PROYECTO

FINAL SQL

CURSO CODERHOUSE

**Lucas Rubio**

INDICE

[INTRODUCCIÓN 2](#_Toc126598900)

[OBJETIVO 2](#_Toc126598901)

[SITAUCION PROBLEMÁTICA 2](#_Toc126598902)

[MODELO DE NEGOCIO 2](#_Toc126598903)

[D.E.R 3](#_Toc126598904)

[DESCRIPCIÓN DE TABLAS 4](#_Toc126598905)

[DESCRIPCIÓN DE VISTAS 10](#_Toc126598906)

[DESCRIPCION DE FUNCIONES 11](#_Toc126598907)

[DESCRIPCION DE STORED PROCEDURES 12](#_Toc126598908)

[DESCRIPCION DE TRIGGERS 13](#_Toc126598909)

[CREACION DE USUARIOS DCL 14](#_Toc126598910)

[CONCLUCIÓN 15](#_Toc126598911)

[HERRAMIENTAS UTILIZADAS 15](#_Toc126598912)

# INTRODUCCIÓN

A lo largo del tiempo como parte del proyecto estuve haciendo una base de datos relacional para un negocio dedicado a la venta de artículos de computación. En esta misma se almacenan los datos de todos los artículos, de los clientes, las ventas, los ingresos, los proveedores y los empleados.

## OBJETIVO

El objetico de esta base de datos es poder tener un registro de todo lo que hace el negocio. Ya sea el ingreso de los artículos, su venta, los datos de los clientes, etc. A su vez su finalidad también es la de poder saber que artículos se venden más, las ganancias o pérdidas que se tuvo en el mes o el año, buscar de que clientes compraron X artículo, entre muchas consultas más que se podrían a hacer de forma más eficiente.

## SITAUCION PROBLEMÁTICA

Todos los registros de venta y compra de estos artículos los tenían en documentación física y se empezó a hacer difícil el poder buscar de forma eficiente y rápida alguna compra en específico por el número de compra o comprar las ganancias que tuvieron en diferentes meses, saber que artículo se vendió más, etc.

## MODELO DE NEGOCIO

Se resolvió implementar al negocio una base de datos relacional con el sistema de gestión MySQL. En esta aplicación podemos crear la base de datos y resolver el problema que tenia el negocio satisfaciendo sus necesidades.

# D.E.R

# DESCRIPCIÓN DE TABLAS

A continuación, voy a empezar a explicar las diferentes tablas de la base de datos a medida que las fui creando.

|  |  |  |  |
| --- | --- | --- | --- |
| **comprobantes** | | | |
| **Descripción** | Esta tabla sirve para saber el tipo de comprobante que se realiza a la hora de hacer la compra o venta de los artículos. | | |
| **Campo** | **Tipo de dato** | **Llave** | **Descripción** |
| id\_comprobante | SMALLINT | PK | Id de la tabla |
| tipo\_comprobante | VARCHAR (15) | - | El tipo de comprobante |

|  |  |  |  |
| --- | --- | --- | --- |
| **tipos\_pagos** | | | |
| **Descripción** | En esta tabla se indica como fue hecho el pago, ya sea con tarjeta, efectivo, transferencia, etc. | | |
| **Campo** | **Tipo de dato** | **Llave** | **Descripción** |
| id\_tipo\_pago | SMALLINT | PK | Id de la tabla |
| tipo\_pago | VARCHAR (20) | - | El tipo de pago |

|  |  |  |  |
| --- | --- | --- | --- |
| **puestos\_trabajo** | | | |
| **Descripción** | En esta tabla se indica cual es el puesto de los trabajadores. | | |
| **Campo** | **Tipo de dato** | **Llave** | **Descripción** |
| id\_puesto | SMALLINT | PK | Id de la tabla |
| puesto | VARCHAR (20) | - | El puesto del empleado |

|  |  |  |  |
| --- | --- | --- | --- |
| **empleados** | | | |
| **Descripción** | En esta tabla se indican los datos de los empleados y hay una referencia a la PK de otra tabla. | | |
| **Campo** | **Tipo de dato** | **Llave** | **Descripción** |
| id\_empleado | SMALLINT | PK | Id de la tabla |
| id\_puesto | VARCHAR (20) | FK | Hace referencia a la tabla de puestos\_trabajo |
| nombre | VARCHAR (20) | - | Nombre del empleado |
| apellido | VARCHAR (20) | - | Apellido del empleado |
| dni | CHAR (10) | - | DNI del empleado |
| teléfono | VARCHAR (35) | - | Teléfono del empleado |
| dirección | VARCHAR (50) | - | Dirección del empleado |
| email | VARCHAR (30) | - | Email del empleado |
| horario | CHAR (13) | - | Horario de trabajo del empleado |
| salario | INT | - | Salario mensual del empleado |

|  |  |  |  |
| --- | --- | --- | --- |
| **clientes** | | | |
| **Descripción** | En esta tabla se indican los datos de los clientes. | | |
| **Campo** | **Tipo de dato** | **Llave** | **Descripción** |
| id\_cliente | SMALLINT | PK | Id de la tabla |
| nombre | VARCHAR (20) | - | Nombre del cliente |
| apellido | VARHCAR (20) | - | Apellido del cliente |
| dni | CHAR (10) | - | DNI del cliente |
| dirección | VARCHAR (50) | - | Dirección del cliente |
| teléfono | VARCHAR (30) | - | Teléfono del cliente |
| email | VARCHAR (30) | - | Email del empleado |

|  |  |  |  |
| --- | --- | --- | --- |
| **proveedores** | | | |
| **Descripción** | En esta tabla se indican los datos de los proveedores donde compra la empresa. | | |
| **Campo** | **Tipo de dato** | **Llave** | **Descripción** |
| id\_proveedor | SMALLINT | PK | Id de la tabla |
| nombre | VARCHAR (50) | - | Nombre de la empresa |
| dirección | VARCHAR (50) | - | Dirección de la empresa |
| teléfono | VARCHAR (35) | - | Teléfono de la empresa |
| email | VARCHAR (30) | - | Email de la empresa |

|  |  |  |  |
| --- | --- | --- | --- |
| **rubros** | | | |
| **Descripción** | Esta tabla sirve para indicar el rubro de los artículos que se venden. | | |
| **Campo** | **Tipo de dato** | **Llave** | **Descripción** |
| id\_rubro | SMALLINT | PK | Id de la tabla |
| nombre | CHAR (17) | - | Nombre del rubro |

|  |  |  |  |
| --- | --- | --- | --- |
| **artículos** | | | |
| **Descripción** | En esta tabla se indican los datos de todos los artículos que se venden. También se encuentran referencias a otras tablas. | | |
| **Campo** | **Tipo de dato** | **Llave** | **Descripción** |
| id\_articulo | SMALLINT | PK | Id de la tabla |
| id\_proveedor | SMALLINT | FK | Hace referencia a la PK de la tabla proveedores |
| id\_rubro | SMALLINT | FK | Hace referencia a la PK de la tabla de rubros |
| detalles | VARCHAR (100) | - | Detalles de los productos a la venta |
| precio | INT | - | Precio de los artículos por unidad |
| stock | SMALLINT | - | Stock de los artículos |
| **ventas** | | | |
| **Descripción** | Esta tabla sirve para saber todas las ventas que hizo la empresa. | | |
| **Campo** | **Tipo de dato** | **Llave** | **Descripción** |
| id\_ventas | SMALLINT | PK | Id de la tabla |
| id\_cliente | SMALLINT | FK | Hace referencia a la PK de la tabla de clientes |
| id\_comprobante | SMALLINT | FK | Hace referencia a la PK de la tabla de comprobantes |
| id\_tipo\_pago | SMALLINT | FK | Hace referencia a la PK de la tabla de tipos\_pagos |
| fecha\_venta | DATETIME | - | Fecha en la cual se realizó la venta |

|  |  |  |  |
| --- | --- | --- | --- |
| **ventas\_detalles** | | | |
| **Descripción** | Esta tabla está relacionada con la tabla ventas, pero con más detalles sobre estas mismas. | | |
| **Campo** | **Tipo de dato** | **Llave** | **Descripción** |
| id\_venta\_detalles | SMALLINT | PK | Id de la tabla |
| id\_venta | SMALLINT | FK | Hace referencia a la PK de la tabla de ventas |
| id\_articulo | SMALLINT | FK | Hace referencia a la PK de la tabla de artículos |
| id\_empleado | SMALLINT | FK | Hace referencia a la PK de la tabla de empleados |
| cantidad | SMALLINT | - | Cantidad de artículos vendidos |
| precio\_unitario | INT | - | Precio de cada articulo |
| precio\_total | INT | - | Precio final de la cantidad y el precio\_unitario |

|  |  |  |  |
| --- | --- | --- | --- |
| **ingresos\_articulos** | | | |
| **Descripción** | Esta tabla sirve para saber que artículos fueron comprados a los proveedores por la empresa para su posterior venta. | | |
| **Campo** | **Tipo de dato** | **Llave** | **Descripción** |
| id\_ingreso | SMALLINT | PK | Id de la tabla |
| id\_comprobante | SMALLINT | FK | Hace referencia a la PK de la tabla de comprobantes |
| id\_tipo\_pago | SMALLINT | FK | Hace referencia a la PK de la tabla de tipos\_pagos |
| fecha\_ingreso | DATETIME | - | Fecha en la cual se hizo la compra de los artículos |

|  |  |  |  |
| --- | --- | --- | --- |
| **ingresos\_art\_detalles** | | | |
| **Descripción** | Esta tabla está relacionada con la tabla ingresos\_articulos, pero con más detalles sobre estas mismas. | | |
| **Campo** | **Tipo de dato** | **Llave** | **Descripción** |
| id\_ingreso\_det | SMALLINT | PK | Id de la tabla |
| id\_ingreso | SMALLINT | FK | Hace referencia a la PK de la tabla de ingresos\_articulos |
| id\_proveedor | SMALLINT | FK | Hace referencia a la PK de la tabla de proveedores |
| id\_articulo | SMALLINT | FK | Hace referencia a la PK de la tabla de artículos |
| cantidad | SMALLINT | - | Cantidad de artículos comprados |
| precio\_unitario | INT | - | Precio de cada articulo |
| precio\_total | INT | - | Precio final de la cantidad y el precio\_unitario con un descuento por proveedor. |

Acá también muestro las tablas que se crearon para utilizar en triggers y poder auditar las acciones que se realizan en dichas tablas

|  |  |  |  |
| --- | --- | --- | --- |
| **auditoria\_clientes** | | | |
| **Descripción** | Esta tabla fue creada específicamente para poder insertar datos de triggers de auditoria de cuando se realicen inserts y updates en la tabla clientes. Tienen campos importantes como “acción” que sirve para saber que se realizó, “tabla” para saber dónde se realizó, “usuario” para saber quien realizo dicha acción y “fecha y hora” para saber cuándo se hizo. | | |
| **Campo** | **Tipo de dato** | **Llave** | **Descripción** |
| id\_auditoria\_cl | SMALLINT | PK | Id de la tabla |
| nombre | VARCHAR (255) | - | Nombre del cliente |
| apellido | VARCHAR (170) | - | Apellido del cliente |
| dni | VARCHAR (80) | - | DNI del cliente |
| dirección | VARCHAR (100) | - | Dirección vieja y nueva del cliente (en caso de update) |
| teléfono | VARCHAR (90) | - | Teléfono nuevo y viejo del cliente (en caso de update) |
| email | VARCHAR (100) | - | Email nuevo y viejo del cliente (en caso de update) |
| acción | CHAR (6) | - | Se indica la acción que se lleva a cabo en la tabla |
| tabla | CHAR (8) | - | Se indica la tabla donde se lleva a cabo la acción |
| usuario | VARCHAR (50) | - | Se indica el usuario que llevo a cabo dicha acción |
| fecha | DATE | - | Fecha en la que se realizó la acción |
| hora | TIME | - | Hora en la que se realizó la acción |

|  |  |  |  |
| --- | --- | --- | --- |
| **auditoria\_articulos** | | | |
| **Descripción** | Esta tabla fue creada específicamente para poder insertar datos de triggers de auditoría de cuando se realizan updates y delete en la tabla artículos. Tienen campos importantes como “acción” que sirve para saber que se realizó, “tabla” para saber dónde se realizó, “usuario” para saber quién realizo dicha acción y “fecha y hora” para saber cuándo se hizo. | | |
| **Campo** | **Tipo de dato** | **Llave** | **Descripción** |
| id\_auditoria\_art | SMALLINT | PK | Id de la tabla |
| id\_articulo | SMALLINT | - | DNI del cliente |
| detalles | VARCHAR (255) | - | Detalles viejos y nuevos del articulo (en caso de delete) |
| precio | VARCHAR (170) | - | Precio nuevo y viejo del articulo (en caso de delete) |
| stock | VARCHAR (170) | - | Stock nuevo y viejo del articulo (en caso de delete) |
| acción | CHAR (6) | - | Se indica la acción que se lleva a cabo en la tabla |
| tabla | CHAR (9) | - | Se indica la tabla donde se lleva a cabo la acción |
| usuario | VARCHAR (50) | - | Se indica el usuario que llevo a cabo dicha acción |
| fecha | DATE | - | Fecha en la que se realizó la acción |
| hora | TIME | - | Hora en la que se realizó la acción |

# DESCRIPCIÓN DE VISTAS

**VW\_EMPLEADOS\_PUESTO:** El objetivo de esta vista es saber cuál es el trabajo que tienen los empleados del negocio.

Esta misma, muestra el nombre y apellido de los empleados y el puesto de trabajo.

Las tablas que se relacionan son las de empleados y la de puestos\_trabajo.

**VW\_EMPLEADOS\_VENTAS:** El objetivo de esta vista es saber qué empleado le vendió X producto a un cliente.

En esta se muestra el nombre y apellido del empleado, los detalles del producto, con la cantidad, el precio total, nombre y apellido del cliente y la fecha de la venta.

Esta misma está compuesta por la tabla de empleados, ventas\_detalles, artículos, ventas y clientes.

**VW\_RUBROS\_ARTICULOS:** Está vista sirve para poder ver el nombre de los artículos que se venden con su precio y su respectivo rubro de forma más ordenada.

Se van a mostrar el rubro de cada artículo, con los detalles y el precio de este mismo.

Está compuesta por la tabla artículos y la de rubros.

**VW\_PROVEEDORES\_ART:** Esta vista sirve para que muestre cuales son los rubros de los artículos que ingresa cada proveedor.

En esta se muestra el rubro de los artículos, los detalles de estos y cuál es el proveedor que ingresa cada artículo.

Está compuesta por la tabla de proveedores, artículos y la de rubros.

**VW\_ARTICULOS\_MAS\_VENDIDOS:** Está vista es para saber cuáles son los 10 artículos más vendidos del negocio ordenado de forma descendente.

En esta vista se muestran los detalles de los artículos y la cantidad de veces que fueron vendidas.

Las tablas que están relacionadas con la vista son la de ventas\_detalles y la de artículos.

**VW\_INGRESOS\_PROVEEDORES:** Esta vista muerta el nombre del proveedor y el precio total que la empresa pago para los ingresos de todos los artículos.

Esta misma muestra el nombre del proveedor y el precio total que se gastó en todas las compras.

En esta vista uso la tabla ingresos\_art\_detalles y la de proveedores.

# DESCRIPCION DE FUNCIONES

**FN\_VENTAS\_HECHAS:** Esta función es para saber las ventas totales que se hicieron en el negocio.

En esta se le debe indicar el parámetro numérico 0 ya que hace referencia al id de venta y se fija todos los números mayores al 0 que introducimos y ver las ventas totales hechas.

Esta función interactúa con la tabla ventas.

**FN\_CANTIDAD\_MARCA:** Esta función sirve para contar el total de la cantidad de artículos que se venden, pero con un determinado nombre.

Previamente se le debe indicar a la entrada el nombre de la marca que se busca para ver el resultado. La función se va a encargar de buscar el pedazo de nombre de la marca que se introduzco para buscar los artículos.

Esta función interactúa con la tabla de artículos.

**FN\_CALCULAR\_SALARIO:** El objetivo de esta función es poder decir si el salario de los empleados es bajo, medio o alto a través del valor que le indiquemos al parámetro.

Se le debe indicar al parámetro numérico el salario que le darían a un empleado así la función les da el resultado.

**FN\_CLIENTES:** El fin de esta función es contar los clientes totales que compraron en nuestro negocio.

En esta se le debe indicar el parámetro numérico 0 ya que hace referencia al id de clientes y se fija todos los números mayores al 0 que introducimos y ver los clientes totales que tiene el negocio.

Esta función interactuar con la tabla clientes.

**FN\_VENTAS:** El objetivo de la función es a través de las entradas insertar datos a la tabla de ventas sin tener que hacer el insert y los valores.  
A las entradas que tienen esta función se le tiene que insertar el (id\_cliente, id\_comprobante y el id\_tipo\_pago) y esta se encarga de insertarlas en la tabla de ventas con la fecha actual.

**FN\_ENTRADA:** Esta función va de la mano con un stored procedure. Sirve para que se declare error en caso de que la variable esta vacía.

Se le debe indicar algún valor a las entradas del stored procedure para que esta función de okey o si las entradas están vacías de error.

# DESCRIPCION DE STORED PROCEDURES

**SP\_ARTICULOS\_ORDEN:** Este stored procedure sirve para ordenar por la columna que elija, los artículos que vendí de forma ascendente o descendente con una consulta que me permite saber el stock de forma actualizada de los productos.

Para esto tengo dos parámetros que son (p\_orden y p\_asc\_desc) en el cual en el primero elijo la columna por la cual quiero ordenar y en la segunda de qué forma ordenarlo. En el caso de que no se ingrese nada en los parámetros la función fn\_entrada daría como error a este stored procedure.

Este SP interactúa con la tabla de artículos, ventas\_detalles y con ingreso\_art\_detalles.

**SP\_CLIENTES:** En este SP lo hice para poder insertar o hacer delete en la tabla de clientes a través de parámetros de entrada.

En este procedure declaro 7 parámetros de entrada (p\_nombre, p\_apellido, p\_dni, p\_direccion, p\_telefono, p\_email y p\_opcion). Si en p\_opcion pongo “i” me hace un insert a la tabla, donde a los valores que tengo que ingresar a la tabla serían los otros parámetros de entradas para poder colocar los datos después. Si en la p\_opcion pongo “d” me hace un delete a la tabla donde el nombre y apellido de la tabla sean iguales al que se ingresa en los parámetros p\_nombre y a p\_apellido.

Este SP interactúa con la tabla de clientes.

**SP\_NOMBRE\_CLIENTE:** Este stored procedure sirve para ver el nombre y apellido de un cliente a través de su id propio que se ingresa en un parámetro de entrada (p\_id\_cliente).

Este SP interactúa con la tabla de clientes.

**SP\_STOCK:** Este stored procedure sirve para obtener los detalles de algunos artículos en específico poniendo una palabra o letra que contenga ese articulo en un parámetro de entrada llamado p\_articulo.

En este SP se trabaja con la tabla de artículos.

# DESCRIPCION DE TRIGGERS

Para empezar para hacer los triggers cree dos nuevas tablas, llamadas auditoria\_articulos y otra aditoria\_clientes que las explique en la descripción de tablas.

**TRG\_LOG\_AUDI\_ART\_UPD:** Al crear este trigger lo hice con el fin de que se guarde después en la tabla de auditoria\_articulos todos los update que se hagan en la tabla de artículos. Entonces a la hora en que se hace un update a la tabla de artículos se inserten a la tabla de auditoría los valores como el id del artículo viejo, los detalles viejos, el precio viejo y el nuevo a través de una concatenación, como también el stock nuevo y viejo, la acción que se tuvo en cuenta (en este caso update), en la tabla artículos, porque usuario fue hecha y que día y que hora se realizó dicha acción.

**TRG\_LOG\_AUDI\_ART\_DEL:** Este trigger básicamente es igual que el anterior, pero en este se inserta también a la tabla de auditoria\_articulos cuando se realiza una acción de delete a la tabla de artículos. Básicamente se insertan todos los campos viejos que tenía esa tabla, pero en la acción en vez de update es delete, el usuario que realizó la acción y la fecha y hora.

**TRG\_LOG\_AUDI\_CL\_INSERT:** Este trigger sirve para que se guarde en la tabla de auditoria\_clientes todos los insert que se hagan en la tabla de clientes. Esto se logra a través de un insert hacia la tabla de auditoría todos los nuevos valores que se agreguen a la tabla, pero en la acción sea insert, en la tabla clientes y el usuario que realizó la acción con la fecha y hora.

**TRG\_LOG\_AUDI\_CL\_UPD:** Este trigger sirve para que se guarde en la tabla auditoria\_clientes todos los updates que se hagan en la tabla clientes. Esto se logra a través de un insert a la tabla de auditoría cuando se realice el update con el nuevo nombre, nuevo apellido, nuevo dni, de forma concatenada la dirección vieja y nueva, el teléfono viejo y nuevo, el email viejo y nuevo, en la acción con el valor de UPDATE, en la tabla clientes y el usuario que realizó dicha acción con la fecha y hora.

# CREACION DE USUARIOS DCL

Todos estos usuarios solo tienen privilegios sobre la base de datos proyectocoder.

**Aprendiz:** Este usuario tiene como finalidad poder hacer **solo lecturas** de las tablas de artículos, clientes, rubros y proveedores ya que es un aprendiz. A medida que tenga más experiencia se le da otros privilegios.

**Novato:** Este usuario al contar con más experiencia tiene como finalidad poder hacer **solo lecturas, inserción y updates** de datos las tablas de artículos, clientes, rubros y proveedores. Pero todavía no tiene la experiencia suficiente para poder ver todas las tablas o cambiar los datos de estas.

**Experto:** Este usuario ya tiene la experiencia necesaria, por lo tanto, puede hacer lecturas, inserción, updates y delete de datos y crear y eliminar tablas.

**Admin:** Este usuario tiene todos los privilegios en toda la base de datos de proyectocoder

# 

# CONCLUCIÓN

Yo creo que si tendría que volver a hacer otro proyecto elegiría el mismo ya que me gusto bastante trabajar sobre esta temática, intente desarrollarlo de la mejor manera posible a mi parecer. Aprendí mucho sobre como sería una base de datos de un negocio aun que seguro me falten cosas y mejorar en otras. A su vez me gustaría hacerlo de otra temática que no sea la base de datos de un negocio para poder practicar con un tema diferente al que vine trabajando todo este tiempo y tener una visión más general de a todas las cosas que se le podría aplicar en una base de datos.

# HERRAMIENTAS UTILIZADAS

Para la creación y la gestión de la base de datos fue MySQL Workbench.

Para la generación del informe fue la herramienta de visualización Power BI.