



# Proyecto Final

---

Curso: **SQL**

Comisión: **47375**

Docente: **Redondo, Camilo Andres**

Tutor designado: **Fernando Giraudo**

Alumno: **Leandro Barrios**

# ÍNDICE

INTRODUCCIÓN.....	3
OBJETIVO.....	3
MODELO DE NEGOCIO.....	4
SITUACIÓN PROBLEMÁTICA.....	4
HERRAMIENTAS IMPLEMENTADAS.....	4
DIAGRAMA ENTIDAD RELACIÓN (DER).....	5
TABLAS.....	7
VISTAS.....	11
FUNCIONES.....	12
PROCEDIMIENTOS.....	14
CONCLUSIÓN.....	16



## INTRODUCCIÓN

El proyecto final VX consiste en la creación de una base de datos relacional utilizando el lenguaje MySQL, la finalidad es tanto el ingreso de datos limpios para acceder bajo demanda así como la creación de procesos y funciones que permitan obtener información relevante. Se inició con la idea de actualizar una base de datos para el control de un comercio ya existente (<https://www.venex.com.ar/>) dedicado a la comercialización de productos informáticos.

## OBJETIVO

Con la creación de este proyecto tiene como finalidad que pueda ser utilizado y gestionado por usuarios sin experiencia mediante un futuro software creado a medida para la comercialización de productos informáticos. La misma está compuesta de:

- Estructuras para el ingreso y almacenamiento de datos.
- Creación de usuarios con distintos niveles de acceso.
- Vinculación de registros para identificar una gestión en sus etapas.
- Copia de seguridad sobre datos sensibles.
- Registro de modificaciones realizadas a la misma.
- Programación de aplicaciones para filtrar datos.

# VX

## MODELO DE NEGOCIO

El modelo de negocio de “VX” es una tienda de productos informáticos con puntos de retiros físicos ubicados en distintas locaciones de la Argentina, cuenta con su propio ecommerce y también ofrece productos mediante otras plataformas. No solo vende artículos relacionados a la computación sino que ofrecen una amplia gama de productos así como servicios o atenciones diferenciales a la competencia.

## SITUACIÓN PROBLEMÁTICA

Actualmente “VX” cuenta ya con una base de datos y un software de gestión creado en un lenguaje de programación desactualizado, esto provoca distintos inconvenientes a la hora de implementar mejoras o actualizaciones. Por lo tanto la idea es desarrollar en paralelo una nueva versión que permita la mantención a largo plazo con características más funcionales y sumar esta misma a un software desarrollado en un lenguaje con mucho futuro por delante, para así llevar una ventaja competitiva frente a un mercado muy reñido.

## HERRAMIENTAS IMPLEMENTADAS

### MySQL Workbench 8.0.34:

Creación, administración y mantenimiento sobre la base de datos relacional.

### Google Slides:

Creación de presentación sobre proyecto.

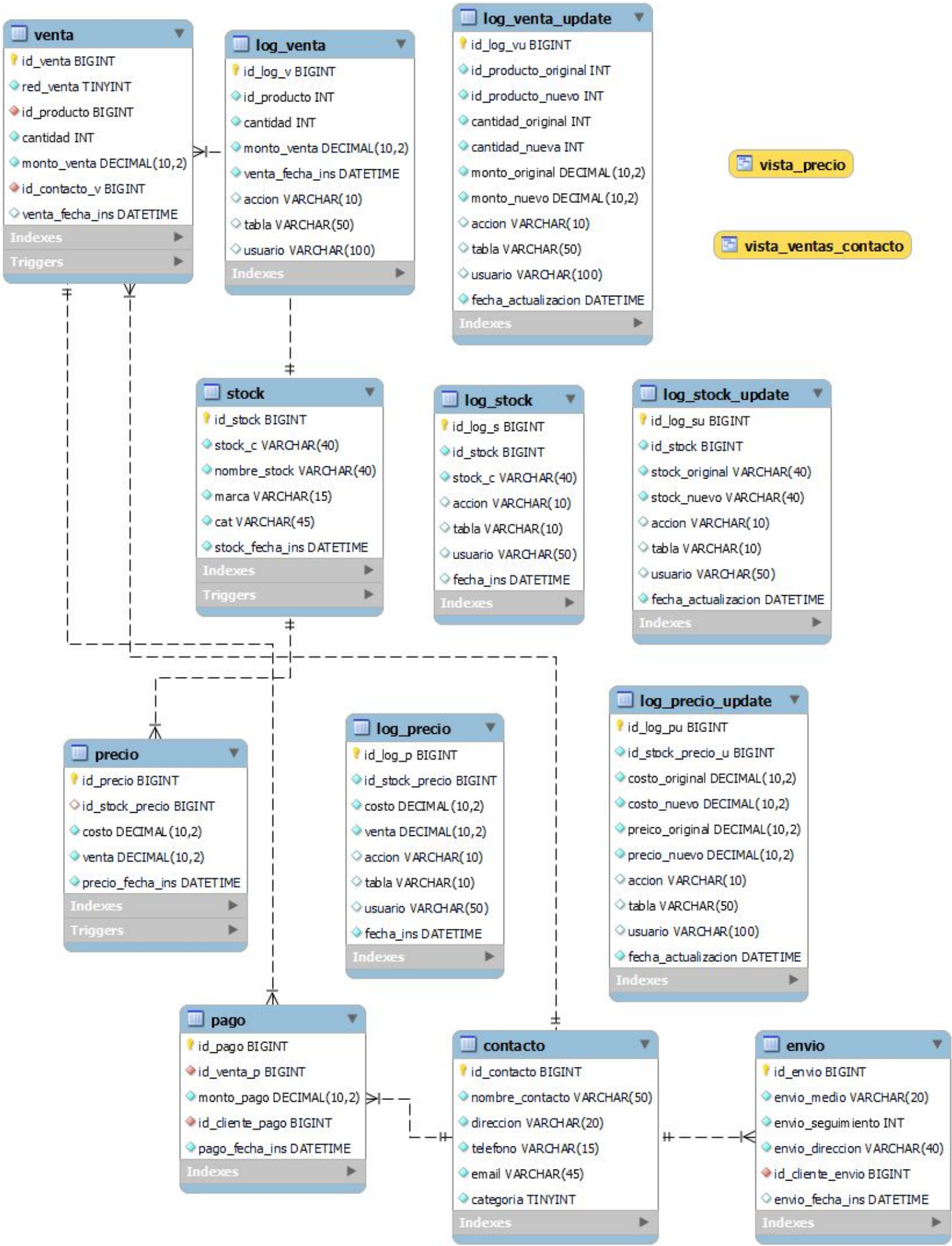
### Google Docs:

Creación de manual para la inserción y implementación de base relacional.

# DIAGRAMA ENTIDAD RELACIÓN

**VX**

—



# DESCRIPCIÓN DE TABLAS

## STOCK

**Stock:** En esta tabla se ingresan todos los productos los cuales van a ser comercializados más adelante indicando distintos datos como cantidad y categorías.

**id\_stock:** identificador de producto

**stock\_c:** cantidad de producto

**nombre\_stock:** nombre de producto

**marca:** marca de producto

**cat:** categoría de producto

**stock\_fecha\_ins:** fecha carga de producto

**Log\_Stock:** Esta tabla se encarga de realizar una copia de seguridad sobre los productos ingresados en la tabla stock.

**id\_log\_s:** identificador de producto secundario

**id\_stock:** copia de identificador producto

**stock\_c:** copia sobre cantidad en stock

**accion:** acción realizada sobre producto en database

**tabla:** nombre de la tabla donde se realiza la acción

**usuario:** usuario quien realizo la acción

**fecha\_ins:** fecha que se realizó la acción

**Log\_Stock\_Update:** Esta tabla se encarga de realizar una copia de seguridad sobre las actualizaciones o eliminaciones de productos ingresados en la tabla stock.

**id\_log\_su:** identificador de producto terciario

**id\_stock:** copia de identificador producto

**stock\_original:** cantidad previa a modificación

**stock\_nuevo:** cantidad post modificación

**accion:** acción realizada sobre producto en database

**tabla:** nombre de la tabla donde se realiza la acción

**usuario:** usuario quien realizo la acción

**fecha\_actualizacion:** fecha que se realizó la acción

# DESCRIPCIÓN DE TABLAS

## PRECIO

**Precio:** En esta tabla se ingresan los valores de productos relacionados a la tabla stock.

**id\_precio:** identificador de unidad

**id\_stock\_precio:** identificador de producto

**costo:** costo real de producto

**venta:** precio de venta al público

**precio\_fecha\_ins:** fecha carga precio

**Log\_Precio:** Esta tabla se encarga de realizar una copia de seguridad sobre los precios ingresados a productos.

**id\_log\_p:** identificador primario

**id\_stock\_precio:** identificador referente a tabla precio

**costo:** costo real de producto

**venta:** precio de venta al público

**accion:** acción realizada sobre producto en database

**tabla:** nombre de la tabla donde se realiza la acción

**usuario:** usuario quien realizo la acción

**fecha\_ins:** fecha carga precio

**Log\_Precio\_Update:** Esta tabla se encarga de realizar una copia de seguridad sobre las actualizaciones o eliminaciones de precios ingresados en la tabla precio.

**id\_log\_pu:** identificador de producto en tabla precio

**id\_stock\_precio\_u:** copia de identificador precio

**costo\_original:** costo de compra previa a modificación

**costo\_nuevo:** costo de compra post modificación

**venta\_original:** monto de venta previa a modificación

**venta\_nuevo:** monto de venta post modificación

**accion:** acción realizada sobre precio en database

**tabla:** nombre de la tabla donde se realiza la acción

**usuario:** usuario quien realizo la acción

**fecha\_actualizacion:** fecha que se realizó la modificacion



# DESCRIPCIÓN DE TABLAS

## VENTA

**Venta:** En esta tabla se ingresan las compras realizadas por clientes.

**id\_venta:** identificador de compra

**red\_venta:** medio utilizado para comprar

**id\_producto:** producto adquirido

**cantidad:** cantidad de unidades seleccionadas

**monto\_venta:** costo a pagar por cliente

**id\_contacto\_v:** identificador de cliente en base de datos

**venta\_fecha\_ins:** fecha y hora de venta

**Log\_Venta:** Esta tabla se encarga de realizar una copia de seguridad sobre las ventas realizadas.

**id\_log\_v:** identificador primario

**id\_producto:** producto adquirido

**cantidad:** cantidad de unidades seleccionadas

**monto\_venta:** costo a pagar por cliente

**venta\_fecha\_ins:** fecha y hora de venta

**accion:** acción realizada por usuario en database

**tabla:** nombre de la tabla donde se realiza la acción

**usuario:** usuario quien realizo la acción

**Log\_Venta\_Update:** Esta tabla se encarga de realizar una copia de seguridad sobre las actualizaciones o eliminaciones de ventas.

**id\_log\_vu:** identificador de actualización sobre venta en base de datos

**id\_producto\_original:** identificador de producto pre modificación

**id\_producto\_nuevo:** identificador de producto modificado

**cantidad\_original:** cantidad de productos pre modificación

**cantidad\_nueva:** cantidad de productos post modificación

**monto\_original:** monto de venta pre modificación

**monto\_nuevo:** monto de venta post modificación

**accion:** acción realizada sobre precio en database

**tabla:** nombre de la tabla donde se realiza la acción

**usuario:** usuario quien realizo la acción

**fecha\_actualizacion:** fecha que se realizó la modificación

# DESCRIPCIÓN DE TABLAS

## CONTACTO

**Contacto:** En esta tabla se ingresan los datos correspondientes a clientes y proveedores.

**id\_contacto:** identificador de cliente/prov

**nombre\_contacto:** nombre de cliente/prov

**dirección:** dirección personal

**teléfono:** medio de contacto

**email:** email de contacto

**categoría:** tipo de cliente, consumidor o proveedor

## PAGO

**Pago:** Esta tabla se encarga de almacenar los montos correspondientes a ventas realizadas y vincularlos con tabla contacto.

**id\_pago:** identificador de pago primario

**id\_venta\_p:** identificador de venta sobre tabla venta

**monto\_pago:** valor abonado por cliente

**id\_cliente\_pago:** identificador de cliente sobre tabla contacto

**pago\_fecha\_ins:** fecha de carga sobre pago

## ENVÍO

**Envio:** Esta tabla se ingresan los datos y el medio para luego el despacho de productos vendidos.

**id\_envio:** identificador de envío

**envio\_medio:** medio por el cual se despacha

**envio\_seguimiento:** número de seguimiento paquete

**envio\_direccion:** domicilio destinatario

**id\_cliente\_envio:** identificador de cliente sobre envío

**envio\_fecha\_ins:** fecha que se realizó el despacho

# VISTAS

**Vista\_Precio:** Se utiliza exclusivamente para visualizar el precio y características de un producto en particular o la lista completa.

**Vista\_Ventas\_Contacto:** Una herramienta creada para la visualización de ventas concretadas por contacto en particular.

---

FUNCIONES

## **FN\_TOTAL\_VENTAS\_POR\_CONTACTO**

Esta funcion es encargada de sumar la totalidad de ingresos concretados por ventas generadas a un cliente en particular que se desee filtrar.

Requerimientos para uso:

- Ingresar el identificador primario del contacto

## **FN\_OBTENER\_ID\_PRODUCTO**

### **FN\_CALCULADORA\_GANANCIA\_VENTA**

Una sumatoria de dos funciones con un único fin, obtener la ganancia neta restando los costos correspondientes ante la venta de un producto.

Requerimientos para uso:

- Ingresar el identificador primario de producto

### **FN\_CALCULADORA\_GANANCIA\_VENTA**

Funcion que facilita el conteo de stock disponible en base a uno solicitado por un producto en particular.

Requerimientos para uso:

- Ingresar el identificador primario de producto
- Ingresar cantidad solicitada

# PROCEDIMIENTOS

—

## **SP\_ACTUALIZAR\_STOCK**

Este procedimiento se encarga de actualizar el stock disponible sobre un producto en particular, al mismo tiempo dejando registro sobre el cambio realizado.

Requerimientos para uso:

- Ingresar el identificador primario del producto en stock
- Ingresar la cantidad nueva a modificar de stock

## **SP\_ACTUALIZAR\_MONTO\_PAGO\_VENTA**

Procedimiento encargado de actualizar el monto abonado por el cliente sobre una venta concreta, dejando a su vez el registro correspondiente sobre el cambio realizado.

Requerimientos para uso:

- Ingresar el identificador primario de la venta
- Ingresar el nuevo monto abonado

## **SP\_ELIMINAR\_CONTACTO\_PAGO\_VENTA**

Gestión encargada de eliminar por completo el registro de tanto la venta como pagos y el cliente en si sobre la base de datos.

Requerimientos para uso:

- Ingresar el identificador primario del contacto a eliminar



# CONCLUSIÓN

---

Desde el inicio del cursado tenía en claro que buscaba con este proyecto, ya que era una idea personal que tenía sobre mi lugar de trabajo actualmente (Venex). A medida que fue corriendo el curso me di cuenta que era una correcta elección, ya sea para aprendizaje únicamente o una futura implementación, los conocimientos adquiridos en este ejemplo me fueron más que útiles para aprender la herramienta MySQL. Volvería a elegir este proyecto sin lugar a dudas y me gustaría seguir expandiendo aprendiendo otros lenguajes como Python. Agradezco al profesor y los tutores que me acompañaron en todo el proceso.