

## Trabajo Practico Nro. 3: Documento Funcional

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

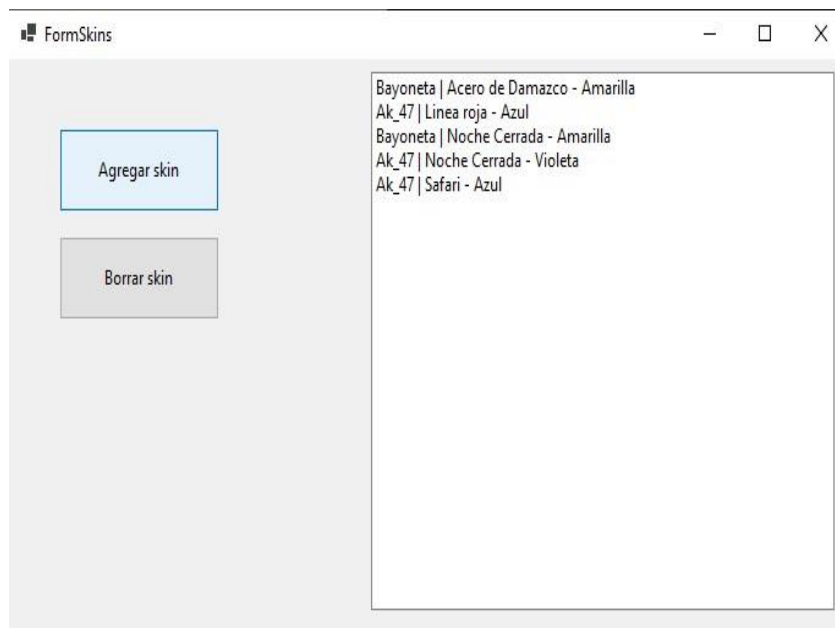
La funcionalidad pretendida es el manejo de **Cientes**, **Armas** y **Ventas**.

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



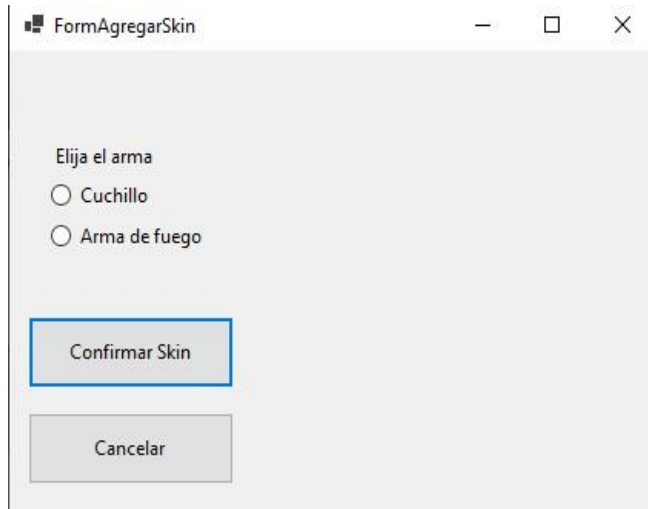
### Skins

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:**

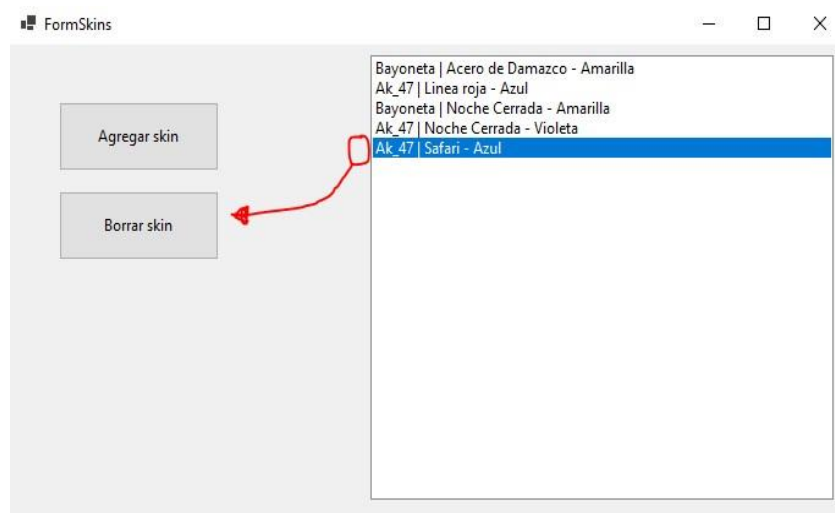
Al clicar el botón se carga otro formulario en el cual debemos cargar los datos que se piden para generar una nueva skin, para confirmar la skin que cargamos debemos clicar “Confirmar Skin” o, caso contrario, Cancelar para evitar la carga de la skin.



The screenshot shows a window titled "FormAgregarSkin" with a standard Windows title bar (minimize, maximize, close buttons). Inside the window, there is a label "Elija el arma" followed by two radio button options: "Cuchillo" and "Arma de fuego". Below these options are two buttons: "Confirmar Skin" (highlighted with a blue border) and "Cancelar".

### **Botón Borrar Skin:**

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



The screenshot shows a window titled "FormSkins" with a standard Windows title bar. On the left side, there are two buttons: "Agregar skin" and "Borrar skin". On the right side, 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:

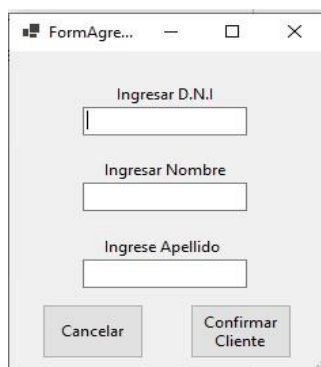
Al clickear el botón aparecerá un TextBox donde se deberá ingresar el DNI del cliente a borrar en cuestión, junto a 2 botones, uno para confirmar el borrado el cual eliminara de la lista el cliente si el DNI coincide con el de algún cliente y un botón para cancelar que ocultara nuevamente el apartado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



The screenshot shows the 'Form1' window with the 'Borrar Cliente' button highlighted. Below it, a text box labeled 'Ingresa DNI' is visible, along with 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 new form titled 'FormAgre...'. It contains three text boxes for input: 'Ingresa D.N.I.', 'Ingresa Nombre', and 'Ingresa Apellido'. At the bottom, there are 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 Windows form titled 'Form1'. At the top, there are two buttons: 'Generar Ticket' and 'Agregar Venta'. Below them 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, se generan en la ruta:

[TP4\Formularios\bin\Debug\net5.0-windows\Tickets](#)

This screenshot is similar to the previous one, but 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 form titled 'FormAgregarVenta'. It contains two DataGridViews and a 'Confirmar Venta' button. The first DataGridView on the left lists clients:

	Dni	Nombre	Apellido	CantidadDeComi
▶	10502931	Walter	Pereyra	0
	45319037	Lazaro	Guzman	0
	38028341	Agustin	Rosales	1

The second DataGridView on the right lists weapons:

	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

A red arrow points from the 'Confirmar Venta' button to the selected row in the second DataGridView.

### Trabajo Practico Nro. 3: Lista temas

**Generics:** Aplicado en la clase "Serializadora.cs" >  
"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" > "btnConfirmarVenta\_Click".  
Pero a su vez hay una Excepcion personalizada en la entidad "Excepciones" > "CampoVacioException.cs", la cual se aplica, por ejemplo, en forms "FormSkins" >  
"FormAgregarSkin.cs" > "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" >  
"FormAgregarVenta.cs" > "btnConfirmarVenta\_Click".  
La lógica de Archivos la aplico en la entidad "Serializacion" > "Archivos.cs", y su aplicación la realizo en forms "FormVentas", form "FormVentas.cs", método "btnCrearTicket\_Click".

**Interfaces:** Se encuentra definido en "Formularios" > "IVolverMenuPrincipal" y se aplica en los forms de Menu (FormMenuClientes, FormMenuVentas, FormMenuArmas) debiendo aplicar un método que vuelva hacia el Menu Principal (FormMenuPrincipal).

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

-Se agregue un arma correctamente a la lista.

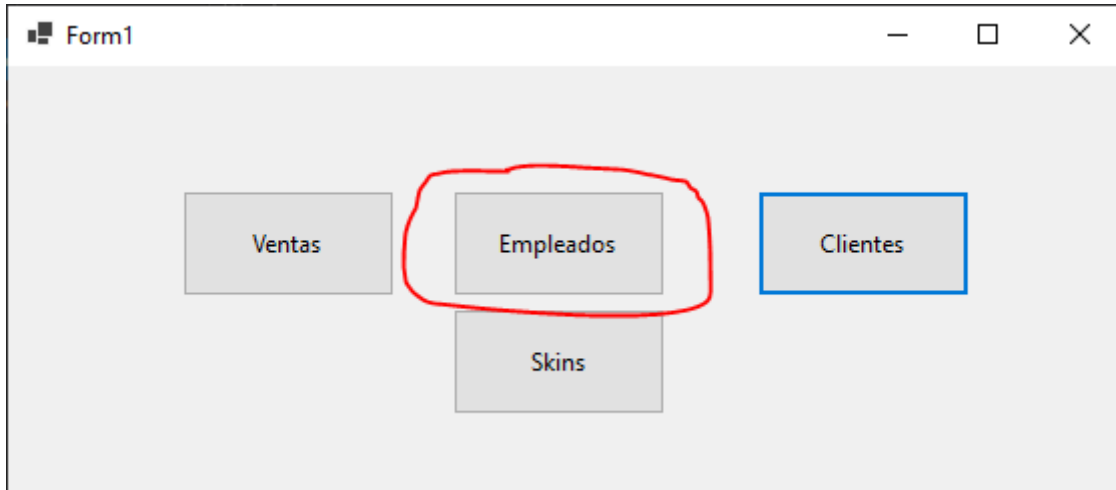
-No se agregue a la lista de Clientes un cliente con campos en null.

### **Correcciones TP3:**

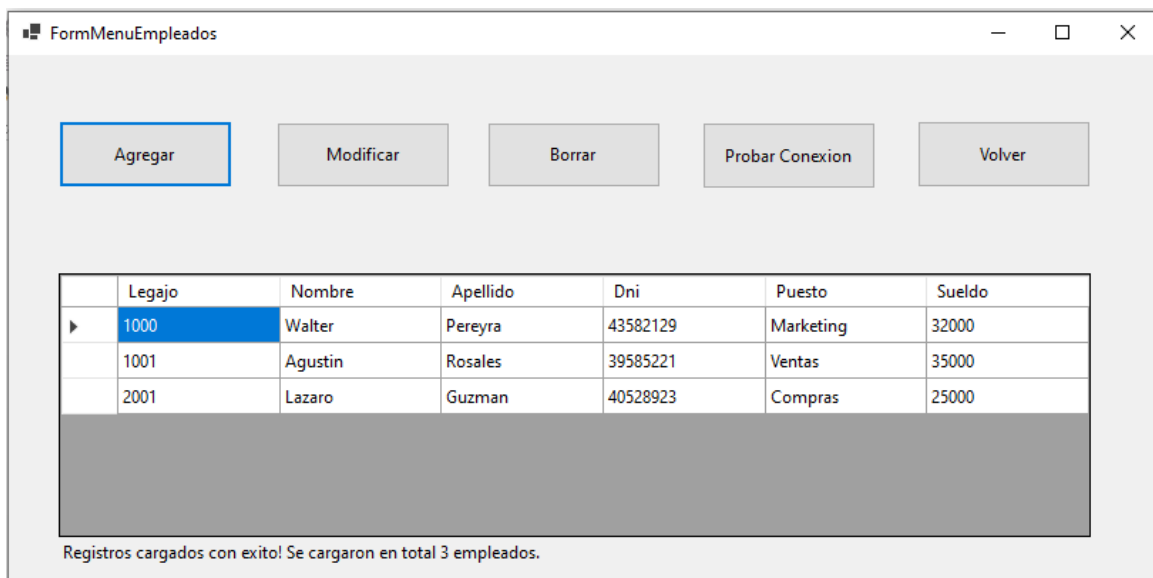
- Aclaración de la ruta donde se genera el ticket en este documento.
- En el form donde se agregan skins al no ingresar un número sale el messagebox de la misma manera pero ya no cierra el form.
- Todos los forms ahora están dentro de la misma entidad.
- En el constructor de clientes ya no está el try catch vacío.
- Se eliminó la interface y se reemplazó por la mencionada en el tema

## Trabajo Practico Nro. 4

Para esta extensión del TP se añadió una nueva funcionalidad, la cual está centrada en el manejo de **Empleados**.



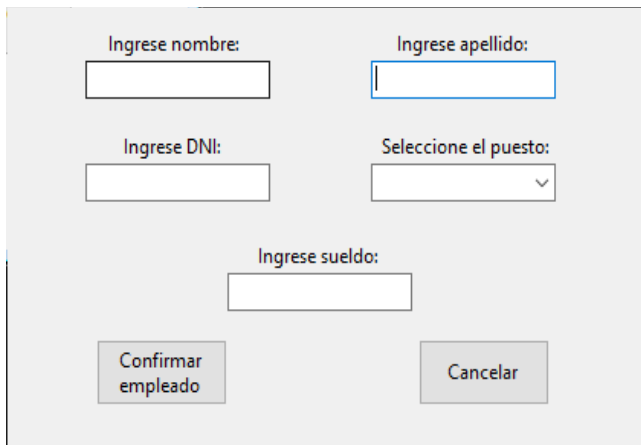
Dicho apartado tiene como objetivo principal integrar el tema de SQL, ya que la lista cargada en el DataGridView proviene de una base de datos, y la funcionalidad de los botones es simplemente un CRUD sin la opción de lectura ya que la carga se hace automáticamente al ingresar al apartado de Empleados.

A screenshot of a Windows application window titled 'FormMenuEmpleados'. The window has a standard Windows title bar. Below the title bar, there is a horizontal row of five buttons: 'Agregar', 'Modificar', 'Borrar', 'Probar Conexion', and 'Volver'. Below these buttons is a DataGridView containing three rows of employee data. The first row is highlighted in blue. Below the DataGridView, there is a message box that says 'Registros cargados con exito! Se cargaron en total 3 empleados.'

	Legajo	Nombre	Apellido	Dni	Puesto	Sueldo
▶	1000	Walter	Pereyra	43582129	Marketing	32000
	1001	Agustin	Rosales	39585221	Ventas	35000
	2001	Lazaro	Guzman	40528923	Compras	25000

Registros cargados con exito! Se cargaron en total 3 empleados.

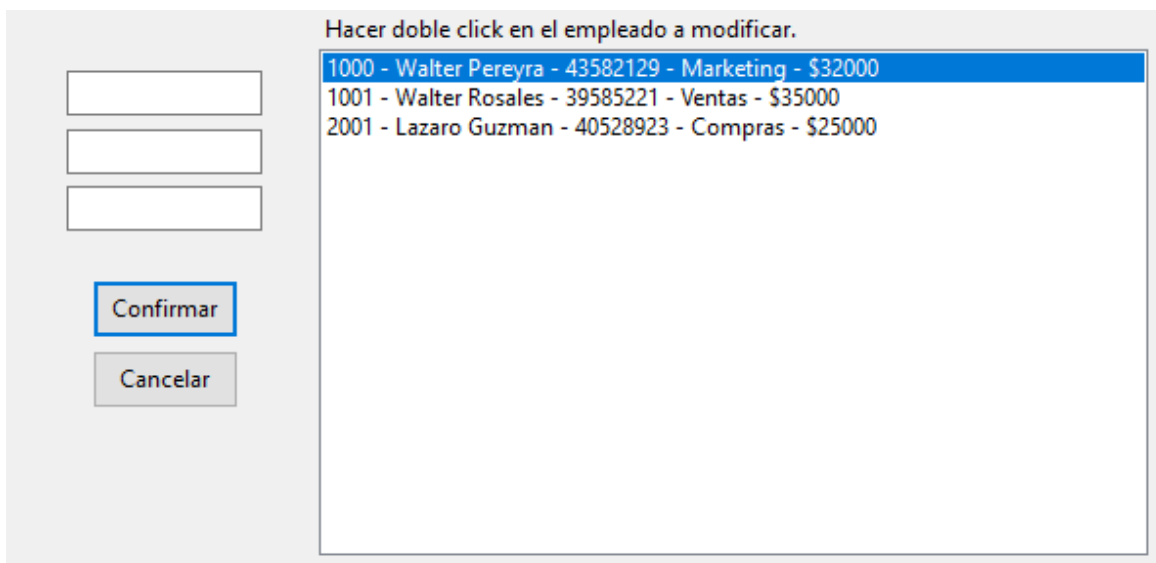
**Botón Agregar:** Abre el formulario “FormAgregarEmpleado” el cual contiene los campos necesarios para crear un nuevo empleado, también se puede cancelar la acción.



Formulario para agregar un empleado. Contiene los siguientes campos:

- Ingrese nombre:
- Ingrese apellido:
- Ingrese DNI:
- Seleccione el puesto:
- Ingrese sueldo:
- Botón Confirmar empleado
- Botón Cancelar

**Botón Modificar:** Abre el formulario “FormModificarEmpleado” en el cual carga un listbox con los elementos de la base de datos, al realizar doble click se importan los datos editables del empleado a los textbox.



Formulario para modificar un empleado. Contiene los siguientes elementos:

- Titulo: Hacer doble click en el empleado a modificar.
- Lista de empleados:

1000	-	Walter Pereyra	-	43582129	-	Marketing	-	\$32000
1001	-	Walter Rosales	-	39585221	-	Ventas	-	\$35000
2001	-	Lazaro Guzman	-	40528923	-	Compras	-	\$25000

- Botón Confirmar
- Botón Cancelar

**Botón Probar Conexión:** Prueba la conexión y devuelve un msbox con el resultado (Éxito/Fallo).

**Botón Volver:** Vuelve al menú principal.



## **Trabajo Practico Nro. 4: Lista de temas**

**Base de datos:** Define su funcionalidad en la clase “EmpleadoDAO” y se implementa en el “FormMenuEmpleados”.

**Hilos:** Definido en la clase “Serializadora” y en “EmpleadoDAO”, se implementa en “FormMenuEmpleados”, “FormMenuArmas”, “FormMenuClientes” y “FormMenuVentas”, todas desde el método “Form\_Load” correspondiente, simulando una “gran carga” de manera asincrónica (aplicando un wait de 2 segundos) desde la base de datos provocando la variación del texto de un label ubicado en los alrededores de la list/datagrid dependiendo de si pudo cargar la lista/datagrid o no, indicando a su vez la cantidad de elementos cargados.

**Eventos y Delegados:** Delegados: Definidos en la clase “Delegados” y aplicados en conjunto con Eventos en “FormMenuEmpleados”

Eventos: Definidos en la clase “EmpleadoDAO” aplicados en conjunto con Delegados en “FormMenuEmpleados”.