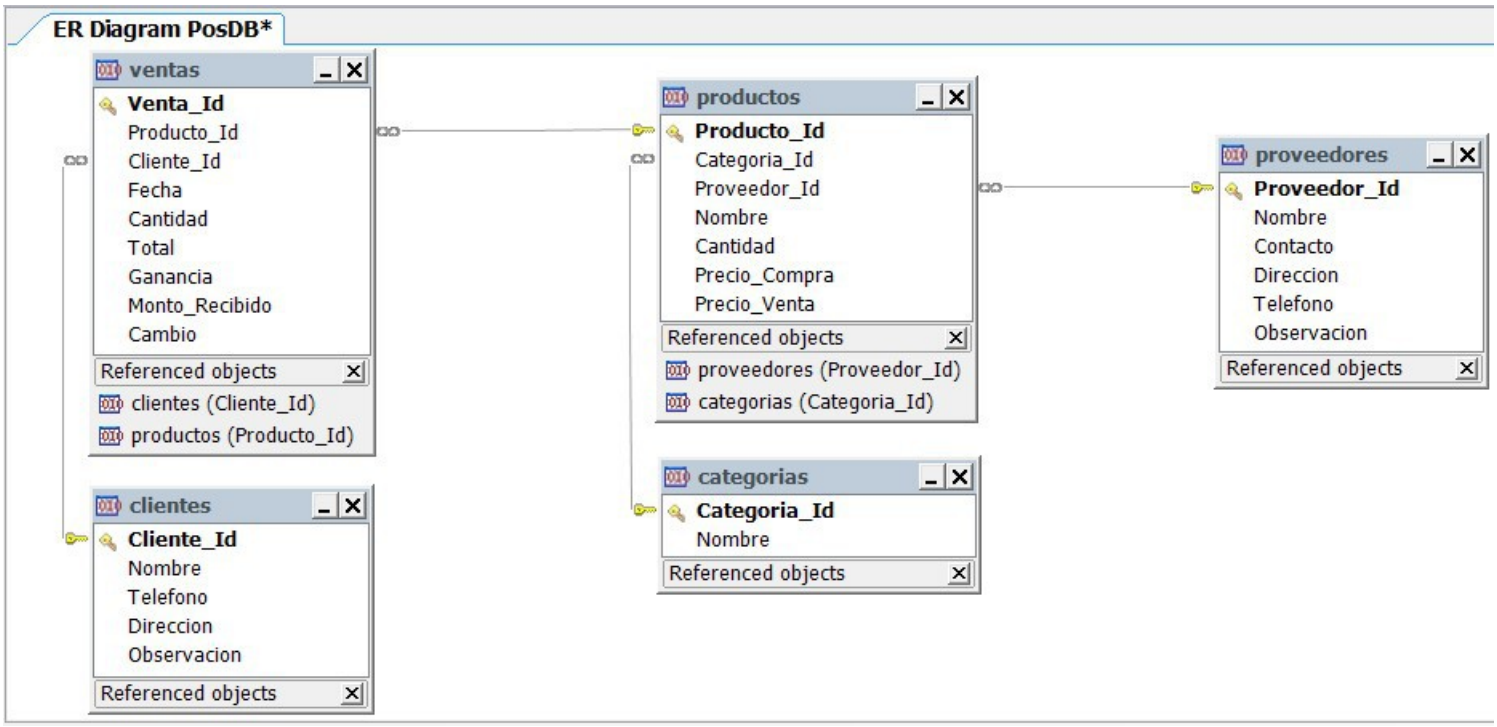
Bien, continuando con los reworks de ejemplos de aplicaciones por ahí expuestas en la Web, esta vez traigo uno de un POS - Sistema de Ventas de comida rápida, el cual está diseñado en PHP - MySql.

La BD original está creada en inglés así como su front, yo cambié ambos al español, y rediseñé la BD que al parecer fue en su momento creada por un principiante que carecía de experiencia; es claro que un pos de ventas tiene que tener una relación entre ventas y clientes, pues toda venta necesariamente estará atada a un cliente que realiza la compra, en la BD original la estructura para almacenar las ventas, tenía una columna llamada: clientes donde almacenaban el nombre del cliente, cuando debería ser el ID del cliente, y volvían a almacenar en dicha estructura, datos del producto implicado en la venta, como su nombre, precios de compra, venta y categoría asociada al producto, esto desde luego es una falla de diseño grave, pues están manejando redundancia de información, que ya está almacenada en otras tablas, y con solo tener el ID del producto, es posible traer toda la información que se requiera del producto implicado.

Lo anterior se suele hacer en escenarios muy particulares, donde el volumen de las tablas a relacionar en un join es alto, y por ende el join se vuelve muy costoso en términos de rendimiento, y se toma la decisión de replicar ciertas columnas, para evitar estar yendo a otra estructura, a traer información que el usuario siempre requiere ver, evitando con ello ralentizar los sistemas, pero reitero esto se hace en escenarios muy particulares, y que ameriten hacerlo así, porque igual no deja de ser una muy mala práctica, que desde luego viola las reglas de normalización en las bases de datos.

Por otra parte la aplicación original no manejaba estructura para categorías, estaban todas "quemadas" en una lista desplegable, lo que impedía un futuro crecimiento y control en este aspecto al sistema. Además de que la BD no tenía constraints en ninguna tabla, entonces ésta carecía de integridad referencial, permitiendo eliminar información sin control alguno; por lo que se hicieron algunos ajustes previendo eso, y normalizándola adecuadamente, quedando el modelo como sigue:



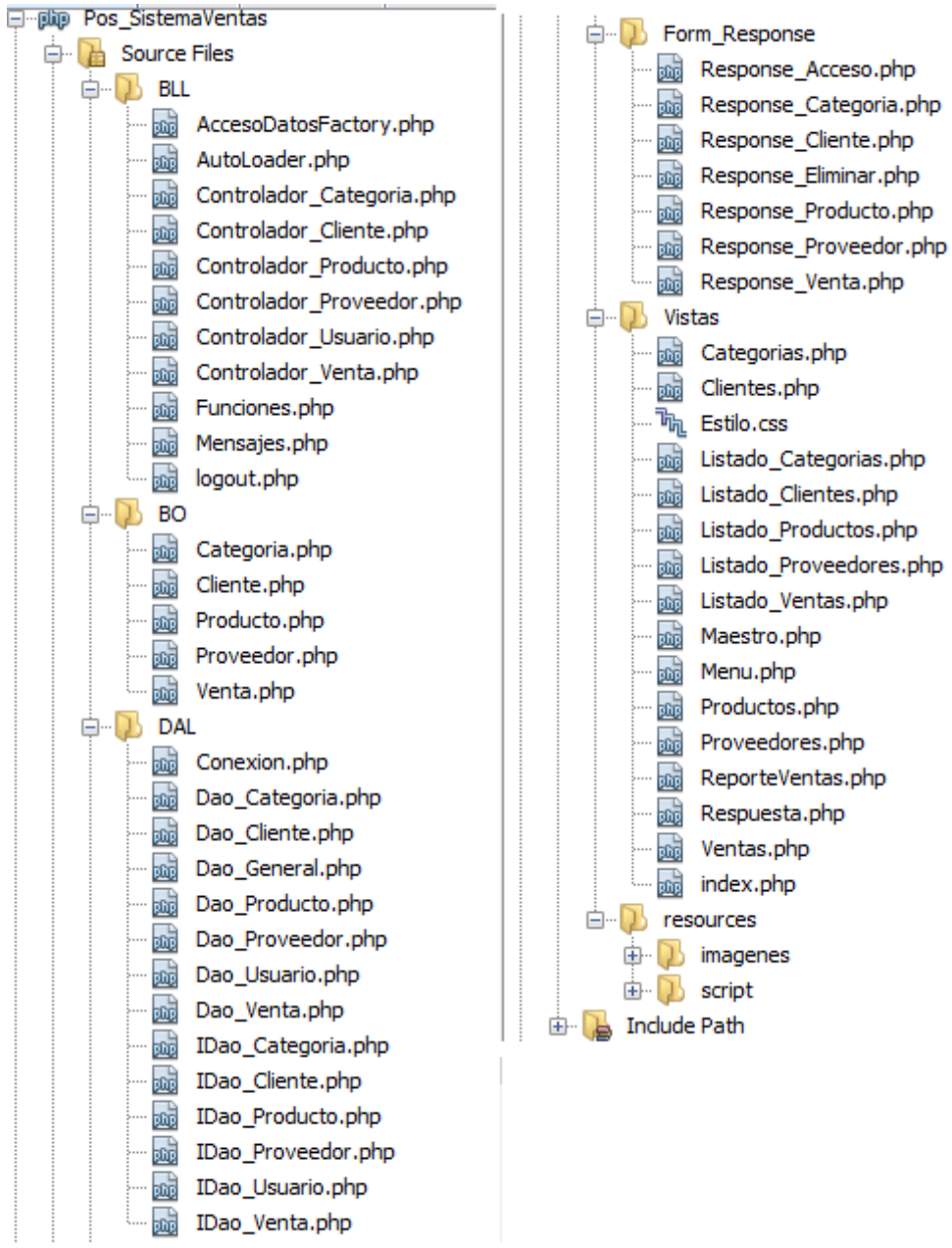
Desde luego esto es solo un ejemplo que se queda corto, pues para que una BD sea capaz de controlar eficientemente, todo lo que implica controlar un punto de venta, tendría que tener más estructuras y por supuesto más columnas en las que se montaron.

No creo que amerite mucha explicación, pero básicamente una venta está relacionada con un cliente y tiene asociado un producto que a su vez tiene relacionada una categoría y un proveedor, por su parte la estructura de usuarios no está relacionada

Se cuidó mucho la confección de los querys que cargan información, para que vayan a la BD solo una vez, y traigan de una todo lo que se requiere, pero por ejemplo en la original cuando generan el reporte de ventas, le envían como parámetros unas fechas y traen la información de ese periodo, y luego vuelven y envían esas mismas fechas dos veces más, para reflejar el total y la utilidad de las ventas, efectuando tres viajes innecesarios a la BD, cuando en una sola ida se puede traer todo como se hizo en esta vez, este aspecto debe ser muy bien cuidado, porque la idea es que se vaya a la BD lo estrictamente necesario, pues no es un solo usuario quien está pegado del sistema, son muchos usuarios a la vez, y cada que se va a ésta a consultar información se genera tráfico de red y lentitud en los sistemas, para eso se pueden construir querys empleando la instrucción Union All que nos permite traer información variada de múltiples tablas, y/o subconsultas anidadas evitando los problemas ya mencionados.

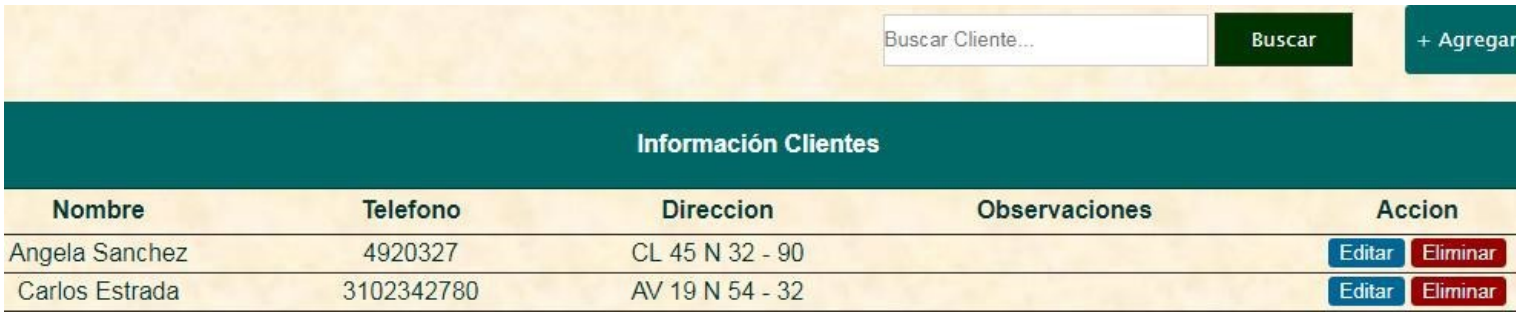
En cuanto al front tenía unas fallas de diseño bien grandes, pues cualquier post lo hacían sobre una pantalla o formulario clonado, desde donde se originaba el post, entonces habían n formularios repetidos que hacían lo mismo, yo separé los formularios de listados para cada estructura, y diseñé un solo formulario por estructura para Insert y Update, reduciendo así el número de formularios a la mitad.
Aparte de un error garrafal en el control del stock, pues si bien descontaban de éste la cantidad involucrada en el pedido, no controlaban que al momento de hacer el pedido, la cantidad no superara la existencia en bodega, pues de lógica no puedo vender de lo que no tengo. La clave es reflejar solo aquellos productos cuya existencia sea mayor de cero, y por supuesto cuidar que cantidad pedida sea menor o igual a cantidad existente; por otra parte no controlaban caracteres no numéricos en datos estrictamente numéricos, permitiendo incluso números negativos y comenzando por cero, entre otros, todo esto se controló en esta ocasión.
Se crearon además clases para cada uno de los objetos de negocio, separando claramente por capas la presentación de este y del acceso a la BD, aplicando MVC dándole un manejo POO, quedando mejor estructurado, organizado y mantenible, porque antes estaba hecho en lo que llamamos "código espagueti", ahora queda distribuido como se ve en las imágenes siguientes:

Proyecto en IDE NetBeans



Menú Principal

Formulario Listado Clientes



Formulario Clientes

Formulario Listado Categorías

Formulario Categorías

Formulario Listado Proveedores

Formulario Proveedores

Formulario Listado Productos

Formulario Productos

Formulario Resumen de ventas

01/02/2019 12/02/2019 Buscar

Resumen de Ventas

Fecha	Cliente	Producto	Cantidad	Venta
2019-02-01 00:00:00	Angela Sanchez	Robalo Asado	2	$ 16000
2019-02-01 00:00:00	Carlos Estrada	Hamburguesa	5	$ 25000

	Ventas
Total Ingresos:	$ 41000

Formulario Listado de selección de Pedidos

Buscar Producto... Busc

Seleccionar Productos

Categoria	Nombre	Precio	Cantidad Stock	Proveedor	Hacer Pedido
Comida Marina	Robalo Asado	4000	28	Inversiones Caceres	Pedir
Comida Marina	Salmon Al Ajillo	5000	20	Inversiones Caceres	Pedir
Comida Marina	Trucha Asada	3000	25	Inversiones Caceres	Pedir
Comidas Rapidas	Hamburguesa	2500	35	Surtitodo	Pedir
Comidas Rapidas	Pizza	3000	35	Surtitodo	Pedir

Formulario Registrar Ventas

Transaccion Venta

Fecha Actual 2019/02/11
Clientes: Carlos Estrada
Categoria Comida Marina
Nombre Producto Robalo Asado
Precio Venta 8000
Cantidad * Cantidad
Monto Total a Pagar — Calcular Total
Ganancia
Monto Cancelado * Monto Recibido — Calcular Cambio
Cambio

Agregar Cancelar