**UNIVERSIDAD NACIONAL DE INGENIERÍA**

**(UNI)**



**Materia: Matemática para computación II Ing. En computación.**

**Tema**: Redimensión de arreglos.

Laboratorio #1

**Integrantes:**

Erick Hazael Morales Jarquin 2023-0862U Denis Josué Wong S. 2019-0919U

**Grupo:** 2M1-CO.

Programa, Registro de estudiantes hecho por: Erick Morales. Programa, Registro de trabajadores hecho por: Denis Wong.

Laboratorio #1: Registro de estudiantes.

Un placer y gracias por su atención, es bien sabido que c# es un amplio lenguaje de programación y complejo de entender en un corto periodo de tiempo pero por eso es nuestro deber compartir la belleza de este lenguaje para trabajar. Antes de pasar a nuevas tecnologías es importante tener dominio de las bajas, en este caso de Windows forms, Windows forms es una tecnología que no se usa ya demasiado pero es una buena alternativa para empezar a desarrollar tu aplicación de escritorio, la ventaja de Windows forms es que tiene una manera de hacer interfaces graficas más eficiente y atractiva a la vista que del propio Java.

En Java necesitas saber hasta como programar un pequeño JFrame, esto puede causar un proceso lento en desarrollar tu de escritorio. Además, sus interfaces no serán muy atractivas si de tiempo no dispones, si eres un programador aficionado en código y no en diseño, c# te gustara muchísimo, aquí tienes una pequeña guía de como realizar un diseño rápido y sencillo de un sistema cualquiera ya sea de ventas o manejo de fabricas, eso depende de tu creatividad:

Aprender a diseñar tu primera app de escrtorio:

<https://youtu.be/eCSbUCL4teE?si=oo--ixrjHHFyIwhJ>

(Copiar y pegar en el buscador)

Aprender a usar formularios secundarios para crear vistas es crucial si quieres tener una restructura decente en tu aplicación, porque habrá una que otra ocasión por no decir siempre que neceistaras una nueva vista para realizar el proceso correctamente, aparte que será mas como de usar y atractivo a la vista, aparte de tener un buen orden, es algo practico de hacer, aquí hay una guía de como crear las vistas secundarias donde se estarán mostrando las distintas secciones de la solución:

Aprender a crear vistas en Windows forms:

<https://youtu.be/lx7aaNazwwU?si=xG9QXnV8OV2AlzU6>

(Copiar y pegar en el buscador)

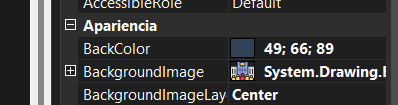
Si tienes claro de como diseñar un formulario, ahora pasemos a la parte interesante, todo lo que pasa por encima y se mira es lo de menos. Empieza diseñando el formulario padre guiándote del video anterior, crea paneles y asignales un color a tu preferencia. Uno para los botones que cambiaran las

vistas, otro para los 3 botones responsables del cambio de tamaño de tu formulario y ppr ultimo y menos importante, el panel donde se mostraran las vistas.

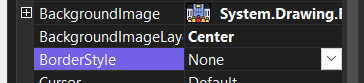
**Modelo a seguir para avanzar al código:**



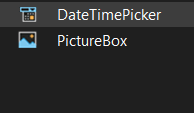
Si te queda la duda de como se han puesto imágenes como por ejemplo en el fondo del panel, ubicándote en propiedades hay una propiedad que te dice:



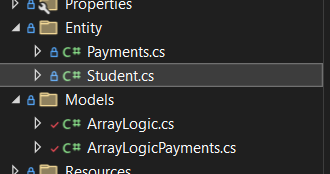
Es importante haber descargado la imagen para exportarla y elegir un fondo adecuado, si un dado caso la imagen esta como no debe de estar, es crucial ajustarla usando la propiedad :



Ajusta la imagen a tu gusto. Sin embargo, puedes poner un logo a tu preferencia usando:



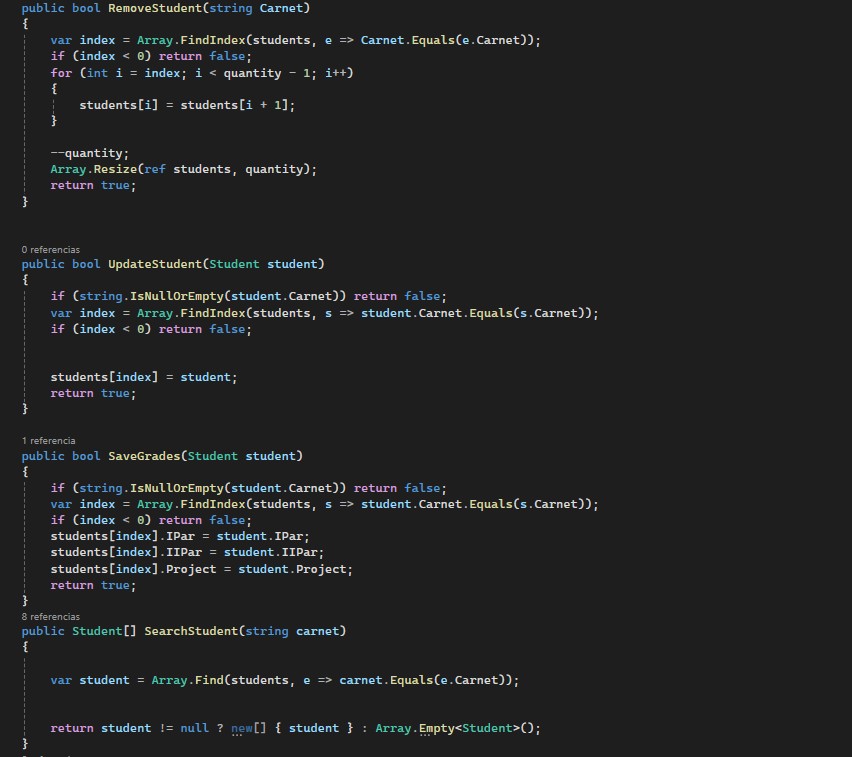
Si te gusta ser creativo puedes darte gusto con eso, aunque es la parte bonita, programar también tiene su toque si estas un poco mal de la cabeza. A continuación se explicara la lógica para guardar, buscar, borrar y actualizar estudiante usando una clase en models:



Anteriormente se ha explicado la estructura para guardar tus clases y métodos, junto a sus vistas. Creamos 2 clases dentro de la carpeta Entity que servirán de nuestros objetos y luego 2 clases mas en la carpeta models donde se manejaran los arreglos como una colección de objeto.



Para poder guardar un estudiante se requiere de métodos ubicados en la clase Studentlogic , esta sección de divide en guardar, actualizar, eliminar y buscar estudiando por medio de su carnet. En el método guardar se agrega una condición que comprueba si la cantidad es mayor al almacenamiento del arreglo, y si no, procede a guardar dicha información. En el siguiente método está construido con el objeto de agrupar a los estudiantes de mayor a menor nota e imprimir.

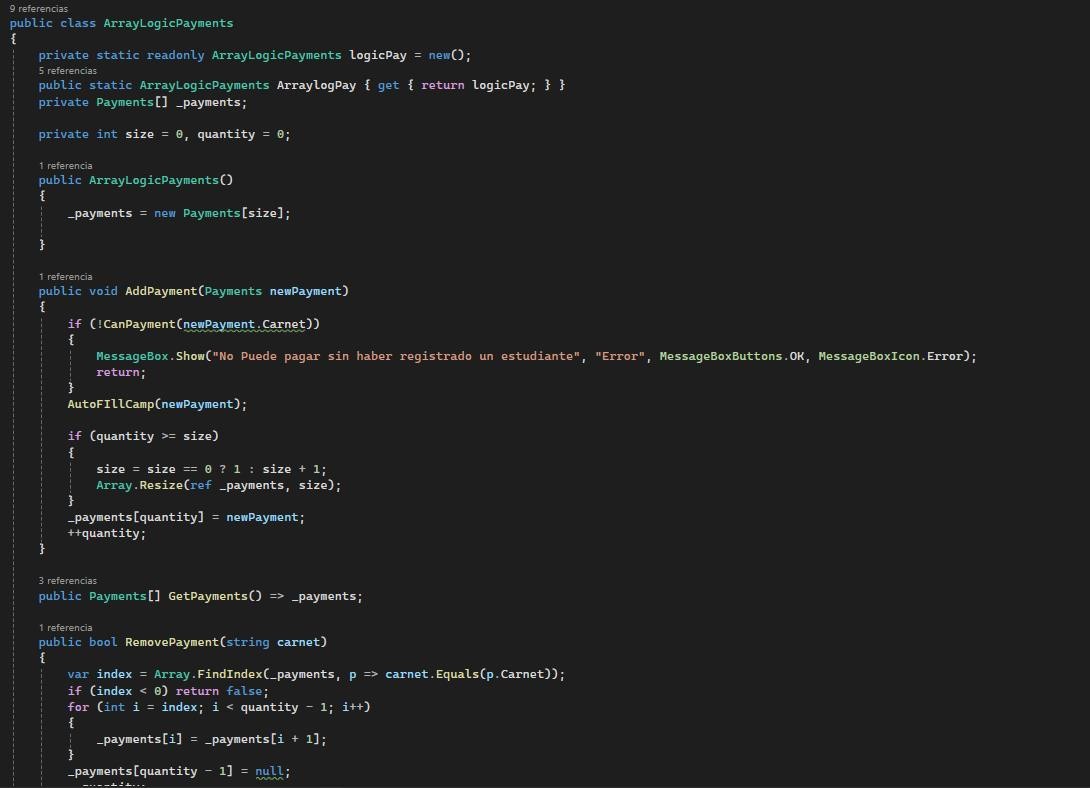


**Método para eliminar, actualizar y buscar estudiante por medio del carnet siguiendo la misma logica**



**Se guardan las notas ubicando al estudiante por su carnet.**

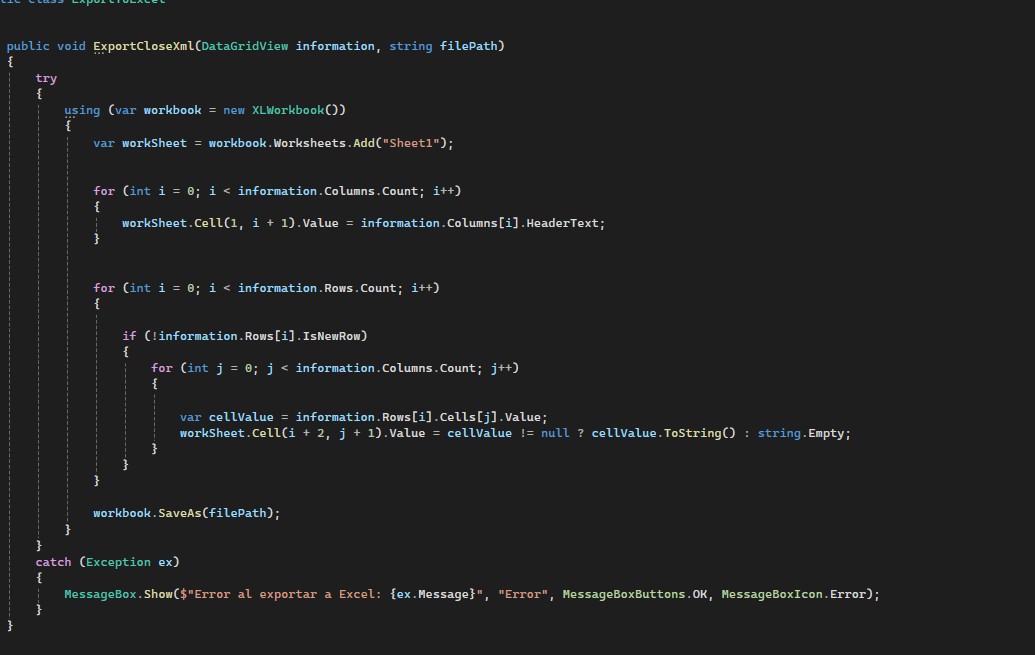
**Existen 2 métodos útiles para validar la entrada de datos, nada mas puede haber un carnet y una cédula dentro del sistema de registro.**



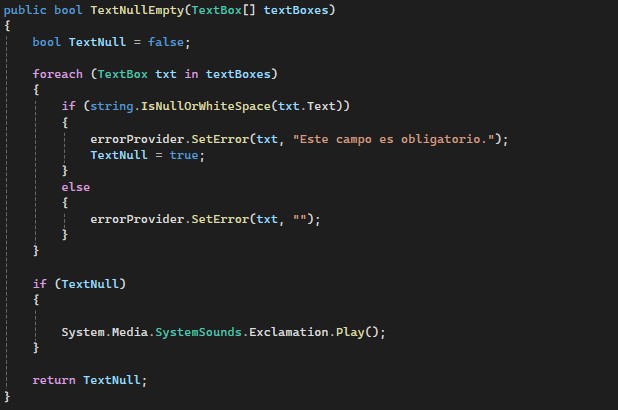
**Mismo proceso para la extracción de información categoría pagos, se guardan los datos, se eliminan, se actualizan y se extraen a un archivo Excel en función del carnet estudiantil junto a su información como el nombre, carnet y fecha de pago**



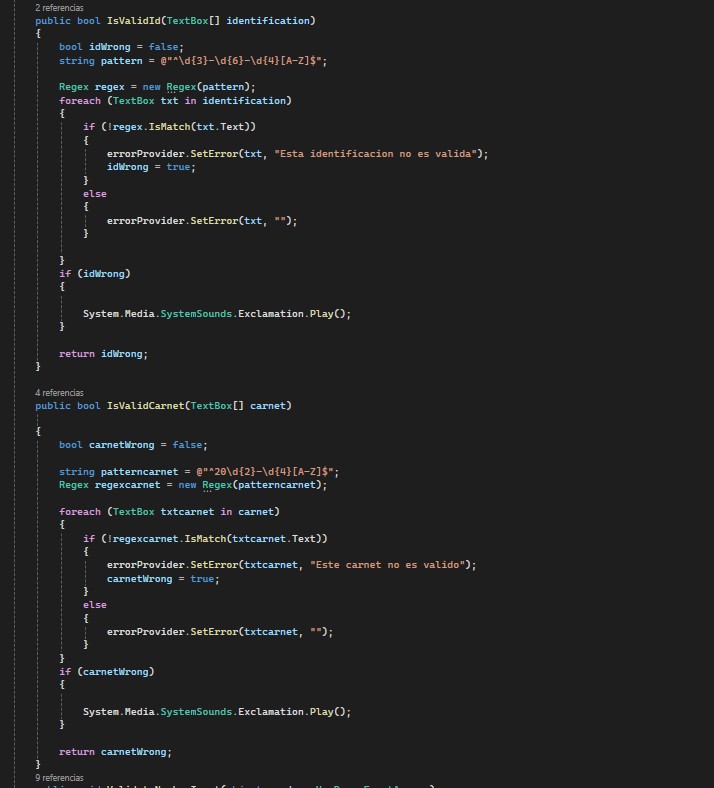
**Método para rellenar los campos de nombre, apellido y carnet para enlazar los datos del usuario y el pago, de este modo se ubica de quien y cuando hizo el pago .**



**Clase para exportar la información de los estudiantes y pagos en un archivo Excel.**



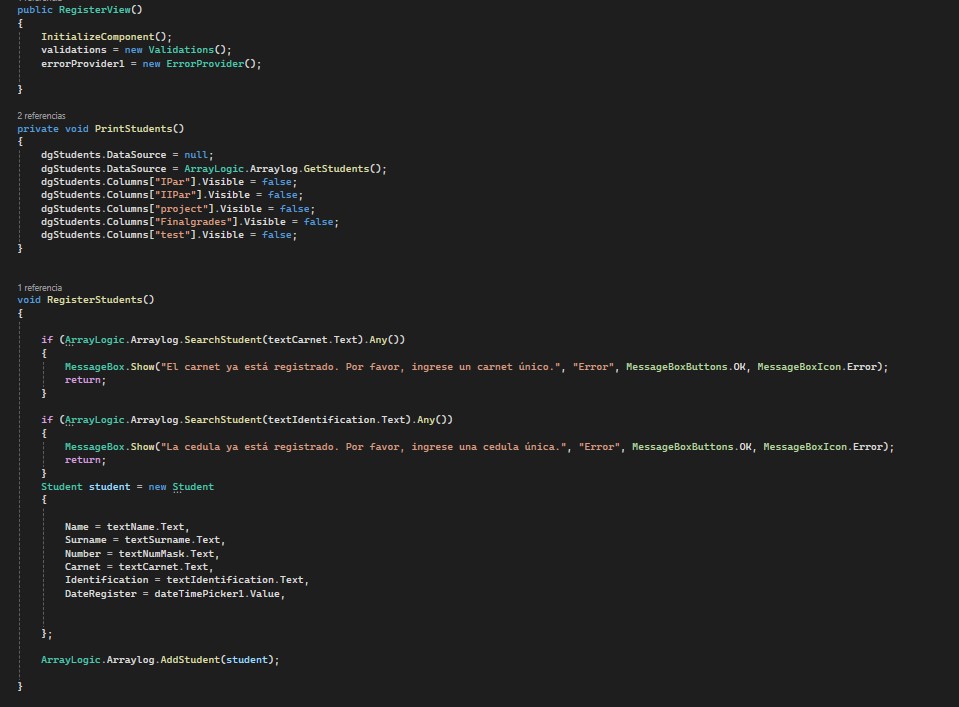
**Validaciones: Campo vacío.**



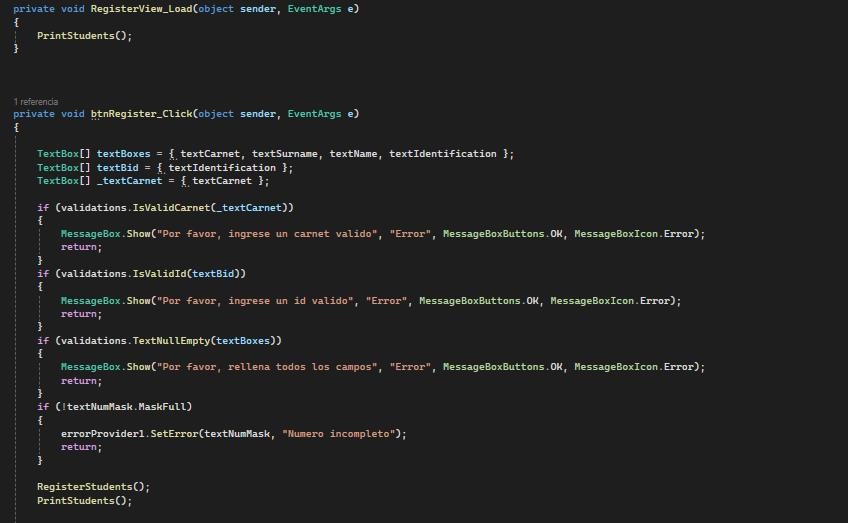
**Validaciones: Carnet y cédula válida.**



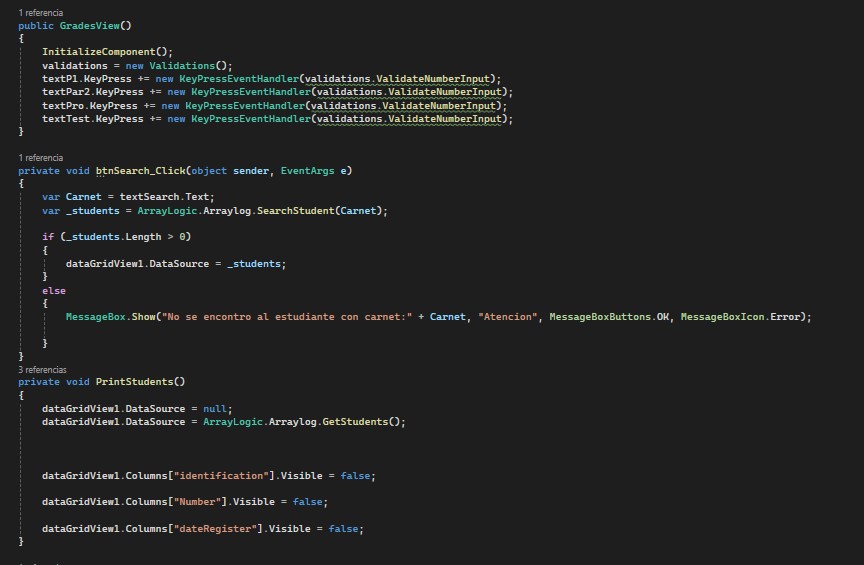
**Aplicamos el tutorial sobre conectar vistas o forma hijos/secundarios a un formulario padre/principal**



**Aplica la lógica de guardar estudiantes en tu formulario junto a sus validaciones.**

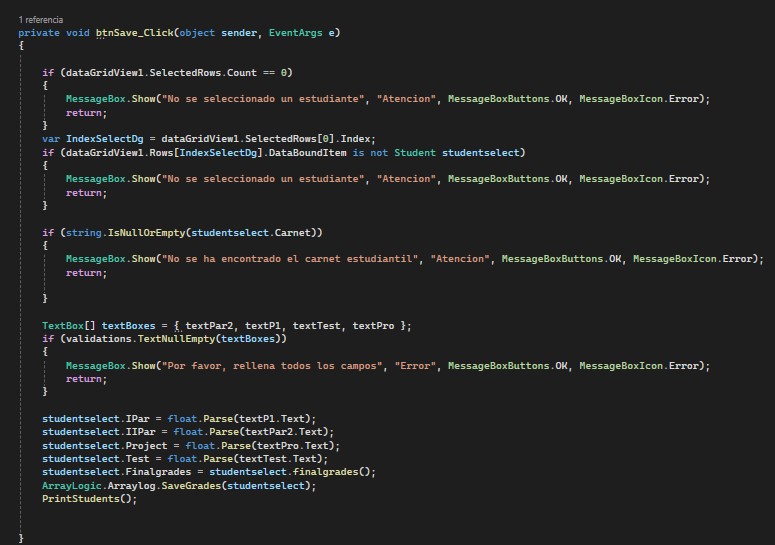


**Validaciones y método guardar.**

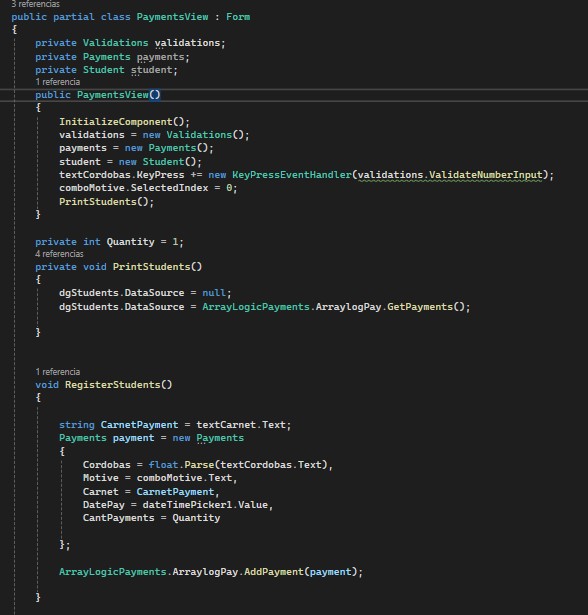


**Llamar validaciones en el constructor.**

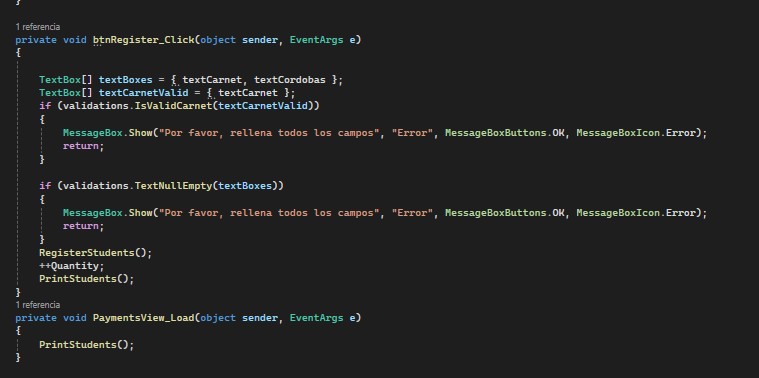
**Validaciones que requieren de un buttom event se pueden instanciar de ese modo, ahorramos líneas de código, esfuerzo y obtenemos una mejor experiencia con el programa, en este caso es para no ingresar algo que sea diferente a un número.**



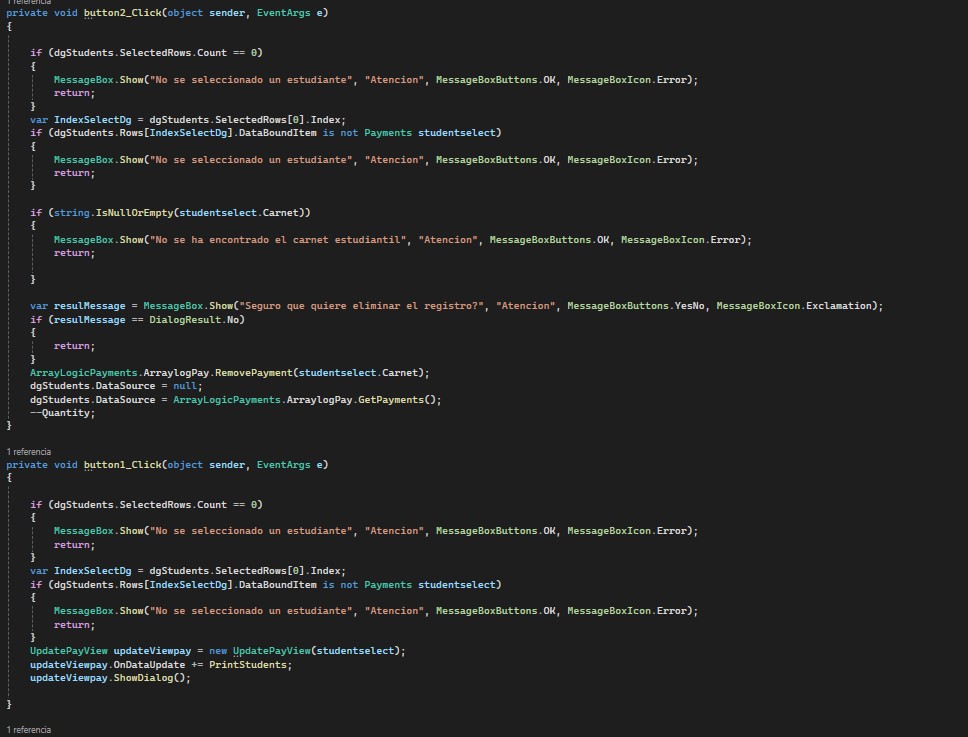
**Obligatoriamente si desea guardar las calificaciones debe de haber ingresado algo.**



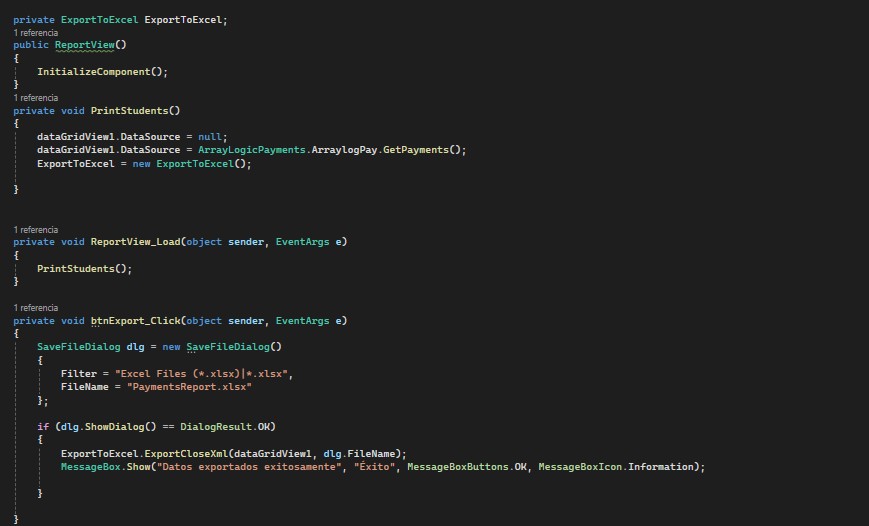
**Registro de pagos usando ArrayLogicPayments**



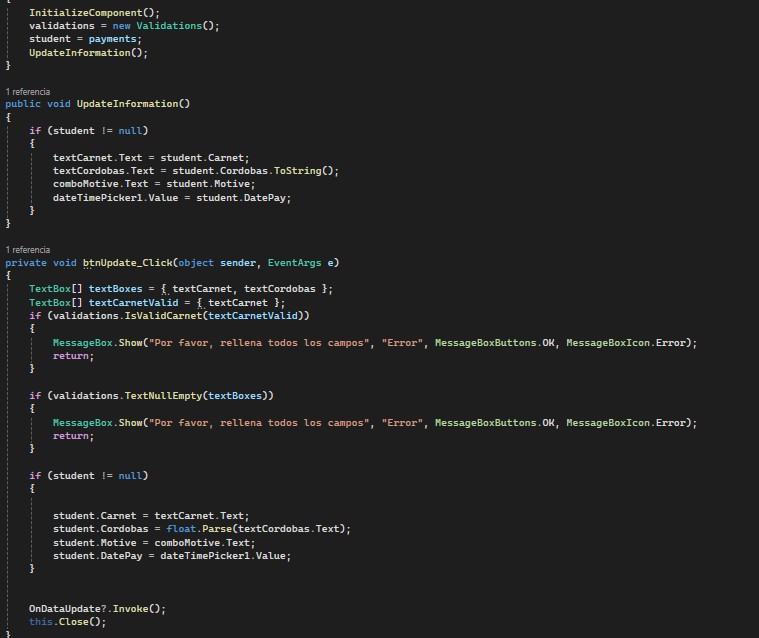
**Debe de haber algo valido para guardar, de casi contrario el método no funcionará.**



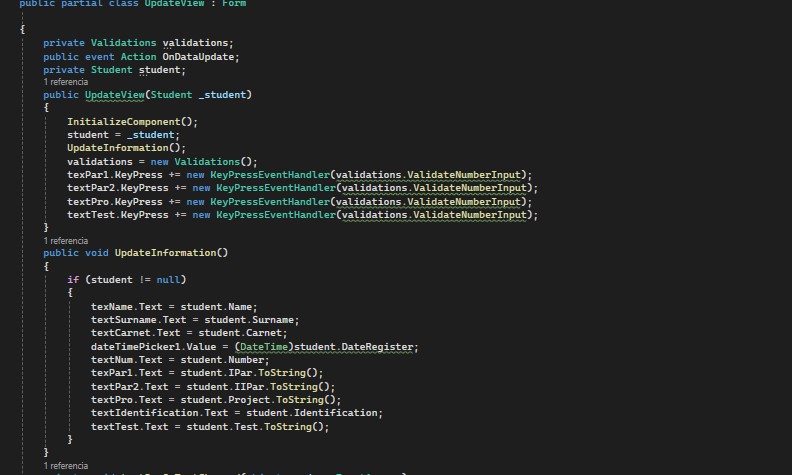
**Eliminar registro de pago extrayendo la información através del carnet de estudiante, se elimina sus pagos pero si desea actualizar se llama otro formulario que lanza un evento al cerrarse, pero al abrirse genera un evento que extrae los datos del pago y los pega en campos de texto para su edición.**



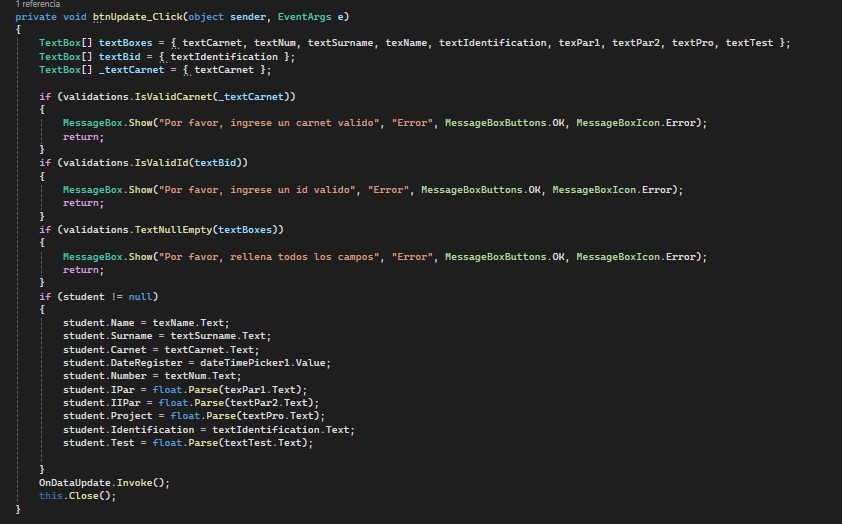
**Uso de la clase para guardar un archivo Excel con el registro. Indicas que será un archivo xlsm y el nombre del archivo .**



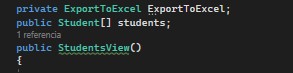
Ubicados en el formulario de actualización, es como cualquier otra excepto por “.Invoke();” y “this.close”, esto dispara el evento antes mencionado la acutalicion de los datos sin haber una intervención.

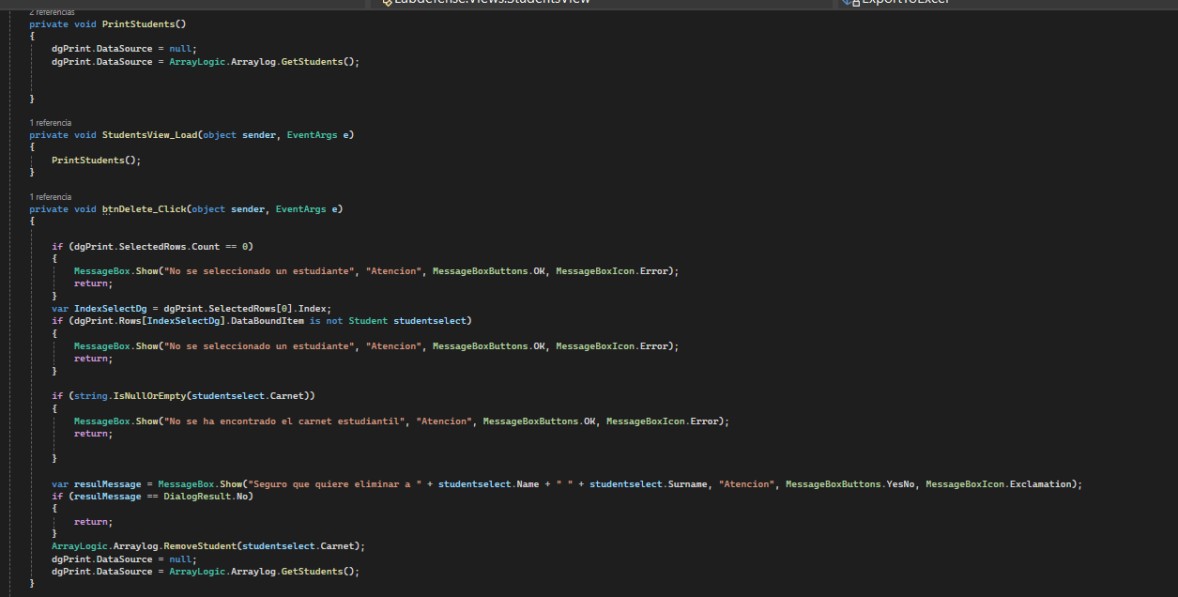


**Usamos el evento keypress en el constructor para su validación al ingresar información. Y actualizamos lo que se haya cambiado .**

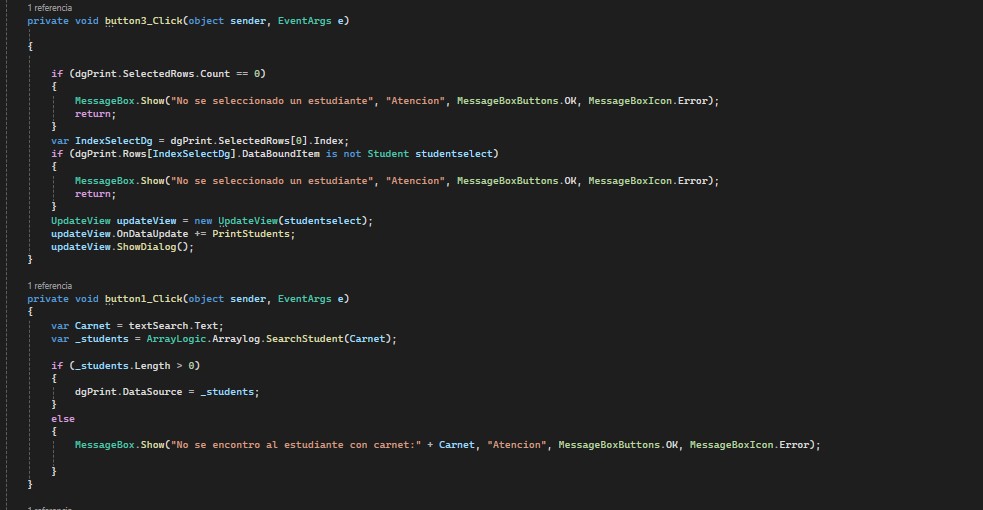


**Nuevamente el evento Invoke es usado, cuando se cierra se extrae la nueva versión y se almacenan en la colección de objetos y en donde corresponde**





Cada vez que uses un método de una clase, en dado caso que al usar dicho método te muestre un error de objeto, una alternativa es crear un objeto de esa clase e instanciarlo.



Uso del evento cerrar formulario para la extracción de datos, por consiguiente si es necesario buscar un estudiante por su carmetSe utiliza el siguiente botón que ocupa el método SearchStudent y manda el objetivo encontrado.

**Resultado:**

