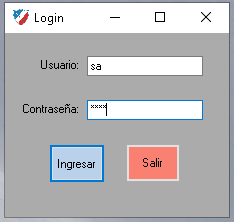
El siguiente informe tiene como objetivo describir el proceso de desarrollo del trabajo practico integrador para la materia Programación de Aplicaciones Visuales a lo largo del cuatrimestre, junto con las decisiones relevantes que tomamos como grupo y las problemáticas que se nos fueron planteando y debimos solucionar.

Para el dominio del problema elegimos a la organización Bochas Sport Club, ubicado en la ciudad de Colonia Caroya. El club cuenta con cerca de 500 jóvenes y adultos que practican distintas disciplinas como fútbol, básquetbol, hockey sobre césped, patín, gimnasia, karate, yoga, paddle, natación, ciclismo y bochas. En las instalaciones también se realizan ventas de indumentaria deportiva.

La primera actividad que realizamos para el desarrollo del sistema fue abstraer las entidades más importantes que pudimos identificar para llevarlas a un modelo de tablas relacional, obteniendo el siguiente modelo: **(ver anexo)**.

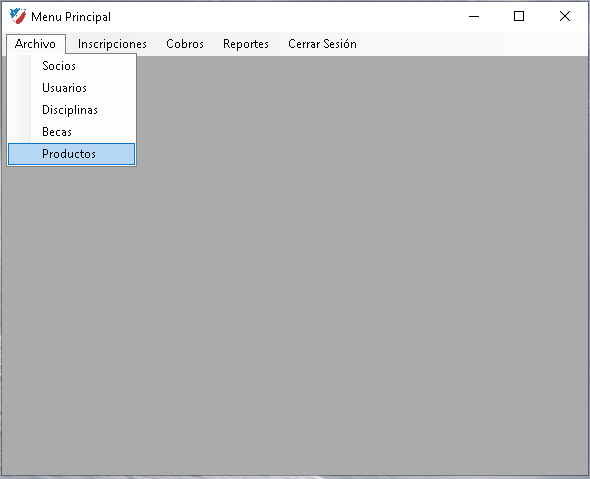
Dentro del diagrama tenemos la tabla de usuarios para manejar el acceso del personal al sistema, luego la tabla de los socios con sus datos relevantes está asociada a las disciplinas a las que estos se inscriben, la información relacionada a estas y los productos deportivos que puedan llegar a comprar, además de las becas que puedan otorgarse a ciertos socios. Con la ayuda de SQL Server Management Studio generamos el script para la creación de la base de datos con la que trabajamos en el programa.

Después de tener el dominio establecido empezamos a darle forma al programa. La primera interacción con el usuario es una ventana de log in en la que se deben ingresar nombre de usuario y contraseña (posteriormente pueden agregarse más usuarios)

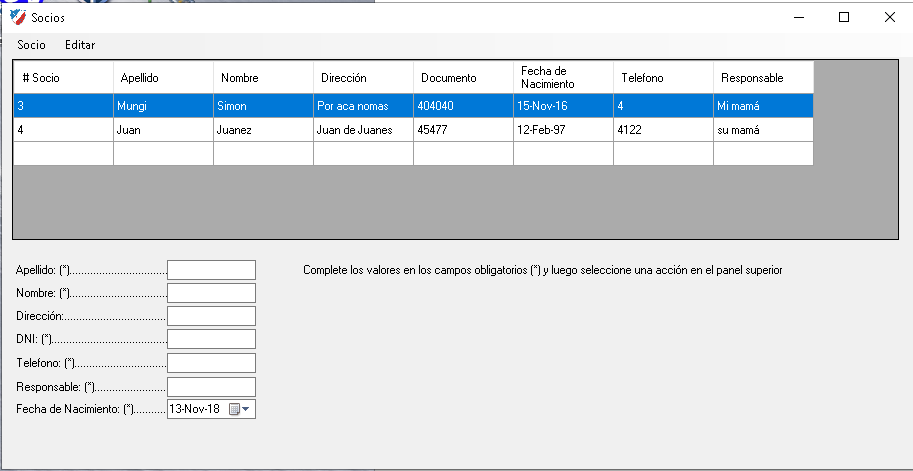


Al ingresar se accede a una ventana principal con menús desplegables. Lo primero que desarrollamos fue el alta, baja y modificación de socios, usuarios, disciplinas, becas y productos, cada uno con su propia form (y éstas a su vez pueden generar más forms para facilitar el ingreso de los datos).

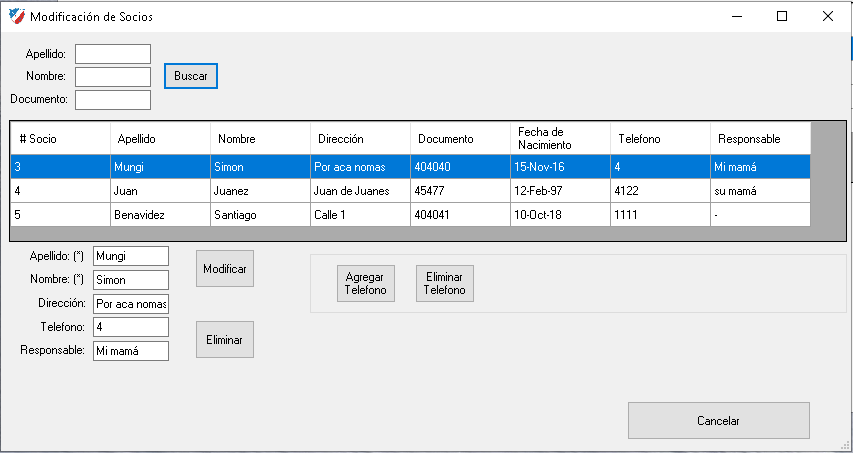
El primer problema con el que nos encontramos fue la inconsistencia visual en las distintas ventanas del programa. Debido a que había un gran número de ellas para construir nos dividimos el trabajo y cada miembro terminó dándole un estilo personalizado.



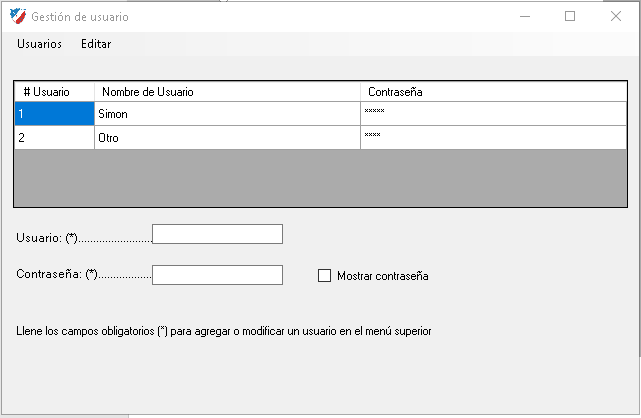
Inscripción y visualización de socios:



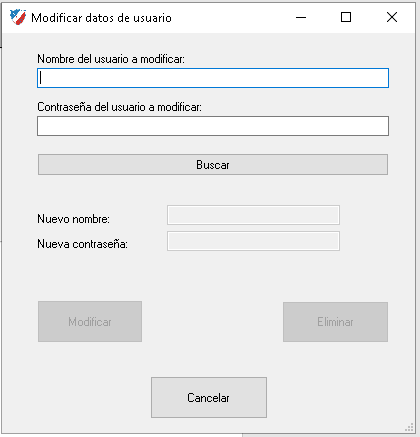
Modificación de socios existentes:



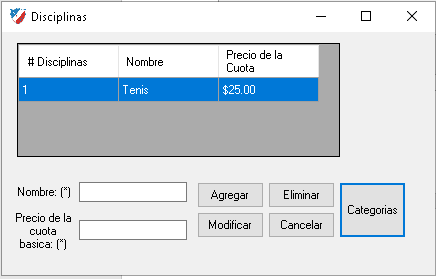
Agregar nuevos usuarios:



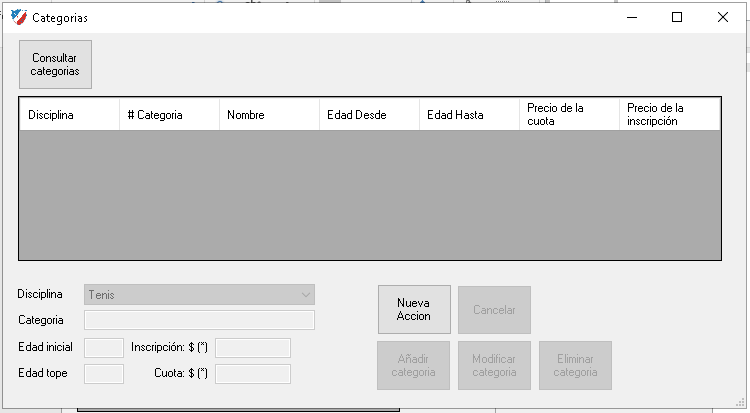
Modificar usuarios existentes:



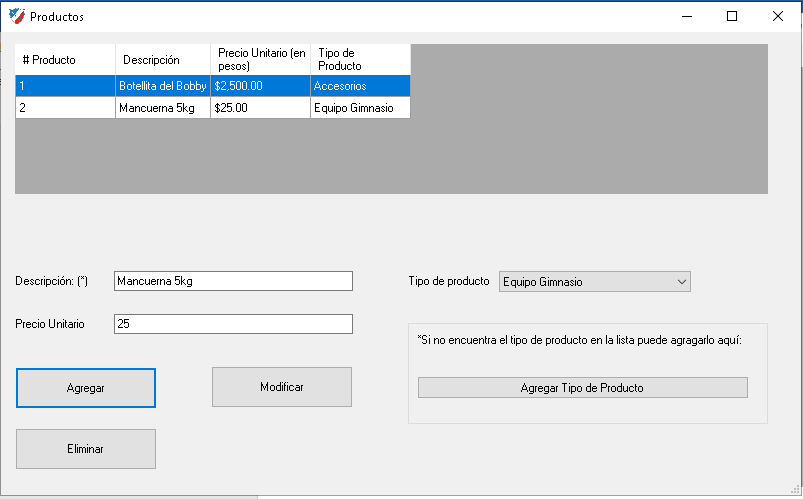
Agregar disciplinas:

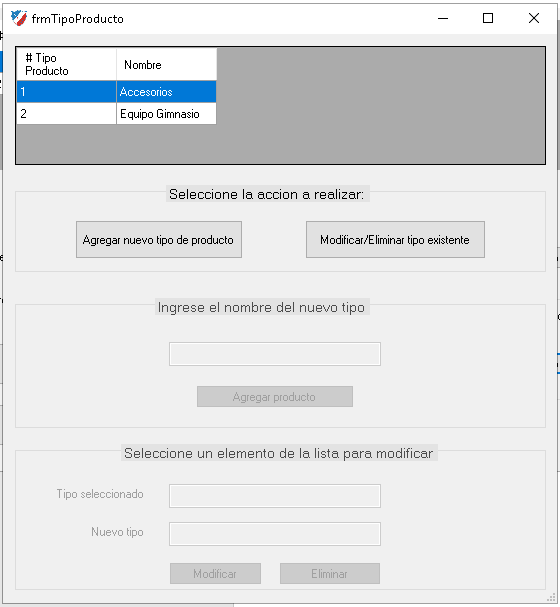


Consultar categoría de las disciplinas:

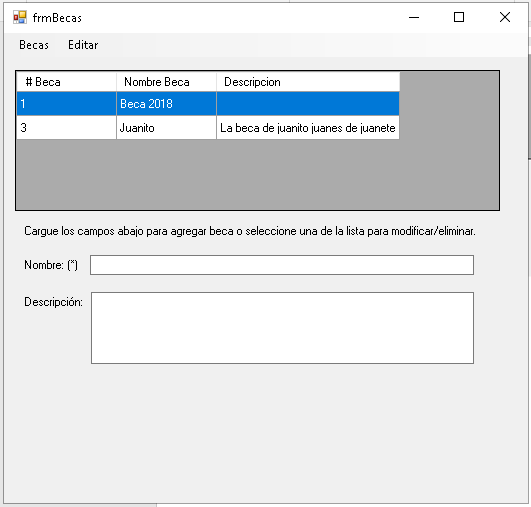


Agregar productos y su tipo:





Agregar, editar y eliminar becas:



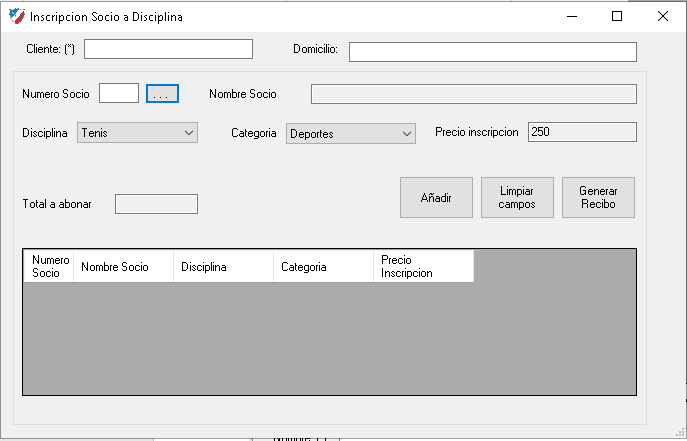
Una vez realizada la parte grafica de los componentes de alta, baja y modificación comenzamos a desarrollar la lógica de negocio asociada a cada form.

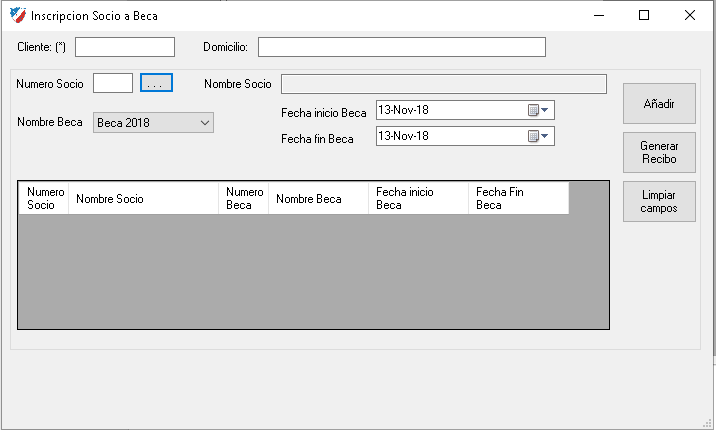
El modo en que se estructura nuestro programa está pensado en capas, cada form (separadas en la carpeta GUI) está relacionada con una clase de entidad que posee los métodos para la carga y eliminación propiamente dichas en la base de datos. Las clases de entidad se comunican con una clase de soporte (BDHelper) que posee el string de conexión para la base de datos junto con otros métodos útiles para realizar consultas, insert, update y delete.

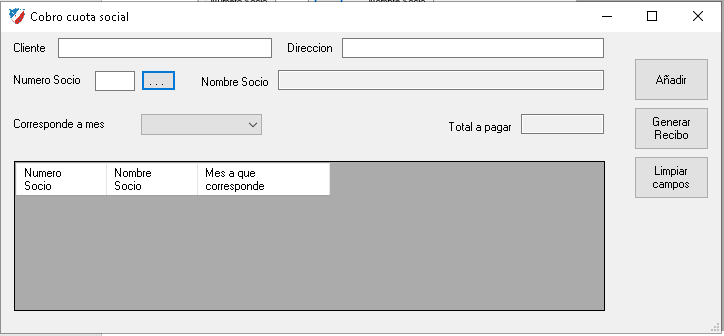
Finalmente, las clases de los forms asociadas a cada operación poseen programadas los comportamientos de botones, menús, grillas, etc.

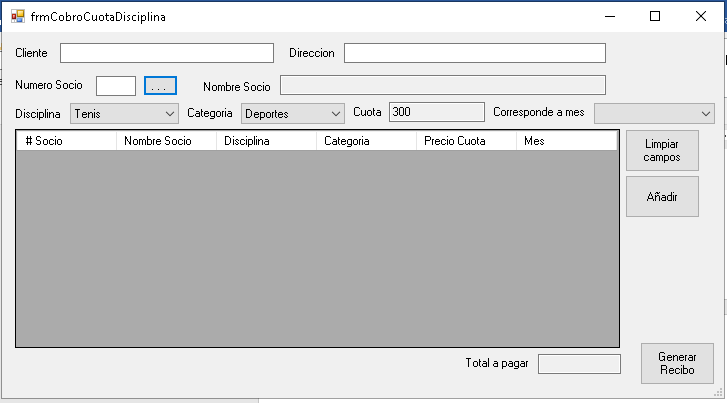
Luego de las tareas simples de alta baja y modificación la siguiente tarea fue programar las transacciones, múltiples operaciones que se ejecutan como una sola (en caso de que una falle no se ejecuta ninguna) y afectan varias tablas.

Las transacciónes de nuestro sistema se ven en las siguientes forms:

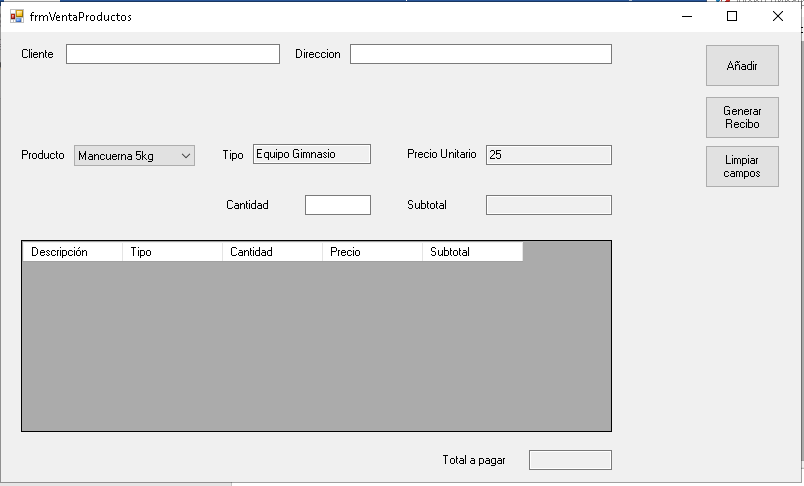




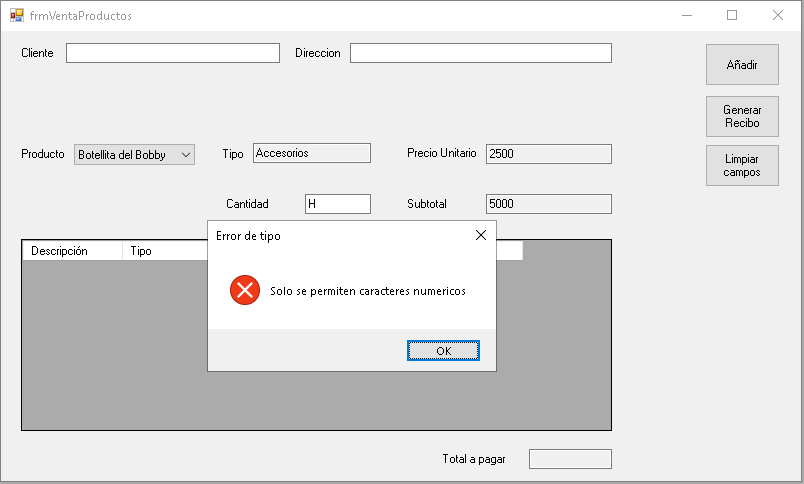




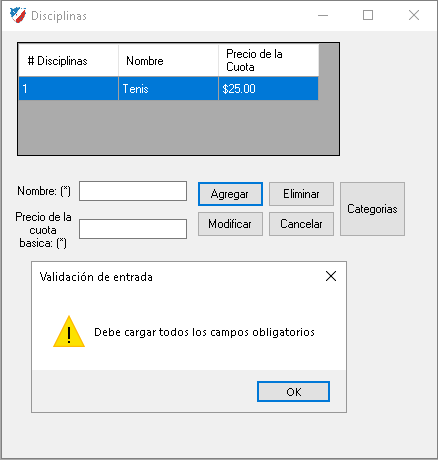
Venta de productos



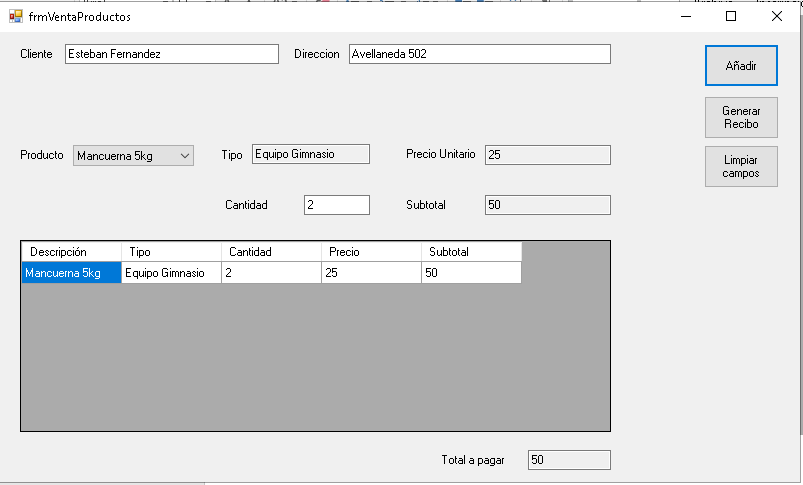
En esta ventana al igual que las demás decidimos validar la entrada de datos de modo que el usuario no pueda ingresar un tipo de dato incorrecto al campo correspondiente. Como ejemplo vemos los que sucede al intentar llenar el área de cantidad con caracteres alfabéticos.

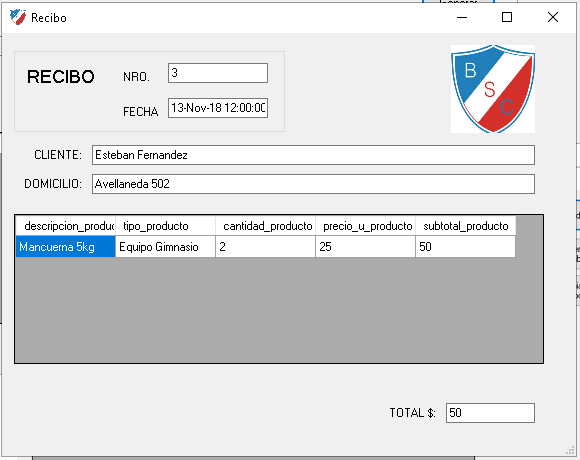


Otro ejemplo de validaciones implementadas fue en los casos en que la base de datos no admitía el ingreso de valores NULL, verificamos que siempre que sea necesario se suministraran los datos necesarios antes de realizar una acción. Por ejemplo, al intentar agregar una nueva disciplina con campos vacíos:



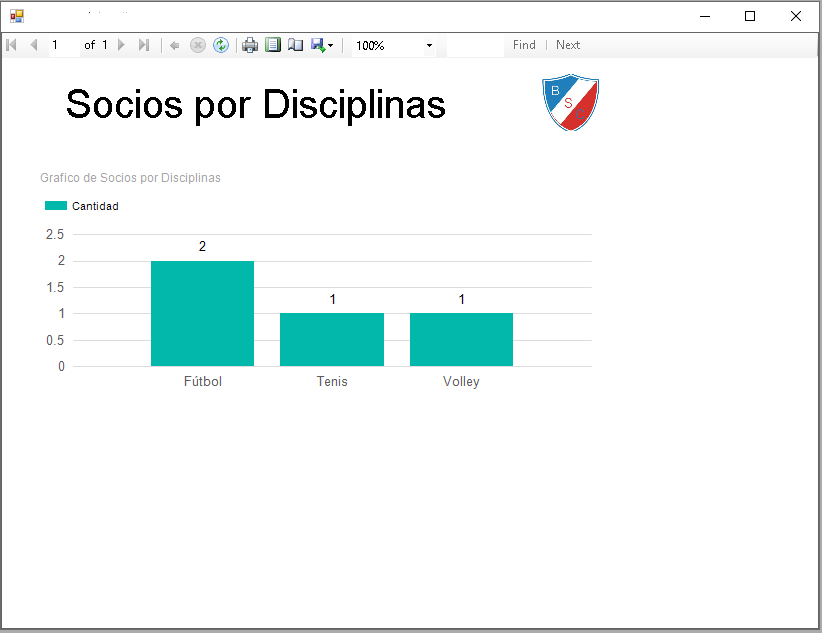
Para cada transacción se agrega la opción de generar un recibo, que levanta los datos desde la form que lo invocó. En el siguiente ejemplo vemos la generación de un recibo para la venta de un producto a un cliente.



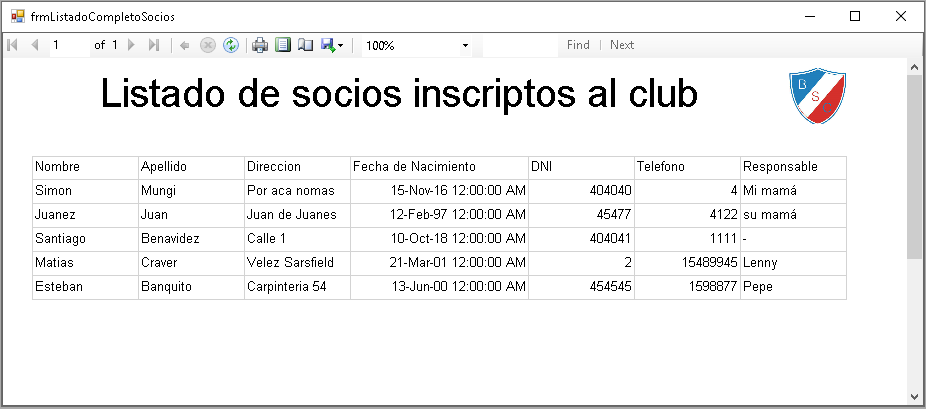


Por ultimo procedimos a generar una serie de reportes que facilitaran la visualización de datos generados por el programa. Para esto empleamos las herramientas de reportes complementarias de Visual Studio y generamos distintos tipos de informes con gráficos, listados y filtros de búsqueda por fecha.

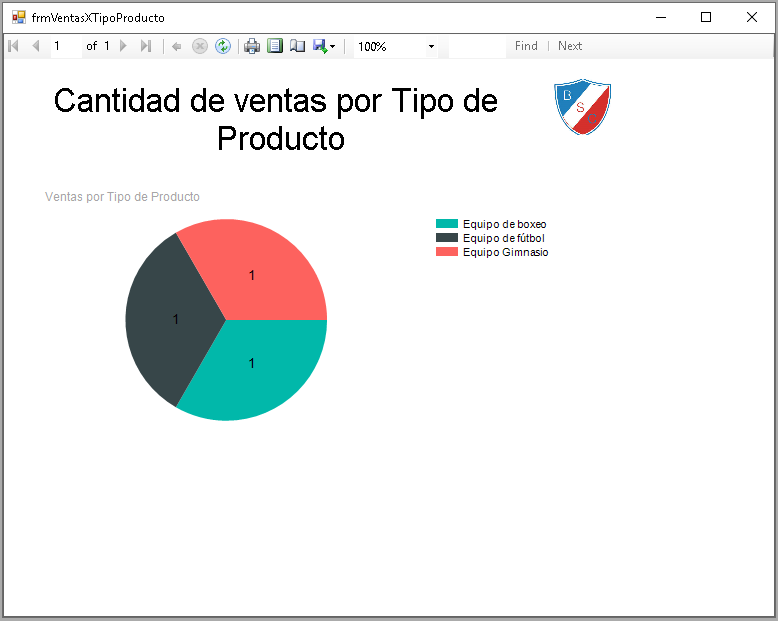
Reporte de Socios por disciplinas con grafico de barras incluido:



Listado de todos los socios inscriptos en la base de datos:



Informe sobre las ventas de productos realizadas con gráfico de torta:



Listado de recibos generados. Se incluye un filtro de búsqueda para ingresar un rango de fechas:

