Base de Datos Pizzeria

# Documentación Técnica

Autor: César Aracena

Documentación del Desarrollo de la base de datos

“pizzeria” creada como ejemplo para la clase de SQL

De Coderhouse.

## Revisión 3 | 30 de Abril de 2023

# Resúmen

En este documento se detalla el modelo de negocio, situación problemática, el objetivo de este trabajo, así como también referente a la parte técnica, los Diagramas de Entidad-Relación, estructura de tablas, funciones, procedimientos, vistas y triggers creados/utilizados para solucionar los problemas listados.

# Introducción

En el mundo existen innumerables modelos de negocios que cada día generan y obtienen infinidad de datos que, almacenados de manera ordenada, permiten realizar análisis y tomar decisiones basadas en datos para permitir operar cada vez mejor.

El siguiente documento detalla técnicamente el proceso de creación de una base de datos para una pizzería imaginaria que le permitiría a su dueño tomar mejores decisiones a la hora de realizar nuevas inversiones o simplemente llevar mejor su negocio día a día.

# Objetivo

Intentar demostrar una manera ordenada, relacional y digital de almacenar los datos generados u obtenidos día a día para lograr obtener mejores indicadores de rendimiento clave (KPIs por sus siglas en inglés).

Una vez creado el sistema empleando un sistema de bases de datos popular y gratuito como es MySQL, se podrán obtener resúmenes objetivos de las métricas que sean útiles para la toma de decisiones.

# Situación Problemática

Basándome en la manera tradicional de llevar un pequeño comercio como lo puede ser una pizzería, pude listar los siguientes problemas:

* Sobreabastecimiento
* Subabastecimiento
* Falta de ofertas correctas según ganancia o abastecimiento
* Falta de seguimiento de pagos impositivos
* Ofertas de productos sin stock

# Modelo de Negocio

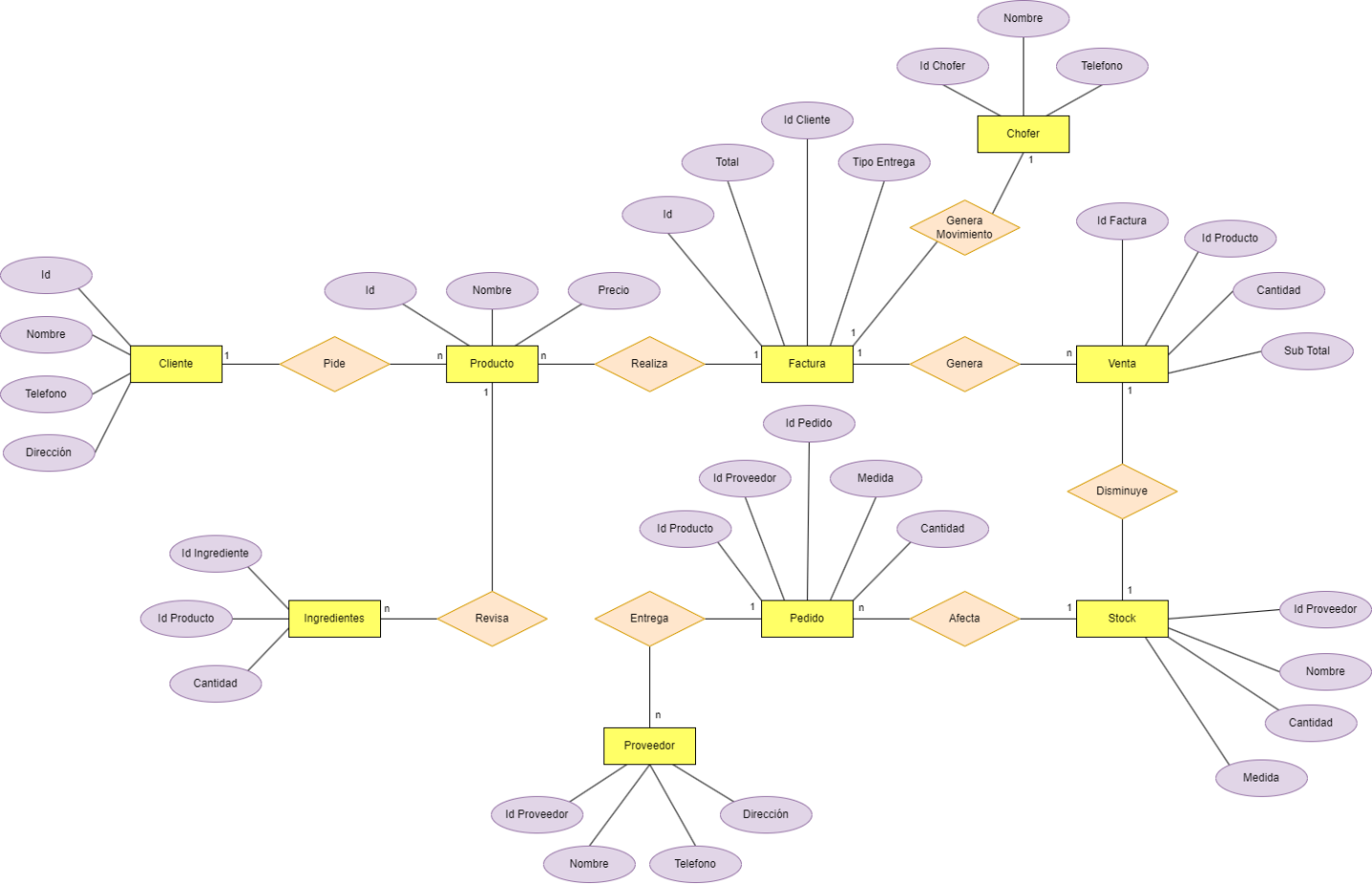
Como se mencionó previamente, el modelo de negocio elegido para este trabajo es una pizzería tradicional que consta de la siguiente estructura:

* Un (1) propietario único
* Cuatro (1) empleados en local
* Capacidad de entrega en sitio
* Capacidad de delivery

En este ejemplo nuestra pizzería trabaja 6 días a la semana (de martes a domingo) y solo por las noches (de 19hr a 24hr).

# Diagrama Entidad-Relación (conecptual)

Se adjunta el DER conceptual para entender/visualizar rápidamente las entidades y sus relaciones:



# Diagrama Entidad-Relación (Esquemático)

Se adjunta DER esquemático (de ingeniería) con las relaciones entre tablas. El mismo está dividido en tres partes para facilitar su lectura:

A picture containing text, diagram, plan, parallel

Description automatically generated

Ilustración - Tablas de auditoría

A picture containing text, diagram, plan, parallel

Description automatically generated

Ilustración - Tablas principales

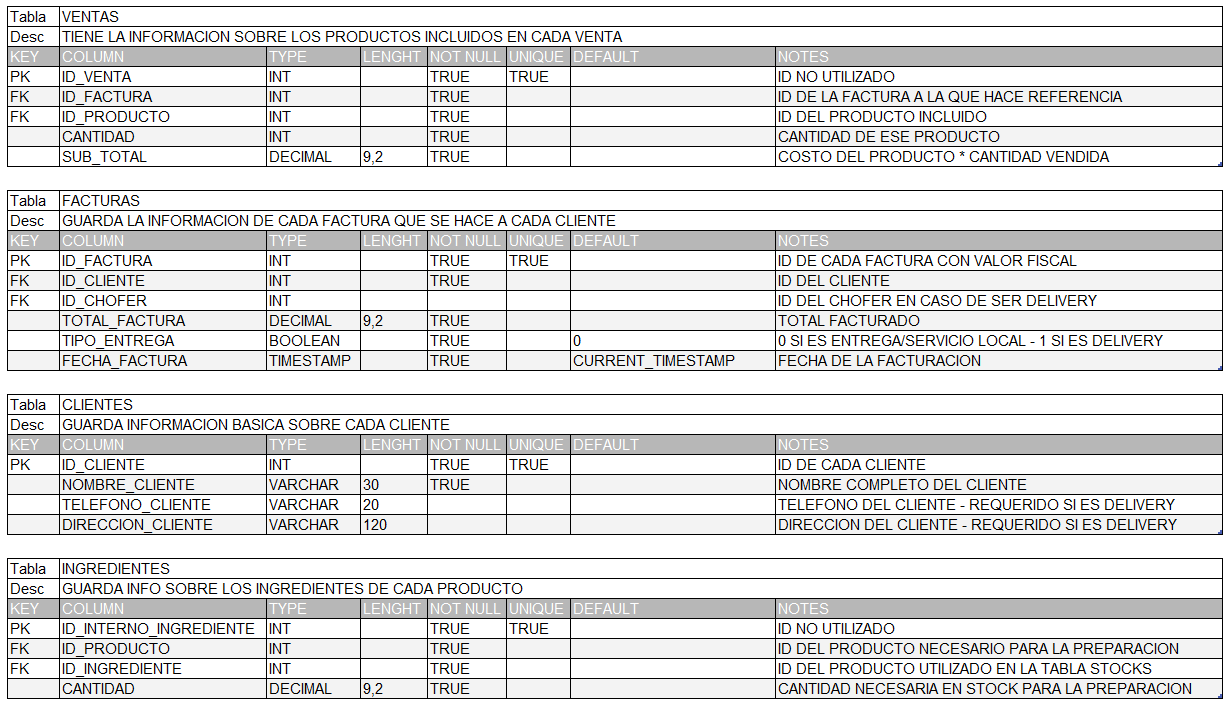
A picture containing text, diagram, plan, parallel

Description automatically generated

Ilustración - Tablas de asistencia (temporales)

# Listado de Tablas

A continuación, se incluye el listado de las tablas creadas con sus atributos:



A screenshot of a computer

Description automatically generated with low confidence

A picture containing text, screenshot, number, font

Description automatically generated

# Funcionalidad

A continuación, se describen los elementos creados a partir de la estructura de la base de datos para asistir en el trabajo con la misma:

## Vistas

Se crearon las siguientes vistas:

* facturación\_mensual: Genera un simple output que muestra el total facturado durante el mes en curso
* facturación\_trimestral: Muestra 3 campos donde se encuentra la facturación del mes en curso, más la facturación total de los dos meses anteriores
* menu: Genera una vista simplificada donde solo muestra los productos que se pueden fabricar según el stock de los componentes necesarios para cada uno
* productos\_a\_pedir: Genera un listado de los productos cuyo stock es menor al mínimo deseado
* productos\_mas\_vendidos: Muestra un listado en orden de los productos más vendidos históricamente

## Stored Procedures

Los siguientes son procedimientos almacenados que se crearon para simular la operación diaria con la base de datos, ya que no cuenta con un programa externo que realice las consultas necesarias:

* sp\_generar\_venta: Simula la creación de una venta a un cliente aleatorio, donde la cantidad de productos y número de cada producto vendido son aleatorios
* sp\_venta\_completo: Realiza los movimientos en la base de datos, dictados por el procedimiento anterior
* sp\_venta\_extra: Es un procedimiento auxiliar que realiza tareas de mantenimiento en la base de datos

|  |  |
| --- | --- |
| Lightbulb with solid fill | Nota: Para simular una venta, solo debe ejecutar/llamar al procedimiento “sp\_generar\_venta” y, siempre y cuando esté disponible un repartidor o, aleatoriamente se realice una venta en el local, se generará una nueva venta |

## Funciones

Las siguientes funciones ayudan con cálculos preestablecidos y de uso frecuente:

* calcular\_provision: Devuelve la cantidad a pedir de un producto o ingrediente, basándose en el 110% de la diferencia entre la cantidad ideal y la cantidad actual
* f\_generar\_cliente: Genera un cliente aleatorio tomando un nombre y un apellido de una tabla auxiliar
* f\_rand\_entrega: Genera un valor aleatorio entre 1 y 2 para determinar si un pedido es para retirar en el local o para ser enviado
* f\_random: Genera un número aleatorio entre 100 y 999999 para utilizar como valor provisional de venta en cada transacción
* fventas: Calcula el total a facturar de una venta
* iva: Calcula el impuesto al valor agregado de cada venta

## Triggers

Los siguientes triggers fueron creados para llevar una auditoría de algunos eventos:

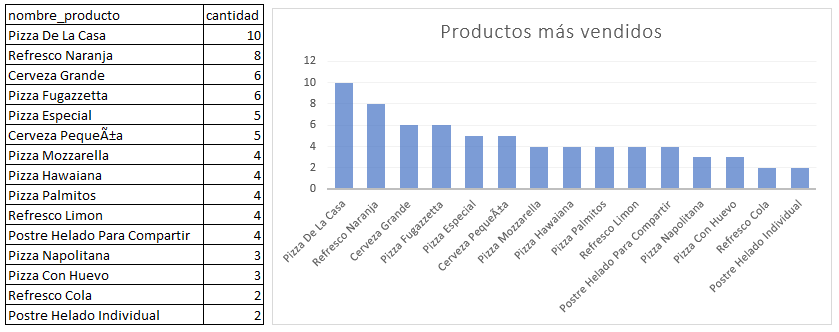
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Trigger | Tabla que monitorea | Acción que monitores | Tabla donde registra | Descripción |
| clientes\_AFTER\_INSERT | clientes | AFTER INSERT | logs | Registra la creación de cada cliente nuevo |
| tr\_logs | facturas | AFTER INSERT | logs | Registra cada factura generada |
| tr\_clientes\_alta | clientes | AFTER INSERT | log\_clientes | Registra el alta de cada cliente |
| tr\_clientes\_eliminacion | clientes | AFTER DELETE | log\_clientes | Registra cuando un cliente es eliminado |
| tr\_nueva\_venta | ventas | AFTER INSERT | log\_ventas | Registra cada venta nueva |
| tr\_agregar\_producto | productos | AFTER INSERT | log\_productos | Registra cada producto que se añade al menú |
| tr\_modificar\_producto | productos | AFTER UPDATE | log\_productos | Registra la modificación o edición de cada producto |
| tr\_eliminar\_producto | productos | AFTER DELETE | log\_productos | Registra la eliminación de un producto |

# Business Analysis

Con las vistas creadas es simple obtener los siguientes resúmenes para su posterior análisis:

## Productos más vendidos

Nos permite saber cuales son los productos más vendidos para así poder coordinar ofertas en días y horarios calculados:



## PRODUCTOS PARA PEDIR

Esta otra vista nos permite saber rápidamente que productos están bajo el mínimo de stock y nos provee datos sobre donde pedir cada uno:

A picture containing text, diagram, screenshot, line

Description automatically generated

# Tecnologías utilizadas

Para la creación y desarrollo de esta base de datos se utilizaron las siguientes herramientas:

|  |  |  |
| --- | --- | --- |
| Herramienta | Versión | Utilidad |
| Servidor | Windows 11 64-bit | Servidor para ejecutar y desarrollar el motor de la base de datos |
| Base de Datos MySQL | CE 8.0.33 | Sistema de DB de uso gratuito |
| Draw.io | 21.1.2 | Diagramación de DERs |
| MS Excel | Excel for Microsoft 365 | Estructura de tablas y Data Analysis |
| MySQL Workbench | CE 8.0.33 | Interacción con DB |