**Base de datos 1**

Alumno: Santiago Vietto

Docente: Leandro Luis Juárez

DNI: 42654882

Institución: UCC

Año: 2021

**Ejercicio 2: Sistema básico de ventas**

**Parte 1**

\_ Se necesita una Base de datos que permita apoyar la gestión de un sistema de ventas. La empresa necesita llevar un control de: proveedores, clientes, productos, ventas.

\_ Un **proveedor** tiene: ID, nombre, dirección, teléfono,  Email.

* *Cada proveedor puede brindar más de un tipo de producto.*
* *Todos los proveedores de la base de datos Brindan al menos un tipo de producto.*
* *El dato dirección se entiende por calle, número, barrio y ciudad.*

\_ Un **cliente** tiene: ID, nombre, direcciones de envío, hasta 2 teléfonos de contacto.

* *El dato dirección se entiende por calle, número, barrio y ciudad.*
* *Cada dirección pertenece solo a 1 cliente.*

\_ Un **producto** tiene: ID, nombre, precio actual, stock, nombre del proveedor (cada producto tiene un solo proveedor).

* *Además, los productos se organizan en categorías, y cada producto va sólo en una categoría mientras que cada categoría puede tener varios productos. No existen categorías vacías.*

\_ Una **categoría** tiene: ID, nombre, descripción.

\_ Por razones de contabilidad, se debe registrar la información de cada **venta** con un id, fecha, cliente, descuento, producto y monto final. El monto final será el precio del producto más el IVA, teniendo en cuenta el descuento y la cantidad de productos.

* *Asumimos que cada venta solo tiene un producto.*

**Además se debe guardar el precio del producto al momento de la venta y la cantidad vendida.**

**Parte 2**

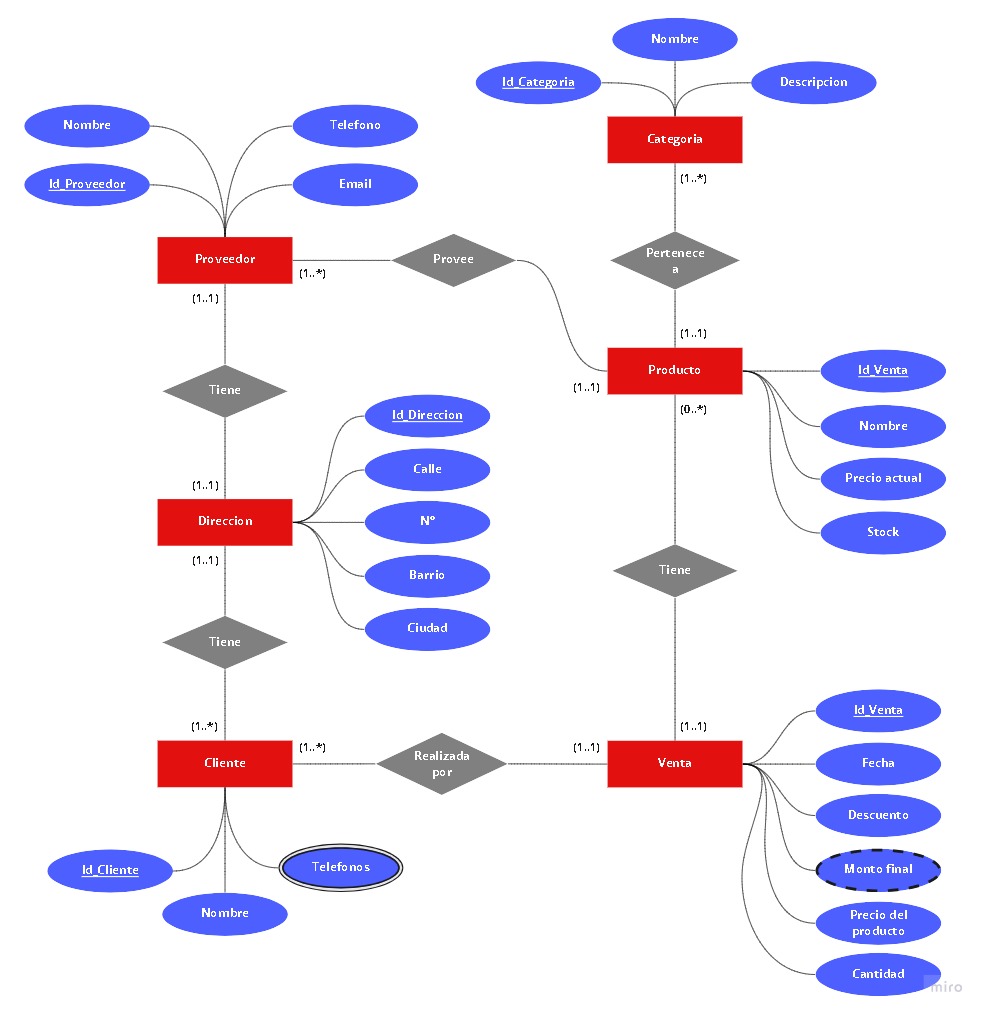
* ¿Cuáles son los cambios que deberíamos realizar en los diagramas para que las ventas tengan distintos productos?
* ¿Cuáles son los cambios que se deben realizar para cargar las direcciones de las sucursales de los proveedores?

**Resolución parte 1**

\_ Como podemos ver, el atributo dirección en la entidad cliente, es multivalorado y compuesto, por lo tanto generamos una entidad dirección que se relacione con cliente. Un cliente puede tener una o muchas direcciones, y una dirección puede tener uno o varios clientes. Como las direcciones, tanto del cliente como del proveedor, estaban formadas de la misma manera con los mismos atributos, creamos la entidad relación que guarde todas las direcciones, entonces las direcciones no tienen idea de quienes son y eso no importa, porque despues el cliente tiene el Id de su dirección, y lo mismo que el proveedor. Podemos hacer esto porque justo coincide que las direcciones de ambos son iguales, ya que si alguno de los dos tiene en el atributo dirección alguna otra cosa distinta, ya este modelo no nos sirve, porque en las columnas de la otra entidad faltarían datos o tendríamos columnas vacías. La ventaja de esto es que si tenemos que agregar atributos a las direcciones solamente cambiamos la entidad dirección. Para graficar esto necesitamos una tabla pasarela por más que la relación muchos a muchos.

\_ En la entidad producto cuando se pide el nombre del proveedor, hace referencia al Id proveedor que se representa con la relación y en el diagrama de clases, y esto hace referencia a una consulta a la base de datos. Un producto tiene una y solo una categoría, y una categoría tiene uno o muchos productos pero nunca va a estar vacía.

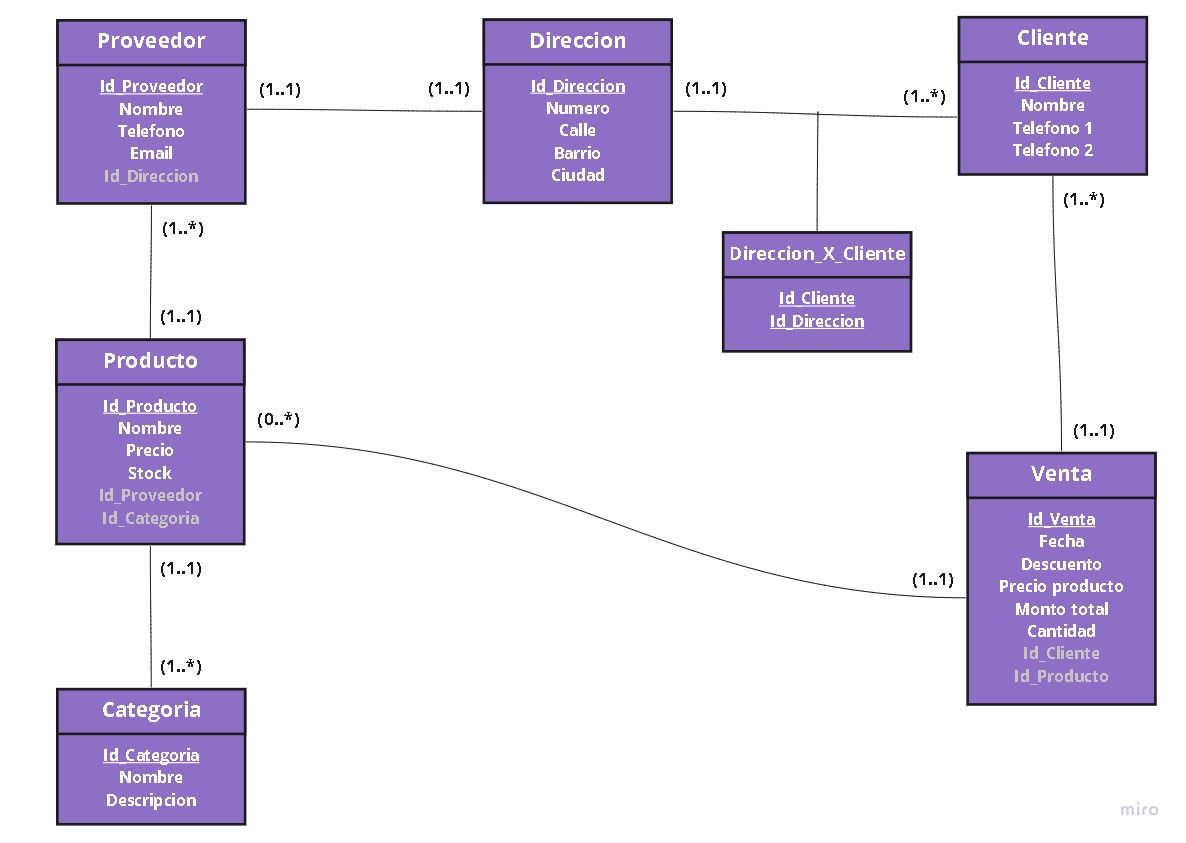
\_ En la entidad venta, la cantidad es cuantos productos, tenemos una fecha, un descuento, el Id del producto, y además el monto final que va a estar calculado por el precio del producto más el IVA teniendo en cuenta el descuento y la cantidad de productos. El atributo descuento es fijo, porque lo que es derivado o calculado es el monto final. Una venta tiene uno y solo un producto, ahora esto leído al revés seria que un producto en mi stock nunca haya sido vendido o puede que lo haya vendido muchas veces. Lo mismo que una venta se realiza para uno y solo un cliente, y uno o muchos clientes pueden realizar una compra, pero nunca ninguno. El precio del producto está puesto, porque que es congelar el precio al momento de la venta, no es lo mismo que el precio actual del producto, porque si se compra 3 veces el producto ahí tenemos el precio actual x 3 que sería el precio del producto en un momento determinado (congelado), y con este calculamos el monto total. Es por eso que tenemos que persistir atributos, es decir, guardar o congelar el valor de un atributo en un momento particular.

\_ El diagrama E-R seria:

\_ Por otro lado en el diagrama de clases la relación entre proveedor, dirección y cliente seria, tenemos la entidad dirección que tiene direcciones dentro, como proveedor tiene una sola dirección y la relación con esta es de uno a uno ponemos el Id de dirección en el proveedor, y para el cliente que tiene muchas, creamos la tabla pasarela, no le ponemos el Id del cliente en la dirección porque haría una interferencia con la dirección del proveedor, es por eso que se crea la tabla pasarela y asi ni el cliente ni la dirección se conocen.

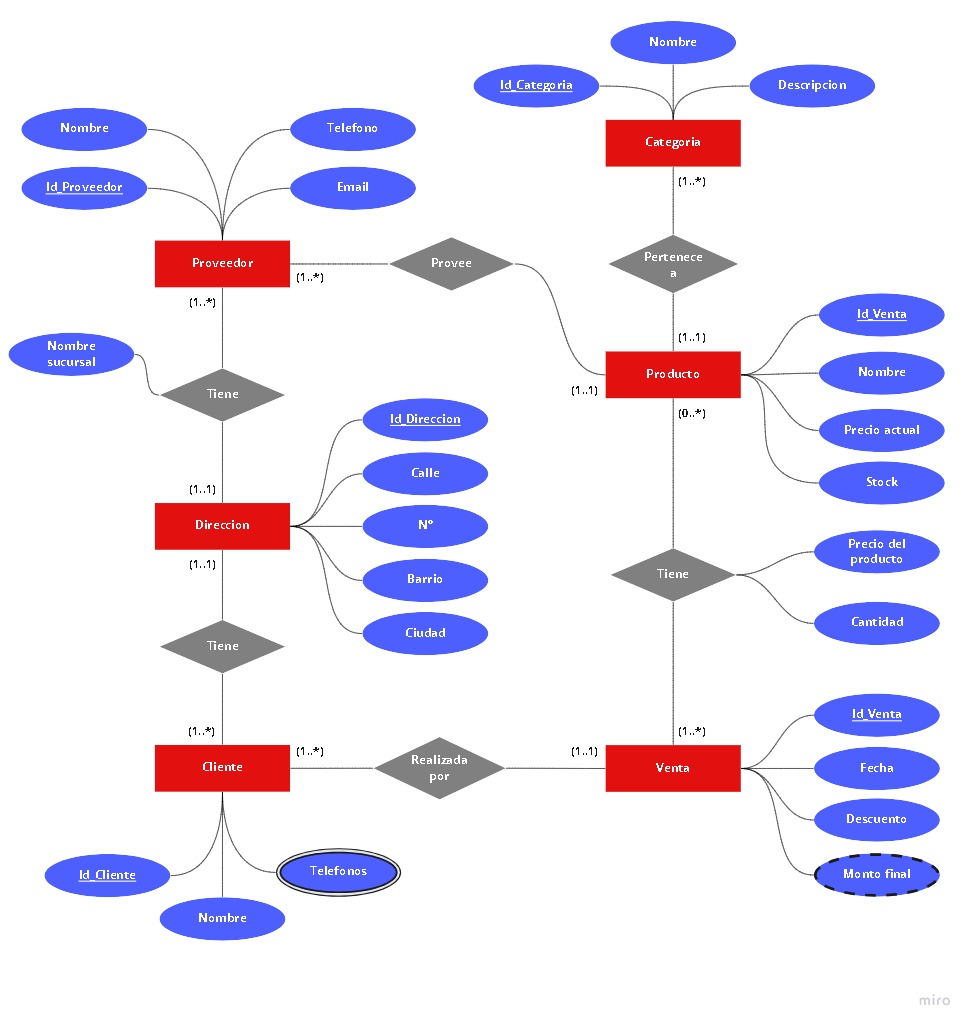
\_ La entidad producto tiene el Id del proveedor y la categoría, por que al ser una relación uno a uno tiene un solo proveedor y una sola categoría.

\_ En el caso de la venta tenemos el Id del cliente porque es de uno a muchos por lo tanto va en venta ya que es (1..1), y al igual con el Id producto porque fue solo uno producto.

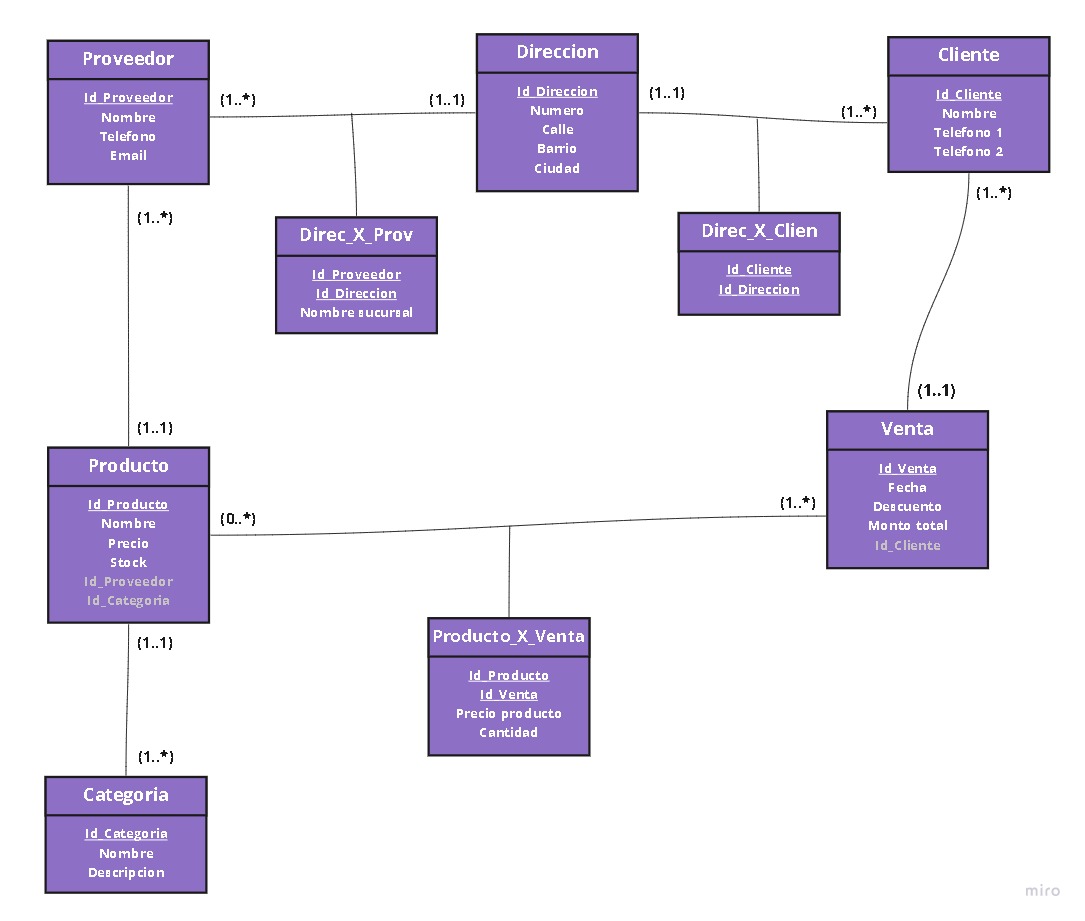


**Resolución parte 2**

\_ Cambia la cardinalidad en ventas (1..\*), no podemos tener el Id producto en la tabla de ventas porque no sabemos que producto es el de que se está hablando, por ende necesitamos una tabla pasarela, en donde además agregamos el precio del producto y la cantidad del producto determinado.

\_ Ahora los proveedores tienen muchas direcciones, y cargamos en la relación el nombre de la sucursal para no modificar la entidad dirección.

\_ Prototipo diagrama E-R:

\_ Prototipo diagrama de clases: