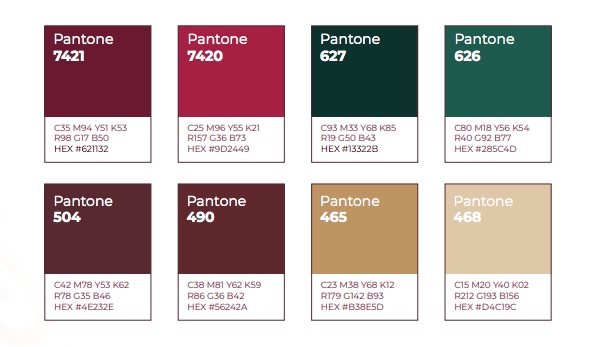
**PALETA DE COLORES DEL UI:**

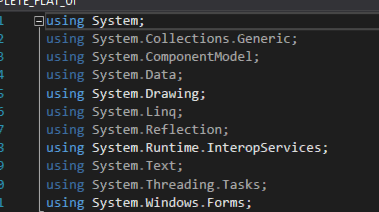
Para la creación de este proyecto se manejó la paleta de colores siguiente:



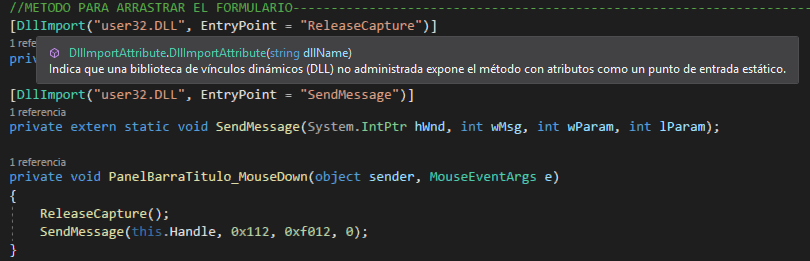
El archivo de esta paleta se incluye en la carpeta de archivos necesarios, se manejo una paleta acorde a los colores oficiales que maneja el IMSS, es importante mencionarlo para explicar el porqué de los colores.

**Sistema.cs:**

Este es el formulario principal y es también el formulario que más código contiene:



Los Using son las librerías que maneja este formulario, las cuales son necesarias para ejecutar cada uno de los componentes que se usan en el formulario, en estas la que librería que destaca es la de Runtime.IteropServices que permite que podamos ocupar librerías externas para poder ejecutar métodos de arrastre del formulario con la barra de título que en este formulario es creada con un panel para poder tener control con la interfaz de usuario así personalizarla aún más.



Tanto este como los otros formularios tienen adecuados los botones de minimizar, maximizar, restaurar y cerrar como iconos externos dentro del panel antes mencionado:

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Para que estos iconos puedan funcionar introdujo el siguiente código:

Texto

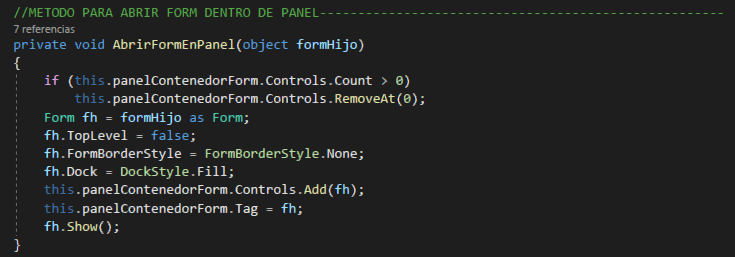
Descripción generada automáticamente

Cómo se puede observar en la acción de clic de cada botón en donde se especifican los tamaños que deberá modificar el Form para la interacción de la ventana, estos tamaños se manejan como en el plano cartesiano con los ejes de la X y de la Y, en el caso de minimizar se invoca el estado de ventana y se cambia a minimized para poder minimizar la ventana. Para acceder a la acción de clic de cada botón solo hay que dar doble clic en el modo diseño para poder modificar este código en cada uno, esto aplicar para todos los formularios del proyecto.

El botón de cerrar condicionamos la acción mandando un MessageBox con una pequeña advertencia en donde si el usuario selecciona si se invocará al método Exit() de la Aplicación.

**Manejo de Forms dentro del panel de cada módulo.**

Cuando damos clic a un módulo en automático nos invoca un formulario el cual se visualiza en la ventana principal esto se logró con el siguiente código:



En donde creamos el método AbrirFormEnPanel() en donde el cual estaremos invocando cada vez que queramos abrir un Form incrustado en alguna sección.

Por ejemplo: al dar clic en el botón de Investigador incrustamos el siguiente código.

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

Donde podemos observar que se debe declarara una variable “frame” para almacenar el llamado de un Formulario a elegir en esta caso es login() posteriormente incluimos el constructor MostrarFormAlCerrarForms() que es un método que encapsula el Método AbrirFormEnPanel(fm) el cual aplicará el código anterior esto con el fin de que cuando cerremos el formulario incrustado se mantenga abierto el formulario principal.

**Login.cs:**

El formulario login nos permitirá que los usuarios administradores se puedan loguear e invocar cada uno de sus formularios.

Para esto es necesario crear un método loguear() el cual espere como retorno dos string uno para el usuario y otro para la contraseña, cómo se muestra en el siguiente código:

Texto

Descripción generada automáticamente

Texto

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

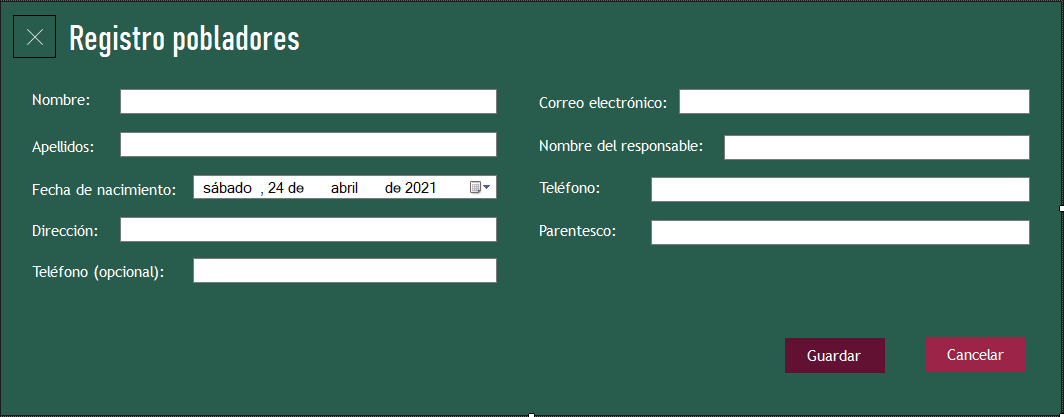
****Para la verificación de las contraseñas y apertura de cada formulario se utilizó una condicional IF ELSE IF anidado para poder validar cada situación esto dentro de un Try Catch que llamará la conexión con la base de datos y que nos respaldará por si algo llega a fallar, en este apartado es importante saber que podemos cambiar las contraseñas de acceso para validar contra la base de datos (**NOTA: También en la base datos hay que cambiar la contraseña para que se pueda hacer la validación de la misma, estas se encuentran en la tabla dentro de la base de datos.)**

Por último, el botón ingresar deberá llamar al constructor loguear(); especificando los valores a enviar para que el método pueda funcionar, estos valores los extraemos de los textbox que se encuentran en el formulario login.cs

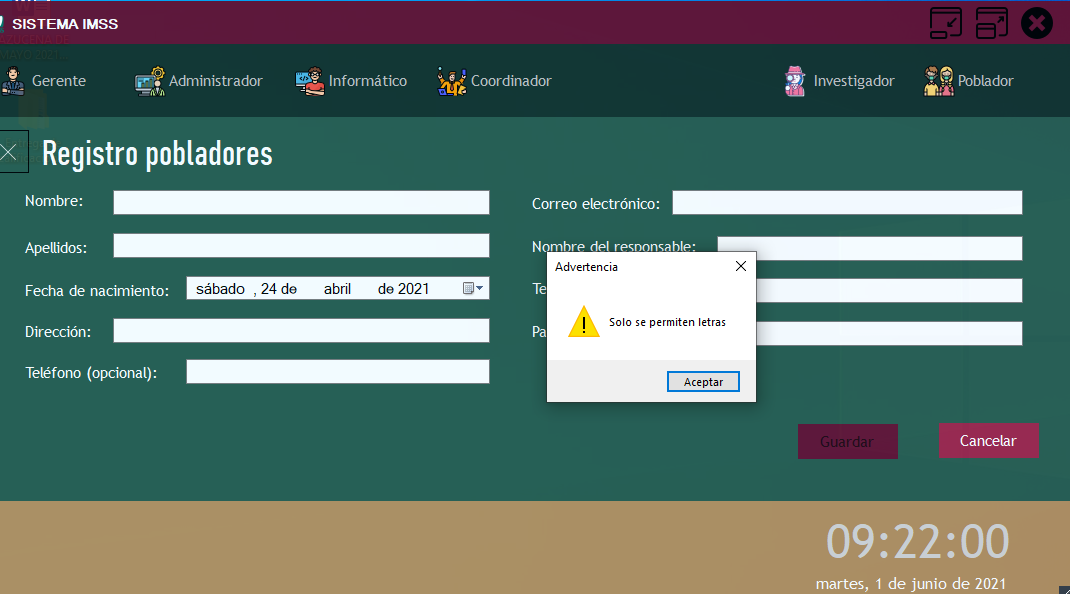
Imagen que contiene Interfaz de usuario gráfica

Descripción generada automáticamente

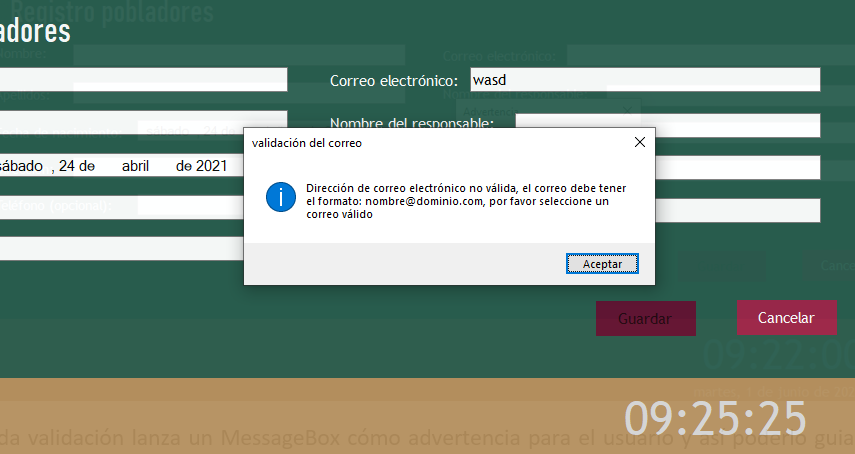
**Formulario Pobladores**

****

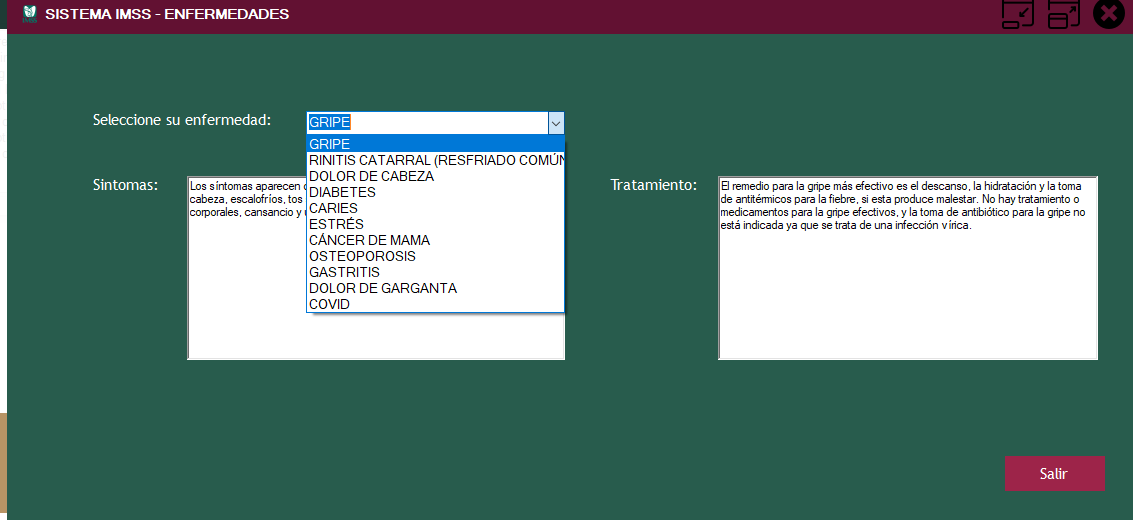
En este formulario se realiza el registro de los pobladores, el cual tiene validación de datos tanto de números cómo de letras así cómo en el formato de correo electrónico:



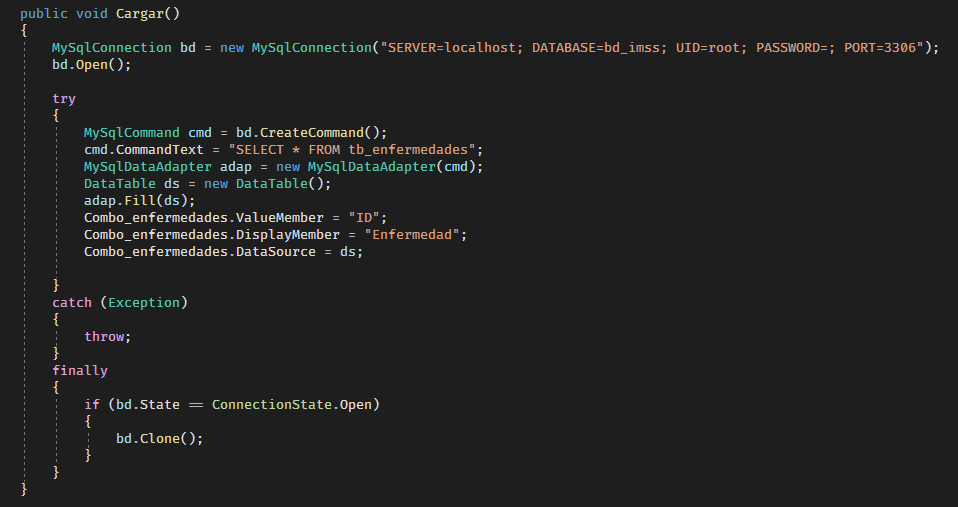
Cada validación lanza un MessageBox cómo advertencia para el usuario y así poderlo guiar en el llenado del formulario.



**Formulario Enfermedades**



Una vez registrado el poblador se mostrará el formulario de enfermedades en donde se encuentra registradas estas y se podrán visualizar sus síntomas y tratamientos este es su código de conexión con la base de datos:

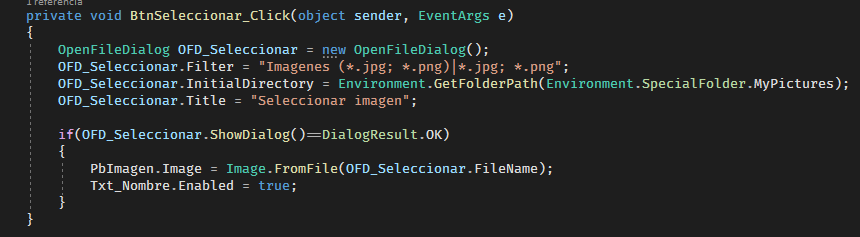


**Formulario investigador.cs**



En el formulario de investigador se encuentra una botonera con diferentes opciones en donde se pueden agregar nuevas enfermedades, cargar entrevista fotografiada, consultar entrevista, cargar fotografía del informe, consultar informe y levantar un tiquet en la mesa de ayuda para la atención del informático.

Este es el fragmento de código en donde se utiliza el OpenFileDialog para poder seleccionar la imagen cómo tal el OpenFileDialog es un modelo gráfico hay que especificarle los tipos de archivos que detectará así los que serán compatibles.



En este otro fragmento se realiza la conversión de la imagen a aBYTE para que se pueda almacenar de forma correcta a la base de datos.

Este proceso se repite tanto en la sección de entrevista cómo de reporte. En el caso del levantamiento de los tiquets se procede a llenar un formulario sencillo para levantar las incidencias que el investigador pudiese detectar.

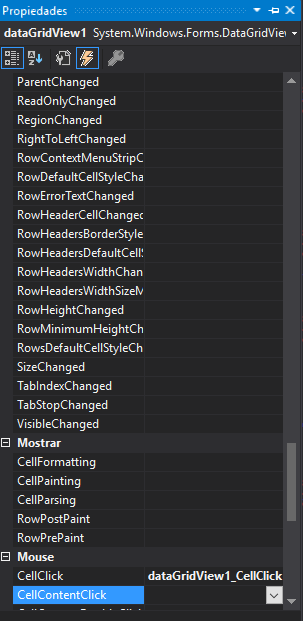
**Formulario gerente.cs**

Esta es la vista del módulo del gerente:

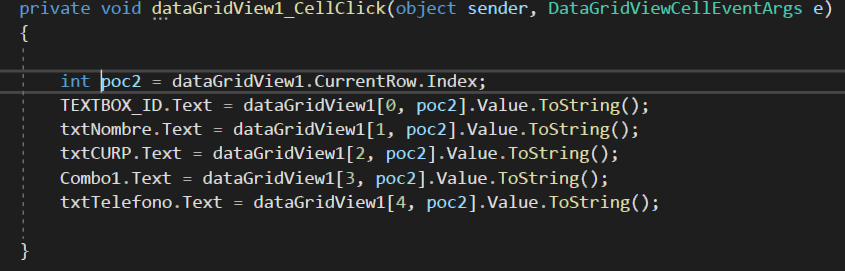


En este perfil encontraremos otra botonera con diferentes opciones entre ellas se podrá realizar la validación de los datos y gestionarlos.

En este modulo se pretende utilizar un DataGridView para poder visualizar los datos y que el usuario pueda administrar los mismo mediante una tabla en la construcción de los formularios cada uno tiene campos con datos textbox se llenarán de forma automática cuando demos clic a algún registro de nuestro DataGridView y así una vez seleccionado podamos eliminarlos el registro mostrador en nuestros textbox. Con el botón de limpiar podemos borrar el contenido de los textbox, el código para todas estas acciones se muestra a continuación.

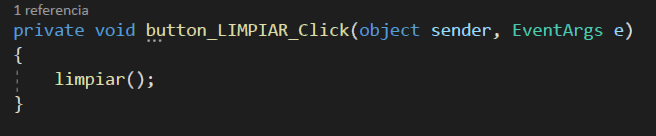


Para el DataGridView nos iremos a sus propiedades en el apartado de “Eventos” daremos clic en el evento de Mouse “CellClick” que es el evento que manipula cuando damos clic en alguna fila de nuestro DataGrid, al hacer eso nos mandará al siguiente código:



En donde declaramos un variable que estará almacenando nuestra posición sobre la fila del dataGrid, posteriormente estaremos creando matrices para indicar que posición se estará extrayendo la información para posteriormente establecerla en la propiedad Text de cada elemento textbox.

El botón Limpiar invoca el siguiente constructor:



En el constructor de limpiar() el cual lo único que hace es establecer vacíos en la propiedad Text de cada elemento textbox y combobox.

El cual contempla la conexión con la base de datos y los parámetros adecuados para la eliminación del registro, sólo cambiará la instrucción DELETE FROM (Por la tabla a elegir).

Este tipo de desarrollo se repite en los formularios **Administrador.cs, Gerente.cs** y **Coordinador.cs**

**Formulario Informatico.cs**



Así cómo en los formularios anteriores el informático podrá visualizar los tiquets que están pendientes de atender así cómo el histórico de los mismos.