

INFORME DE PROYECTO

GESTIÓN DE EMPRESA DE VENTA DE CELULARES

1er cuatrimestre - Año 2023



Profesores

Simón, Angel

Lara, Brian

Gonzalez, Juan

Integrantes

Arroyo, Ailén Marina

García, Lucio Agustín

Alvez, Nahuel

Galdós, Iñaki



Contenido

1. Introducción	3
2. Descripción del sistema.....	3
3. Salidas del sistema.....	4
3.1. Listados.....	4
3.2. Consultas.....	5
3.3. Informes.....	5
4. Modelos orientativos de pantalla	5
4.1. Pantallas de menú	5
4.2. Pantallas de informes.....	7
5. Declaración de clases a utilizar	9

1. Introducción

El sistema por desarrollar estará destinado a la gestión de ventas de un local que se dedica al comercio de equipos celulares, tablets y accesorios.

A través de distintos menús se podrán realizar acciones de carga de venta, stock de productos y gestionar accesos de usuarios, a los cuales se les vincularán las ventas que realicen.

2. Descripción del sistema

El objetivo de la aplicación es sistematizar el control de la gestión del negocio. Permitirá cargar y controlar el stock, las ventas, contabilizar las transacciones que haya realizado cada vendedor e informar la recaudación total de la empresa. Para lograr esta misión, se tendrán en cuenta las siguientes características:

- Para trabajar con el programa el usuario requerirá contar con credenciales. Estas credenciales estarán vinculadas a cada usuario, que con acuerdo a su rol tendrá distintos permisos de acceso.
- Cada usuario será individualizado a través de su número de DNI y se le vincularán datos como el Nombre, Apellido, Mail y Fecha en la que se realizó el registro de ese usuario. Al mismo tiempo, contará con un alias y una contraseña, las cuales servirán para poder acceder al sistema y vincular todas aquellas transacciones asociadas a las ventas de productos del local a cada usuario que la haya realizado.
- Cada registro de venta se vinculará al cliente que realice la compra. El cliente será individualizado mediante el tipo de documento y el nro. de documento, esto permitirá registrar a una persona física o a una persona jurídica, por lo que se podrán registrar tanto DNI como nro. de CUIT según corresponda. Al mismo tiempo se almacenarán datos como el Nombre y Apellido, o razón social, mail, datos de domicilio y fecha de registro.

- Cada producto que se cargue en el sistema tendrá una serie de atributos para poder identificarlos y gestionarlos: ID, Categoría, Marca, Nombre, Descripción, Precio de venta y Stock. Los mismos serán listados por ID, Marca y precio y podrán ser consultados por los mismos atributos.
- Cada registro de venta podrá ser cargado mediante la búsqueda de un cliente y en caso de que no exista le permitirá cargarlo para luego continuar con el proceso de venta. Asimismo, se añadirán productos que serán vinculados mediante un ID de venta. Cargar una venta implicará el descuento correspondiente de stock de productos con acuerdo a la cantidad de productos seleccionados para la venta.
- Cada venta realizada generará un comprobante en archivo de texto (.txt), que podrá ser enviado al cliente una vez realizada la transacción.

3. Salidas del sistema

3.1. Listados

Productos

- Ordenado por ID
- Ordenado por Marca
- Ordenado por Precio

Usuarios

- Ordenado por alias
- Filtrados por activos e inactivos

Clientes

- Ordenado por nombre

Ventas

- Ordenado por ID
- Ordenado por fecha de venta (descendente)

3.2. Consultas

Productos

- Por ID
- Por Precio
- Por Marca

Usuarios

- Por Alias
- Por Nro. de documento
- Por Búsqueda libre

Clientes

- Por Nombre
- Por Nro. De DNI / CUIT
- Por Búsqueda libre

Ventas

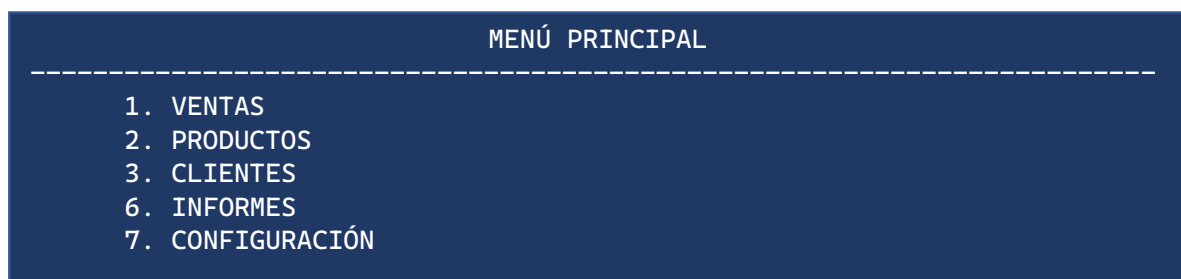
- Por ID
- Por importe
- Por fecha

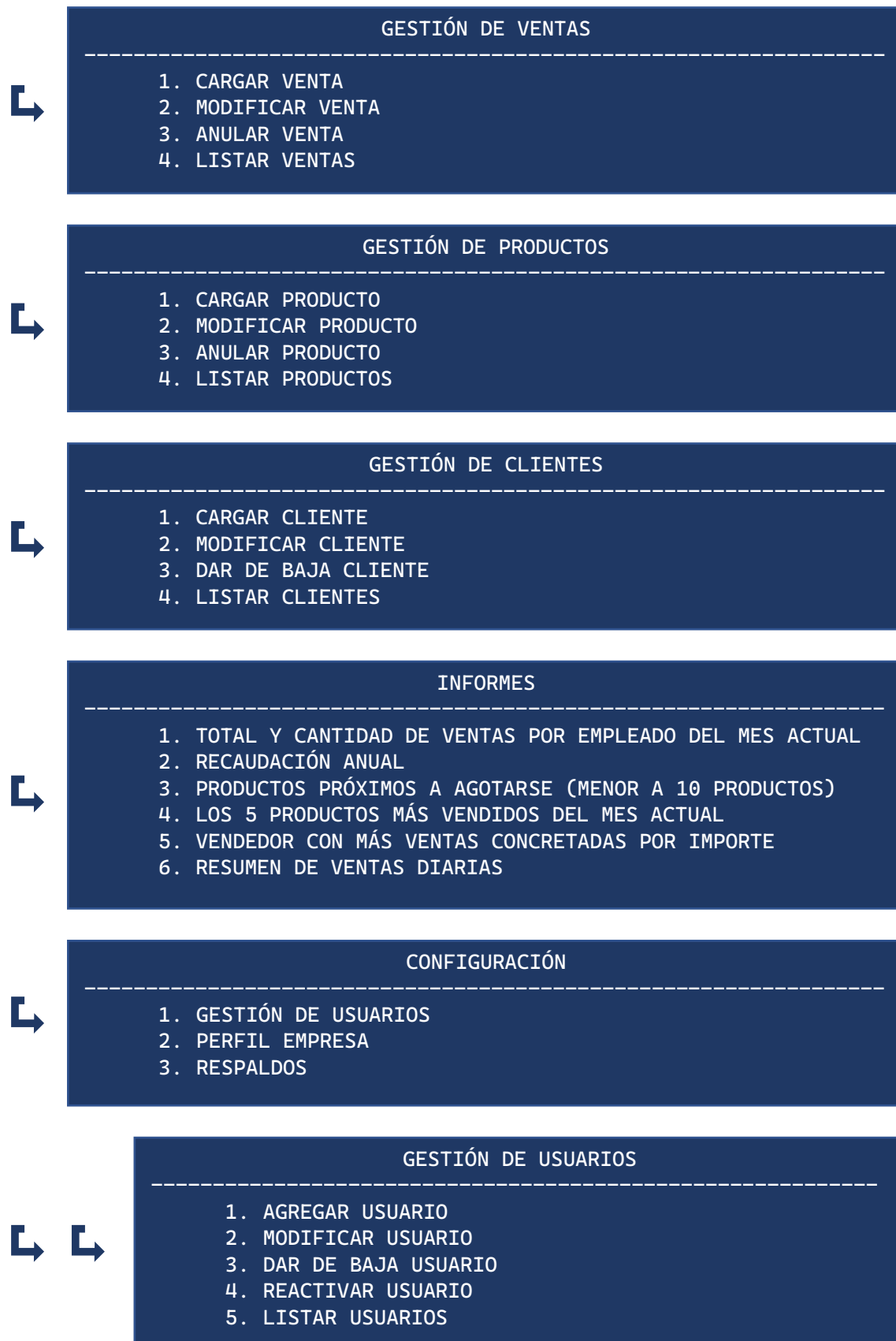
3.3. Informes

- Importe total y cantidad de ventas por empleado durante el mes actual
- Recaudación anual (con detalle mensual)
- Productos próximos a agotarse (Menor a 10 productos)
- 5 productos más vendidos del mes actual
- Vendedor con más ventas concretadas por importe
- Resumen de ventas diarias

4. Modelos orientativos de pantalla

4.1. Pantallas de menú





4.2. Pantallas de informes

***** SISTEMA DE GESTION DE STOCK Y VENTAS *****				

IMPORTE TOTAL DE VENTAS POR EMPLEADO DEL MES ACTUAL				

ID EMPLEADO	NOMBRE	APELLIDO	CANTIDAD DE VENTAS	IMPORTE

40360789	PABLO	MARTINEZ	12	\$1.080.000
22358481	LUCAS	GALVAN	15	\$1.610.400
26361419	ROCIO	AYALA	12	\$980.850
30350585	MARTINA	DIAZ	24	\$2.184.000
39360810	ISMAEL	ESPOSITO	20	\$1.780.500

***** SISTEMA DE GESTION DE STOCK Y VENTAS *****	

RECAUDACION ANUAL DE VENTAS DEL AÑO 2022	

MES	IMPORTE

ENERO	\$654.000
FEBRERO	\$886.500
MARZO	\$815.000
ABRIL	\$1.502.000
MAYO	\$1.560.000
JUNIO	\$1.854.400
JULIO	\$2.354.000
AGOSTO	\$1.969.000
SEPTIEMBRE	\$1.541.900
OCTUBRE	\$1.554.000
NOVIEMBRE	\$1.988.500
DICIEMBRE	\$2.659.000

***** SISTEMA DE GESTION DE STOCK Y VENTAS *****				

DETALLE DE PRODUCTOS PROXIMOS A AGOTAR STOCK				

ID PRODUCTO	MARCA	MODELO	STOCK ACTUAL	CATEGORIA

303030	SAMSUNG	A12	8	CELULARES
303033	SAMSUNG	A20	6	CELULARES
303045	SAMSUNG	S50	9	CELULARES
303010	SAMSUNG	S32	5	CELULARES
303040	MOTOROLA	G40	8	CELULARES
303070	MOTOROLA	G42	6	CELULARES
303011	IPHONE	12 PRO MAX	5	CELULARES
303030	IPHONE	10	4	CELULARES
404030	SAMSUNG	A7	8	TABLETS
404070	SAMSUNG	S6	7	TABLETS
202070	TECH	FUNDA SAMSUNG A20	5	ACCESORIOS
202082	TECH	CARGADOR TIPO C	5	ACCESORIOS

***** SISTEMA DE GESTION DE STOCK Y VENTAS *****

TOP 5 PRODUCTOS MAS VENDIDOS DEL MES ACTUAL

ID PRODUCTO	MARCA	MODELO	CANTIDAD VENDIDA (Unidades)
303030	IPHONE	10	120
303011	IPHONE	12 PRO MAX	110
303070	MOTOROLA	G42	86
303010	SAMSUNG	S32	81
303030	SAMSUNG	A12	79

***** SISTEMA DE GESTION DE STOCK Y VENTAS *****

VENDEDOR CON MAYOR IMPORTE DE VENTAS POR MES Y ANIO INGRESADO

ID EMPLEADO	NOMBRE	APELLIDO	IMPORTE
30350585	MARTINA	DIAZ	\$2.184.000

***** SISTEMA DE GESTION DE STOCK Y VENTAS *****

RESUMEN DE VENTAS DIARIAS

FECHA	CANTIDAD DE PRODUCTOS	MEDIO DE PAGO	IMPORTE
30/5/2023	2	EFFECTIVO	\$150.000
30/5/2023	1	EFFECTIVO	\$135.000
30/5/2023	3	TARJETA DE CREDITO	\$280.000
30/5/2023	1	TARJETA DE CREDITO	\$195.000
30/5/2023	2	TRANSFERENCIA	\$6.000
TOTAL:			\$766.000

5. Declaración de clases a utilizar

Persona
<ul style="list-style-type: none"> - int _tipoDocumento; - char _nroDocumento[12]; - char _nombre[30]; - char _apellido[30]; - char _email[320]; - bool _estado; - Fecha _fechaRegistro;
<ul style="list-style-type: none"> + Persona(int tipoDocumento = 0, const char* nroDocumento = "null", const char* nombre = "null", const char* apellido = "null", const char* email = "null", bool estado = "false", Fecha fechaRegistro = Fecha()); + void Mostrar(); + void setTipoDocumento(int tipoDocumento); + void setNroDocumento(const char* nroDocumento); + void setNombre(const char* nombre); + void setApellido(const char* apellido); + void setEmail(const char* email); + void setEstado(bool estado); + void setFechaRegistro(Fecha fechaRegistro); + int getTipoDocumento(); + std::string getTipoDocumentoDescripcion(); + const char* getNroDocumento(); + const char* getNombre(); + const char* getApellido(); + const char* getEmail(); + bool getEstado(); + Fecha getFechaRegistro();

Usuario
<ul style="list-style-type: none"> - char _alias[15]; - char _contrasenia[15]; - int _rol;
<ul style="list-style-type: none"> + Usuario(int tipoDocumento = 0, const char* nroDocumento = "null", const char* nombre = "null", const char* apellido = "null", const char* email = "null", bool estado = false, Fecha fechaRegistro = Fecha(), const char* alias = "null", const char* contrasenia = "null", int rol = -1); + void setAlias(const char* alias); + void setContrasenia(const char* contrasenia); + void setRol(int rol); + const char* getAlias(); + const char* getContrasenia(); + int getRol(); + std::string getRolDescripcion();

UsuarioArchivo

```

- UsuarioArchivo(std::string nombre);

+ int buscar(int tipoDocumento, std::string nroDocumento);
+ int buscar(std::string alias);
+ bool crear(Usuario reg);
+ Usuario leer(int posicion);
+ bool modificar(Usuario reg, int posicion);
+ bool darDeBaja (Usuario reg, int posicion);
+ void setNombre(std::string nombre);
+ int getCantidadDeUsuarios();
    
```

UsuarioManager

```

- UsuarioArchivo _archivo;
- bool _permisos[3];
- void listar(Usuario usuario, int tipoListado);
- void listarPorAlias();
- void listarPorDNI();
- void listarActivos();
- void listarInactivos();

+ void cargar();
+ void listarUsuarios();
+ void ordenarPorAlias();
+ void modificar();
+ void darDeBaja();
+ void setPermisos(bool adm = false, bool sup = false, bool ven = false);
    
```

UsuarioMenu

```

- UsuarioManager _usuarioManager;

+ void mostrar();
    
```

Cliente

```

- char _domicilio[15];
- char _localidad[15];
- int _provincia[15];

+ Cliente(int tipoDocumento = 0, const char* nroDocumento = "null", const char* nombre = "null", const char* apellido = "null", const char* email = "null", bool estado = false, Fecha fechaRegistro = Fecha(), const char* domicilio = "null", const char* localidad = "null", provincia = "null");
+ void setDomicilio(const char* domicilio);
+ void setLocalidad(const char* localidad);
+ void setProvincia(const char* provincia);
+ const char* getDomicilio();
+ const char* getLocalidad();
+ const char* getProvincia();
    
```

ClienteArchivo

```

-   std::string _ruta;

+   ClienteArchivo();
+   ClienteArchivo(std::string ruta);
+   int getCantidadRegistros();
+   bool guardar(Cliente reg);
+   bool guardar(Cliente reg, int posicion);
+   Cliente leer(int posicion);
+   int buscar(int tipoDocumento, std::string
nroDocumento);

```

ClienteManager

```

-   int _permisos[3];
-   ClienteArchivo _archivo;
-   ClienteArchivo _archivoBkp;
-   int generarID();

+   void Cargar();
+   void Modificar();
+   void Eliminar();
+   void Listar (Producto producto);
+   void listarPorNombre();
+   void HacerCopiaDeSeguridad();
+   void RestaurarCopiaDeSeguridad();

```

ClienteMenu

```

-   ClienteManager _clienteManager;

+   void mostrar();

```

Fecha

```

-   int _dia;
-   int _mes;
-   int _anio;
-   void AgregarDia();
-   void RestarDia();
-   bool esBisiest(int anio);

+   Fecha(int dia = 1, int mes = 1, int anio =
2023);
+   void Mostrar();
+   void AgregarDias(int dias);
+   std::string toString(std::string
formatoFecha = "DD/MM/YYYY");
+   Fecha fechaActual();
+   void setDia(int dia);
+   void setMes(int mes);
+   void setAnio(int anio);
+   int getDia();
+   int getMes();
+   int getAnio();

```

Producto
<pre> - int _ID; - int _categoria; - Marca _marca; - char _modelo[30]; - char _descripcion [50]; - float _precioVenta; - int _stock; - bool _activo; </pre>
<pre> + Producto(int ID = 0, Marca marca=Marca(), const char* modelo = "null", const char* descripcion= "null", float precioVenta, int stock = 0, bool activo = false); + void Mostrar(); + void setID(int ID); + void setCategoria (int categoria); + void setMarca (Marca marca); + void setModelo(const char* modelo); + void setDescripcion(const char* descripcion); + void setPrecio (floatar precioVenta); + void setStock (int stock); + void setActivo (bool activo); + int getID(); + int getCategoria (); + Marca getMarca (); + const char* getModelo(); + const char* getDescripcion(); + float getPrecio (); + int getStock (); + bool getActivo (); </pre>

ProductoArchivo
<pre> - std::string _ruta; </pre>
<pre> + ProductoArchivo(); + ProductoArchivo(std::string ruta); + int getCantidadRegistros(); + bool guardar(Producto reg); + bool guardar(Producto reg, int posRegAREemplazar); + Producto leer(int nroRegistro); + int buscar(int ID); + void vaciar(); </pre>

ProductoManager
<ul style="list-style-type: none"> - int _permisos[3]; - ProductoArchivo _archivo; - ProductoArchivo _archivoBkp; - int generarID();
<ul style="list-style-type: none"> + void Cargar(); + void Modificar(); + void Eliminar(); + void Listar (Producto producto); + void ModificarStock(); + void listarXID(); + void listarXMarca(); + void listarXTopePrecio(); + void listarOrdenadoXStock(); + void listarXProxAgotarse(); + void listarXMasVendidos(); + void HacerCopiaDeSeguridad(); + void RestaurarCopiaDeSeguridad();

ProductoMenu
<ul style="list-style-type: none"> - ProductoManager _productoManager;
<ul style="list-style-type: none"> + void Mostrar();

Marca
<ul style="list-style-type: none"> - int _ID; - char _nombre[30];
<ul style="list-style-type: none"> + Marca(int ID = 0, const char* modelo = "null"); + void setID(int ID); + void setNombre(const char* nombre); + int getID(); + const char* getNombre();

MarcaArchivo
<ul style="list-style-type: none"> - std::string _ruta;
<ul style="list-style-type: none"> + ProductoMarca(); + ProductoMarca(std::string ruta); + int getCantidadRegistros(); + bool guardar(Marca reg); + bool guardar(Marca reg, int posRegAReemplazar); + Marca leer(int nroRegistro); + int buscar(int ID); + void vaciar();

MarcaManager

```

-   int _permisos [3];
-   MarcaArchivo _archivo;
-   MarcaArchivo _archivoBkp;
-   int generarID();

+   void Cargar();
+   void Modificar();
+   void Eliminar();
+   void Listar (Marca marca);
+   void ModificarStock();
+   void listarXID();
+   void HacerCopiaDeSeguridad();
+   void RestaurarCopiaDeSeguridad();
    
```

Venta

```

-   int _IDPedido;
-   Cliente _cliente;
-   Fecha _fechaCompra;
-   Producto _productos[10]
-   int _unidadesXProducto[10]
-   float _montoTotal;
-   int _metodoPago;

+   Venta();
+   Venta( int IDPedido, Cliente cliente, Fecha
    fechaCompra, Producto productos[10], int
    unidadesXProducto[10], float montoTotal,
    int metodoPago[3]);
+   void setIDPedido(int IDPedido);
+   void setCliente(Cliente cliente);
+   void setFechaCompra(Fecha fechaCompra);
+   void setProducto(int pos, Producto
    producto);
+   void setUnidadesXProducto(int pos, int
    unidades);
+   void setMontoTotal(float montoTotal);
+   void setMetodoPago(int metodoPago)
+   int getIDPedido();
+   Cliente getCliente();
+   Fecha getFechaCompra();
+   Producto getProducto(int pos);
+   int getUnidadesXProducto(int pos);
+   float getMontoTotal();
+   int getMetodoPago()
    
```

VentaArchivo

```

-   std::string _ruta;

+   VentaArchivo();
+   VentaArchivo(std::string ruta);
+   int getCantidadRegistros();
+   bool guardar(Venta reg);
+   bool guardar(Venta reg, int
        posRegAreemplazar);
+   Venta leer(int nroRegistro);
+   int buscar(int ID);
+   void vaciar();
    
```

VentaManager

```

-   int _permisos[3];
-   VentaArchivo _archivo;
-   VentaArchivo _archivoBkp;
-   int generarID();

+   void Cargar();
+   void Modificar();
+   void Eliminar();
+   void Listar (Venta venta);
+   void ventasXEmpleado();
+   void recaudacionAnual();
+   void listarVentasXMedioPago();
+   void HacerCopiaDeSeguridad();
+   void RestaurarCopiaDeSeguridad();
    
```

VentaMenu

```

-   VentaManager _ventaManager;

+   void Mostrar();
    
```