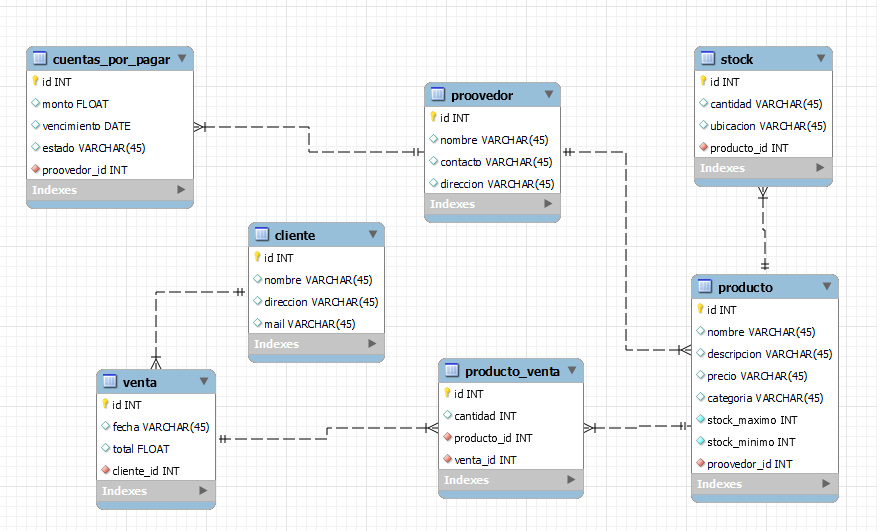
**TP Consultas SQL**

Modelo entidad relación:



La base de datos está diseñada para gestionar información relacionada con un sistema de ventas, incluyendo proveedores, productos, clientes, ventas, detalles de ventas, niveles de stock y cuentas por pagar.  
  
Tabla "**proovedor**":

Almacena información sobre los proveedores.

La clave primaria es la columna id.

Tabla "**producto**":

Contiene información detallada sobre los productos disponibles para la venta.

La clave primaria es la columna id.

La columna proovedor\_id establece una relación con la tabla "proovedor".

Tabla "**cliente**":

Guarda información acerca de los clientes.

La clave primaria es la columna id.

Tabla "**venta**":

Registra información sobre las ventas realizadas.

La clave primaria es la columna id.

La columna cliente\_id establece una relación con la tabla "cliente".

Tabla "**producto\_venta**":

Mantiene los detalles específicos de los productos vendidos en cada transacción.

La clave primaria es la columna id.

Las columnas producto\_id y venta\_id establecen relaciones con las tablas "producto" y "venta" respectivamente.

Tabla "**stock**":

Almacena información sobre los niveles de stock de cada producto.

La clave primaria es la columna id.

La columna producto\_id establece una relación con la tabla "producto".

Tabla "**cuentas\_por\_pagar**":

Registra las cuentas pendientes de pago a proveedores.

La clave primaria es la columna id.

La columna proovedor\_id establece una relación con la tabla "proovedor".

Script para crear la base de datos:

*En este caso, se crea y usa un esquema mydb, se puede remplazar por el que mas le guste, acuérdese de hacerlo en casa una de la queries.*

|  |
| --- |
| CREATE SCHEMA mydb;  USE mydb;  CREATE TABLE proovedor (  id INT NOT NULL AUTO\_INCREMENT,  nombre VARCHAR(45),  contacto VARCHAR(45),  direccion VARCHAR(45),  PRIMARY KEY (id)  );  CREATE TABLE producto (  id INT NOT NULL AUTO\_INCREMENT,  nombre VARCHAR(45),  descripcion VARCHAR(256),  precio FLOAT,  categoria VARCHAR(45),  stock\_maximo INT NOT NULL,  stock\_minimo INT NOT NULL,  proovedor\_id INT NOT NULL,  PRIMARY KEY (id),  CONSTRAINT fk\_producto\_proovedor  FOREIGN KEY (proovedor\_id)  REFERENCES proovedor (id)  );  CREATE TABLE cliente (  id INT NOT NULL AUTO\_INCREMENT,  nombre VARCHAR(45),  direccion VARCHAR(45),  email VARCHAR(45),  PRIMARY KEY (id)  );  CREATE TABLE venta (  id INT NOT NULL AUTO\_INCREMENT,  fecha VARCHAR(45),  total FLOAT,  cliente\_id INT NOT NULL,  PRIMARY KEY (id),  CONSTRAINT fk\_venta\_cliente  FOREIGN KEY (cliente\_id)  REFERENCES cliente (id)  );  CREATE TABLE producto\_venta (  id INT NOT NULL AUTO\_INCREMENT,  cantidad INT,  producto\_id INT NOT NULL,  venta\_id INT NOT NULL,  PRIMARY KEY (id),  CONSTRAINT fk\_producto\_venta\_producto  FOREIGN KEY (producto\_id)  REFERENCES mydb.producto (id),  CONSTRAINT fk\_producto\_venta\_venta  FOREIGN KEY (venta\_id)  REFERENCES venta (id)  );  CREATE TABLE stock (  id INT NOT NULL AUTO\_INCREMENT,  cantidad VARCHAR(45),  ubicacion VARCHAR(45),  producto\_id INT NOT NULL,  PRIMARY KEY (id),  CONSTRAINT fk\_stock\_producto  FOREIGN KEY (producto\_id)  REFERENCES producto (id)  );  CREATE TABLE cuentas\_por\_pagar (  id INT NOT NULL AUTO\_INCREMENT,  monto FLOAT,  estado VARCHAR(45),  vencimiento DATE,  proovedor\_id INT NOT NULL,  PRIMARY KEY (id),  CONSTRAINT fk\_cuentas\_por\_pagar\_proovedor  FOREIGN KEY (proovedor\_id)  REFERENCES proovedor (id)  ); |

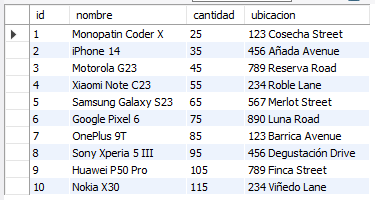
Script para poblar la base de datos:

|  |
| --- |
| -- Insertar 10 registros en la tabla proovedor  INSERT INTO proovedor (nombre, contacto, direccion) VALUES  ('Quantum Innovations Tech', 'contacto@quantuminnovations.tech', '1234 Innovator Street'),  ('Code Nexus Solutions', 'codenexus@gmail.com', '789 Byte Avenue'),  ('CyberSphere Dynamics', 'contacto@cyberspheredynamics.net', '567 Matrix Street'),  ('NanoByte Technologies', 'info@nanobyte-tech.com', '234 Nano Lane'),  ('MicroByte Innovations', 'contact@microbyteinnovations.net', '123 Micro Lane'),  ('QuantumSphere Technologies', 'info@quantumspheretech.com', '345 Quantum Road'),  ('NanoInnovate Technologies', 'contact@nanoinnovatetech.co', '678 Nano Stree'),  ('QuantumByte Innovations', 'info@quantumbyteinnovations.net', '234 Quantum Lane'),  ('CodeSphere Dynamics', 'contact@codespheredynamics.com', '567 Code Street'),  ('ByteNexus Technologies', 'info@bytenexustechnologies.co', '890 Byte Road');  -- Insertar 10 registros en la tabla producto  INSERT INTO producto (nombre, descripcion, precio, categoria, stock\_maximo, stock\_minimo, proovedor\_id) VALUES  ('Monopatin Coder X', 'Estilo y tecnología convergen en el Monopatín Coder X, la fusión perfecta de movilidad eléctrica con conectividad Bluetooth y diseño vanguardista.', 2500, 'Tecnologia', 20, 10, 1),  ('iPhone 14', 'El iPhone 14 redefine la excelencia con un diseño elegante y tecnología de vanguardia. Experimenta la innovación en tus manos.', 900, 'Tecnologia', 45, 20, 2),  ('Motorola G23', 'El Motorola G23 ofrece rendimiento y estilo en un solo dispositivo. Descubre la última tecnología en un diseño moderno y asequible.', 700, 'Tecnologia', 45, 15, 3),  ('Xiaomi Note C23', 'El Xiaomi Note C23 combina rendimiento potente y diseño elegante en un dispositivo accesible. Experimenta la innovación con estilo.', 800, 'Tecnologia', 150, 20, 4),  ('Samsung Galaxy S23', 'El Samsung Galaxy S23 redefine la excelencia con su diseño sofisticado y tecnología de vanguardia. Vive la innovación en cada detalle.', 950, 'Tecnologia', 200, 50, 5),  ('Google Pixel 6', 'Una fusión de potencia y fotografía excepcional, el Google Pixel 6 lleva la experiencia Android a nuevas alturas con su diseño elegante y funciones inteligentes.', 1100, 'Tecnologia', 100, 30, 6),  ('OnePlus 9T', 'El OnePlus 9T ofrece un rendimiento extraordinario y una pantalla fluida, combinados con una potente configuración de cámaras, proporcionando una experiencia premium a un precio atractivo.', 1050, 'Tecnologia', 70, 40, 7),  ('Sony Xperia 5 III', 'El Sony Xperia 5 III destaca con su pantalla cinematográfica y capacidad de captura profesional, ofreciendo un rendimiento impresionante en un formato compacto y elegante.', 1150, 'Tecnologia', 80, 45, 8),  ('Huawei P50 Pro', 'Con un diseño innovador y un sistema de cámaras excepcional, el Huawei P50 Pro redefine la fotografía móvil, brindando una experiencia visual única y un rendimiento de vanguardia.', 1000, 'Tecnologia', 90, 50, 9),  ('Nokia X30', 'El Nokia X30 combina durabilidad clásica con tecnología moderna, ofreciendo un rendimiento sólido y una experiencia Android pura en un dispositivo asequible y confiable.', 1000, 'Tecnologia', 100, 55, 10);  -- Insertar 10 registros en la tabla cliente  INSERT INTO cliente (nombre, direccion, email) VALUES  ('Sofia Rodriguez', 'Av Rivadavia 237', 'sofirodriguez@gmail.com'),  ('Nahuel Mosse', 'Av Santa Fe 849', 'nahuelmosse@gmail.com'),  ('Facundo Cogan Repetti', 'Av Italia 529', 'coganfacundo@gmail.com'),  ('Sebastian Martinez Amaray', 'Av Brasil 942', 'sebastianmartinez@gmail.com'),  ('Lautaro Martinez', 'Av Cordoba 345', 'martzlautaro@gmail.com'),  ('Juan Roman Riquelme', 'Av Alto Peru 123', 'jrriquelme@gmail.com'),  ('Antonela Rocuzzo', 'Ocampo 848', 'rocuzzoanto@gmail.com'),  ('Eduardo Feinmann', 'Irala 113', 'feinmanne@mail.com'),  ('Jorge Rial', 'Argerich 227', 'rialjorge@gmail.com'),  ('Susana Gimenez', 'Pedro Diaz 998', 'gimenezsusana@gmail.com');  -- Insertar 10 registros en la tabla venta  INSERT INTO venta (fecha, total, cliente\_id) VALUES  ('2023-11-01', 1500, 1),  ('2023-11-20', 2500, 2),  ('2023-11-18', 3500, 3),  ('2023-11-22', 4500, 4),  ('2023-11-21', 5500, 5),  ('2023-12-20', 6500, 6),  ('2023-11-23', 7500, 7),  ('2023-12-20', 8500, 8),  ('2023-12-24', 9500, 9),  ('2023-3-11', 10500, 10);  -- Insertar 10 registros en la tabla producto\_venta  INSERT INTO producto\_venta (cantidad, producto\_id, venta\_id) VALUES  (10, 1, 1),  (22, 2, 2),  (35, 3, 3),  (42, 4, 4),  (50, 5, 5),  (65, 6, 6),  (17, 7, 7),  (28, 8, 8),  (69, 9, 9),  (100, 10, 10);  -- Insertar 10 registros en la tabla stock  INSERT INTO stock (cantidad, ubicacion, producto\_id) VALUES  ('25', '123 Cosecha Street', 1),  ('35', '456 Añada Avenue', 2),  ('45', '789 Reserva Road', 3),  ('55', '234 Roble Lane', 4),  ('65', '567 Merlot Street', 5),  ('75', '890 Luna Road', 6),  ('85', '123 Barrica Avenue', 7),  ('95', '456 Degustación Drive', 8),  ('105', '789 Finca Street', 9),  ('115', '234 Viñedo Lane', 10);  -- Insertar 10 registros en la tabla cuentas\_por\_pagar  INSERT INTO cuentas\_por\_pagar (monto, vencimiento, estado, proovedor\_id) VALUES  (1000, '2023-12-01', 'Pendiente', 1),  (2000, '2023-12-02', 'Pendiente', 2),  (3000, '2023-12-03', 'Pendiente', 3),  (4000, '2023-12-04', 'Pagado', 4),  (5000, '2023-12-05', 'Pagado', 5),  (6000, '2023-12-06', 'Pagado', 6),  (7000, '2023-05-07', 'Vencido', 7),  (8000, '2023-06-08', 'Vencido', 8),  (9000, '2023-08-09', 'Vencido', 9),  (10000, '2023-03-10', 'Vencido', 10); |

Consultas:

* Ver todo el stock disponible

|  |
| --- |
| SELECT  stock.id,  producto.nombre,  stock.cantidad,  stock.ubicacion  FROM stock  INNER JOIN producto ON producto.id = stock.producto\_id; |



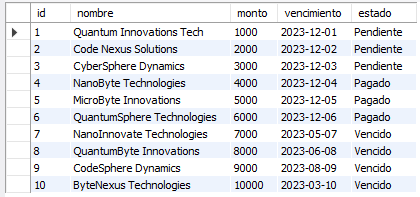
* Ventas realizadas en un período específico

|  |
| --- |
| SELECT  venta.id,  cliente.nombre,  venta.fecha,  venta.total  FROM venta  INNER JOIN cliente ON venta.cliente\_id = cliente.id  WHERE venta.fecha BETWEEN '2023-01-01' AND '2023-12-31'; |



* Estado de cuentas por pagar a proveedores

|  |
| --- |
| SELECT  cuentas\_por\_pagar.id,  proovedor.nombre,  cuentas\_por\_pagar.monto,  cuentas\_por\_pagar.vencimiento,  cuentas\_por\_pagar.estado  FROM cuentas\_por\_pagar  INNER JOIN proovedor ON cuentas\_por\_pagar.proovedor\_id = proovedor.id; |



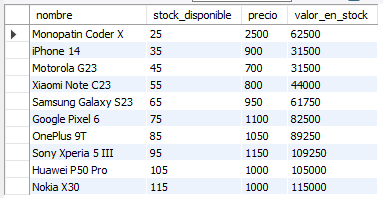
* Productos comprados por cada cliente

|  |
| --- |
| SELECT  cliente.nombre AS nombre\_cliente,  producto.nombre AS nombre\_producto,  producto\_venta.cantidad  FROM producto\_venta  INNER JOIN producto ON producto\_venta.producto\_id = producto.id  INNER JOIN venta ON producto\_venta.venta\_id = venta.id  INNER JOIN cliente ON venta.cliente\_id = cliente.id; |



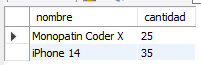
* Valorización del stock

|  |
| --- |
| SELECT  producto.nombre,  SUM(stock.cantidad) AS stock\_disponible,  producto.precio,  producto.precio \* SUM(stock.cantidad) AS valor\_en\_stock  FROM stock  INNER JOIN producto ON producto.id = stock.producto\_id  GROUP BY producto.id; |



* Productos para reposición

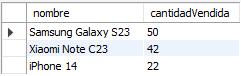
|  |
| --- |
| SELECT  producto.nombre,  stock.cantidad  FROM stock  INNER JOIN producto ON producto.id = stock.producto\_id  WHERE stock.cantidad < 40; |



Informes:

* Informes que muestren los productos más vendidos en un período

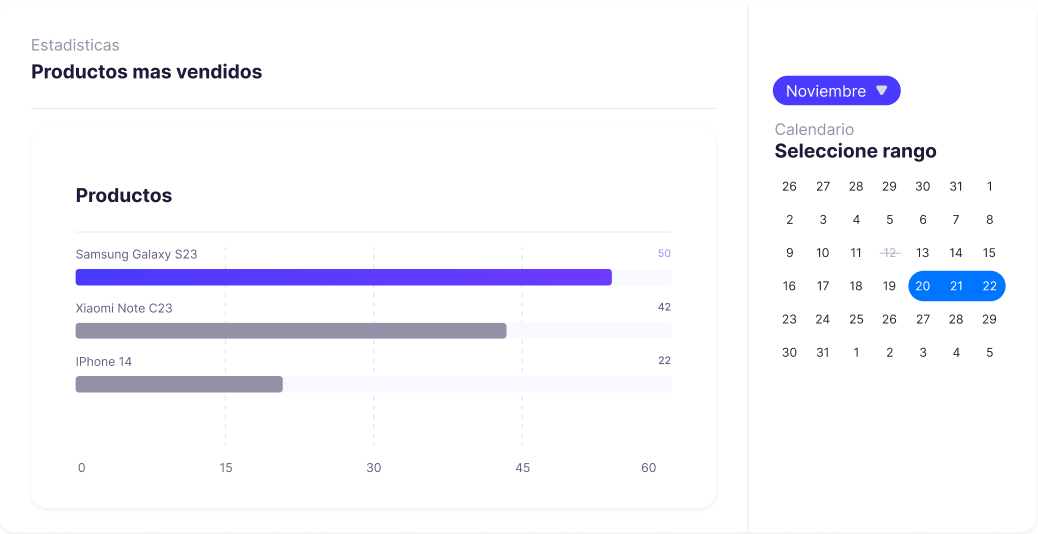
|  |
| --- |
| SELECT  producto.nombre,  SUM(producto\_venta.cantidad) AS cantidadVendida  FROM producto\_venta  INNER JOIN producto ON producto.id = producto\_venta.producto\_id  INNER JOIN venta ON producto\_venta.venta\_id = venta.id  WHERE venta.fecha BETWEEN '2023-11-20' AND '2023-11-22'  GROUP BY producto\_venta.producto\_id  ORDER BY cantidadVendida DESC; |



En esta pantalla, se puede seleccionar las fechas de las cueles se va a hacer la consulta, en este caso desde el 20/11 hasta el 22/11.

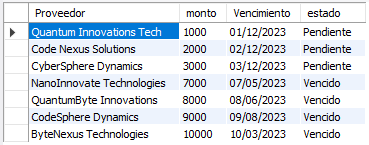
Una vez que llegan los datos de la base de datos, se ve una lista de productos, rankeados por cantidad vendida, informando el nombre del mismo, y la cantidad que se vendió en el periodo marcado.

Te muestra la cantidad vendida a través de un gráfico, que te muestra la cantidad vendida de manera visual.

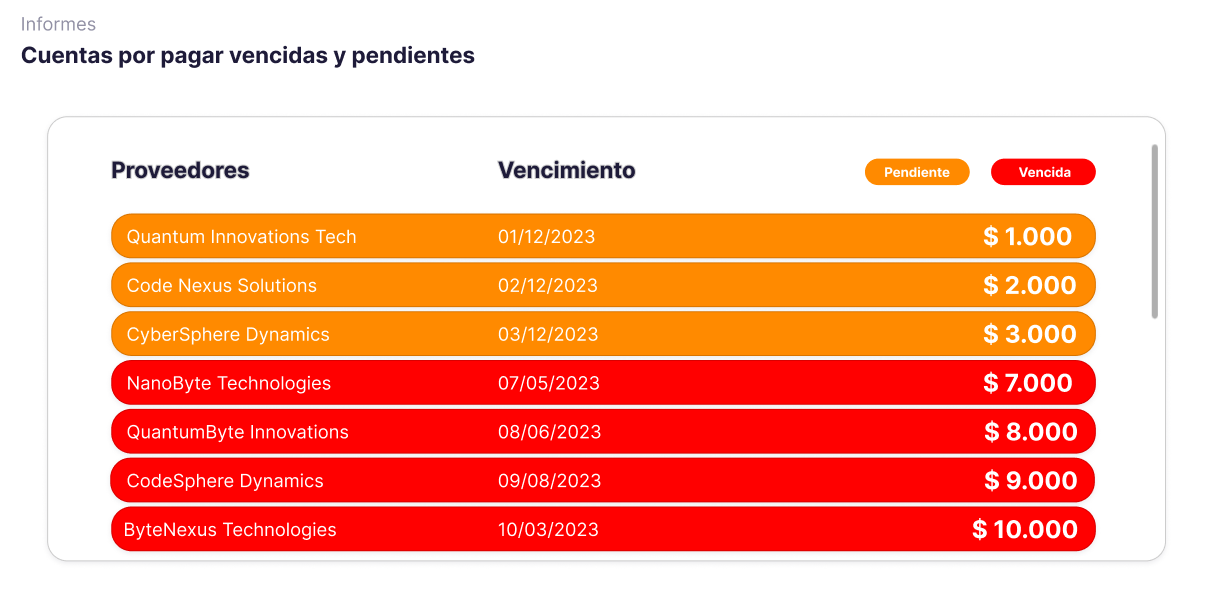


* Informes de cuentas por pagar pendientes y vencidas

|  |
| --- |
| SELECT  proovedor.nombre AS Proveedor,  cuentas\_por\_pagar.monto,  DATE\_FORMAT(cuentas\_por\_pagar.vencimiento, '%d/%m/%Y') AS Vencimiento,  cuentas\_por\_pagar.estado  FROM cuentas\_por\_pagar  INNER JOIN proovedor ON cuentas\_por\_pagar.proovedor\_id = proovedor.id  WHERE cuentas\_por\_pagar.estado = 'Pendiente' OR cuentas\_por\_pagar.vencimiento < CURDATE(); |

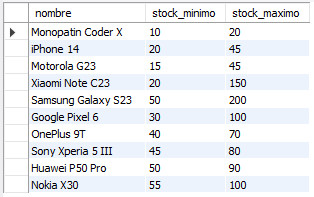


En esta pantalla, se ven todos los pagos que hacen falta realizar, teniendo el nombre del proveedor, fecha de vencimiento y monto, además de tener un color que identifica el estado de cada cuenta por pagar.



* Informes de stock mínimo y máximo de productos

|  |
| --- |
| SELECT  producto.nombre,  producto.stock\_minimo,  producto.stock\_maximo  FROM producto; |

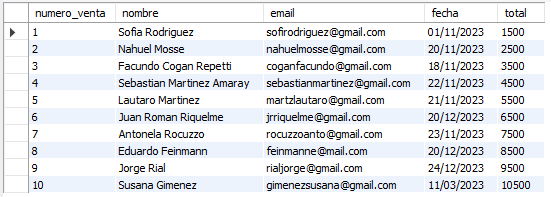


En esta pantalla, se nuestra a través de un gráfico, de azul, el stock máximo, y de gris el mínimo, igualmente, del lado derecho, se ven los dos datos de forma numérica, siendo el valor de la izquierda el valor mínimo, y el de la derecha el máximo.



* Informes que relacionen las ventas con la información de clientes

|  |
| --- |
| SELECT  venta.id AS numero\_venta,  cliente.nombre,  cliente.email,  DATE\_FORMAT(venta.fecha, '%d/%m/%Y') AS fecha,  venta.total  FROM venta  INNER JOIN cliente ON cliente.id = venta.cliente\_id; |



En esta pantalla, podemos ver en forma de tabla, el numero de venta, el cliente, email del cliente, fecha y total de cada una de las ventas realizadas utilizando el sistema.

