

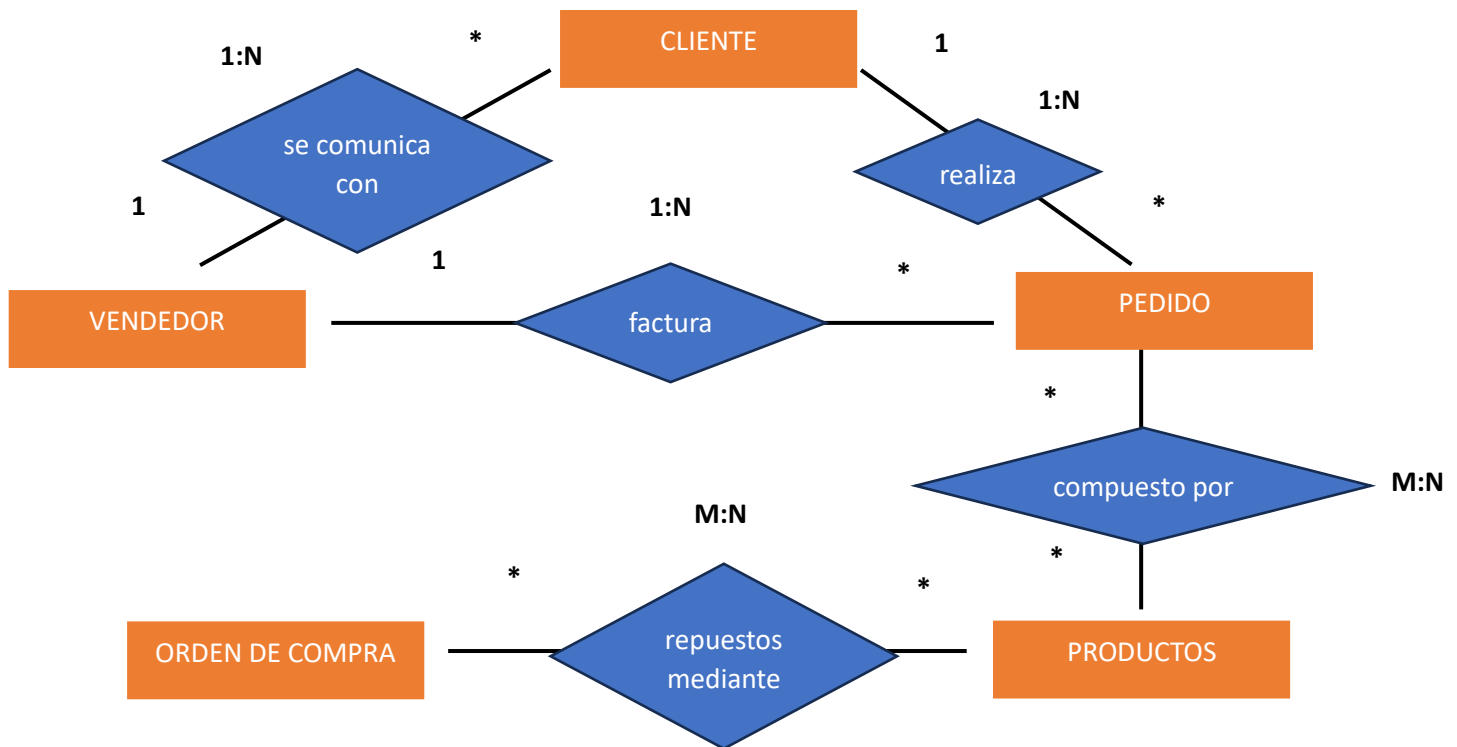
# PROYECTO MYSQL

## PRIMERA PREENTREGA

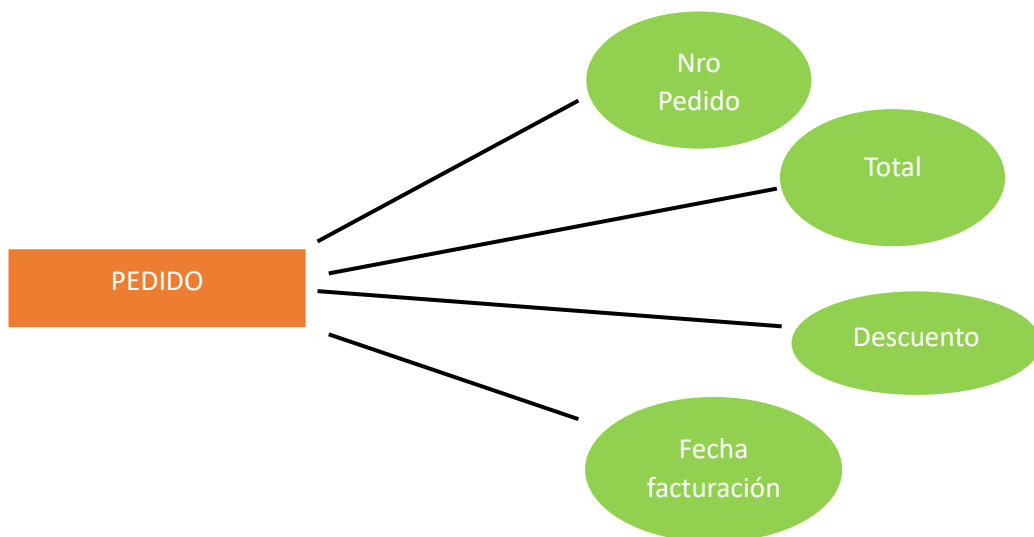
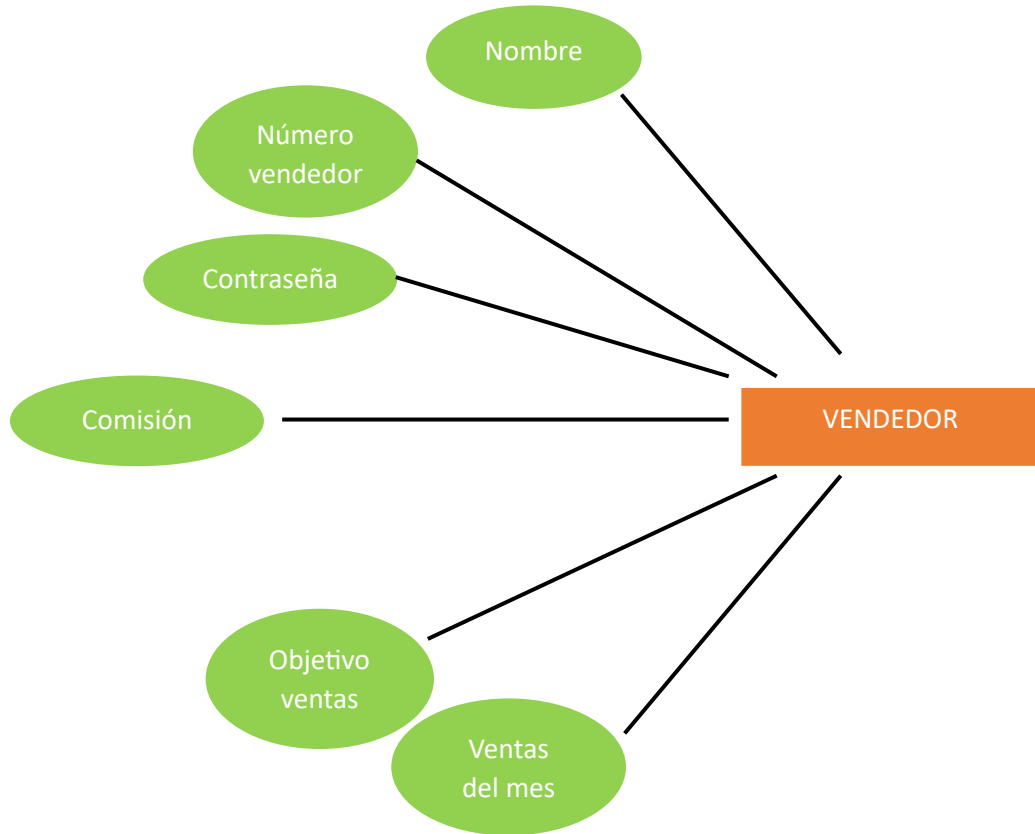
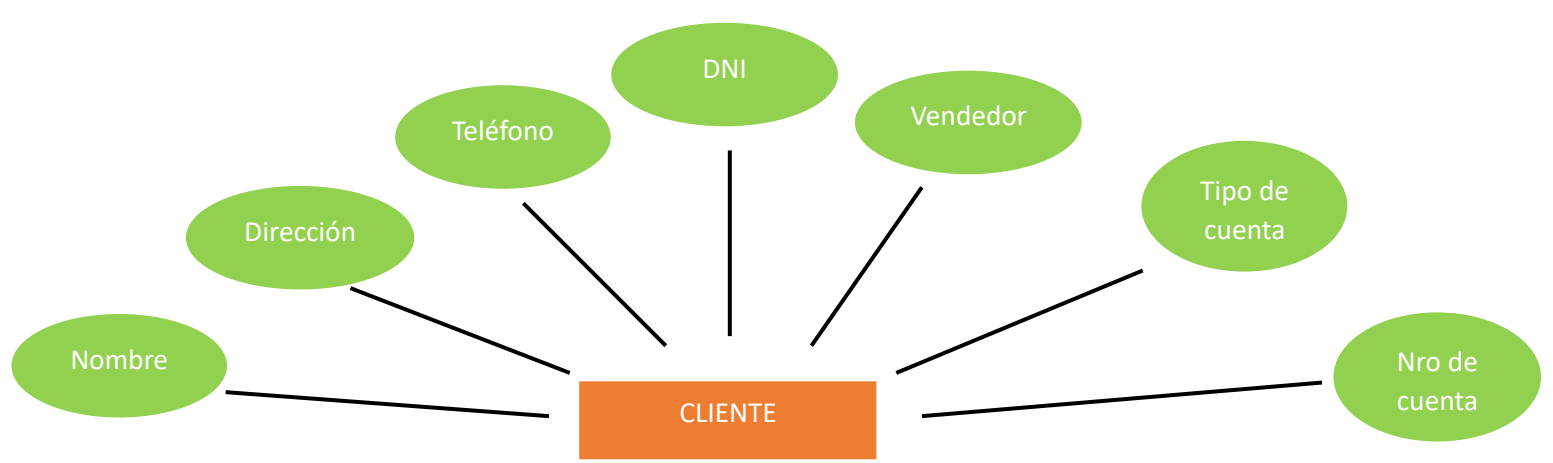
La Base de Datos que elaboraré a lo largo de este curso utilizando la herramienta el lenguaje SQL será una base de datos destinada a complementar la aplicación web desarrollada como proyecto en el curso de Javascript de esta misma plataforma, que es un simulador de facturación de materiales eléctricos.

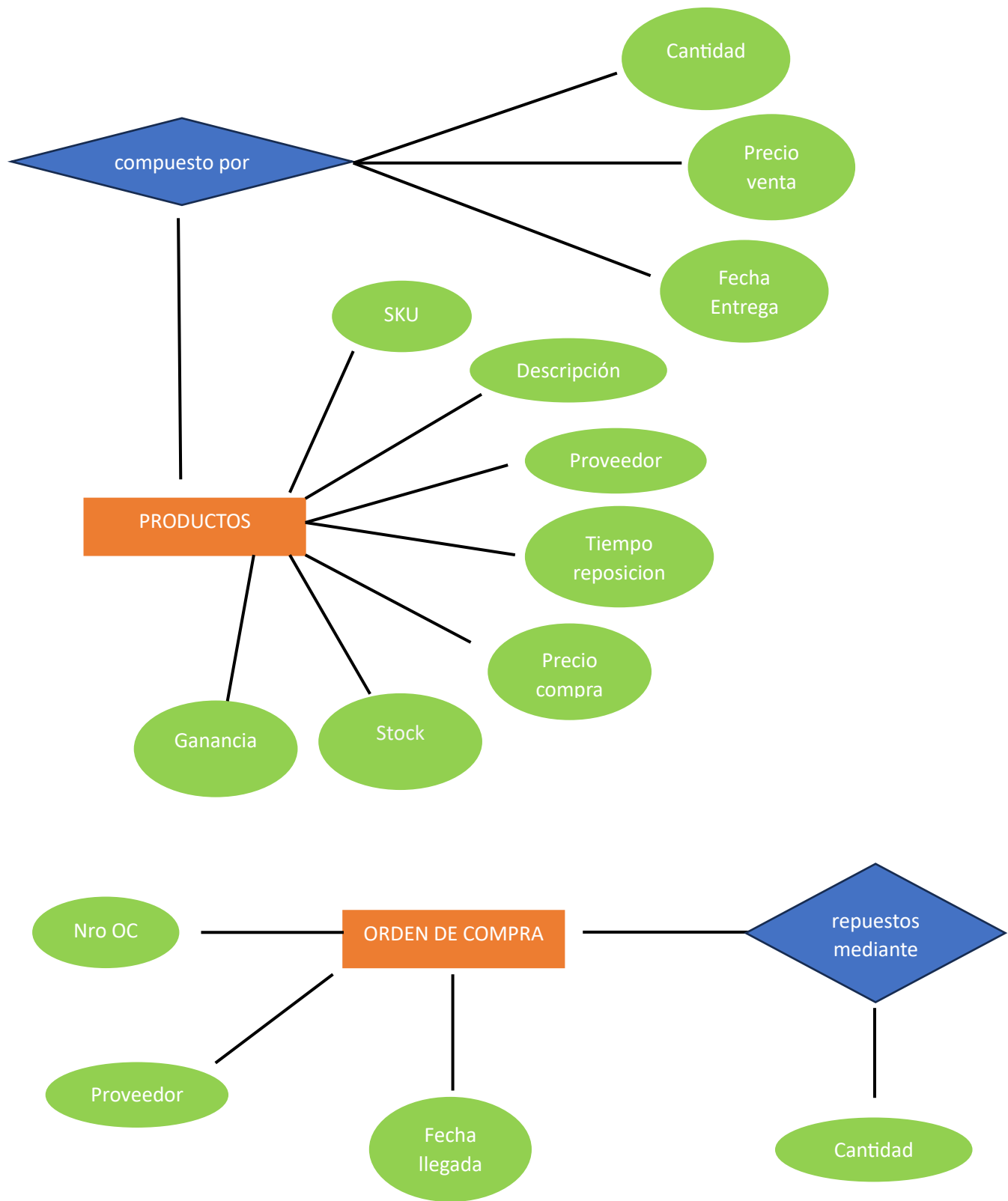
Esta base de datos contará con información sobre los artículos disponibles, precios de compra/venta, stock, datos del proveedor, etc. Así mismo, contará con información de los clientes, tales como datos de envío, tipo de cuenta, compras anteriores, y demás; información sobre pedidos y facturas ya realizadas; datos sobre los vendedores y órdenes de compra.

El diagrama entidad-relación asociado será el siguiente:



Por cuestiones de espacio, separé el diagrama en dos partes: la parte anterior contiene las entidades y sus relaciones; la parte posterior mostrará los campos asociados a cada entidad.





Así, siguiendo este diagrama, las tablas que compondrán la DB serán las siguientes

CLIENTS						
(Tabla que recolecta datos de clientes, útiles para registro contable y fines logísticos)						
accountNumber	dni	name	adress	phoneNumber	accountType	salesman
El número de cuenta asociado a cada cliente. Número único. Primary key (index key)	Número de DNI del titular de la cuenta	Nombre de la cuenta. Puede ser nombre real o razón social	Dirección a la que se entrega el material	Teléfono de contacto para coordinar entregas y/o pagos	Tipo de cuenta para registro de contabilidad. Puede ser cliente eventual o cta Corriente	Número de vendedor del vendedor que registró al cliente. Si es cta Corriente, este cliente es exclusivo suyo. Foreign key

SALESMEN					
(Tabla que recoge datos del vendedor, útiles para llevar un registro del rendimiento de dicho vendedor)					
sManNumber	password	name	salesTarget	commission	monthlySales
Número asignado a cada vendedor de la empresa, utilizado para su login. Primary key (index key)	Contraseña que utiliza el vendedor para su login	Nombre completo del vendedor	Objetivo de ventas mensuales que tiene el vendedor	Comisión que cobra el vendedor sobre el total de cada venta. Expresada en porcentaje	Total de ventas facturadas por el vendedor en el mes actual.

ORDERS					
(registro de todas las facturas hechas. Permite llevar control contable y continuar el proceso logístico)					
traceability	clientNumber	salesman	facturationDate	discount	totalPrice
Número de trazabilidad de la factura. Primary Key (index key)	Número de cuenta que encargó este pedido. Foreign key	Número de vendedor del vendedor que facturó este pedido. Foreign Key	Fecha en la cual se facturó este pedido.	Descuento que se aplica al total de la factura. Expresado en porcentaje	Precio final de la factura, contando la sumatoria de [precioArticulo]*[cantidad], y a ese precio multiplicarlo por discount.

ARTICLES (datos de un producto en concreto, que permiten llevar un control de ventas y logístico para entregar al cliente en tiempo y forma)						
SKU	articleDescription	supplier	repositionCost	stock	profit	repositionDelay
Código numérico interno único del artículo en cuestión. PrimaryKey (index key)	Descripción breve del artículo, que permite ubicarlo fácilmente	Razón social del proveedor al cual se le compra el artículo en cuestión	Precio total que cobra el proveedor por cada unidad de este artículo	Cantidad de unidades disponibles de ese artículo	Porcentaje de ganancia que la empresa le saca a ese artículo	Cantidad de días que tarda el proveedor en fabricar, enviar, y nuestra empresa en recibir el artículo

PURCHASE_ORDERS (datos que permiten un seguimiento logístico de cuando se encarga reposición de un producto al proveedor)		
orderNumber	supplier	dateOfArrival
Número de orden de compra con el cual llevar seguimiento contable y logístico del envío del proveedor. Primary key (index key)	Razón social del proveedor al cual se le encargó esta orden de compra	Fecha en la cual llegará a nuestra empresa los productos incluidos en esta orden de compra.

ARTICLES*ORDER (tabla intermedia que refleja la relación muchos a muchos entre los artículos y los pedidos facturados)				
orderNumber	SKU	quantity	salePrice	dateOfDelivery
Número de pedido, tomado de la tabla ORDERS, del cual se registran los productos encargados. Foreign key	Código numérico interno de cada producto encargado en el pedido, tomado de la tabla ARTICLES. Único por orden. Foreign Key	Cantidad de unidades encargadas del producto en cuestión	Precio al que se vende cada unidad del producto en cuestión, tomado de [costo de reposición]*[porcentaje de ganancia establecido]*quantity	Fecha en la cual se pactó con el cliente entregar cada producto particular del pedido. No es necesario que sea el mismo para toda la orden.

ARTICLES*PURCHASE_ORDER (tabla intermedia que refleja los artículos incluidos en una orden de compra a un proveedor)		
orderNumber	SKU	quantity
Número de orden de compra, tomado de la tabla PURCHASE_ORDERS, de la cual se registran los productos incluidos. Foreign key	Código numérico interno de cada producto encargado en la orden de compra, tomado de la tabla ARTICLES. Único por orden. Foreign Key	Cantidad de unidades encargadas al proveedor del producto en cuestión.

Se incluirá un archivo en el enlace de GitHub al final de este documento con un script de inserción de datos para la inicialización de esta DB.

Además de las tablas mencionadas, se incluirán las siguientes vistas en la base de datos

**RESUMEN DE VENTAS POR VENDEDOR:** esta vista permite resumir el progreso en ventas de un vendedor, contando cuantos pedidos facturó en determinado mes y año, y el total de dinero que facturó en dicho período. Así mismo, agrupa estos resultados para obtener un resumen por vendedor, año, y mes. Su utilidad es poder analizar el rendimiento de cada vendedor, hacer comparaciones entre ellos, y así determinar, por ejemplo, el valor de las comisiones de cada uno. Está compuesta a partir de la tabla ORDERS

**MEJORES CLIENTES:** esta vista permite analizar cuáles son los clientes más recurrentes de la empresa, basándose simplemente en cantidad de pedidos realizados, y luego los ordena de mayor a menor cantidad de pedidos realizados. Muestra tanto el número de cuenta como el nombre/razón social del cliente. Su utilidad es permitir analizar qué clientes están más fidelizados, para así poder ofrecerles mayores beneficios, como descuentos, ofertas, prioridad en las ventas, etc. Está compuesta a partir de las tablas CLIENTS y ORDERS

**RESUMEN DE STOCK:** esta vista permite lisa y llanamente analizar el stock disponible de todos los artículos de la empresa, mostrando tanto el código del artículo como su descripción. Su utilidad radica en poder analizar qué disponibilidad hay de cada artículo para ser facturado, o ver si es necesario realizar una nueva orden de compra. Está formada únicamente de la tabla ARTICLES.

**RESUMEN DE ÓRDENES DE COMPRA:** esta vista permite ver un resumen de las órdenes de compra pendientes de ser recibidas. Permite visualizar el número de orden, el proveedor al que fue hecha, la fecha estimada de llegada, y un cálculo de cuántos días faltan para que llegue esa orden de compra a la empresa y poder entregar el material. Su utilidad radica tanto para el seguimiento por parte de los compradores del proceso de recepción de material, como por parte de los vendedores para informar correctamente a sus clientes y tomar mejores decisiones a la hora de facturar. Está formada a partir de la tabla PURCHASE\_ORDERS

Los siguientes Stored Procedures son procedimientos a partir de los cuales se realizan las operaciones necesarias para mantener esta base de datos correctamente

**AÑADIR NUEVO CLIENTE:** este procedimiento permite añadir una nueva entrada a la tabla de clientes, tomando como parámetros de entrada todos los datos necesarios menos el número de cuenta que se genera automáticamente por el AUTO\_INCREMENT de la PK de esta tabla.

Permite a los vendedores dar de alta a un nuevo cliente en cualquier momento, y así ampliar la base de clientes de la empresa. Interactúa únicamente con la tabla CLIENTS.

**NUEVO PEDIDO:** este procedimiento permite generar una nueva factura, y una vez generada añadirle todos los artículos que el cliente solicite. Toma como parámetros todos los datos necesarios de facturación, y además crea una tabla accesoria donde añade artículo por artículo la cantidad pedida por el cliente, a la que luego se le calcula el precio artículo por artículo y el de toda la factura. Su utilidad es evidente, pues es la principal función del asistente de facturación, generar una nueva factura con todo el detalle de artículos. Este procedimiento interactúa con las tablas ORDERS y ARTICLES\_ORDER.

**AÑADIR NUEVO ARTÍCULO:** este procedimiento permite agregar a la DB un nuevo producto a ser vendido, solicitando descripción, proveedor, precio y tiempo de entrega, pues el código se genera automáticamente por el AUTO\_INCREMENT de la tabla de artículos. Su utilidad es la de permitir crear nuevas entradas de artículos, con lo cual es posible vender nuevos artículos que no estaban antes en la DB. Interactúa únicamente con la tabla ARTICLES

**AÑADIR NUEVO VENDEDOR:** este procedimiento permite actualizar la DB de vendedores, añadiendo a un nuevo vendedor a la empresa, tomando los datos necesarios del empleado, menos el número de identificación, generado automáticamente. Su utilidad es que permite ser flexible con la incorporación de nuevo personal, sin estorbar las estadísticas de vendedores previos o posteriores, y poder llevar un registro de cada uno de los vendedores. Solo interactúa con la tabla SALESMEN

**ACTUALIZAR COMISIÓN:** este procedimiento permite actualizar el valor de la comisión que un vendedor tiene asignada, a partir de su número de vendedor. Es útil dado que puede funcionar como un incentivo o un correctivo, a partir de que el rendimiento del vendedor en cuanto a facturación sea el adecuado o no. Este procedimiento solo trabaja con la tabla SALESMEN

**ACTUALIZAR OBJETIVO DE VENTAS:** de manera similar al procedimiento anterior, este permite mantener actualizado el objetivo de ventas de cada vendedor, con lo cual a partir de su rendimiento periódico se pueden establecer objetivos realistas teniendo en cuenta su desempeño, el flujo de clientes, la inflación, etc. Igualmente, solo interactúa con la tabla SALESMEN

**\*NOTA:** decidí no incluir procedimientos para modificar el stock de los artículos existentes dado que la principal función de esta base de datos es servir de complemento a un asistente de facturación, y el control del stock es responsabilidad del área logística de una empresa, no de sus vendedores. Por este mismo motivo, no incluí funciones ni procedimientos contables, ni de eliminación de datos, dado que no es la principal función del asistente y por lo tanto de la DB.

Así mismo, las siguientes funciones serán necesarias para calcular algunos ítems importantes en algunas tablas:

**VER STOCK POR ARTÍCULO:** simplemente toma como parámetro el SKU de un artículo, y con él devuelve el stock disponible de dicho artículo. Útil para realizar consultas rápidas referentes a un artículo. No manipula ninguna tabla; solo lee valores de la tabla ARTICLES.

**CALCULAR PRECIO TOTAL FACTURA:** esta función permite calcular el precio total a pagar en una determinada factura. Es útil para que el cliente pueda realizar un solo pago por todo el pedido,

en vez de pagar artículo por artículo, así como para llevar un registro contable. Funciona con la fórmula  $\text{Sumatoria}(\text{precioDeVenta por artículo})$ . Toma sus datos de las tablas ORDERS y ARTICLES\_ORDER, y el parámetro retornado pasa a ser incluido en la propia tabla ORDERS

CALCULAR PRECIO DE VENTA: esta función permite separar el precio de venta de cada artículo, con lo cual se pueden establecer márgenes de venta diferentes y mantenerlos actualizados según oferta demanda. Permite a su vez tener una visión clara del precio real de un artículo a la hora de venderlo. Utiliza la fórmula  $[\text{costoReposicion}] * (1 + \text{porcentajeGanancia}) * [\text{cantidad}]$ , que toma de las tablas ARTICLES, e inserta el resultado en la tabla ARTICLES\_ORDER

En el siguiente link de GitHub se encuentra un archivo SQL con el script para generar las tablas mencionadas en este documento:

<https://github.com/L4U7YDLC/Proyecto-final-curso-SQL-Coderhouse>