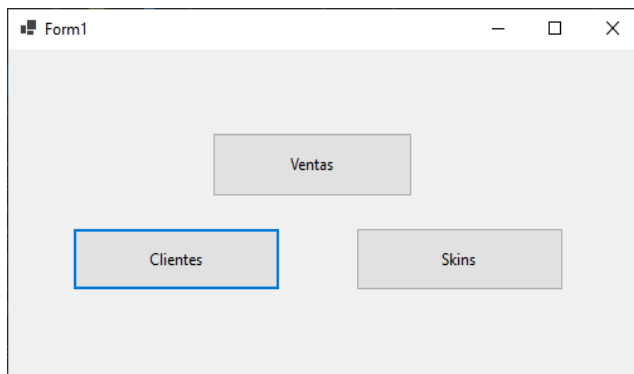


## Trabajo Practico Nro. 3: Documento Funcional

**Temática:** Tienda de skins de Counter Strike: Global Offensive.

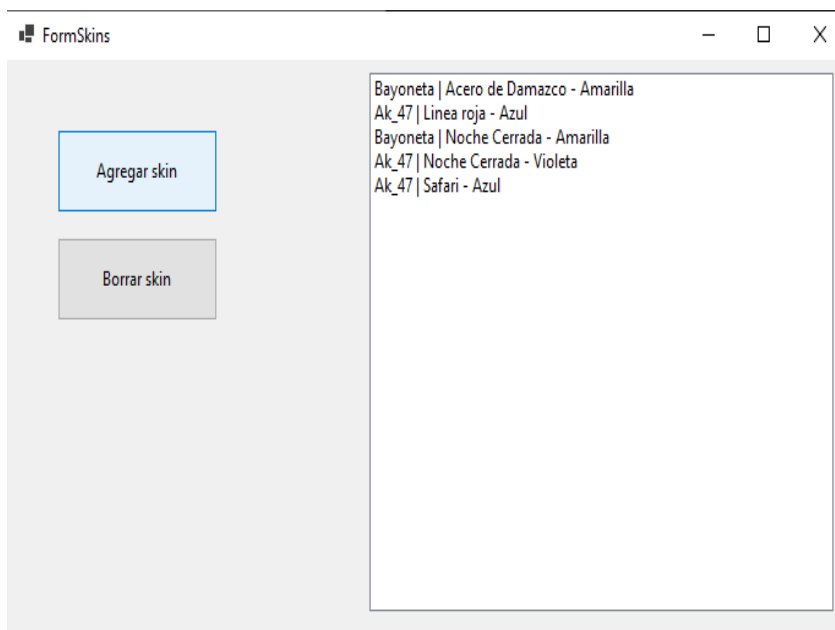
La funcionalidad pretendida es el manejo de **Cientes**, **Armas** y **Ventas**.

Al iniciar la aplicación lo primero que se ve es un menú principal, cada botón deriva al respectivo formulario.



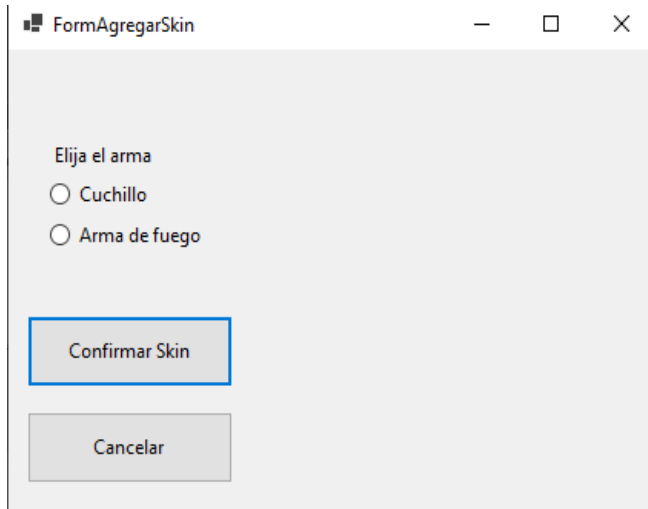
### Skins

Al ingresar al form de Skins se cargan 2 botones y una ListBox la cual contiene los elementos del archivo Serializado “SerializacionArmasXML\_lista.xml”



### **Botón Agregar Skin:**

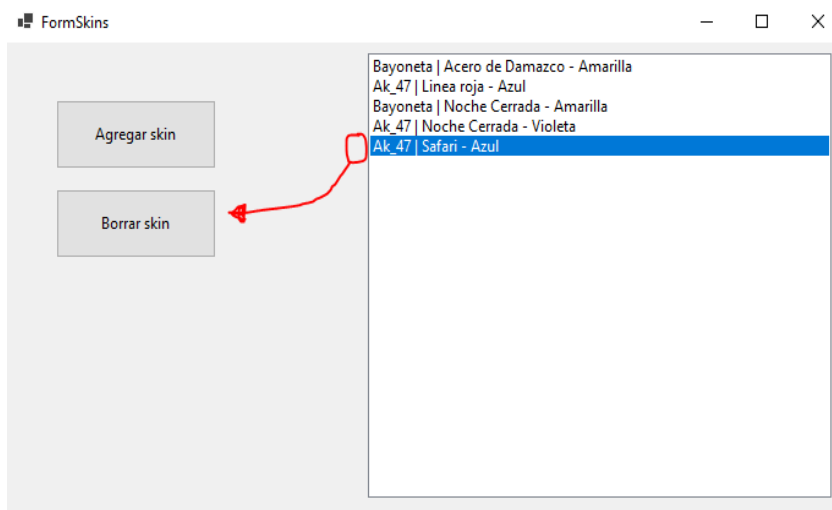
Al clicar el botón se carga otro formulario en el cual debemos cargar los datos que se piden para generar una nueva skin, para confirmar la skin que cargamos debemos clicar “Confirmar Skin” o, caso contrario, Cancelar para evitar la carga de la skin.



The screenshot shows a Windows-style dialog box titled "FormAgregarSkin". It contains two radio buttons under the label "Elija el arma": "Cuchillo" and "Arma de fuego". Below the radio buttons are two buttons: "Confirmar Skin" and "Cancelar". The "Confirmar Skin" button is highlighted with a blue rectangular border.

### **Botón Borrar Skin:**

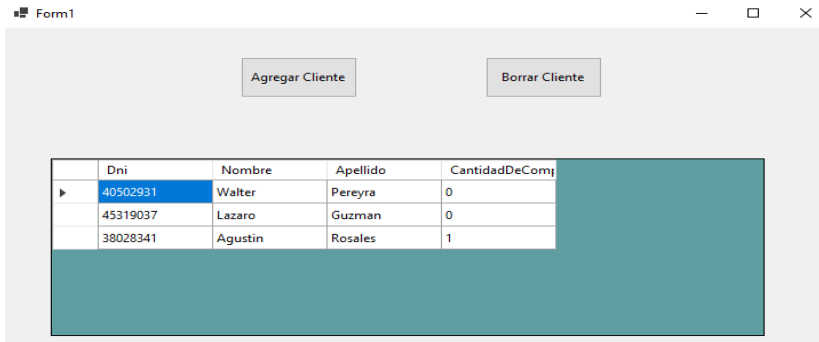
Para borrar una skin debemos primero seleccionarla en el ListBox y luego clicar el botón respectivamente.



The screenshot shows a Windows-style window titled "FormSkins". On the left, there are two buttons: "Agregar skin" and "Borrar skin". On the right, there is a list box containing five items: "Bayoneta | Acero de Damazco - Amarilla", "Ak\_47 | Linea roja - Azul", "Bayoneta | Noche Cerrada - Amarilla", "Ak\_47 | Noche Cerrada - Violeta", and "Ak\_47 | Safari - Azul". The last item, "Ak\_47 | Safari - Azul", is selected and highlighted in blue. A red arrow points from the selected item in the list box to the "Borrar skin" button.

## Cientes

Al ingresar al apartado de clientes se carga un formulario el cual contiene 2 botones y un DataGridView el cual carga la lista de Clientes del archivo Serializado “SerializacionClientesJSONlistaClientes”

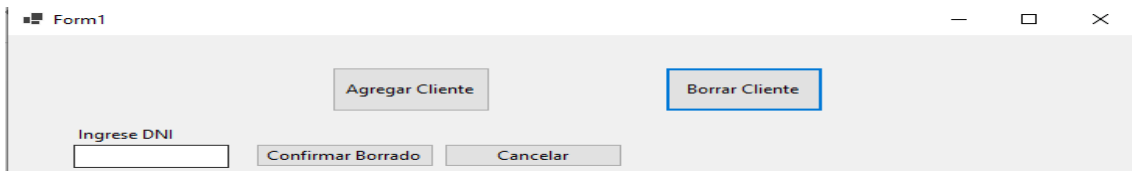


The screenshot shows a Windows form titled 'Form1'. At the top, there are two buttons: 'Agregar Cliente' and 'Borrar Cliente'. Below them is a DataGridView table with the following data:

	Dni	Nombre	Apellido	CantidadDeCom
▶	40502931	Walter	Pereyra	0
	45319037	Lazaro	Guzman	0
	38028341	Agustin	Rosales	1

### Botón Borrar Cliente:

Al clickear el botón aparecerá un TextBox donde se deberá ingresar el DNI del cliente a borrar en cuestión, junto a 2 botones, uno para confirmar el borrado el cual eliminara de la lista el cliente si el DNI coincide con el de algún cliente y un botón para cancelar que ocultara nuevamente el apartado de borrado.

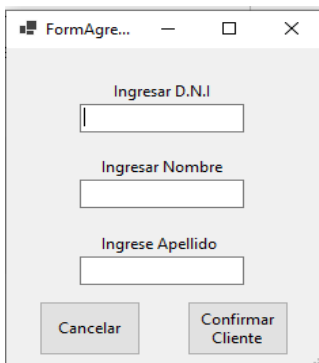


The screenshot shows the 'Form1' window with the 'Borrar Cliente' button highlighted. Below it, a sub-form is visible with the following elements:

- A text box labeled 'Ingrese DNI'.
- Two buttons: 'Confirmar Borrado' and 'Cancelar'.

### Boton Agregar Cliente:

Al clickear el botón se cargara el formulario correspondiente a la carga de datos del nuevo Cliente, luego de rellenar todos los campos se debe clickear en el botón Confirmar Cliente o, caso contrario, click en Cancelar para detener la carga en cuestión



The screenshot shows a sub-form titled 'FormAgre...'. It contains the following elements:

- A text box labeled 'Ingresar D.N.I'.
- A text box labeled 'Ingresar Nombre'.
- A text box labeled 'Ingresar Apellido'.
- Two buttons: 'Cancelar' and 'Confirmar Cliente'.

## Ventas

Al ingresar al apartado de Ventas carga un formulario el cual contiene 2 botones y un DataGridView el cual carga la lista de Ventas del archivo Serializado “SerializacionVentasXML\_ventas”

The screenshot shows a window titled 'Form1' with two buttons at the top: 'Generar Ticket' and 'Agregar Venta'. Below the buttons is a DataGridView containing the following data:

	FechaYHora	Dni	Nombre	Apellido	NombreTipoArm	CalidarArma	NombreSkin	Precio
▶	04/06/2022 07:0...	45319037	Lazaro	Guzman	Ak_47	Azul	Linea roja	2500
	04/06/2022 07:0...	38028341	Agustin	Rosales	Karambit	Amarilla	Bosque Boreal	2500
	04/06/2022 07:2...	43576834	Hugo	Sanchez	Desert_Eagle	Rosa	Hipnotico	2500
	04/06/2022 11:0...	38028341	Agustin	Rosales	Bayoneta	Amarilla	Noche Cerrada	2500
	05/06/2022 05:5...	45319037	Lazaro	Guzman	Ak_47	Azul	Safari	2500
	05/06/2022 06:0...	45319037	Lazaro	Guzman	Bayoneta	Amarilla	Acero de Dama...	2500

### Botón Generar Ticket:

Para generar un ticket debemos seleccionar una de las filas previo al click de dicho botón

This screenshot shows the same 'Form1' window. A red arrow points from the 'Generar Ticket' button to the second row of the DataGridView, which is highlighted in blue. The data in the DataGridView is the same as in the previous screenshot.

### Boton Agregar Venta:

Al clickear el botón se abre un nuevo formulario en el cual al igual que en el generar ticket debemos seleccionar una fila de armas y una fila de clientes, acto seguido Confirmar Venta.

The screenshot shows a new window titled 'FormAgregarVenta'. It contains two DataGridViews and a 'Confirmar Venta' button. The first DataGridView on the left lists clients, and the second DataGridView on the right lists weapons. Red arrows indicate the selection of a client and a weapon.

	Dni	Nombre	Apellido	CantidadDeComi
▶	40502931	Walter	Pereyra	0
	45319037	Lazaro	Guzman	0
	38028341	Agustin	Rosales	1

	TipoCalidad	TipoArma	NombreSkin	Precio
	Amarilla	Bayoneta	Acero de Dama...	2500
▶	Azul	Ak_47	Linea roja	2500
	Amarilla	Bayoneta	Noche Cerrada	2500
	Violeta	Ak_47	Noche Cerrada	2500
	Azul	Ak_47	Safari	2500

### **Trabajo Practico Nro. 3: Lista temas**

**Generics:** Aplicado en la entidad "Serializacion", clase genérica "Serializadora.cs", métodos "Escribir", "EscribirVentas", "Leer", "EscribirJson", "LeerJson".

**Exceptions:** Hay diversos try-catch genericos repartidos en los diversos Forms y Entidades, por dar un ejemplo hay uno aplicado en el forms "FormVentas", form "FormAgregarVenta.cs", método "btnConfirmarVenta\_Click". Pero a su vez hay una Excepcion personalizada en la entidad "Excepciones", clase "CampoVacioException.cs", la cual se aplica, por ejemplo, en forms "FormSkins", form "FormAgregarSkin.cs", método "btnConfirmarSkin\_Click".

**Archivos y Serializacion:** La lógica de la Serializacion la aplico en la ruta escrita en Generics, y su aplicación la hago en los forms "FormVentas", "FormClientes", "FormSkins" y en sus respectivos formularios ya que cada modificación, carga, borrado, etc de las listas se refleja automáticamente en los archivos, por nombrar un ejemplo de su aplicación ir a forms "FormVentas", form "FormAgregarVenta.cs", método "btnConfirmarVenta\_Click".

La lógica de Archivos la aplico en la entidad "Serializacion", clase "Archivos.cs", y su aplicación la realizo en forms "FormVentas", form "FormVentas.cs", método "btnCrearTicket\_Click".

**Interfaces:** Se encuentra definido en la entidad "Serializacion", interfaz "IDato.cs" y se aplica en "Serializacion", clase "Archivos.cs", método "AplicarInterface", al estar contrarreloj con el tp no pude realizar una aplicación más funcional de interfaces por lo cual realice ello para "cubrir" el tema.

**UnitTesting:** Se encuentra en la entidad “UnitTesting”, clase “Testeos.cs” y evalua que:

- Se agregue un arma correctamente a la lista.
- No se agregue a la lista de Clientes un cliente con campos en null.