



**Proyecto Final - Curso SQL**  
**Ryu-Technology E-commerce**

Luiggi Marquez  
Profesor: Camilo Redondo  
Comisión 31275

Buenos Aires, septiembre de 2022

## Índice

Introducción .....	3
Objetivo .....	3
Situación problemática .....	4
Modelo de negocio .....	4
Diagrama de entidad de relación .....	5
Listado de tablas .....	6
Objetos de la base de datos .....	9
Inserción de datos .....	11
Informes generados .....	12
Información adicional .....	15
Herramientas utilizadas .....	16

## **Introducción**

Una base de datos relacional es una recopilación de elementos de datos con relaciones predefinidas entre ellos, en donde se organizan como un conjunto de tablas con columnas y filas. Las tablas se utilizan para guardar información sobre los objetos que se van a representar en la base de datos. Cada columna de una tabla guarda un determinado tipo de datos y un campo almacena el valor real de un atributo; las filas de la tabla representan una recopilación de valores relacionados de un objeto o una entidad. SQL o lenguaje de consulta estructurada es la interfaz principal utilizada para comunicarse con bases de datos relacionales; se utiliza para agregar, actualizar o eliminar filas de datos, por lo que se recuperan subconjuntos de datos para aplicaciones de análisis y procesamiento de transacciones, y para administrar todos los aspectos de la base de datos.

El siguiente proyecto abarca la creación de una base de datos relacional sobre un e-commerce ficticio llamado Ryu Technology, que consiste en un website dedicado a la venta de productos que trabajan bajo tecnología TCP/IP tales como cámaras de seguridad, routers, switches, firewalls. A través de este informe se especifica como está basada la base de datos, cuáles son sus componentes y se explica su importancia para el funcionamiento del sitio.

## **Objetivo**

Realizar el diseño e implementación de una base de datos relacional para un e-commerce que permita el almacenamiento análisis y uso de los datos para su óptimo funcionamiento.

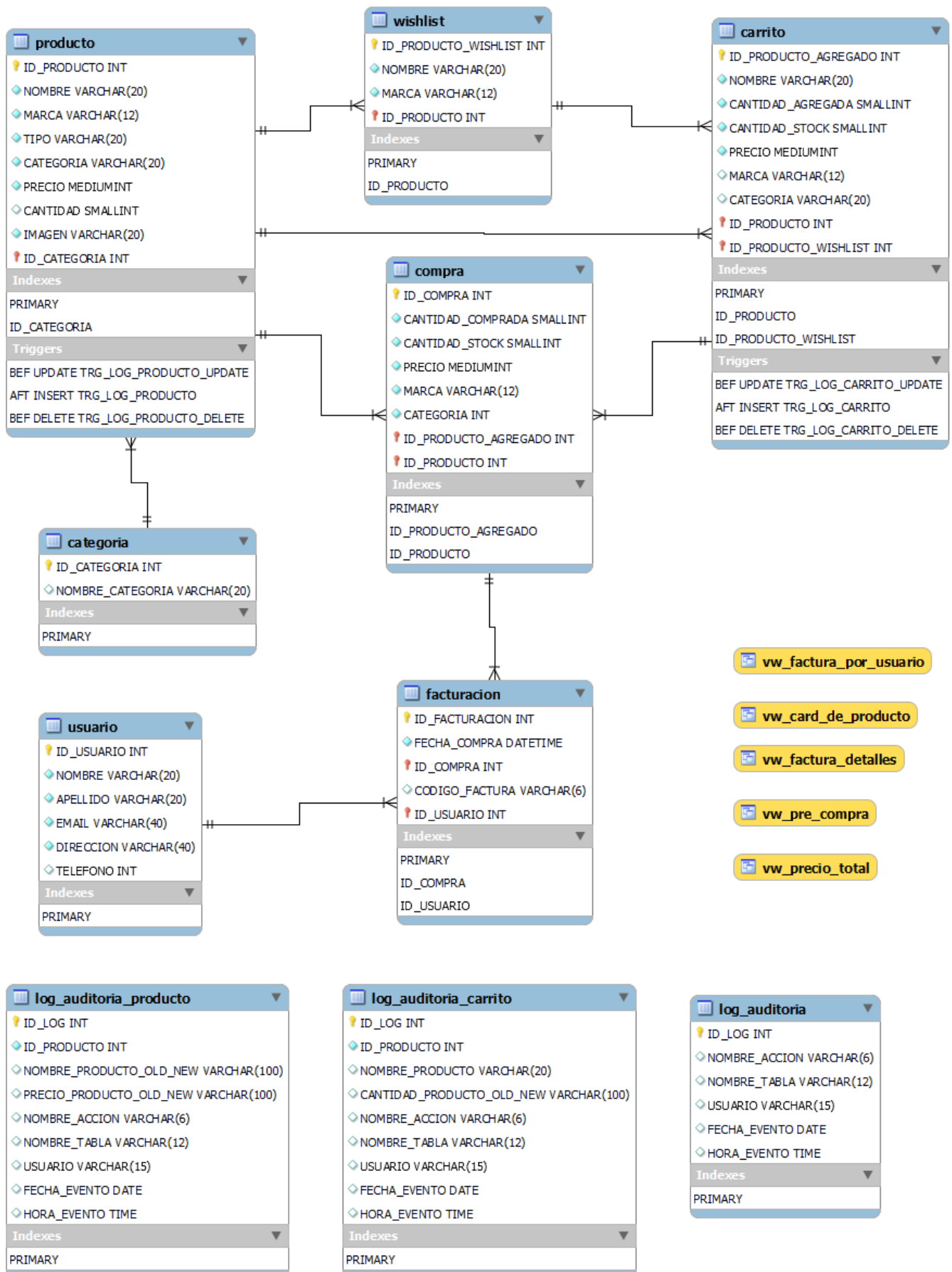
## **Situación problemática**

La ejecución del Front-end para un E-commerce basado en React.js requiere que las variables manejadas por su lenguaje de programación, Javascript, sean almacenadas en una base de datos, para que cada vez que se acceda al website, los elementos sean cargados y mostrados al usuario: menús, imágenes, productos, entre otros. Por otra parte, se necesita almacenar temporalmente los datos del carrito de compras del sitio, en caso de que el usuario cambie de página o se desconecte y lo mismo sucede con la lista de deseos (favoritos o wishlist). Otro aspecto importante, es tener un registro de las compras realizadas y los datos del usuario, para la parte legal, fiscal y por si el usuario desea consultar en futuro las compras realizadas en la tienda. Se necesita de una base de datos para poner en funcionamiento la página web y para realizar las actividades de la misma, en este caso, realizar compra de productos de modo online y llevar registro de la información financiera.

## **Modelo de Negocio**

Ryu Technology es un e-commerce, es decir, su modelo es la práctica de comprar y vender productos a través de internet; es una tienda que se dedica a este negocio. Este sector ha revolucionado el comercio tradicional ofreciendo un nivel de comodidad y personalización sin precedentes: por ello, el comercio electrónico se ha convertido en la base del modelo de negocio para muchísimas marcas porque han supuesto toda una revolución en la manera de vender, y pueden adaptarse a todo tipo de sectores. Para este caso en particular, Ryu Technology está dirigido directamente al consumidor final ofreciendo tecnología de punta del sector de vigilancia y comunicaciones a nivel digital y está apoyado en las principales ventajas del ecommerce: mayor alcance ( al ser tienda online, está automáticamente accesible desde todo el mundo), representa menores costes (montar una tienda online tiene un coste inicial muy bajo comparado con una tienda física) y está disponible 24/7.

## Diagrama de entidad de relación



## Listado de tablas

- **Tabla categoría:** la función de esta tabla es agregar dinámicamente en el menú de la web, en el Front-end, las categorías existentes para no estar modificando el código HTML cada vez que se agregue una categoría nueva y también cumple la función de limitar las categorías que se pueden agregar en la descripción de los productos. Se refiere a las diferentes clases de productos, por ejemplo, routers, switches, entre otros.

Nombre de Campo	Descripción	Tipo de Dato	Tipo de Clave
ID_CATEGORIA	Identificación única de la tabla	INT	PRIMARY KEY
NOMBRE_CATEGORIA	Nombre de las categorías de los productos	VARCHAR(20)	-

- **Tabla usuario:** esta tabla contiene los datos del usuario para generar la factura (ID, nombre, apellido, mail, dirección) y con su ID, consultar a futuro todas las compras realizadas.

Nombre de Campo	Descripción	Tipo de Dato	Tipo de Clave
ID_USUARIO	Identificación única de la tabla, su valor es el DNI del usuario	INT	PRIMARY KEY
NOMBRE	Nombre del cliente	VARCHAR(20)	
APELLIDO	Apellido del cliente	VARCHAR(20)	
EMAIL	Email del cliente	VARCHAR(40)	
DIRECCION	Dirección del cliente	VARCHAR(40)	
TELEFONO	Telefono del cliente	INT	

- **Tabla producto:** contiene el stock de todos los productos de la tienda; de aquí se descuentan la cantidad que haya sido comprada y se cargan los datos de los artículos mostrados en el front-end. Las variables representan todas las características de los productos vendidos

Nombre de Campo	Descripción	Tipo de Dato	Tipo de Clave
ID_PRODUCTO	Identificación única de la tabla	INT	PRIMARY KEY
NOMBRE	Nombre del producto	VARCHAR(20)	
MARCA	Marca del producto	VARCHAR(12)	
TIPO	Modelo de cada producto	VARCHAR (20)	
CATEGORIA	Clases de productos vendidos	VARCHAR(20)	
PRECIO	Valor de venta del producto	MEDIUMINT	
CANTIDAD	Número de productos disponibles	SMALLINT	
IMAGEN	Ruta de la imagen en el servidor	VARCHAR(20)	
ID_CATEGORIA	Valor que hace referencia a la tabla Categoría	INT	FOREIGN KEY

- **Tabla wishlist:** almacena productos que pueden ser agregados en un futuro al carrito de compras. Asegura que los datos persistan en el front-end para el usuario

Nombre de Campo	Descripción	Tipo de Dato	Tipo de Clave
ID_PRODUCTO_WISHLIST	Identificación única de la tabla	INT	PRIMARY KEY
NOMBRE	Nombre del producto agregado al wishlist	VARCHAR(20)	
MARCA	Marca del producto	VARCHAR(12)	
ID_PRODUCTO	Valor que hace referencia a la tabla Producto	INT	FOREIGN KEY

- **Tabla carrito:** contiene la información de los productos seleccionados por el usuario para ser comprados. Asegura que los datos persistan en el front-end para el usuario para realizar la compra después.

Nombre de Campo	Descripción	Tipo de Dato	Tipo de Clave
ID_PRODUCTO_AGREGADO	Identificación única de la tabla	INT	PRIMARY KEY
NOMBRE	Nombre del producto agregado al carrito de la página	VARCHAR(20)	
CANTIDAD_AGREGADA	Cantidad seleccionada por el usuario para comprar	SMALLINT	
CANTIDAD_STOCK	Cantidad total de los productos en stock	SMALLINT	
PRECIO	Valor de venta del producto	MEDIUMINT	
MARCA	Marca del producto	VARCHAR(12)	
CATEGORIA	Clases de productos vendidos	VARCHAR(20)	
ID_PRODUCTO	Valor que hace referencia a la tabla Producto	INT	FOREIGN KEY
ID_PRODUCTO_WISHLIST	Valor que hace referencia a la tabla Wishlist	INT	FOREIGN KEY

- **Tabla compra:** contiene el contenido de la compra final realizada por el usuario.

Nombre de Campo	Descripción	Tipo de Dato	Tipo de Clave
ID_COMPRA	Identificación única de la tabla	INT	PRIMARY KEY
CANTIDAD_COMPRADA	Cantidad del producto comprado	SMALLINT	
CANTIDAD_STOCK	Cantidad en el stock de productos	SMALLINT	
PRECIO	Valor de los productos comprados	MEDIUMINT	
MARCA	Marca de los productos comprados	VARCHAR(12)	
CATEGORIA	Clases de productos vendidos	INT	
ID_PRODUCTO_AGREGADO	Valor que hace referencia a la tabla carrito	INT	FOREIGN KEY
ID_PRODUCTO	Valor que hace referencia a la tabla Producto	INT	FOREIGN KEY

- **Tabla facturacion:** posee todos los registros de compras realizados por usuario.

Nombre de Campo	Descripción	Tipo de Dato	Tipo de Clave
ID_FACTURACION	Identificación única de la tabla	INT	PRIMARY KEY
FECHA_COMPRA	Fecha de la compra	DATETIME	
CODIGO_FACTURA	Identificador de la factura	VARCHAR(6)	
ID_COMPRA	Valor que hace referencia a la tabla Compra	INT	FOREIGN KEY
ID_USUARIO	Valor que hace referencia a la tabla Usuario	INT	FOREIGN KEY

- **Tabla log\_auditoria:** en esta tabla se registran los eventos que ocurren cuando se activa los triggers TRG\_LOG\_PRODUCTO y TRG\_LOG\_CARRITO, que monitorean cuando se insertan nuevos artículos en la tabla Producto y la tabla Carrito respectivamente.

Nombre de Campo	Descripción	Tipo de Dato	Tipo de Clave
ID_LOG	Identificación única de la tabla	INT	PRIMARY KEY
NOMBRE_ACCION	Nombre de la operación realizada: update, delete, insert	VARCHAR(6)	
NOMBRE_TABLA	Nombre de la tabla monitoreada	VARCHAR(12)	
USUARIO	Usuario que realiza la acción	VARCHAR(15)	
FECHA_EVENTO	Fecha que ocurrió la acción	DATE	
HORA_EVENTO	Hora que ocurrió la acción	TIME	

- **Tabla log\_auditoria\_producto:** en esta tabla se registran los eventos de suceden cuando se activan loss triggers TRG\_LOG\_PRODUCTO\_UPDATE y TRG\_LOG\_PRODUCTO\_DELETE cuando se varían con UPDATE la tabla Producto o se borra con DELETE, es decir, se estará auditando los valores de NOMBRE\_PRODUCTO y PRECIO\_PRODUCTO almacenando los valores anteriores y los nuevos valores si es que se produce cambio, así como el usuario, la fecha y la hora del evento.

Nombre de Campo	Descripción	Tipo de Dato	Tipo de Clave
ID_LOG	Identificación única de la tabla	INT	PRIMARY KEY
ID_PRODUCTO	ID del producto modificado	INT	
NOMBRE_PRODUCTO_OLD_NEW	Nombre anterior y/o nuevo si se modificó	VARCHAR(100)	
PRECIO_PRODUCTO_OLD_NEW	Precio anterior y/o nuevo si se modificó	VARCHAR(100)	
NOMBRE_ACCION	Nombre de la operación realizada: update, delete, insert	VARCHAR(6)	
NOMBRE_TABLA	Nombre de la tabla monitoreada	VARCHAR(12)	
USUARIO	Usuario que realiza la acción	VARCHAR(15)	
FECHA_EVENTO	Fecha que ocurrió la acción	DATE	
HORA_EVENTO	Hora que ocurrió la acción	TIME	



- **Tabla log\_auditoria\_carrito:** registra los eventos capturados por los triggers TRG\_LOG\_CARRITO\_UPDATE y TRG\_LOG\_CARRITO\_DELETE, que está pendiente de cambios en la tabla Carrito en cuando actualizaciones o borrado.

Nombre de Campo	Descripción	Tipo de Dato	Tipo de Clave
ID_LOG	Identificación única de la tabla	INT	PRIMARY KEY
ID_PRODUCTO	ID del producto modificado	INT	
NOMBRE_PRODUCTO	Nombre del producto modificado	VARCHAR(20)	
CANTIDAD_PRODUCTO_OLD_NEW	Cantidad anterior y/o nueva si se modificó	VARCHAR(100)	
NOMBRE_ACCION	Nombre de la operación realizada: update, delete, insert	VARCHAR(6)	
NOMBRE_TABLA	Nombre de la tabla monitoreada	VARCHAR(12)	
USUARIO	Usuario que realiza la acción	VARCHAR(15)	
FECHA_EVENTO	Fecha que ocurrió la acción	DATE	
HORA_EVENTO	Hora que ocurrió la acción	TIME	

## Scripts de creación de cada objeto de la base de datos

**Vistas:** El proyecto posee cinco vistas:

- **Vista VW\_PRE\_COMPRA:** esta vista está formada por la unión de campos de la tabla Carrito y la tabla Wishlist. Tiene como objetivo mostrar cual sería la compra total de haberse seleccionado los elementos del wishlist. Los elementos que tienen id\_producto\_wishlist tienen en valor en NULL cuando han sido agregados directamente de la tabla Producto, para referenciar de donde proviene cada artículo. Aquellos agregados de la tabla Wishlist tienen su valor correspondiente.
- **Vista VW\_FACTURA\_DETALLES:** esta vista está formada por la unión de las tablas Compra, Carrito y Facturacion, con los campos detallados de los productos adquiridos recientemente, tomados de la tabla Compra, el id de los productos tomado de la tabla Carrito y los datos del usuario tomado de Facturacion. Su objetivo es mostrar los datos necesario para hacer la factura de la compra.
- **Vista VW\_FACTURA\_POR\_USUARIO:** esta vista tiene como objetivo listar todas las facturas que tenga registradas los usuarios a modo histórico. Está formada por los elementos de las tablas Usuario Y Facturacion.

- **Vista VW\_CARD\_DE\_PRODUCTO:** esta vista agrupa los campos de la tabla Producto que se visualizarán en el Front-End en el card detallado de cada producto. Su objetivo es armar los detalles del producto que aparecerán en la página.
- **Vista VW\_PRECIO\_TOTAL:** esta vista tiene el objetivo de mostrar el costo de cada producto del stock, sin IVA y costo final con IVA. Está formada por la tabla Producto.

**Store procedure:** consta de dos procedimientos:

- **Procedimiento almacenado SP\_ORDENAR:** ordena ascendente o descendente dependiendo del valor elegido en ORDEN\_ASCENDENTE\_DESCENDENTE en la columna designada en el parámetro PARAMETRO\_POR\_ORDENAR, aplicado en la tabla Producto. Su uso simplifica búsqueda y consulta de valores en el stock del e-commerce, bien sea para actualizar, corregir y/o borrar datos.
- **Procedimiento almacenado SP\_INSERT\_DELETE\_PRODUCT\_CATEGORY:** es de vital importancia en el Front-end, porque alimenta dinámicamente el menú del navbar en el header del sitio, es decir, el website cambiará automáticamente el menú dependiendo del contenido de la tabla Categoría. El objetivo de este store procedure es agregar o borrar categorías de esta tabla, que adicionalmente se usa en las tablas Producto, Carrito Y Compra.

**Triggers:** consta de seis disparadores:

- **Disparador TRG\_LOG\_PRODUCTO:** monitorea cambios por inserción de datos en la tabla Producto, registrando usuario, la fecha y hora realizada. Se usa con instrucción AFTER porque interesa la información después de agregar los nuevos datos, y se guarda en la tabla Log\_Auditoria.
- **Disparador TRG\_LOG\_PRODUCTO\_UPDATE:** cada vez que se modifique con UPDATE la tabla PRODUCTO estará auditando los valores de NOMBRE\_PRODUCTO y PRECIO\_PRODUCTO almacenando los valores anteriores y los nuevos valores si es que se produce cambio, así como el usuario, fecha y hora del evento, en la tabla Log\_Auditoria\_Producto
- **Disparador TRG\_LOG\_PRODUCTO\_DELETE:** usa la instrucción before porque necesitamos los datos antes de que sean borrados. Se activa cuando se borran datos de la tabla Producto, registrando en la tabla Log\_Auditoria\_Producto los datos con .OLD (datos anteriores) de nombre del producto, precio, usuario y fecha del borrado.

- **Disparador TRG\_LOG\_CARRITO:** se activa con la inserción de nuevos datos a la tabla carrito, tomando solo registro de la acción, el nombre del usuario y cuando ocurrió, guardando esto en la tabla Log\_Auditoria.
- **Disparador TRG\_LOG\_CARRITO\_UPDATE:** encargado de monitorear las operaciones de actualización en la tabla CARRITO, toma registro del nombre del producto, la cantidad antes y después de la actualización, usuario, fecha y hora. Usamos la instrucción Before para tomar nota antes de la actualización y del nuevo valor, en CANTIDAD\_PRODUCTO\_OLD\_NEW. Almacena los logs en Log\_Auditoria\_Carrito.
- **Disparador TRG\_LOG\_CARRITO\_DELETE:** se activa al borrar un registro de la tabla CARRITO, por lo que usa el comando BEFORE para almacenar los datos que se van a borrar. Se almacena en la table Log\_Auditoria\_Carrito.

**Funciones:** consta de dos funciones:

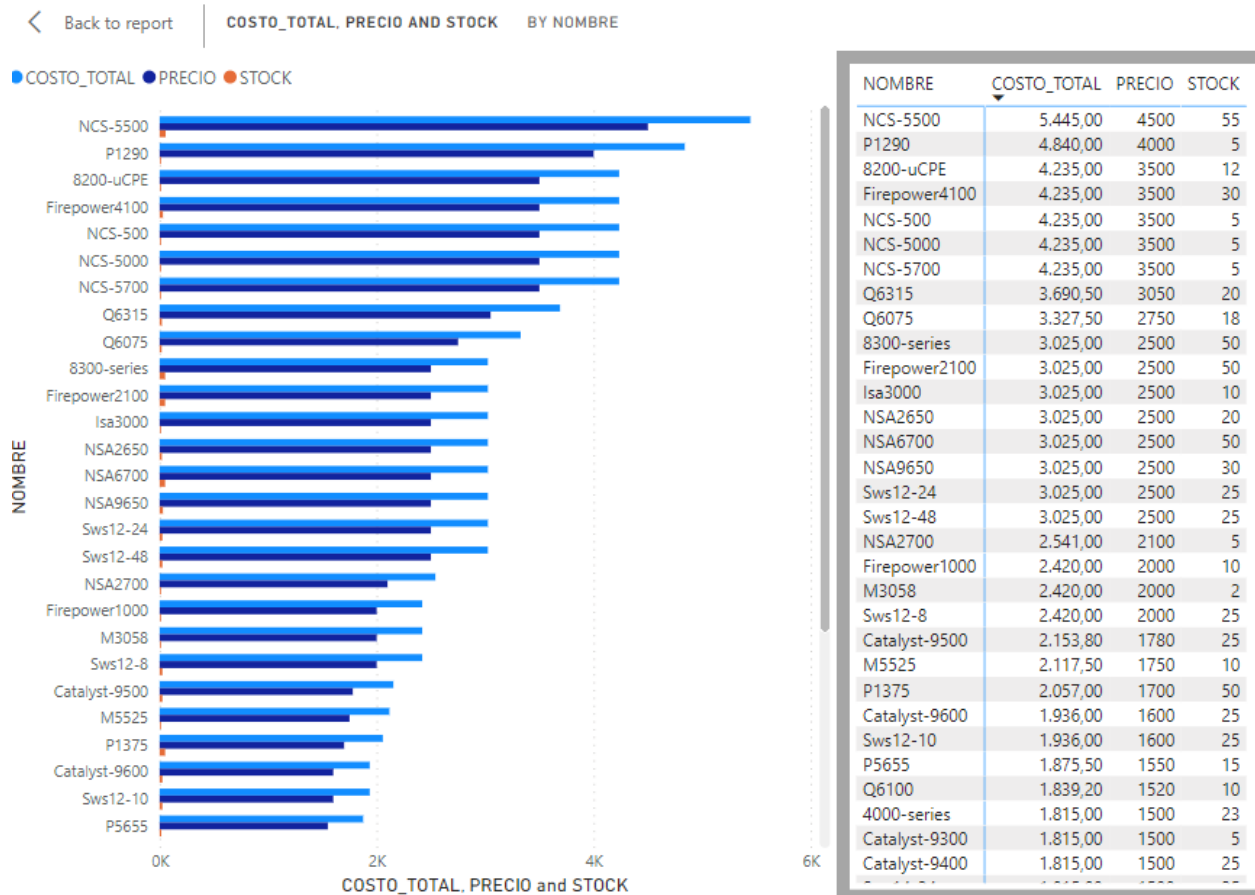
- **Función FN\_CONSULTA\_MARCA:** está función está diseñada para buscar cuántos tipos de artículos hay por marca y qué cantidades se disponen en stock de cada uno.
- **Función FN\_IVA\_POR\_CATEGORIA\_COMPRA:** esta función tiene por objetivo calcular el total, con el IVA incluido, a pagar en una compra, filtrada solo por categoría.

### **Inserción de datos**

En el proyecto se incluye un script donde están todos los datos necesarios para el funcionamiento de las tablas del proyecto, que debe ser usado en el orden especificado al correr los scripts (segundo, después de la inserción de las tablas)

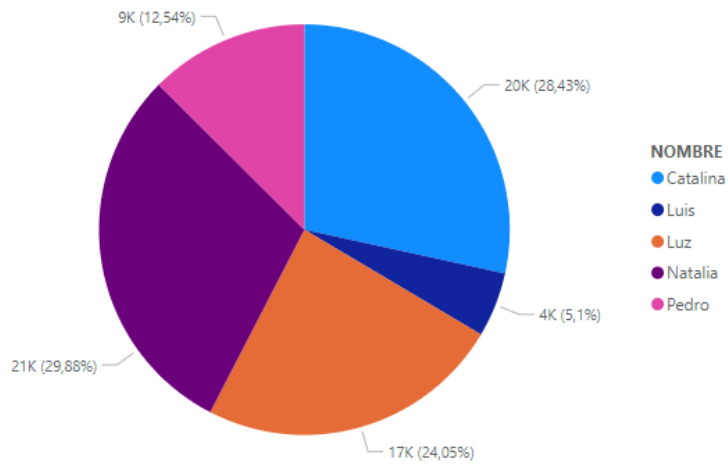
## Informes generados

Se usó la herramienta Power BI para graficar los resultados de algunas vistas. Por ejemplo la siguiente gráfica corresponde a la vista VW\_PRECIO\_TOTAL; podemos ver fácilmente cual es nuestro producto más costo, así como la variación de precios de nuestros productos y el costo final para el cliente:



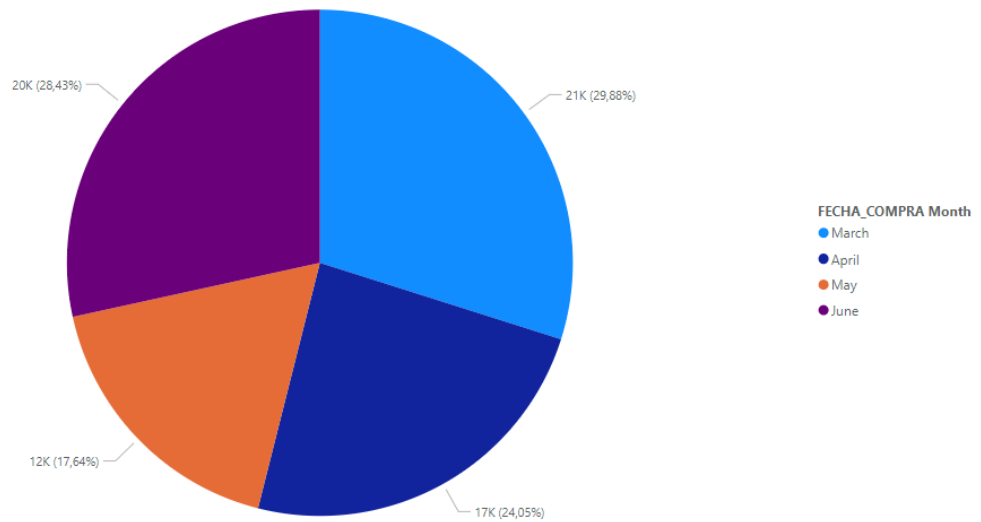
Con la vista VW\_FACTURA\_POR\_USUARIO podemos tener información de quien ha sido el usuario que más ha comprado o por ejemplo cual es el mes que más ganancias ha dejado

< Back to report | MONTO\_TOTAL\_PRODUCTO BY NOMBRE

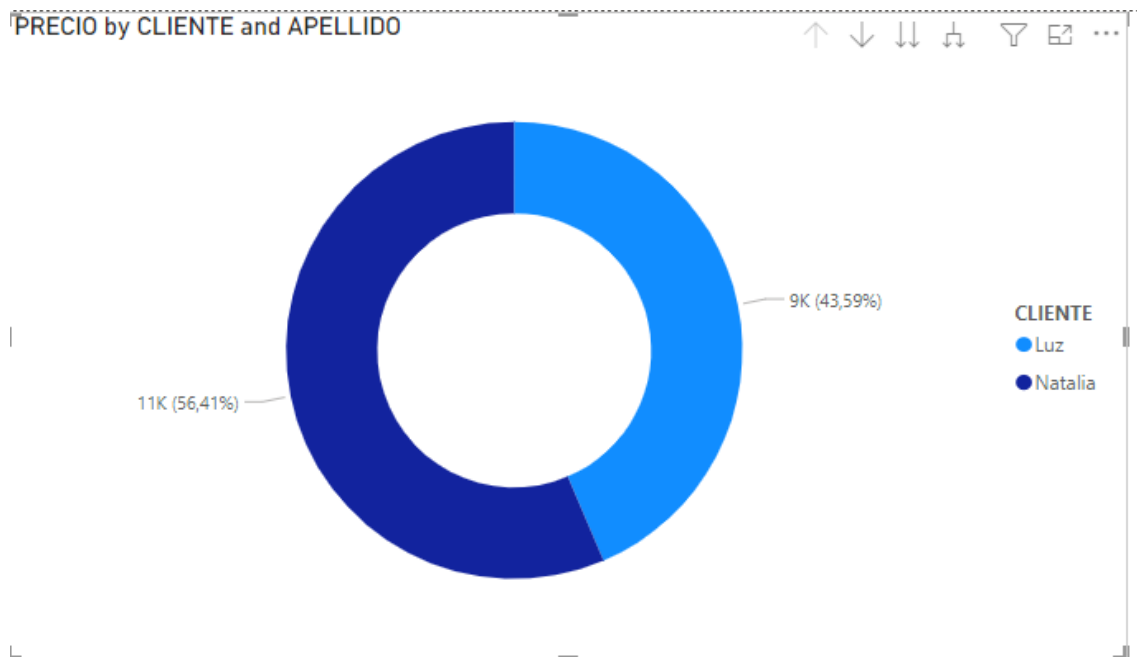


NOMBRE	MONTO_TOTAL_PRODUCTO
Catalina	19500
Luis	3500
Luz	16500
Natalia	20500
Pedro	8600

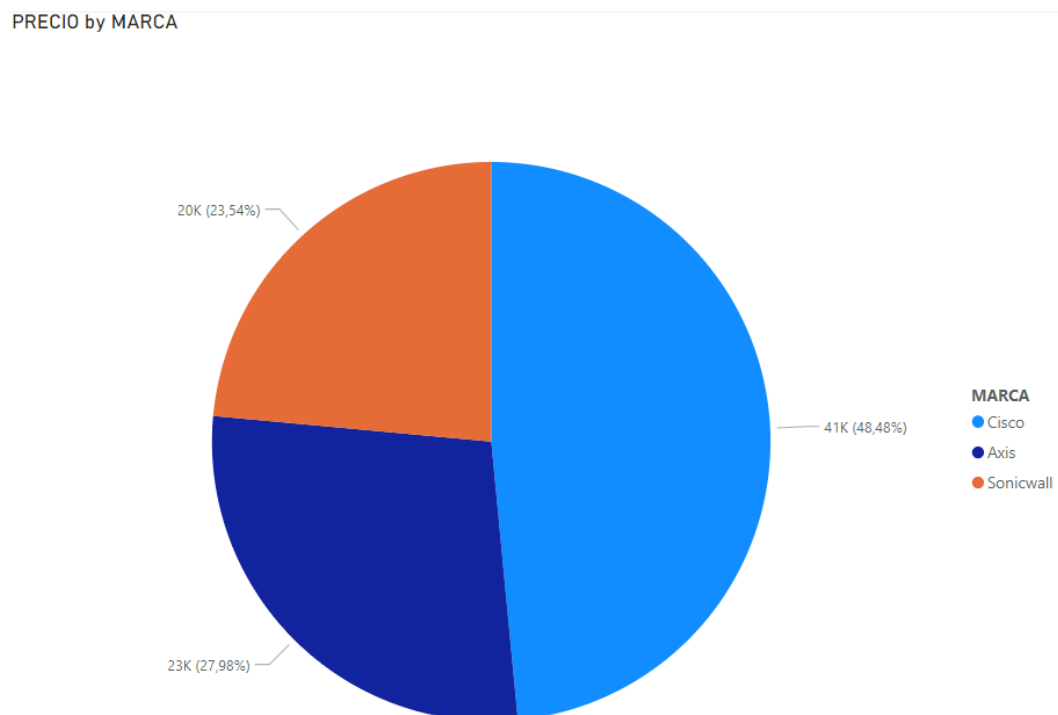
MONTO\_TOTAL\_PRODUCTO by Month



La vista VW\_FACTURA\_DETALLES muestra rápidamente un total de lo comprado recientemente del carrito:

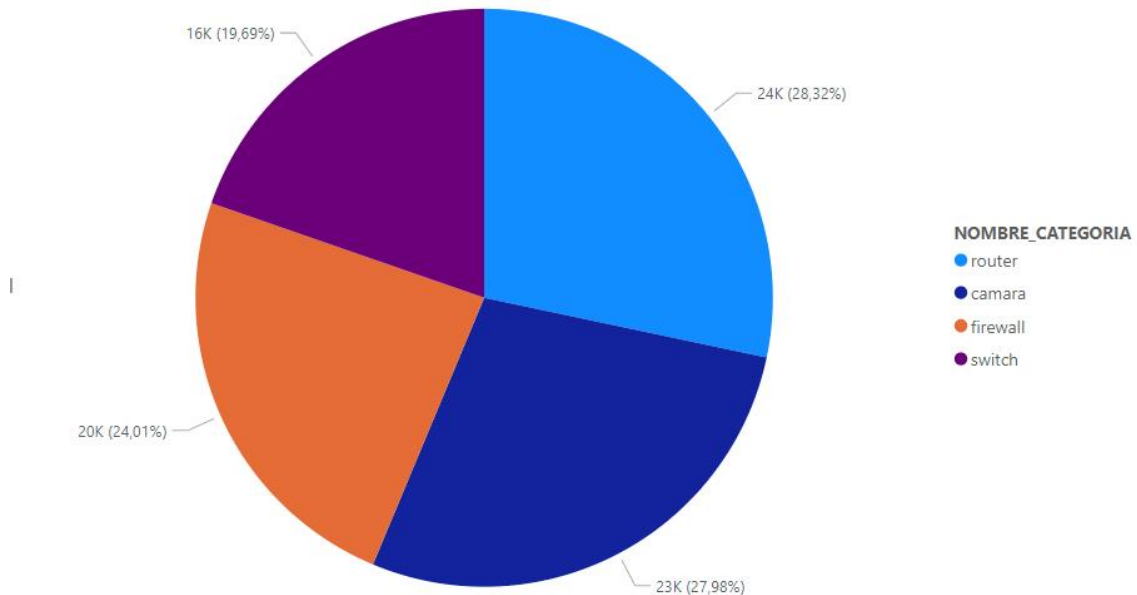


También se pueden obtener información de interés, como por ejemplo, de la vista VW\_CARD\_DE\_PRODUCTO se puede obtener cual es la marca más costosa,



El valor total de los productos por categoría:

'PRECIO by NOMBRE\_CATEGORIA and CATEGORIA



### Información Adicional

- **Sub-Lenguaje DCL**: se incluye un script donde se explica paso a paso la creación de dos usuarios, colocarle y cambiarle la contraseña, asignar diferentes privilegios y con ciertas operaciones demostrar su funcionalidad.
- **Sub-Lenguaje TCL**: se incluye un script que contiene operaciones de inserción de datos para demostrar el uso de los savepoint y del rollback, así como el uso del autocommit.
- **Backup tablas**: también, debido a la importancia que te tiene en la protección de los datos, se incluye un script para realizar el restablecimiento de la base de datos a través de un backup de las tablas y otro con las tablas y estructura.

## **Herramientas y tecnologías utilizadas**

El desarrollo de la base de datos Ryu-Technology se ha hecho con el software MySQL Workbench 8.0.29. Para la visualización de los informes se usó Power Bi versión 2.108.997.0. Para subir los scritps a GitHub se ha usado Visual Studio Code 1.70.1