimo de 0 |
| proveedor | object | con datos del proveedor |
| id\_proveedor | int | ID del proveedor |
| nombre\_prov | string | Nombre del proveedor |
| **recurso** | **arrays** | **Lista de objetos con información de los recursos** |
| id\_recurso | int | ID del recurso |
| nombre\_recurso | string | Nombre del recurso |

## Proveedores

|  |  |  |
| --- | --- | --- |
| **Dato** | **Tipo** | **Descripción** |
| id\_proveedor | int | ID único del proveedor |
| nombre\_prov | string | Nombre del proveedor |
| apellido\_paterno\_prov | string | Apellido paterno del proveedor |
| apellido\_materno\_prov | string | Apellido materno del proveedor |
| direccion\_prov | string | Dirección del proveedor |
| telefono\_prov | string | Teléfono del proveedor |
| correo\_prov | string | Correo electrónico del proveedor |
| empresa | string | Empresa del proveedor |
| horario | string | Horario laboral del proveedor en formato HH  AM/PM |

## Mesas

|  |  |  |
| --- | --- | --- |
| **Dato** | **Tipo** | **Descripción** |
| id\_mesa | int | ID único de la mesa |
| lugar | string | Ubicación de la mesa (valores posibles: "PATIO EXTERIOR", "PATIO INTERIOR", "INTERIOR", "LLEVAR") |
| capacidad | int | Capacidad de la mesa |

## Accesos

|  |  |  |
| --- | --- | --- |
| **Dato** | **Tipo** | **Descripción** |
| id\_empleado\_acceso | int | ID único de acceso del empleado |
| empleado | objeto | Información del empleado que accede |
| id\_empleado | int | ID del empleado |
| nombre\_empleado | string | Nombre del empleado |
| puesto | string | Puesto del empleado |
| tipo\_acceso | objeto | Información del tipo de acceso |
| id\_acceso | int | ID del tipo de acceso |
| descripcion | string | Descripción del tipo de acceso |
| fecha | date | Fecha del acceso |
| hora | string | Hora del acceso (formato HH:MM) |

## Tipos\_De\_Accesos

|  |  |  |
| --- | --- | --- |
| **Dato** | **Tipo** | **Descripción** |
| **id\_acceso** | int | ID único del tipo de acceso |
| **descripcion** | string | Descripción del tipo de acceso (de 1 a 100 caracteres) |
| **nivel\_acceso** | string | Nivel de acceso del tipo de acceso (puede ser: BÁSICO, INTERMEDIO, AVANZADO, ADMINISTRADOR) |

## Ventas

|  |  |  |
| --- | --- | --- |
| **Dato** | **Tipo** | **Descripción** |
| id\_venta | int | ID único de la venta |
| metodo\_pago | string | Método de pago (EFECTIVO, TARJETA, CREDITO) |
| cliente | objeto | Información del cliente que realiza la compra |
| id\_cliente | int | ID del cliente |
| nombre\_cliente | string | Nombre del cliente |
| pedido | objeto | Información del pedido realizado |
| id\_pedido | int | ID del pedido |
| total | double | Total del pedido (mínimo 0) |
| subtotal | double | Subtotal del pedido (mínimo 0) |

## Clientes

|  |  |  |
| --- | --- | --- |
| **Dato** | **Tipo** | **Descripción** |
| id\_cliente | int | ID único del cliente |
| nombre\_cliente | string | Nombre del cliente, por defecto 'ANONIMO' (de 1 a 50 caracteres) |
| apellidos\_cliente | string | Apellidos completos del cliente (máximo 70 caracteres) |
| nacionalidad | string | Nacionalidad del cliente (máximo 30 caracteres) |

# 

# Modelo de Datos

bebidas

ingredientes

Array

empleados

puesto

Objeto Anidado

ingredientes

recurso

Objeto Anidado

Postres

recurso

Objeto Anidado



entregas

proveedor

Objeto Anidado

recursos

Array







Ventas

cliente

Objeto Anidado

pedido

Objeto Anidado

Accesos

empleado

Objeto Anidado

Tipo\_acceso

Objeto Anidado

postres

Array

bebidas

Array

empleados

Objeto Anidado

mesa

Objeto Anidado

pedidos



## Justificación

Bebidas

En nuestro modelo relacional de SQL se tenía una relación n-n con ingredientes, así que se decido optar por embebido en la colección para realizar las consultas más rápidas, y así los empleados al ver una bebida inmediatamente puedas observar que ingredientes requieres.

Empleados

En nuestro modelo relacional de SQL se tenía una relación 1-1 con la tabla puestos, pero decidimos realizar embebido porque los administradores deben conocen en todo momento que puesto desarrolla el empleado. De la misma forma requiere saber todos los puestos que existes. Es por esto que tenemos como objeto anidado puesto y además la tabla Puestos.

Ingredientes

En nuestro modelo relacional se tenía una relación 1-1 con la tabla recursos, se decido implementar embebido debido a que muchos campos que se poseen en recurso son necesarios tenerlos al consultar ingredientes (como por ejemplo la caducidad, o la cantidad en existencia)

Postres

Misma justificación que ingredientes

Entregas

En nuestro modelo relacional se tenía una relación n-n con la tabla recursos, realizamos embebidos porque consideramos mas factible a la hora de hacer las consultar al buscar una entrega tener ya que recursos abastecimos. Además de la relación 1-1 con la tabla proveedores como es una entrega sus valores no cambian así que el proveedor que deje esa entrega será ese para esa entrega y punto, es por esto que se coloco un objeto anidado proveedor.

Pedidos

En esta tabla se tenían muchas relaciones por lo que la desglosaremos una por una

Pedidos – empleado

Esta relación en SQL server es 1 a 1 pero son muchas las consultas que se hacen en el día sobre un pedido así que hacerla relacional no era una opción tan factible como embebida

Pedidos – mesa

Misma razón que la anterior

Pedidos – Bebidas

Esta relación en SQL server es de n a n, por lo que hacer una consulta era muy larga al tener la tabla intermedia, y al hacerlo embebido se soluciona ya que al tratarse de un array se observa muy claro cuales son las bebidas que tuvo ese pedido.

Pedidos – Postres

Misma razón que la anterior

Accesos

En la tabla accesos tenía relación n-n con empleados por lo que, en la tabla intermedia en SQL, se convierte en la principal para convertirla en colección debido a que tenía muchos campos importantes, campos que al hacer embebido con tipos\_de\_acceso (tabla accesos en SQL) y empleados se obtienen de manera más ágil.

Ventas

En SQL se tenía una relación 1-1 con las tablas pedidos y clientes, por lo que decidimos hacer embebido para tener los campos necesarios de una vez en ventas, en vez de solo poner el id de pedidos y clientes

# Información en Colecciones

# Script de Validación y Script para el llenado de las colecciones