# Gestión para Mueblería



## Introducción:

El sistema de gestión a desarrollar controlará la facturación y el stock de una mueblería, descontando las unidades que se venden del mismo y generando información sobre todos los ingresos por ventas y los movimientos de productos en el depósito. En este caso la mueblería consta de 5 artículos a medida para su venta y en caso de adquirir nuevos productos se podrán añadir a lista.

# Descripción detallada del sistema:

En vista de la gran proliferación de emprendimientos y cuentapropistas en el AMBA desde hace algunos años, este proyecto se propone crear un sistema de gestión simple, intuitivo y de fácil utilización para los usuarios que permita facilitar el proceso de facturación y generación de pedidos. A efectos de cumplir los requisitos de la materia y poner en práctica los conceptos aprendidos, hemos decidido orientar este software de emprendedores a quienes vendan bienes y cuenten con un stock de los mismos, por lo que a medida que el sistema vaya confirmando los pedidos, irá descontando productos del stock y tendrá varios informes y listados configurables para facilitar la gestión de la empresa o emprendimiento.

Los clientes quedarán guardados en una base de datos con todos los datos que permitan la facturación de los pedidos y la entrega de la mercadería. Luego se podrá listar, modificar, dar de baja al Cliente. Por otro lado, los artículos se podrán listar y ordenar por más o menos vendido, cantidad disponible en stock o número de artículos.

Las ventas, que es lo principal de nuestro programa, se tomará un cliente por Nro de cliente y se completarán todos los campos adicionales del "encabezado" de cualquier factura normal, así como también se podrá listarlas, modificarlas e incluso dar de baja. Luego, en el detalle o cuerpo del comprobante se podrán agregar varios ítems del tipo "artículos" los cuales a su vez se irán descontando del stock. El programa podrá reimprimir estos comprobantes ya emitidos en el momento que se desee.

# Salidas del sistema:

El software tendrá una serie de listados y reportes detallados a continuación:

## Listados

### Listado de clientes ordenados por:

- Número de cliente
- Nombre
- Apellido
- DNI
- Teléfono
- Mail
- Localidad
- Dirección

## Listado de ventas ordenadas por:

- Número de factura
- Número de cliente
- Forma de pago
- Comentario
- Fecha
- Monto total
- Seña

### Listado de artículos ordenados por:

- Número de artículo
- Stock
- Descripción del articulo

## Listado específico:

- Número Cliente
- Número Factura
- Por Fecha
- Por Artículo

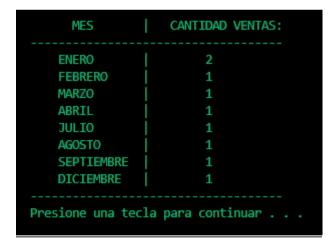
## Reportes

A continuación, se detallan los reportes que el sistema generará:

- Ventas totales
- Clientes totales
- Ventas por Día
- Ventas por Mes
- Ventas por Año
- Recaudación Mensual
- Recaudación Anual
- Recaudación por métodos de pago
- Recaudación por tipo de artículo
- Artículo más vendido
- Articulo menos vendido
- Cerrar Caja

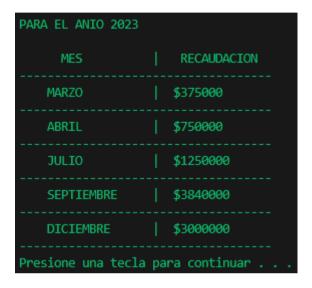
## Ventas por Mes:

El usuario seleccionará un año y podrá ver las ventas realizadas de los meses que tuvieron ventas.



#### • Recaudación Mensual:

El usuario podrá seleccionar un año a su elección donde podrá ver la facturación de todos los meses.



### • Recaudación Anual:

Por otro lado, el usuario podrá seleccionar un año a su elección donde podrá ver la facturación total del año elegido.

```
EL ANIO 2023 TIENE UN TOTAL DE : $9215000

Presione una tecla para continuar . . .
```

## • Recaudación Forma de pago:

El usuario podrá visualizar en más detalle la facturación que el cliente hizo según sus tres tipos de forma de pago.

```
PARA LA FORMA DE PAGO 1 - DEBITO HUBO UNA RECAUDACION DE: $2375000
PARA LA FORMA DE PAGO 2 - TARJETA HUBO UNA RECAUDACION DE: $7790000
PARA LA FORMA DE PAGO 3 - EFECTIVO HUBO UNA RECAUDACION DE: $1120000
Presione una tecla para continuar . . .
```

## • Recaudación por Artículo:

El usuario podrá tener una vista en más detalle de la facturación de cada artículo.

NUMERO ARTICULO	I	RECAUDACION:
1	Ī	\$2020000
2	I	\$1500000
3	I	\$9840000
4	I	\$1250000
5	I	\$5760000
Presione una tecla para continuar		

## Configuraciones

El sistema contará con un menú de configuración que permitirá realizar el backup de los datos a guardar, tanto para ventas como clientes.

## Anexo: Diagrama de clases

-nroCliente: int nombre: char [30] apellido: charf301 -telefono: char (30) -direction: char [100] mail: char[100] dni: char [12] -localidad: char [30] eliminado: bool -estado: bool aetNroCliente(): int getNombre(): char \* getApellido(): char \* getTelefono(): char \* getDireccion(): char \* getMail(): char \* getDni(): char \* getLocalidad(): char \* getEliminado(): bool getEstado(); bool setNroCliente(valor: int): void setNombre(valor: const char \*): void setApellido(valor: const char\*): void setTelefono(valor: const char \*): void setDireccion(valor: const char \*): void setMail(valor: const char \*): void setDni(valor: const char \*): void setLocalidad(valor: const char \*): void setEliminado(valor: bool): void setEstado(valor: bool): void

VentaPedido nroFactura: int nroCliente: int -formaPago: int -fecha: Fecha -comentarios: char [100] montoTotal: long -estado: bool -eliminado: bool getNroFactura(): int getNroCliente(): int getFormaPago(): int getFecha(): Fecha getMontoTotal: long getSenia: long getEstado: bool getEliminado: bool setNroFactura(valor: int): void setNroCliente(valor: int): void setFormaPago(valor: int): void setFecha(valor: Fecha): void setComentarios(valor: const char \*): void setMontoTotal: (valor: long): void setSenia(valor: long): void setEstado(valor: bool): void setEliminado(valor: bool): void

VentaPedidoDetalle nroFactura: int nroArticulo: int -cantidad: int -precio: lona -estado: bool getNroFactura(): int getNroArticulo(): int getCantidad(): int getPrecio(): long getEstado(): bool setNroFactura(valor: int): void setNroArticulo(valor: int): void setCantidad(valor: int): void setPrecio(valor: long): void setEstado(valor: bool): void Fecha -dia: int -mes: int -anio: int

Fecha

-dia: int
-mes: int
-anio: int
setDia(valor: int): void
setMes(valor: int): void
setAnio(
getDia(): int
getMes(): int
getMes(): int

Articulo
-nroArticulo: int
-stock: int
-descripcion: char[50]
-estado: bool
getNroArticulo(): int
getStock(): int
getStock(): int
getDescripcion(): char \*
getEstado(): bool
setNroArticulo(valor: int): void
setStock(valor: int): void
setEbescripcion(valor: const char \*): void
setEstado(valor: bool): void