**Introducción**

Este proyecto consiste en el diseño de una base de datos para una repostera, con el objetivo de organizar y gestionar eficientemente la información relacionada con productos, pedidos, ingredientes y clientes. El enfoque del modelo es práctico y apunta a simplificar el manejo diario de datos clave como recetas, ventas y stock.

**Objetivo**

El objetivo principal del proyecto es implementar una estructura de base de datos relacional que permita a la repostera registrar sus pedidos, administrar los ingredientes utilizados en cada receta y obtener un mayor control sobre sus productos y clientes. Esto también posibilita gestionar de forma ordenada los aspectos logísticos del negocio, como la fecha de entrega de pedidos, el seguimiento de productos vendidos y los costos por unidad.

**Situación Problemática**

La repostera necesita un sistema que le permita dejar de depender de registros manuales, mensajes o planillas desorganizadas, y centralizar toda la información crítica de su negocio. Actualmente, la falta de una estructura ordenada provoca errores en los pedidos, pérdida de información sobre ingredientes, dificultades para calcular cantidades o precios, y complicaciones al momento de gestionar los clientes o hacer análisis básicos. La implementación de esta base de datos busca cubrir esas brechas, permitiendo una administración clara, escalable y confiable de todos los procesos clave del emprendimiento.

**Modelo de Negocio**

La organización que se representa en este proyecto es un emprendimiento individual de repostería. Su modelo de negocio se centra en la producción y venta personalizada de productos como tortas, cupcakes y otros postres, preparados por pedido.

Los pedidos pueden tener particularidades como decoraciones especiales, cantidades personalizadas o ingredientes adicionales, lo cual hace necesario un sistema flexible y detallado para registrar cada transacción.

El negocio opera bajo demanda, es decir, produce solo lo que vende, por lo que resulta fundamental controlar con precisión los ingredientes usados por receta, los costos asociados, la planificación de entregas y la gestión de los clientes. La solución propuesta permite gestionar todos estos aspectos desde una base de datos relacional estructurada.

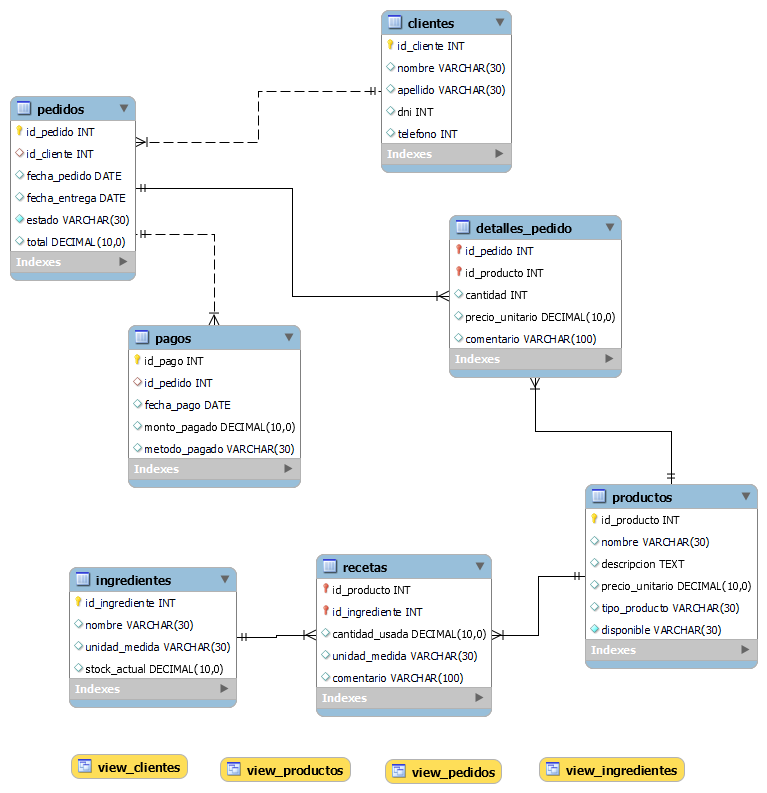
**Diagrama Entidad-Relación (E-R)**

El diagrama E-R representa de manera fiel la estructura de la base de datos generada a partir del script SQL del proyecto Tortas Mariluz. Incluye todas las entidades del sistema: productos, clientes, pedidos, detalles\_pedido, pagos, ingredientes y recetas.

Se muestran también las relaciones entre ellas, destacando:

* La relación entre clientes y pedidos, donde un cliente puede tener múltiples pedidos.
* La tabla intermedia detalles\_pedido, que vincula pedidos con productos, permitiendo registrar múltiples productos por pedido, su cantidad, precio y un comentario adicional personalizado.
* La relación de pedidos con pagos, permitiendo registrar diferentes métodos y fechas de pago.
* La relación entre productos e ingredientes, modelada mediante la tabla intermedia recetas, que especifica qué ingredientes y en qué cantidades se utilizan para cada producto.

Diagrama:



**Listado de Tablas:**

CREATE TABLE IF NOT EXISTS productos (

    id\_producto INT AUTO\_INCREMENT PRIMARY KEY,

    nombre VARCHAR(30),

    descripcion TEXT,

    precio\_unitario DECIMAL,

    tipo\_producto VARCHAR(30),  *-- "Torta", "cupcake", "tarta", etc*

    disponible BOOLEAN

)

CREATE TABLE IF NOT EXISTS clientes (

    id\_cliente INT AUTO\_INCREMENT PRIMARY KEY,

    nombre VARCHAR(30),

    apellido VARCHAR(30),

    dni INT,

    telefono INT

)

CREATE TABLE IF NOT EXISTS pedidos (

    id\_pedido INT AUTO\_INCREMENT PRIMARY KEY,

    id\_cliente INT,

    fecha\_pedido DATE,

    fecha\_entrega DATE,

    estado BOOLEAN,     *-- si esta listo para la entrega*

    total DECIMAL,

    FOREIGN key (id\_cliente) REFERENCES clientes(id\_cliente)

)

*/\* Tabla intermedia entre pedidos y productos \*/*

CREATE TABLE IF NOT EXISTS detalles\_pedido (

    id\_detalle INT AUTO\_INCREMENT PRIMARY KEY,

    id\_pedido INT,

    id\_producto INT,

    cantidad INT,

    precio\_unitario DECIMAL,

    comentario varchar(100),    *-- Si desea agregar algun extra al producto por ej: "chips"*

    FOREIGN key (id\_pedido) REFERENCES pedidos(id\_pedido),

    FOREIGN key (id\_producto) REFERENCES productos(id\_producto)

)

CREATE TABLE IF NOT EXISTS pagos (

    id\_pago INT AUTO\_INCREMENT PRIMARY key,

    id\_pedido INT,

    fecha\_pago DATE,

    monto\_pagado DECIMAL,

    metodo\_pagado VARCHAR(30)

    FOREIGN key (id\_pedido) REFERENCES pedidos(id\_pedido)

)

CREATE TABLE IF NOT EXISTS ingredientes (

    id\_ingrediente INT AUTO\_INCREMENT PRIMARY key,

    nombre VARCHAR(30),

    unidad\_medida VARCHAR(30),

    stock\_actual DECIMAL

)

*/\* Tabla intermedia entre productos y ingredientes \*/*

CREATE TABLE IF NOT EXISTS recetas (

    id\_receta INT AUTO\_INCREMENT PRIMARY key,

    id\_producto INT,

    id\_ingrediente INT,

    cantidad\_usada DECIMAL,

    unidad\_medida VARCHAR(30),

    descripcion\_uso VARCHAR(100),

    FOREIGN key id\_producto REFERENCES productos(id\_producto),

    FOREIGN key id\_ingrediente REFERENCES ingredientes(id\_ingrediente)

)