



Tecnicatura Universitaria en Programación Base de Datos II

Base de Datos para Gestión de Reseñas Gastronómicas

Equipo # 34

Integrantes:

**Parlangeli, Marcos – Legajo 31191
Della Savia, Natali - Legajo 24960
Perosa Croses, Lisandro - Legajo 29859
Salas, Gabriel Edgardo - Legajo 31190**

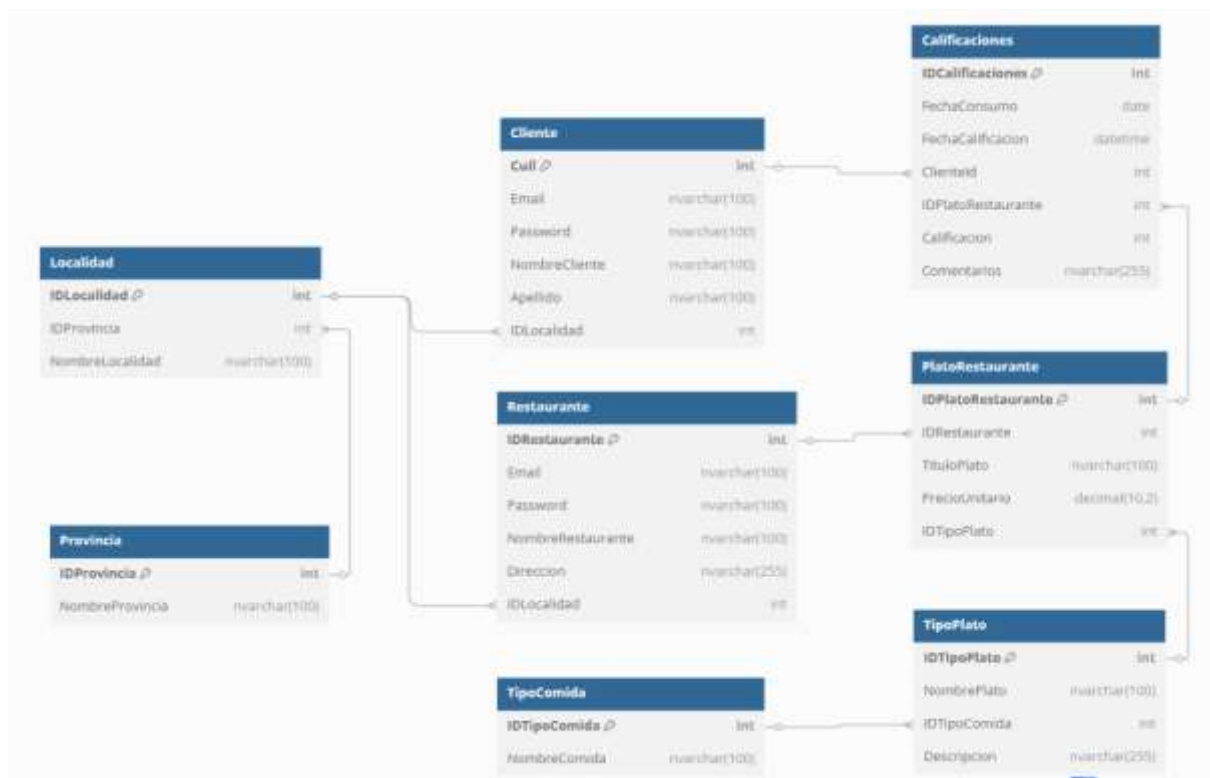
Explicación del Sistema

En este trabajo se ha desarrollado un sistema para una aplicación de reseñas de platos de restaurantes. La base de datos permite almacenar la información de los usuarios clientes y restaurantes, donde los clientes pueden navegar entre distintas opciones gastronómicas según sus preferencias, calificar y comentar sobre los platos consumidos, y los restaurantes pueden gestionar su oferta mostrando sus platos, revisando sus reseñas y analizando cuál es su plato más destacado o aquellos que presentan oportunidades de mejora.

Entre las funcionalidades principales tenemos:

- Listar restaurantes ubicados en la misma localidad que el usuario cliente.
- Permitir a los restaurantes conocer cuáles de sus platos tienen mejores calificaciones.
- Facilitar la consulta de las reseñas y el análisis de tendencias, gracias a la utilización de vistas.
- Garantizar la integridad de los datos y automatizar procesos críticos mediante procedimientos almacenados y triggers (como por ejemplo, prevenir inconsistencias en las fechas de reseñas o transformar la eliminación física en una baja lógica).

Diagrama de Entidad Relación



Código

La sección de Código abarca los aspectos fundamentales para el desarrollo de la base de datos, incluyendo la creación de la base de datos y sus tablas, así como la definición de relaciones entre ellas. También se detallan los procesos de carga de datos, las consultas SQL más relevantes, el uso de triggers para automatizar acciones y la implementación de procedimientos almacenados para optimizar operaciones repetitivas, todo orientado a la eficiencia y funcionalidad del sistema.

Creación de Tablas

```

Create Database TPFinal
[ Collate Latin1_General_CI_AI
Go
Use TPFinal
Go

Create Table Provincia(
    IDProvincia int not null primary key identity(1,1),
    NombreProvincia nvarchar(100) not null
)
Go

Create Table Localidad(
    IDLocalidad int not null primary key identity(1,1),
    IDProvincia int not null foreign key references Provincia(IDProvincia),
    NombreLocalidad nvarchar(100) not null
)
Go

Create Table Cliente(
    Cuil int not null primary key,
    Email nvarchar(100) not null,
    Password nvarchar(100) not null,
    NombreCliente nvarchar(100) not null,
    ApellidoCliente nvarchar(100) not null,
    IDLocalidad int not null foreign key references Localidad(IDLocalidad)
)
Go

Create Table Restaurante(
    IDRestaurante int not null primary key identity(1,1),
    Email nvarchar(100) not null,
    Password nvarchar(100) not null,
    NombreRestaurante nvarchar(100) not null,
    Direccion nvarchar(255) not null,
    IDLocalidad int not null foreign key references Localidad(IDLocalidad)
)
Go

Create Table TipoComida(
    IDTipoComida int not null primary key identity(1,1),
    NombreComida nvarchar(100) not null
)
Go
```

```

Create Table TipoPlato(
    IDTipoPlato int not null primary key identity(1,1),
    NombrePlato nvarchar(100) not null,
    IDTipoComida int not null foreign key references TipoComida(IDTipoComida),
    Descripcion nvarchar(255)
)
Go

Create Table PlatoRestaurante(
    IDPlatoRestaurante int not null primary key identity(1,1),
    IDRestaurante int not null foreign key references Restaurante(IDRestaurante),
    TituloPlato nvarchar(100) not null,
    PrecioUnitario decimal(10,2) not null,
    IDTipoPlato int not null foreign key references TipoPlato(IDTipoPlato),
    FechaBaja datetime NULL
)
Go

Create Table Calificaciones(
    IDCalificaciones int not null primary key identity(1,1),
    FechaConsumo date not null,
    FechaCalificacion datetime not null,
    ClienteId int not null foreign key references Cliente(Cuil),
    IDPlatoRestaurante int not null foreign key references PlatoRestaurante(IDPlatoRestaurante),
    Calificacion int not null check( Calificacion between 1 and 5 ),
    Comentarios nvarchar(255)
)
Go

```

Carga de Datos

```
-- INSERTS --
-- Agregar Provincias
INSERT INTO Provincia (NombreProvincia)
VALUES
    ('Buenos Aires'),
    ('Catamarca'),
    ('Chaco'),
    ('Chubut'),
    ('Cordoba'),
    ('Corrientes'),
    ('Entre Ríos'),
    ('Formosa'),
    ('Jujuy'),
    ('La Pampa'),
    ('La Rioja'),
    ('Mendoza'),
    ('Misiones'),
    ('Neuquen'),
    ('Rio Negro'),
    ('Salta'),
    ('San Juan'),
    ('San Luis'),
    ('Santa Cruz'),
    ('Santa Fe'),
    ('Santiago del Estero'),
    ('Tierra del Fuego'),
    ('Tucuman');
GO

-- Agregue 5 localidades por provincia
-- Provincia 1: Buenos Aires
INSERT INTO Localidad (IDProvincia, NombreLocalidad)
VALUES
    (1, 'La Plata'),
    (1, 'Mar del Plata'),
    (1, 'Tandil'),
    (1, 'Bahia Blanca'),
    (1, 'Tigre');
GO
```

```

-- Provincia 2: Catamarca
INSERT INTO Localidad (IDProvincia, NombreLocalidad)
VALUES
    (2, 'San Fernando del Valle de Catamarca'),
    (2, 'Fray Mamerto Esquiú'),
    (2, 'Belen'),
    (2, 'Santa María'),
    (2, 'Hualfín');
GO

-- Provincia 3: Chaco
INSERT INTO Localidad (IDProvincia, NombreLocalidad)
VALUES
    (3, 'Resistencia'),
    (3, 'Barranqueras'),
    (3, 'Fontana'),
    (3, 'Villa Ángela'),
    (3, 'Charata');
GO

-- Provincia 4: Chubut
INSERT INTO Localidad (IDProvincia, NombreLocalidad)
VALUES
    (4, 'Rawson'),
    (4, 'Trelew'),
    (4, 'Comodoro Rivadavia'),
    (4, 'Puerto Madryn'),
    (4, 'Sarmiento');
GO

-- Provincia 5: Córdoba
INSERT INTO Localidad (IDProvincia, NombreLocalidad)
VALUES
    (5, 'Córdoba'),
    (5, 'Villa María'),
    (5, 'Villa Carlos Paz'),
    (5, 'Río Cuarto'),
    (5, 'Bell Ville');
GO

-- Provincia 6: Corrientes
INSERT INTO Localidad (IDProvincia, NombreLocalidad)
VALUES
    (6, 'Corrientes'),
    (6, 'Goya'),
    (6, 'Paso de los Libres'),
    (6, 'Mercedes'),
    (6, 'Bella Vista');
GO

```

```

-- Provincia 7: Entre Ríos
INSERT INTO Localidad (IDProvincia, NombreLocalidad)
VALUES
    (7, 'Parana'),
    (7, 'Concordia'),
    (7, 'Gualeguaychu'),
    (7, 'Gualeguay'),
    (7, 'Colon');
GO

-- Provincia 8: Formosa
INSERT INTO Localidad (IDProvincia, NombreLocalidad)
VALUES
    (8, 'Formosa'),
    (8, 'Clorinda'),
    (8, 'Pirane'),
    (8, 'Las Lomitas'),
    (8, 'Laguna Blanca');
GO

-- Provincia 9: Jujuy
INSERT INTO Localidad (IDProvincia, NombreLocalidad)
VALUES
    (9, 'San Salvador de Jujuy'),
    (9, 'Palpala'),
    (9, 'Perico'),
    (9, 'El Carmen'),
    (9, 'San Francisco');
GO

-- Provincia 10: La Pampa
INSERT INTO Localidad (IDProvincia, NombreLocalidad)
VALUES
    (10, 'Santa Rosa'),
    (10, 'General Pico'),
    (10, 'Realico'),
    (10, 'Rancul'),
    (10, 'Catrilo');
GO

-- Provincia 11: La Rioja
INSERT INTO Localidad (IDProvincia, NombreLocalidad)
VALUES
    (11, 'La Rioja'),
    (11, 'Chilecito'),
    (11, 'Villa Union'),
    (11, 'Famatina'),
    (11, 'Sanagasta');
GO

```



```

-- Provincia 12: Mendoza
INSERT INTO Localidad (IDProvincia, NombreLocalidad)
VALUES
    (12, 'Mendoza'),
    (12, 'San Rafael'),
    (12, 'Godoy Cruz'),
    (12, 'Lujan de Cuyo'),
    (12, 'Maipu');
GO

-- Provincia 13: Misiones
INSERT INTO Localidad (IDProvincia, NombreLocalidad)
VALUES
    (13, 'Posadas'),
    (13, 'Obera'),
    (13, 'Eldorado'),
    (13, 'Iguazu'),
    (13, 'Apostoles');
GO

-- Provincia 14: Neuquen
INSERT INTO Localidad (IDProvincia, NombreLocalidad)
VALUES
    (14, 'Neuquen'),
    (14, 'San Martin de los Andes'),
    (14, 'Villa La Angostura'),
    (14, 'Zapala'),
    (14, 'Plottier');
GO

-- Provincia 15: Rio Negro
INSERT INTO Localidad (IDProvincia, NombreLocalidad)
VALUES
    (15, 'Viedma'),
    (15, 'San Carlos de Bariloche'),
    (15, 'General Roca'),
    (15, 'Cipolletti'),
    (15, 'Catriel');
GO

-- Provincia 16: Salta
INSERT INTO Localidad (IDProvincia, NombreLocalidad)
VALUES
    (16, 'Salta'),
    (16, 'San Lorenzo'),
    (16, 'Rosario de la Frontera'),
    (16, 'Cerrillos'),
    (16, 'Campo Quijano');
GO

```

```

-- Provincia 17: San Juan
INSERT INTO Localidad (IDProvincia, NombreLocalidad)
VALUES
    (17, 'San Juan'),
    (17, 'Caucete'),
    (17, 'Jachal'),
    (17, 'Zonda'),
    (17, 'Santa Lucia');
GO

-- Provincia 18: San Luis
INSERT INTO Localidad (IDProvincia, NombreLocalidad)
VALUES
    (18, 'San Luis'),
    (18, 'Villa Mercedes'),
    (18, 'La Punta'),
    (18, 'Merlo'),
    (18, 'Juana Koslay');
GO

-- Provincia 19: Santa Cruz
INSERT INTO Localidad (IDProvincia, NombreLocalidad)
VALUES
    (19, 'Rio Gallegos'),
    (19, 'El Calafate'),
    (19, 'Caleta Olivia'),
    (19, 'Puerto Deseado'),
    (19, 'Pico Truncado');
GO

-- Provincia 20: Santa Fe
INSERT INTO Localidad (IDProvincia, NombreLocalidad)
VALUES
    (20, 'Santa Fe'),
    (20, 'Rosario'),
    (20, 'Villa Constitución'),
    (20, 'Firmat'),
    (20, 'Rafaela');
GO

-- Provincia 21: Santiago del Estero
INSERT INTO Localidad (IDProvincia, NombreLocalidad)
VALUES
    (21, 'Santiago del Estero'),
    (21, 'La Banda'),
    (21, 'Termas de Río Hondo'),
    (21, 'Loreto'),
    (21, 'Bandera');
GO

```

```

-- Provincia 22: Tierra del Fuego
INSERT INTO Localidad (IDProvincia, NombreLocalidad)
VALUES
    (22, 'Ushuaia'),
    (22, 'Río Grande'),
    (22, 'Tolhuin'),
    (22, 'Porvenir'),
    (22, 'Rio San Juan');
GO

-- Provincia 23: Tucumán
INSERT INTO Localidad (IDProvincia, NombreLocalidad)
VALUES
    (23, 'San Miguel de Tucuman'),
    (23, 'Concepción'),
    (23, 'Yerba Buena'),
    (23, 'Tafi Viejo'),
    (23, 'Ranchillos');
GO

```

```

INSERT INTO TipoComida (NombreComida)
VALUES
    ('Parrilla'),
    ('Minutas'),
    ('Pastas'),
    ('Pizzas'),
    ('Empanadas'),
    ('Regional'),
    ('Vegetariana'),
    ('Mariscos'),
    ('Postres'),
    ('Sopas');

INSERT INTO TipoPlato (NombrePlato, IDTipoComida, Descripcion)
VALUES
    ('Empanadas', 2, 'Masa rellena de carne, pollo u otros ingredientes, horneada o frita.'),
    ('Milanesa', 2, 'Filete empanado de carne o pollo, frito o al horno.'),
    ('Asado', 7, 'Variedad de carnes asadas a la parrilla, típico de reuniones.'),
    ('Locro', 3, 'Guiso espeso de maíz, zapallo y carne, tradicional en fechas patrias.'),
    ('Humita', 10, 'Pasta de maíz con queso, cocida en chala al vapor.'),
    ('Tarta', 4, 'Masa rellena con verduras, jamón, queso o combinación, al horno.'),
    ('Pizza', 10, 'Masa con salsa, queso y diversos toppings, de estilo argentino.'),
    ('Provoleta', 10, 'Queso provolone fundido con especias a la parrilla.'),
    ('Pastel de papas', 6, 'Capas de puré de papa con carne picada y condimentos.'),
    ('Ñoquis', 8, 'Pasta de papa servida con salsas como tuco, crema o pesto.'),
    ('Choripán', 8, 'Pan con chorizo asado y salsa criolla o chimichurri.'),
    ('Puchero', 10, 'Estofado con carne, verduras y legumbres, cocido en agua.'),
    ('Matambre a la pizza', 8, 'Matambre cocido con salsa y queso como una pizza.'),
    ('Fainá', 4, 'Torta salada de harina de garbanzo, acompañante de la pizza.'),
    ('Ensalada rusa', 10, 'Ensalada fría de papa, zanahoria y arvejas con mayonesa.'),
    ('Tamales', 9, 'Bollos de maíz rellenos de carne, cocidos en hojas de maíz.'),
    ('Lomito', 10, 'Sándwich de carne, huevo, tomate y lechuga en pan francés.'),
    ('Ravioles', 1, 'Pasta rellena, servida con salsas variadas.'),
    ('Matambre arrollado', 6, 'Carne enrollada con verduras y huevo, cocida al horno.'),
    ('Flan con dulce de leche', 8, 'Postre de huevo y leche, acompañado con dulce de leche.');

```

```

INSERT INTO Cliente (Cuil, Email, Password, NombreCliente, ApellidoCliente, IDLocalidad)
VALUES
(201000008, 'marcosparla@hotmail.com', 'Hash1234', 'Marcos', 'Parlangeli', 28),
(201000018, 'lucas.rodriguez@mail.com', 'Hash1234', 'Lucas', 'Rodríguez', 33),
(201000028, 'sofia.fernandez@mail.com', 'Hash1234', 'Sofia', 'Fernández', 73),
(201000038, 'mateo.lopez@mail.com', 'Hash1234', 'Mateo', 'López', 1),
(201000048, 'valentina.martinez@mail.com', 'Hash1234', 'Valentina', 'Martinez', 17),
(201000058, 'thiago.gomez@mail.com', 'Hash1234', 'Thiago', 'Gómez', 78),
(201000068, 'camila.perez@mail.com', 'Hash1234', 'Camila', 'Pérez', 42),
(201000078, 'agustin.sanchez@mail.com', 'Hash1234', 'Agustin', 'Sánchez', 16),
(201000088, 'julietta.romero@mail.com', 'Hash1234', 'Julietta', 'Romero', 38),
(201000098, 'benjamin.torres@mail.com', 'Hash1234', 'Benjamin', 'Torres', 103),
(201000108, 'lautaro.alvarez@mail.com', 'Hash1234', 'Lautaro', 'Álvarez', 82),
(201000118, 'lucia.ruiz@mail.com', 'Hash1234', 'Lucía', 'Ruiz', 95),
(201000128, 'tomas.ramirez@mail.com', 'Hash1234', 'Tomás', 'Ramírez', 33),
(201000138, 'mia.flores@mail.com', 'Hash1234', 'Mía', 'Flores', 101),
(201000148, 'santino.acosta@mail.com', 'Hash1234', 'Santino', 'Acosta', 54),
(201000158, 'catalina.silva@mail.com', 'Hash1234', 'Catalina', 'Silva', 1),
(201000168, 'joaquin.molina@mail.com', 'Hash1234', 'Joaquín', 'Molina', 112),
(201000178, 'isabella.castro@mail.com', 'Hash1234', 'Isabella', 'Castro', 39),
(201000188, 'francisco.ortiz@mail.com', 'Hash1234', 'Francisco', 'Ortiz', 48),
(201000198, 'emma.nunez@mail.com', 'Hash1234', 'Emma', 'Núñez', 24),
(201000208, 'martin.gonzalez@mail.com', 'Hash1234', 'Martín', 'González', 38),
(201000218, 'lucas.rodriguez@mail.com', 'Hash1234', 'Lucas', 'Rodríguez', 95),
(201000228, 'sofia.fernandez@mail.com', 'Hash1234', 'Sofia', 'Fernández', 110),
(201000238, 'mateo.lopez@mail.com', 'Hash1234', 'Mateo', 'López', 4),
(201000248, 'valentina.martinez@mail.com', 'Hash1234', 'Valentina', 'Martinez', 64),
(201000258, 'thiago.gomez@mail.com', 'Hash1234', 'Thiago', 'Gómez', 63),
(201000268, 'camila.perez@mail.com', 'Hash1234', 'Camila', 'Pérez', 43),
(201000278, 'agustin.sanchez@mail.com', 'Hash1234', 'Agustin', 'Sánchez', 18),
(201000288, 'julietta.romero@mail.com', 'Hash1234', 'Julietta', 'Romero', 80),
(201000298, 'benjamin.torres@mail.com', 'Hash1234', 'Benjamin', 'Torres', 1),
(201000308, 'lautaro.alvarez@mail.com', 'Hash1234', 'Lautaro', 'Álvarez', 11),
(201000318, 'lucia.ruiz@mail.com', 'Hash1234', 'Lucía', 'Ruiz', 40),
(201000328, 'tomas.ramirez@mail.com', 'Hash1234', 'Tomás', 'Ramírez', 16),
(201000338, 'mia.flores@mail.com', 'Hash1234', 'Mía', 'Flores', 31),
(201000348, 'santino.acosta@mail.com', 'Hash1234', 'Santino', 'Acosta', 101),
(201000358, 'catalina.silva@mail.com', 'Hash1234', 'Catalina', 'Silva', 96),
(201000368, 'joaquin.molina@mail.com', 'Hash1234', 'Joaquín', 'Molina', 38),
(201000378, 'isabella.castro@mail.com', 'Hash1234', 'Isabella', 'Castro', 62),
(201000388, 'francisco.ortiz@mail.com', 'Hash1234', 'Francisco', 'Ortiz', 4),
(201000398, 'emma.nunez@mail.com', 'Hash1234', 'Emma', 'Núñez', 5),
(201000408, 'martin.gonzalez@mail.com', 'Hash1234', 'Martín', 'González', 46),
(201000418, 'lucas.rodriguez@mail.com', 'Hash1234', 'Lucas', 'Rodríguez', 57),
(201000428, 'sofia.fernandez@mail.com', 'Hash1234', 'Sofia', 'Fernández', 71),
(201000438, 'mateo.lopez@mail.com', 'Hash1234', 'Mateo', 'López', 55),
(201000448, 'valentina.martinez@mail.com', 'Hash1234', 'Valentina', 'Martinez', 108),
(201000458, 'thiago.gomez@mail.com', 'Hash1234', 'Thiago', 'Gómez', 86),
(201000468, 'camila.perez@mail.com', 'Hash1234', 'Camila', 'Pérez', 4),
(201000478, 'agustin.sanchez@mail.com', 'Hash1234', 'Agustin', 'Sánchez', 35),
(201000488, 'julietta.romero@mail.com', 'Hash1234', 'Julietta', 'Romero', 99),
(201000498, 'benjamin.torres@mail.com', 'Hash1234', 'Benjamin', 'Torres', 18),
(201000508, 'lautaro.alvarez@mail.com', 'Hash1234', 'Lautaro', 'Álvarez', 101),
(201000518, 'lucia.ruiz@mail.com', 'Hash1234', 'Lucía', 'Ruiz', 73),
(201000528, 'tomas.ramirez@mail.com', 'Hash1234', 'Tomás', 'Ramírez', 52),
(201000538, 'mia.flores@mail.com', 'Hash1234', 'Mía', 'Flores', 34),
(201000548, 'santino.acosta@mail.com', 'Hash1234', 'Santino', 'Acosta', 23),
(201000558, 'catalina.silva@mail.com', 'Hash1234', 'Catalina', 'Silva', 48),
(201000568, 'joaquin.molina@mail.com', 'Hash1234', 'Joaquín', 'Molina', 103),
(201000578, 'isabella.castro@mail.com', 'Hash1234', 'Isabella', 'Castro', 13),
(201000588, 'francisco.ortiz@mail.com', 'Hash1234', 'Francisco', 'Ortiz', 17),
(201000598, 'emma.nunez@mail.com', 'Hash1234', 'Emma', 'Núñez', 3);

```



```

INSERT INTO Restaurante (Email, Password, NombreRestaurante, Direccion, IDLocalidad)
VALUES
('contacto1@resto.com', 'ClaveResto5606', 'Parrilla El Gaucho', 'Calle Ayacucho 258', 44),
('contacto2@resto.com', 'ClaveResto4205', 'Sabor Criollo', 'Calle Castro Barros 417', 30),
('contacto3@resto.com', 'ClaveResto3593', 'La Esquina de Doña Tota', 'Av. 9 de Julio 581', 94),
('contacto4@resto.com', 'ClaveResto6336', 'La Fonda', 'Av. Entre Ríos 643', 31),
('contacto5@resto.com', 'ClaveResto5826', 'Don Julio', 'Av. Triunvirato 748', 98),
('contacto6@resto.com', 'ClaveResto8231', 'La Cabrera', 'Calle Larrea 769', 73),
('contacto7@resto.com', 'ClaveResto8384', 'El Buen Sabor', 'Calle Lavalle 910', 25),
('contacto8@resto.com', 'ClaveResto3013', 'Lo de Beto', 'Calle Pueyrredón 321', 83),
('contacto9@resto.com', 'ClaveResto6190', 'Patio Argentino', 'Av. Triunvirato 367', 69),
('contacto10@resto.com', 'ClaveResto7402', 'Ña Serapia', 'Calle French 579', 108),
('contacto11@resto.com', 'ClaveResto4529', 'El Rinconcito', 'Calle Sarmiento 964', 112),
('contacto12@resto.com', 'ClaveResto5828', 'Resto San Telmo', 'Calle Güemes 743', 96),
('contacto13@resto.com', 'ClaveResto8347', 'Cocina del Parque', 'Av. Callao 258', 23),
('contacto14@resto.com', 'ClaveResto4539', 'Sabor Norteño', 'Calle Alberdi 279', 11),
('contacto15@resto.com', 'ClaveResto2827', 'Mister Parrilla', 'Av. Santa Fe 151', 51),
('contacto16@resto.com', 'ClaveResto2611', 'La Picada', 'Calle Lavalle 598', 111),
('contacto17@resto.com', 'ClaveResto7534', 'Viejo Almacén', 'Calle Esmeralda 184', 98),
('contacto18@resto.com', 'ClaveResto4879', 'Fogón del Sur', 'Calle Castro Barros 137', 87),
('contacto19@resto.com', 'ClaveResto5861', 'Delicias Argentinas', 'Av. San Juan 129', 3),
('contacto20@resto.com', 'ClaveResto3306', 'Comedor Criollo', 'Calle Olazábal 848', 91),
('contacto21@resto.com', 'ClaveResto1905', 'Bife y Malbec', 'Av. Entre Ríos 333', 50),
('contacto22@resto.com', 'ClaveResto1755', 'Asado y Vino', 'Calle Bolívar 708', 32),
('contacto23@resto.com', 'ClaveResto8927', 'Sabores de mi Tierra', 'Calle Suipacha 569', 114),
('contacto24@resto.com', 'ClaveResto3256', 'Don Goyo', 'Calle Güemes 125', 40),
('contacto25@resto.com', 'ClaveResto9409', 'La Posta de los Tiempos', 'Calle Suipacha 661', 22),
('contacto26@resto.com', 'ClaveResto5690', 'Empanadas La Reina', 'Calle Sarmiento 133', 101),
('contacto27@resto.com', 'ClaveResto9791', 'Pan y Vino', 'Calle Güemes 816', 109),
('contacto28@resto.com', 'ClaveResto2982', 'Don Zoilo', 'Av. Corrientes 319', 58),
('contacto29@resto.com', 'ClaveResto7367', 'La Ruralita', 'Av. Córdoba 659', 112),
('contacto30@resto.com', 'ClaveResto4332', 'Lo de Cacho', 'Calle Lavalle 164', 76);

```

```

INSERT INTO PlatoRestaurante (IDRestaurante, TituloPlato, PrecioUnitario, IDTipoPlato)
VALUES
(15, 'Provoleta', 1567.72, 9),
(17, 'Roquis', 5894.1, 17),
(22, 'Polenta', 6944.03, 19),
(19, 'Milanesa', 3698.53, 12),
(22, 'Ravioles', 3482.72, 1),
(1, 'Pizza fugazzeta', 4164.82, 4),
(20, 'Panqueques', 6124.16, 3),
(19, 'Melado artesanal', 5614.1, 4),
(10, 'Choripán', 2865.59, 11),
(24, 'Polenta', 5408.12, 1),
(27, 'Humita', 4421.61, 18),
(1, 'Provoleta', 3844.76, 14),
(13, 'Choripán', 5994.05, 17),
(6, 'Humita', 6872.69, 6),
(7, 'Matambre a la pizza', 3240.43, 4),
(16, 'Panqueques', 4795.7, 9),
(9, 'Tortilla de papas', 4719.59, 11),
(6, 'Sopa Paraguaya', 3617.76, 10),
(24, 'Milanesa', 5599.47, 6),
(18, 'Locro', 2272.15, 7),
(13, 'Panqueques', 6599.46, 1),
(11, 'Ensalada Criolla', 1840.1, 5),
(5, 'Choripán', 6212.56, 4),
(23, 'Milanesa', 2056.27, 9),
(14, 'Choripán', 6799.02, 11),
(20, 'Matambre a la pizza', 2439.86, 4),
(8, 'Empanadas', 4537.07, 5),
(29, 'Provoleta', 6111.71, 13),
(26, 'Asado', 2796.91, 14),
(12, 'Provoleta', 4849.82, 2),
(13, 'Empanadas', 5911.36, 6),
(18, 'Milanesa', 3898.1, 12),
(21, 'Tortilla de papas', 6085.25, 13),
(13, 'Tarta de Verdura', 6230.44, 9),
(7, 'Polenta', 4871.77, 13),
(15, 'Choripán', 3461.51, 7),
(10, 'Guiso de lentejas', 6413.88, 19),
(6, 'Polenta', 5496.0, 4),
(14, 'Ravioles', 2047.14, 14),
(3, 'Flan', 1859.04, 2),
(11, 'Tortilla de papas', 2240.3, 15),
(16, 'Polenta', 1992.62, 6),
(22, 'Flan', 1903.2, 12),
(7, 'Ravioles', 3012.12, 6),
(21, 'Choripán', 5846.11, 18),
(23, 'Flan', 6339.97, 13),
(8, 'Panqueques', 4571.45, 12),
(30, 'Flan', 6286.18, 14),
(8, 'Matambre a la pizza', 1879.06, 4),
(22, 'Tortilla de papas', 2830.95, 8),
(10, 'Roquis', 5316.12, 5),
(4, 'Pizza fugazzeta', 4959.11, 20),
(3, 'Panqueques', 6207.51, 16),
(19, 'Humita', 1639.97, 6),
(13, 'Guiso de lentejas', 5324.1, 2),
(21, 'Choripán', 2661.15, 17),
(9, 'Ensalada Criolla', 1947.39, 3),
(17, 'Panqueques', 1945.75, 6),
(12, 'Guiso de lentejas', 6558.5, 5),
(15, 'Ravioles', 3857.18, 19);

```

```

INSERT INTO Calificaciones (FechaConsumo, FechaCalificacion, ClienteId, IDPlatoRestaurante, Calificacion, Comentarios)
VALUES
('2024-11-06', '2024-11-06 12:41:00', 201000008, 32, 1, 'Perfecto'),
('2024-01-21', '2024-01-21 15:30:00', 201000018, 14, 3, 'Riquísimo'),
('2024-11-09', '2024-11-09 14:53:00', 201000028, 37, 4, 'Zafa'),
('2024-03-16', '2024-03-16 11:07:00', 201000038, 35, 2, 'Falta sabor'),
('2024-03-14', '2024-03-14 13:44:00', 201000048, 5, 2, 'Excelente atención'),
('2024-12-14', '2024-12-14 19:45:00', 201000058, 21, 1, 'Volvería de nuevo'),
('2024-07-01', '2024-07-01 22:32:00', 201000068, 15, 5, 'La porción era chica'),
('2024-02-24', '2024-02-24 20:41:00', 201000078, 1, 2, 'Una bomba'),
('2024-09-19', '2024-09-19 16:30:00', 201000088, 14, 1, 'Bien servido'),
('2024-08-04', '2024-08-04 21:49:00', 201000098, 33, 1, 'La porción era chica'),
('2024-07-25', '2024-07-25 15:44:00', 201000108, 15, 1, 'Bien servido'),
('2024-12-18', '2024-12-18 11:03:00', 201000118, 20, 5, 'Precio justo'),
('2024-06-19', '2024-06-19 18:38:00', 201000128, 5, 5, 'Una bomba'),
('2024-03-20', '2024-03-20 11:58:00', 201000138, 17, 3, 'Zafa'),
('2024-10-11', '2024-10-11 17:57:00', 201000148, 24, 1, 'Volvería de nuevo'),
('2024-12-10', '2024-12-10 20:03:00', 201000158, 17, 3, 'Volvería de nuevo'),
('2024-06-19', '2024-06-19 11:07:00', 201000168, 38, 3, 'Volvería de nuevo'),
('2024-07-08', '2024-07-08 18:48:00', 201000178, 29, 5, 'La porción era chica'),
('2024-11-09', '2024-11-09 10:45:00', 201000188, 29, 1, 'Riquísimo'),
('2024-08-18', '2024-08-18 15:14:00', 201000198, 24, 4, 'Zafa'),
('2024-04-11', '2024-04-11 20:43:00', 201000208, 16, 3, 'Demasiado salado'),
('2024-05-13', '2024-05-13 21:34:00', 201000218, 18, 1, 'Muy sabroso'),
('2024-02-10', '2024-02-10 13:52:00', 201000228, 13, 2, 'Falta sabor'),
('2024-02-13', '2024-02-13 14:14:00', 201000238, 25, 1, 'Riquísimo'),
('2024-01-19', '2024-01-19 11:02:00', 201000248, 40, 3, 'Riquísimo'),
('2024-10-12', '2024-10-12 10:48:00', 201000258, 26, 5, 'Muy rico'),
('2024-04-18', '2024-04-18 19:34:00', 201000268, 7, 2, 'Volvería de nuevo'),
('2024-09-26', '2024-09-26 12:05:00', 201000278, 12, 2, 'Precio justo'),
('2024-08-23', '2024-08-23 19:12:00', 201000288, 12, 3, 'Zafa'),
('2024-08-18', '2024-08-18 14:11:00', 201000298, 9, 1, 'Riquísimo'),
('2024-05-13', '2024-05-13 12:53:00', 201000308, 24, 1, 'Riquísimo'),
('2024-05-05', '2024-05-05 10:44:00', 201000318, 7, 3, 'Muy sabroso'),
('2024-09-21', '2024-09-21 11:20:00', 201000328, 9, 1, 'Excelente atención'),
('2024-07-20', '2024-07-20 21:41:00', 201000338, 16, 3, 'No me gustó'),
('2024-09-05', '2024-09-05 20:13:00', 201000348, 1, 5, 'Precio justo'),
('2024-12-13', '2024-12-13 13:29:00', 201000358, 2, 5, 'La porción era chica'),
('2024-02-27', '2024-02-27 21:08:00', 201000368, 10, 5, 'Demasiado salado'),
('2024-08-13', '2024-08-13 15:40:00', 201000378, 3, 3, 'Muy sabroso'),
('2024-12-21', '2024-12-21 23:04:00', 201000388, 34, 2, 'No me gustó'),
('2024-10-22', '2024-10-22 21:16:00', 201000398, 31, 5, 'Una bomba'),

```



```
( '2024-08-02', '2024-08-02 20:39:00', 201000408, 18, 1, 'Falta sabor'),
( '2024-01-01', '2024-01-01 22:52:00', 201000418, 40, 5, 'Muy rico'),
( '2024-12-22', '2024-12-22 13:35:00', 201000428, 11, 5, 'Excelente atención'),
( '2024-10-28', '2024-10-28 18:22:00', 201000438, 21, 4, 'Volvería de nuevo'),
( '2024-06-22', '2024-06-22 20:54:00', 201000448, 36, 1, 'Una bomba'),
( '2024-10-04', '2024-10-04 10:28:00', 201000458, 17, 2, 'Volvería de nuevo'),
( '2024-12-08', '2024-12-08 17:10:00', 201000468, 11, 3, 'No me gustó'),
( '2024-05-04', '2024-05-04 10:53:00', 201000478, 38, 1, 'Muy rico'),
( '2024-09-19', '2024-09-19 12:41:00', 201000488, 8, 1, 'Bien servido'),
( '2024-04-04', '2024-04-04 15:14:00', 201000498, 5, 3, 'Bien servido'),
( '2024-10-24', '2024-10-24 17:51:00', 201000508, 1, 2, 'Zafa'),
( '2024-08-05', '2024-08-05 23:49:00', 201000518, 5, 4, 'Falta sabor'),
( '2024-02-24', '2024-02-24 10:09:00', 201000528, 38, 5, 'La porción era chica'),
( '2024-04-11', '2024-04-11 17:26:00', 201000538, 39, 4, 'Buena onda'),
( '2024-11-06', '2024-11-06 23:51:00', 201000548, 9, 1, 'Estaba frío'),
( '2024-05-04', '2024-05-04 19:24:00', 201000558, 19, 2, 'Buena onda'),
( '2024-07-04', '2024-07-04 23:27:00', 201000568, 24, 2, 'Falta sabor'),
( '2024-10-09', '2024-10-09 18:18:00', 201000578, 24, 4, 'Muy sabroso'),
( '2024-03-11', '2024-03-11 11:26:00', 201000588, 16, 5, 'Buena onda'),
( '2024-06-01', '2024-06-01 14:27:00', 201000598, 37, 3, 'Recomendado'),
( '2024-10-01', '2024-10-24 09:30:00', 201000008, 3, 5, 'Riquísimo'),
( '2024-09-10', '2024-09-10 12:41:00', 201000008, 3, 5, 'Excelete'),
( '2024-12-22', '2024-12-22 10:41:00', 201000008, 43, 5, 'Excelete'),
( '2024-05-05', '2024-05-05 12:41:00', 201000008, 5, 4, 'Zafable'),
( '2024-11-12', '2024-11-12 12:41:00', 201000008, 43, 3, 'Muy bueno'),
( '2024-12-12', '2024-12-12 12:41:00', 201000008, 50, 5, 'Perfecto');
```

Objetos de Base de Datos clave en el sistema

Vista 1:

La vista **View_MejorPlato** que muestra para cada restaurante, la siguiente información:
 IDRestaurante, IDPlatoRestaurante, Cantidad de calificaciones y Promedio de calificaciones
 Esta vista permite identificar el plato que más “suena” en función del volumen y la calidad de las reseñas, facilitando a los restaurantes conocer cuál de sus platos destaca dentro de la plataforma.


```

--(3) Se diseñará una vista que permita identificar el plato con la calificación promedio más alta;
--considerando únicamente aquellos platos que también tienen un volumen significativo de reseñas (el que "más suena").
--De esta manera, se garantiza que el resultado sea relevante tanto en calidad (calificación)
--como en popularidad (cantidad de opiniones recibidas).

CREATE VIEW View_MejorPlato AS
SELECT
    p.IDPlatoRestaurante,
    p.TituloPlato,
    count(t.IDTipoPlato) AS CantidadCalificacion,
    avg(c.Calificacion) AS PromedioCalificacion
FROM
    Calificaciones AS c
    inner join PlatoRestaurante AS p ON p.IDPlatoRestaurante = c.IDPlatoRestaurante
    inner join TipoPlato AS t ON t.IDTipoPlato = p.IDTipoPlato
WHERE
    p.IDRestaurante = 22
GROUP BY
    p.IDPlatoRestaurante,
    p.TituloPlato
GO

SELECT * FROM View_MejorPlato;

```

Vista 2:

Se utiliza la vista **View_PromedioPlatoPorRestaurante** que consolida, por cada restaurante y por tipo de plato, el promedio de calificaciones recibidas. La información mostrada incluye: IDRestaurante, NombreRestaurante, IDTipoPlato, NombrePlato y Promedio de las calificaciones.

Esta vista permite analizar y comparar la percepción de los clientes sobre las diferentes categorías del menú, dejando más en claro las fortalezas y áreas de mejora de cada restaurante.

```

-- (2) se creará una vista que calcule el promedio de calificación recibido por cada tipo de plato ofrecido en el restaurante.
--Esta información permitirá analizar la percepción de los clientes sobre las diferentes categorías del menú
--y detectar fortalezas o posibles áreas de mejora por tipo de plato.

CREATE VIEW View_PromedioPlatoPorRestaurante AS
SELECT
    t.IDTipoPlato,
    t.NombrePlato,
    AVG(c.Calificacion) AS Promedio
FROM
    Calificaciones AS c
    INNER JOIN PlatoRestaurante AS p ON p.IDPlatoRestaurante = c.IDPlatoRestaurante
    INNER JOIN TipoPlato AS t ON t.IDTipoPlato = p.IDTipoPlato
WHERE
    p.IDRestaurante = 22
GROUP BY
    t.IDTipoPlato,
    t.NombrePlato
GO

SELECT * FROM View_PromedioPlatoPorRestaurante

```

Vista 3

Se utiliza la vista **View_ClienteMasCalificaciones** que identifica al cliente con la mayor cantidad de calificaciones realizadas en el sistema. La vista presenta los siguientes datos: Cuil, NombreCliente, ApellidoCliente y Total de calificaciones.

Esta información es clave para reconocer a los usuarios más activos en la plataforma, lo que puede ayudar en el análisis del comportamiento del cliente.

```
--(1) Se generará una vista que permita identificar al cliente que ha aportado la mayor cantidad de calificaciones en el sistema.
--Esta información resulta valiosa para reconocer a los clientes más activos y comprometidos con el proceso de retroalimentación,
--quienes juegan un rol importante en la construcción de la reputación del restaurante.

CREATE VIEW View_ClienteMasCalificaciones AS
SELECT TOP 1
    c.Cuil,
    c.NombreCliente,
    c.ApellidoCliente,
    count(k.IDCalificaciones) AS TotalCalificaciones
FROM
    Cliente AS c
    inner join Calificaciones AS k ON c.Cuil = k.ClienteId
    inner join PlatoRestaurante AS p ON k.IDPlatoRestaurante = p.IDPlatoRestaurante
WHERE
    p.IDRestaurante = 22
GROUP BY
    c.Cuil,
    c.NombreCliente,
    c.ApellidoCliente
ORDER BY
    TotalCalificaciones DESC;
GO

SELECT * FROM View_ClienteMasCalificaciones
```

Procedimiento Almacenado 1

El procedimiento almacenado **SP_AgregarPlato** para ingresar un nuevo plato en la base de datos. Este procedimiento recibe los siguientes parámetros: IDRestaurante, TituloPlato, PrecioUnitario e IDTipoPlato.

Al ejecutarse, el procedimiento inserta el nuevo registro en la tabla PlatoRestaurante y muestra un mensaje de éxito, validando la operación de inserción del plato en el sistema.

```
Create Or Alter procedure SP_AgregarPlato
    @IDRestaurante int,
    @TituloPlato nvarchar(100),
    @PrecioUnitario decimal(10,2),
    @IDTipoPlato int
As
Begin
    Insert into PlatoRestaurante (IDRestaurante, TituloPlato, PrecioUnitario, IDTipoPlato)
    Values (@IDRestaurante, @TituloPlato, @PrecioUnitario, @IDTipoPlato);

    Print 'Plato insertado correctamente';
End
Go
```

Procedimiento Almacenado 2

Se utiliza el procedimiento almacenado **SP_ModificarDireccionIDLocalidad** para actualizar la dirección y la localidad de un restaurante. Este procedimiento recibe: IDRestaurante, Nueva Dirección y Nuevo IDLocalidad.

De esta manera, se facilita la gestión de la información de ubicación, permitiendo a los restaurantes mantener actualizada su dirección y localidad en la plataforma.

```
CREATE OR ALTER PROCEDURE SP_ModificarDireccionIDLocalidad
    @IDRestaurante int,
    @NuevaDireccion nvarchar(255),
    @NuevoIDLocalidad int
AS
BEGIN
    UPDATE Restaurante
    SET Direccion = @NuevaDireccion,
        IDLocalidad = @NuevoIDLocalidad
    WHERE IDRestaurante = @IDRestaurante;

    PRINT 'Direccion y localidad actualizadas correctamente';
END
GO
```

Procedimiento Almacenado 3

Se utiliza el procedimiento almacenado **SP_EliminarPlato** para eliminar un plato. Sin embargo, en conjunto con el trigger **BorrarPlato**, que se explicara más abajo, este procedimiento no borra físicamente el registro, sino que activa una baja lógica. Al ejecutarse, se actualiza el campo FechaBaja de la tabla PlatoRestaurante, preservando el historial y evitando la pérdida de información.

```
CREATE OR ALTER PROCEDURE SP_EliminarPlato
    @IDPlatoRestaurante int
AS
BEGIN
    DELETE FROM PlatoRestaurante
    WHERE IDPlatoRestaurante = @IDPlatoRestaurante;

    PRINT 'Plato eliminado correctamente';
END
GO
```

Trigger 1

El trigger **TR_Calificaciones_ValidarFechas**, el cual se ejecuta automáticamente tras un INSERT o UPDATE en la tabla Calificaciones. Este trigger verifica que el valor de FechaCalificacion no sea anterior a FechaConsumo.

Si se detecta que la fecha de la calificación es menor a la fecha del consumo, el trigger emite un mensaje de error y revierte la transacción mediante un rollback.

```
CREATE TRIGGER TR_Calificaciones_ValidarFechas
ON Calificaciones
after insert, update
AS
BEGIN
    if exists (SELECT null FROM INSERTED WHERE FechaCalificacion < FechaConsumo)
    BEGIN
        print ('Error: La FechaCalificacion no puede ser anterior a la FechaConsumo.');
```

Trigger 2

Finalmente, se utiliza el trigger **TR_BorrarPlato**, que se activa en lugar de la eliminación física (DELETE) de un registro en la tabla PlatoRestaurante. La función de este trigger es la siguiente:

Al intentarse eliminar un plato, se captura el ID del plato desde la tabla virtual *deleted*. Así, en lugar de borrar el registro, se actualiza el campo FechaBaja con la fecha actual, registrando así la baja lógica del plato.

```
Create Or Alter Trigger TR_BorrarPlato ON PlatoRestaurante
Instead Of Delete
As
Begin
    Declare @IDPlatoRestaurante int;

    -- Obtenemos el ID del plato a borrar
    Select @IDPlatoRestaurante = IDPlatoRestaurante from deleted;

    -- En lugar de eliminar, marcamos la baja logica asignando la fecha actual
    Update PlatoRestaurante
    Set FechaBaja = getdate()
    Where IDPlatoRestaurante = @IDPlatoRestaurante
        And FechaBaja IS NULL;

    Print 'Plato dado de baja lógica correctamente';
End;
Go
```

Trigger 3 y 4

Dentro del sistema de gestión de reseñas gastronómicas, se implementaron dos triggers que garantizan la unicidad del campo Email en las tablas Cliente y Restaurante. Estas validaciones son fundamentales para asegurar la integridad y confiabilidad del sistema, especialmente considerando que el correo electrónico es un identificador clave para el ingreso a la plataforma, la recuperación de credenciales y la comunicación con los usuarios.

TR_EmailUnicoCliente

Este trigger de tipo INSTEAD OF INSERT actúa sobre la tabla Cliente y tiene como propósito evitar la inserción de clientes con un correo electrónico que ya esté registrado. Al interceptar la operación de inserción:

- Comprueba si en la tabla ya existe un registro con el mismo email (Email = i.Email).
- Si detecta una coincidencia, cancela la operación con ROLLBACK e imprime un mensaje de advertencia, evitando la duplicación.
- Si no hay conflictos, realiza la inserción de manera controlada a partir de la tabla virtual inserted.

Esta medida permite mantener la unicidad lógica de los clientes incluso cuando no se haya definido una restricción UNIQUE directamente sobre el campo Email.

TR_EmailUnicoRestaurante

El funcionamiento de este trigger es análogo al del cliente, pero aplicado a la tabla Restaurante. Su objetivo es impedir que dos restaurantes compartan la misma dirección de correo electrónico, lo cual podría comprometer la autenticación o gestión de contenido dentro del sistema. Al igual que el anterior:

- Realiza una verificación previa antes de insertar.
- En caso de duplicación, detiene el proceso y emite un mensaje.
- Permite la inserción solamente si se respeta la unicidad del email.

Importancia de estos triggers

Implementar esta lógica mediante triggers presenta varias ventajas:

- Permite centralizar la validación, evitando tener que repetirla desde la aplicación cliente.
- Ofrece una capa adicional de seguridad ante errores humanos o scripts que intenten insertar registros incorrectos.
- Ambos triggers fueron desarrollados utilizando bloques TRY...CATCH para el manejo seguro de errores y son parte fundamental del diseño robusto del sistema.

```

33 --evitar duplicados en clientes
34
35 CREATE TRIGGER TR_EmailUnicoCliente
36 ON Cliente
37 INSTEAD OF INSERT
38 AS
39 BEGIN
40     BEGIN TRY
41         -- Verificamos si ya existe un email igual
42         IF EXISTS (
43             SELECT 1
44             FROM Cliente c
45             JOIN inserted i ON c.Email = i.Email
46         )
47         BEGIN
48             PRINT 'Error: Ya existe un cliente con ese email.';
49             ROLLBACK TRANSACTION;
50             RETURN;
51         END;
52
53         -- Insert si no existe conflicto
54         INSERT INTO Cliente (Cuil, Email, Password, NombreCliente, ApellidoCliente, IDLocalidad)
55         SELECT Cuil, Email, Password, NombreCliente, ApellidoCliente, IDLocalidad
56         FROM inserted;
57
58         COMMIT;
59         PRINT 'cliente insertado correctamente.';
60     END TRY
61     BEGIN CATCH
62         PRINT 'Se produjo un error inesperado en la inserción del cliente.';
63         ROLLBACK TRANSACTION;
64     END CATCH
65 END;
66 GO

```

Trigger 5

En el contexto de una aplicación de reseñas de restaurantes, es esencial asegurar que los valores de calificación ingresados por los usuarios sean válidos y consistentes. Por ello, se diseñó e implementó el trigger TR_ValidarRangoCalificacion, cuyo objetivo es **verificar que toda calificación registrada esté dentro del rango permitido de 1 a 5**.

Funcionamiento del trigger

Este trigger es de tipo INSTEAD OF INSERT y actúa sobre la tabla Calificaciones. La lógica que implementa es la siguiente:

- Al producirse una operación de inserción (INSERT), el trigger analiza los valores que llegarían a la base mediante la tabla virtual inserted.
- Si encuentra que **alguna de las calificaciones propuestas no está entre 1 y 5**, entonces:
 - Cancela la operación con un ROLLBACK.

- Emite un mensaje con PRINT que indica que los valores están fuera del rango aceptado.
- Si todos los valores están dentro del rango permitido, entonces:
 - Inserta los datos en la tabla Calificaciones de manera segura.
 - Confirma la transacción con un COMMIT.

```

tp_final_armafo.sql
--
830 CREATE OR ALTER TRIGGER TR_Calificacion_RangoValido
831 ON Calificaciones
832 INSTEAD OF INSERT
833 AS
834 BEGIN
835     BEGIN TRY
836         -- Validación: ¿hay alguna calificación fuera de rango?
837         IF EXISTS (
838             SELECT 1 FROM Inserted
839             WHERE Calificacion < 1 OR Calificacion > 5
840         )
841         BEGIN
842             PRINT 'Error: la calificación debe estar entre 1 y 5.';
843             ROLLBACK TRANSACTION;
844             RETURN;
845         END
846
847         -- Si todo está OK, se hace el insert real
848         INSERT INTO Calificaciones (
849             FechaConsumo,
850             FechaCalificacion,
851             ClienteId,
852             IDPlatoRestaurante,
853             Calificacion,
854             Comentarios
855         )
856         SELECT
857             FechaConsumo,
858             FechaCalificacion,
859             ClienteId,
860             IDPlatoRestaurante,
861             Calificacion,
862             Comentarios
863         FROM Inserted;
864
865         PRINT 'Calificación ingresada correctamente.';
866     END TRY
867

```

Links a los recursos

Script de creación de base de datos con datos

https://github.com/gabrielrskm/BD2_TPFinal/blob/main/Base%20de%20datos/BD%20Restaurante%20Final.sql

Video demo del sistema

<https://drive.google.com/drive/folders/1QcoDVszoybEXJ1I73v8zUDK7We3P8ZL-?usp=sharing>