



Objetivo

Crear un programa que optimice los sistemas del hospital ya que los mismos no aceptan softwares genéricos de optimización.

Objetivo alternativo

Tomando en cuenta la especialidad más todos los problemas disponibles en el área, el programa es lo único más rentable de realizar, ya que es el único problema que ningún otro pasante había planteado anteriormente.

Software

Visual Studio Community 2019

Es un entorno de desarrollo integrado para Windows y macOS. Es compatible con múltiples lenguajes de programación, tales como C++, C#, Visual Basic.

Se escogió este programa porque es el más utilizado y popular entre todos los programadores por la amplia cantidad de recursos que tiene disponible, además que tiene una interfaz bastante agradable a la vista, y como un dato adicional es que se puede cambiar la tonalidad oscura o clara.





Adobe Photoshop

Es un editor de fotografías desarrollado por Adobe Systems Incorporated. Usado principalmente para el retoque de fotografías y gráficos.

Este programa será esencial para realizar de manera correcta el programa, ya que con este se realizarán los cuadros de información, el diseño de los botones y demás iconos.



Entorno operativo

Según la información obtenida en “Acerca de” en todo el sistema del hospital se obtuvo los datos en los que se deberá de guiar el programa para poder ser compatible en todos los sistemas operativos del hospital. Los datos son los siguientes:

Procesador: AMD E-450 APU with Radeon™ HD Graphics 1.65 GHz

Memoria instalada (RAM): 4.00 GB

Tipo de sistema: Sistema operativo de 32 bits

Edición de Windows: Windows 7 Home Basic

Lenguaje de programación

Se dio la oportunidad de poder escoger el lenguaje a gusto del programador, por lo que se optó por el lenguaje de C# o mejor conocido como Visual Basic.



3.9.2 Análisis y evaluación de riesgos

NET Framework 4.7.2

Se ha detectado que el sistema del hospital, está en gran parte desactualizado, sobre todo aquellos equipos en donde tenga el sistema operativo Windows 7. El componente más importante es el **NET Framework 4.7.2**.

Cualquier nuevo proyecto hecho, tomará por defecto la plataforma de destino el **NET Framework 4.7.2**, esto es debido a que es el más reciente y la mejor opción hasta el momento.

Nombre del ensamblado:	Espacio de nombres predeterminado:
CIEL	Programa_CIEL
Plataforma de destino:	Tipo de salida:
.NET Framework 4.7.2	Aplicación Windows

Generar redirecciones de enlace automáticamente

Riesgos

- Si el sistema operativo no se le actualiza este componente, marcará errores de instalación y será imposible que el programa sea instalado de manera correcta.
- Si se cambia a una plataforma de destino menos reciente idéntica a la del sistema de todo el hospital (**NET Framework 3.0 o 3.5** por ejemplo), el programa no alcanzará toda su potencia y por consecuencia puede generar errores continuos en el programa y ser vulnerable a posibles ataques.

Alternativas

Tener listo un paquete de instalación en donde incluya el **NET Framework 4.7.2** e instalarlo en cada sistema que lo necesite.



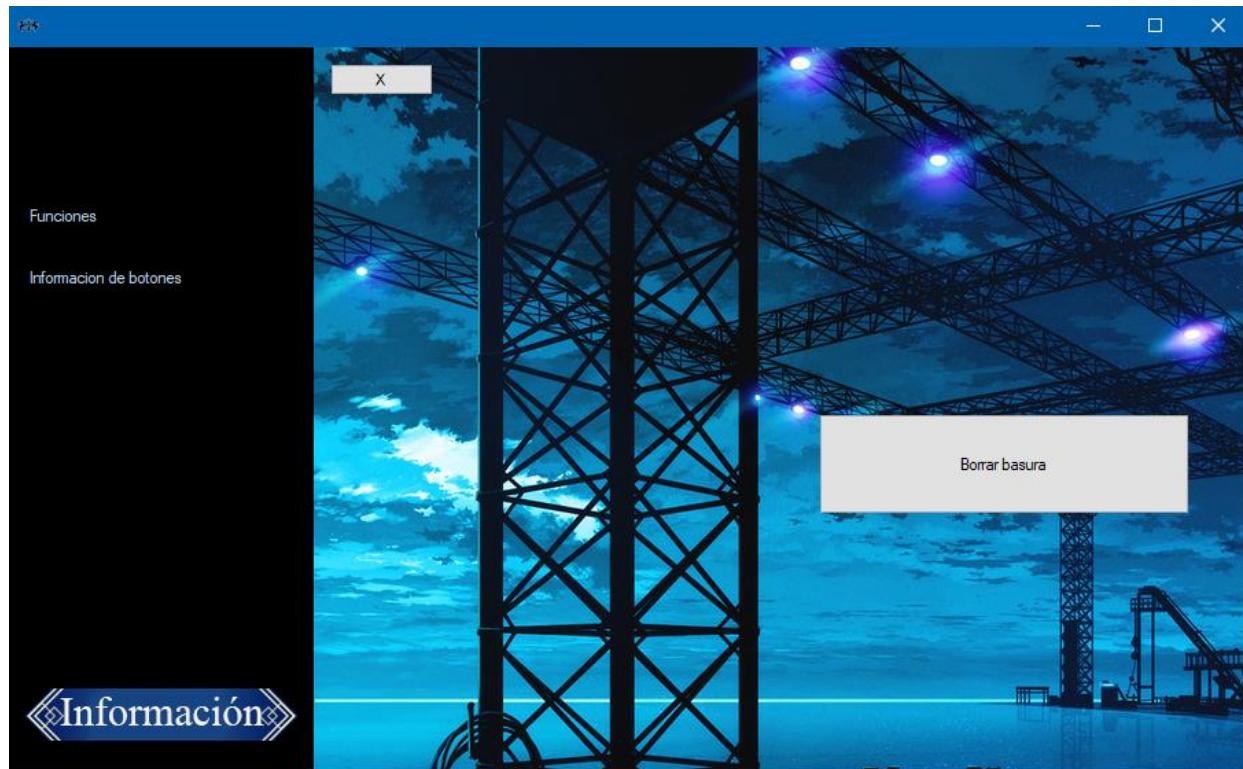
Prototipo

Existe un prototipo en fase inicial del programa, este se utilizará para hacer pruebas de instalación en aquellos sistemas en donde tengan desactualizados la plataforma **NET Framework**.





Los botones están mal colocados, sin diseño y sin funcionalidad. Esto es debido para corroborar si sufre algún tipo de falla al momento de ser instalado en una PC que sea distinta a la del programador.





3.9.3 Diseño

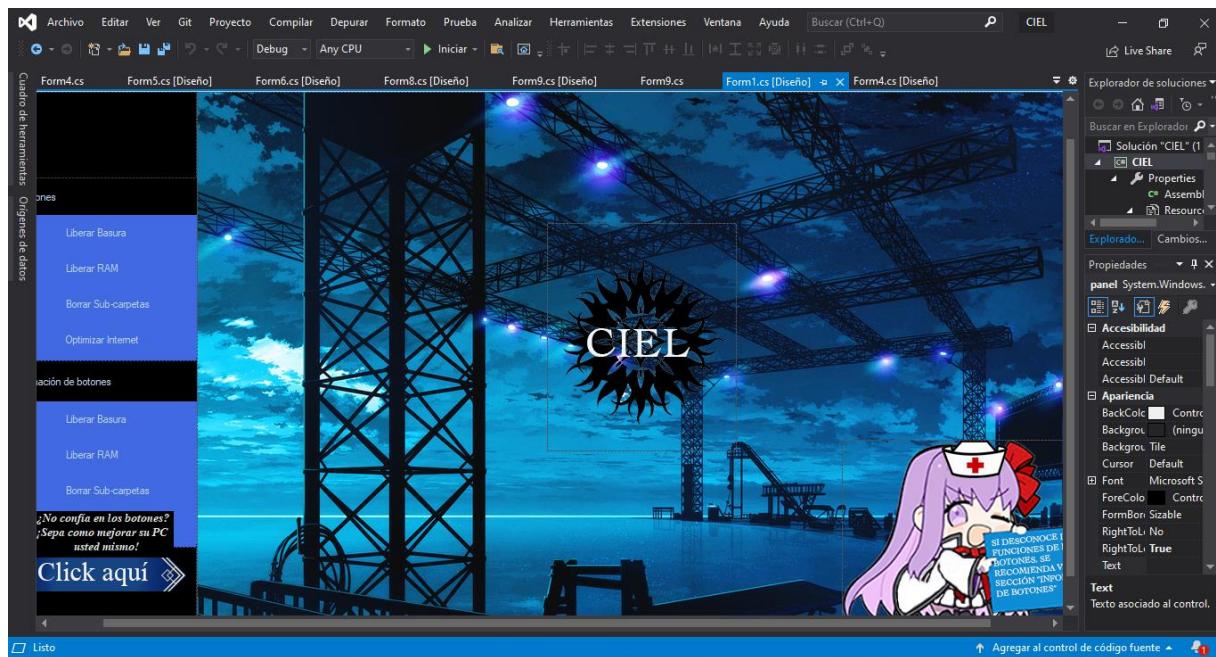
Diseño de interfaces, codificación, prueba y ejecución del programa

Diseño de interfaces

Las pantallas fueron en parte diseñadas con la ayuda del programa **Photoshop**, pero solo en ámbito del diseño del fondo de las pantallas.

Estas fueron creadas a partir del formato .cs, en estas estarán conectadas por medio de los botones. Sobre todo, estas tendrán el nombre de “Form” más el número que le corresponde.

*Para la pantalla principal, así se ve diseñada a partir del programa **Visual Studio**:*



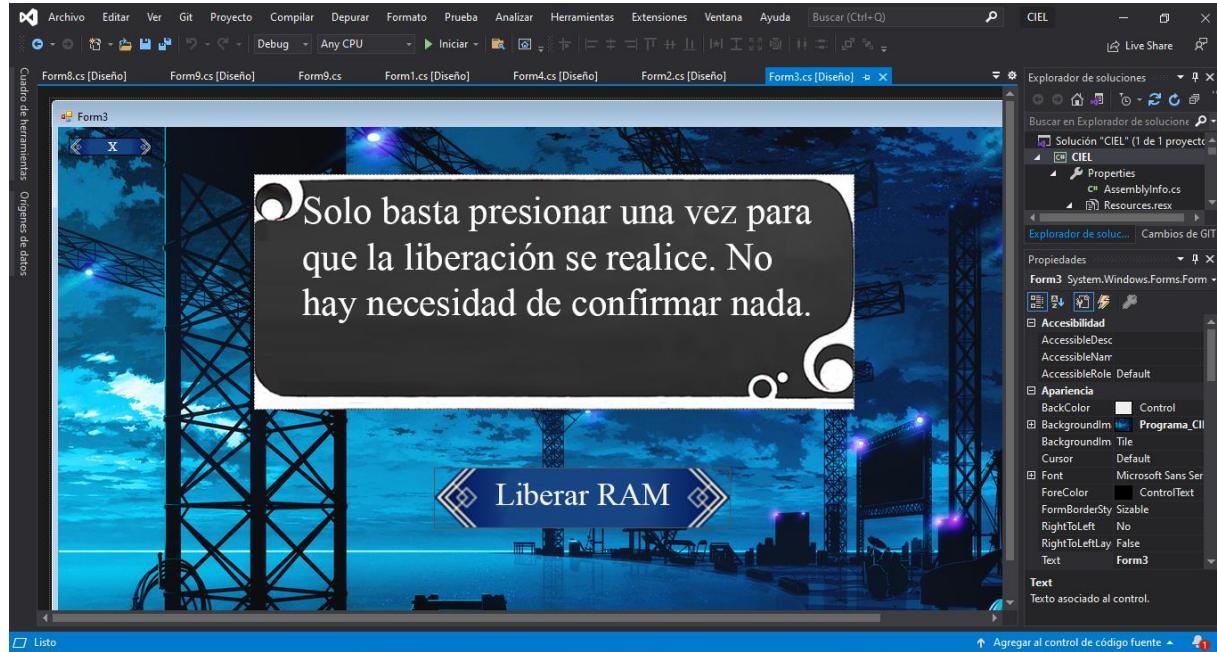


A continuación, se mostrarán los botones que contienen las funciones principales del programa:

Para el primer botón con función:

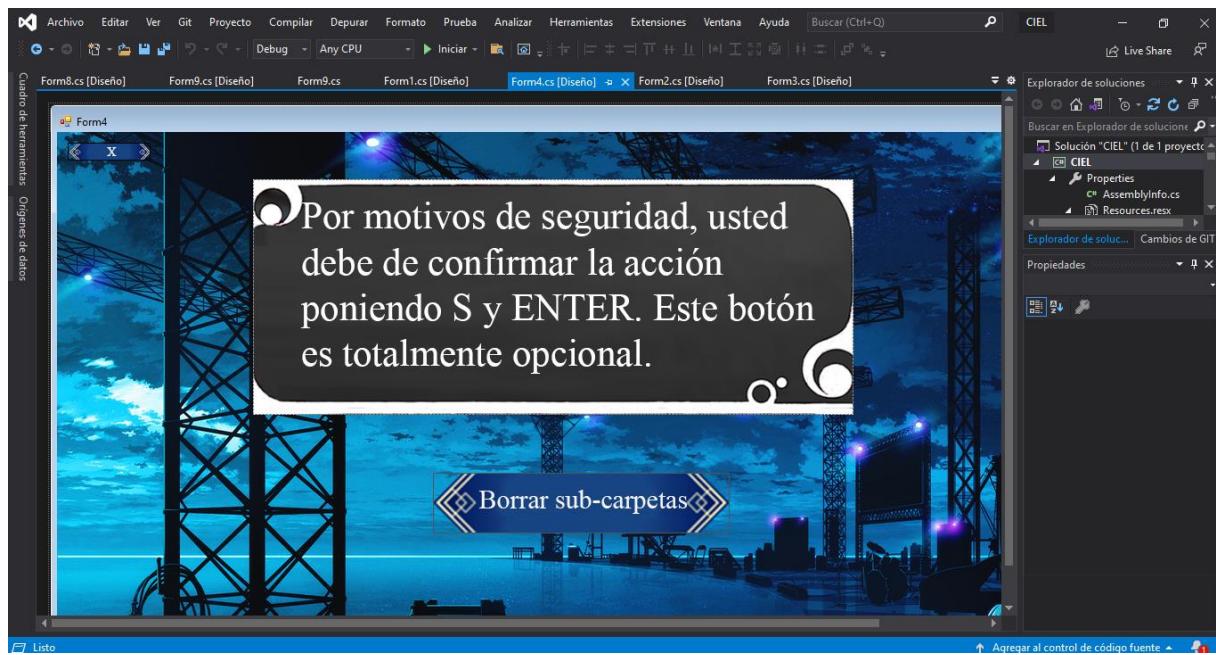


Para el segundo botón con función:

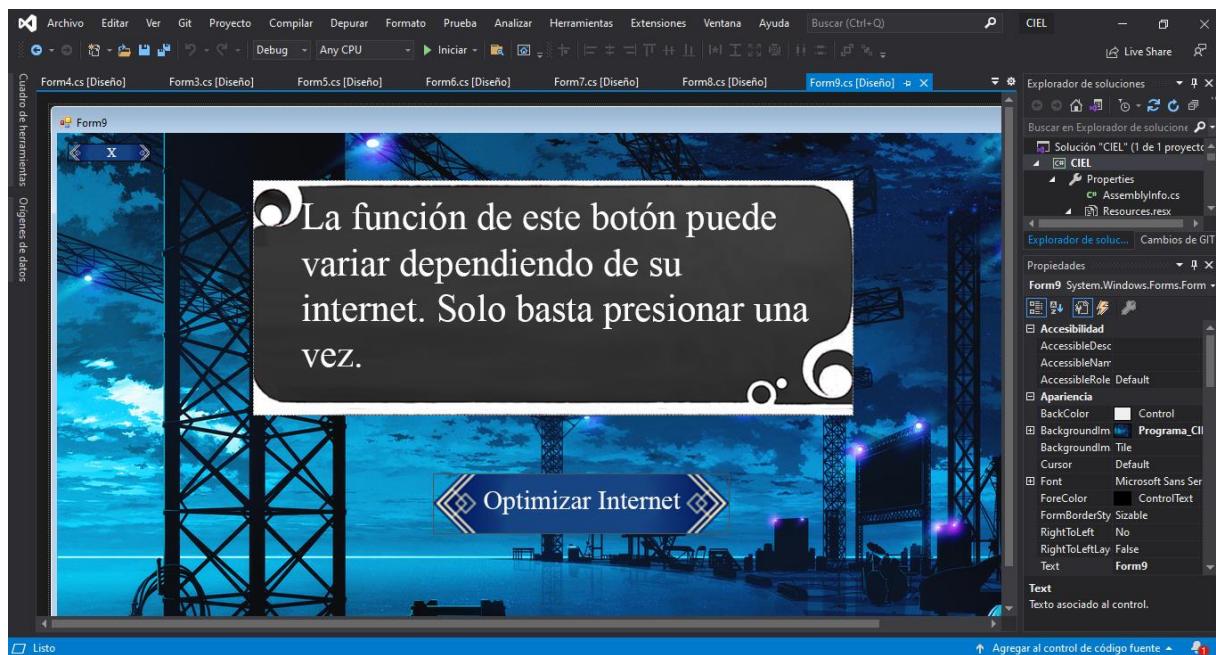




Para el tercer botón con función:



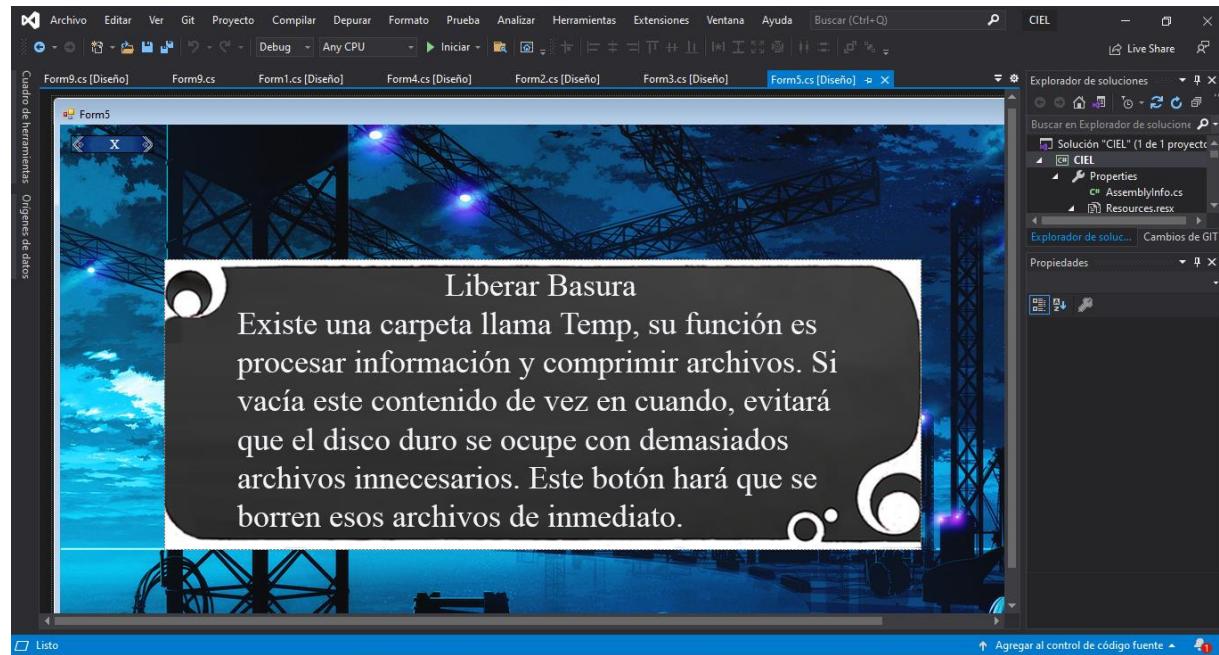
Para el cuarto botón con función:



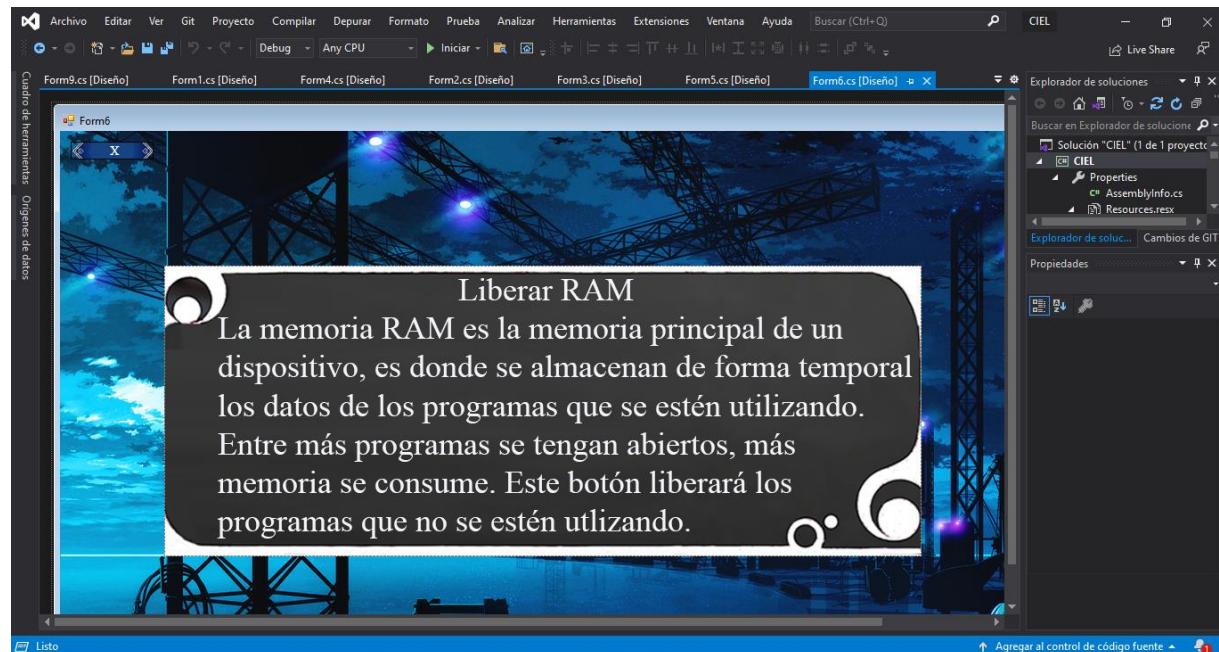


Lo que sigue son los botones que contienen información a detalle de los botones con funciones:

Para el primer botón informativo:

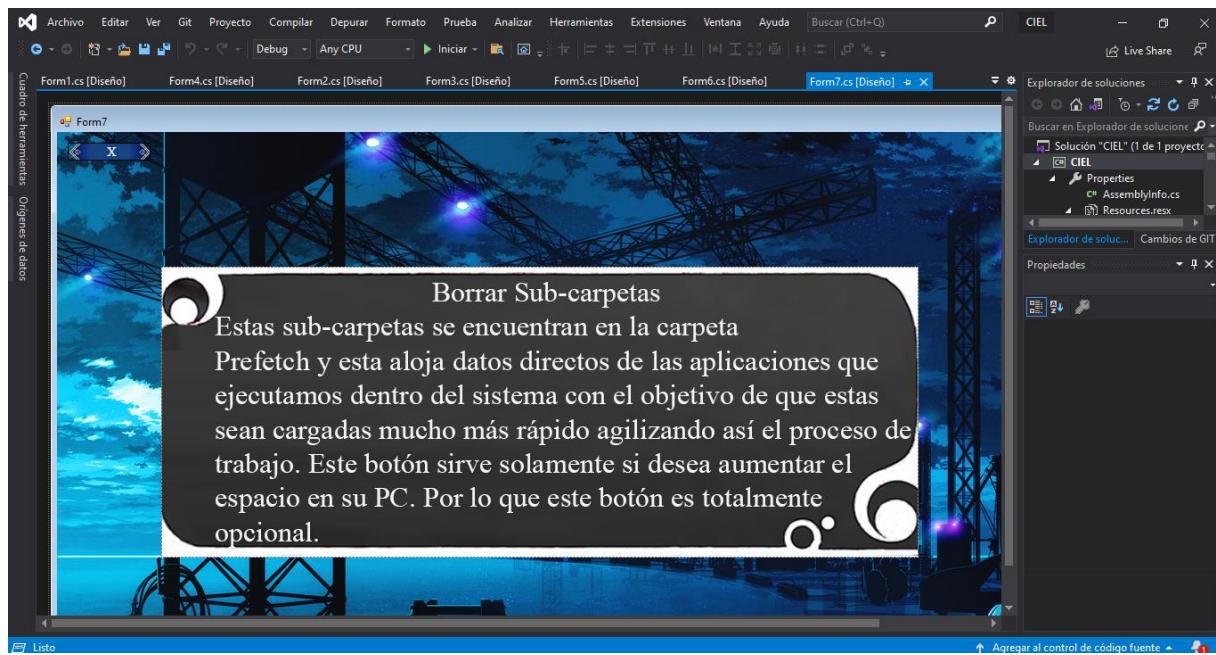


Para el segundo botón informativo:

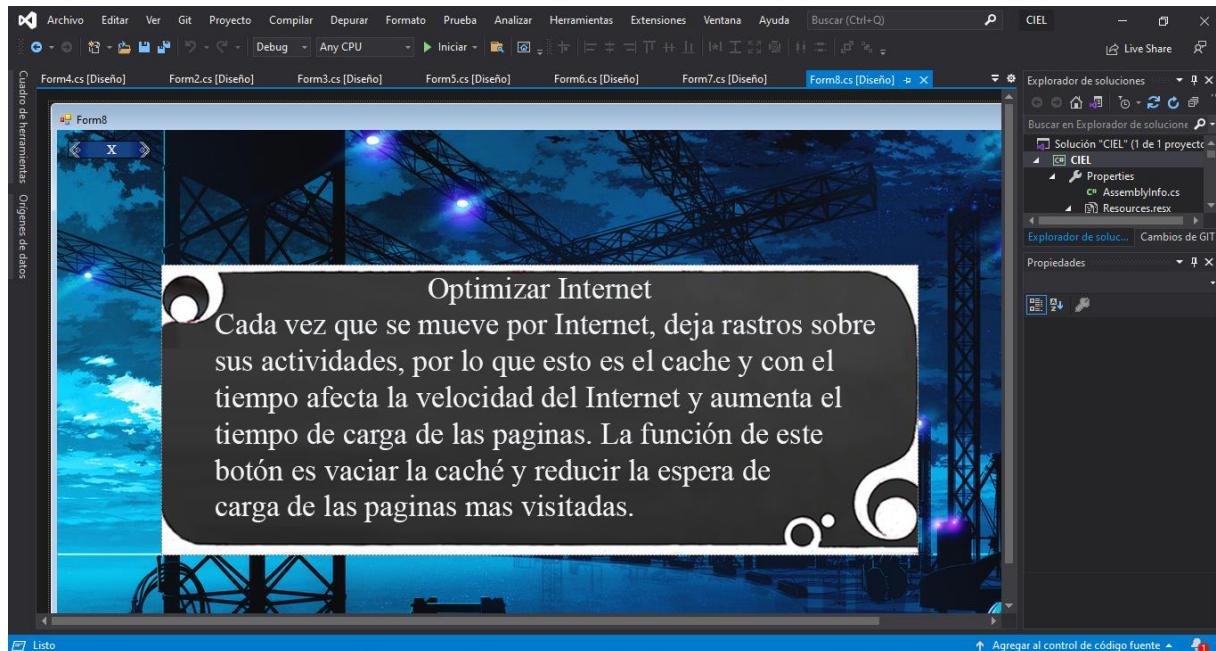




Para el tercer botón informativo:



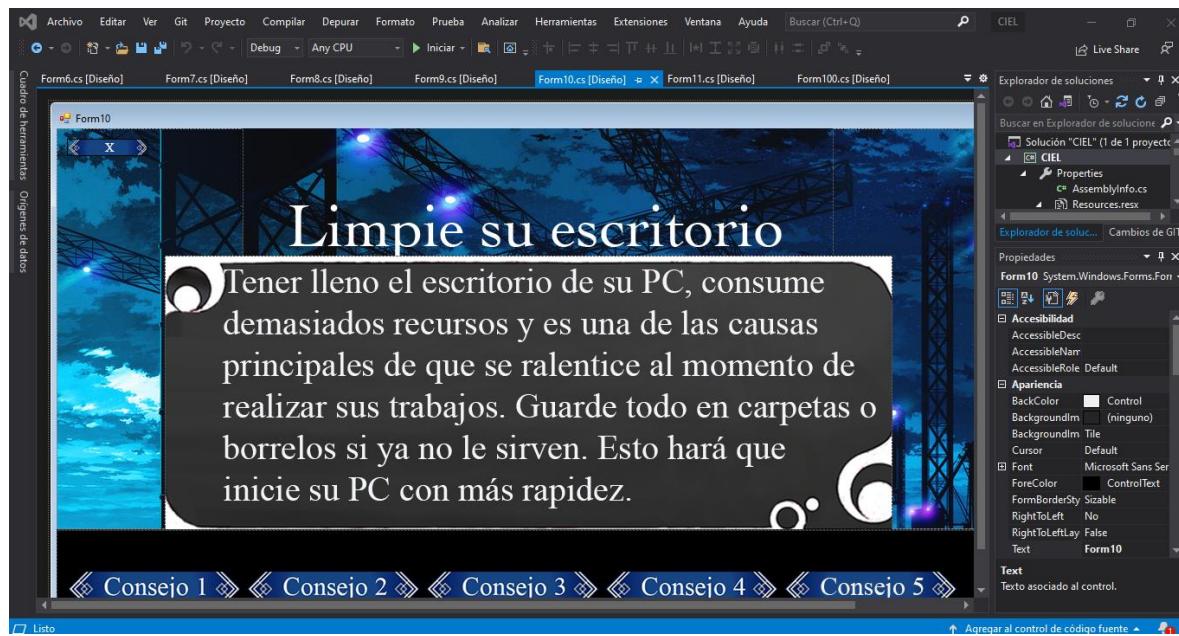
Para el cuarto botón informativo:





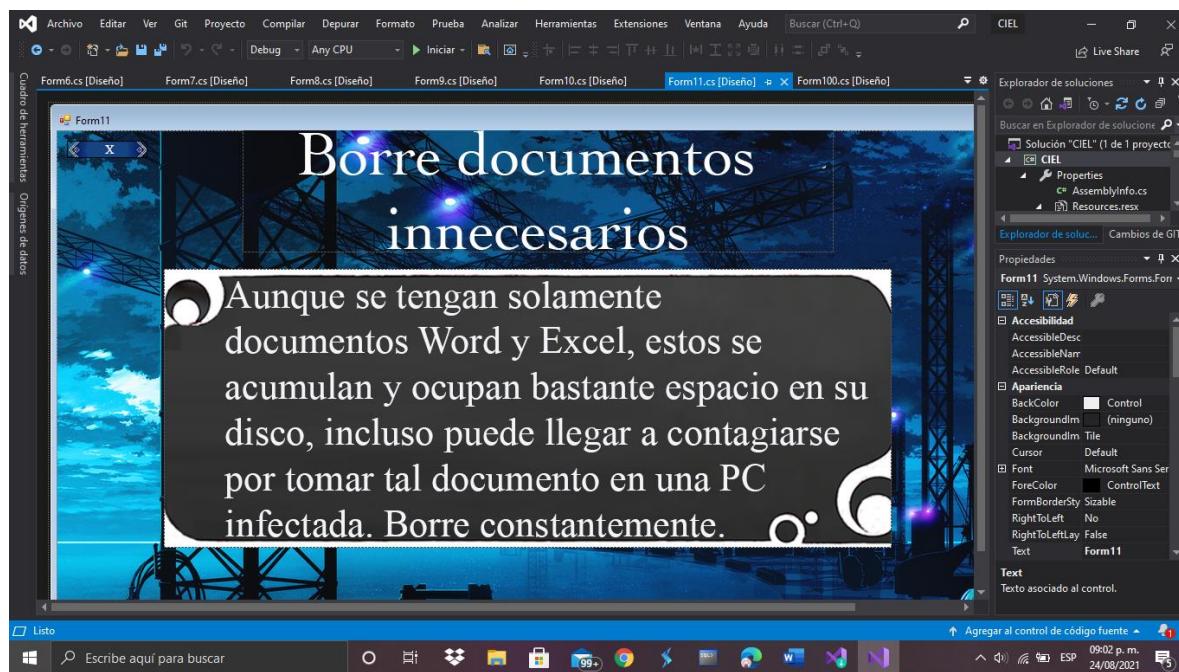
En la parte inferior de la pantalla principal, se encuentra un botón de “Click aquí”, en este aparece con varios botones en la parte inferior mas información adicional del programa.

Así se muestra al momento de darle al botón inferior, esta información pertenece al primer botón con consejo:



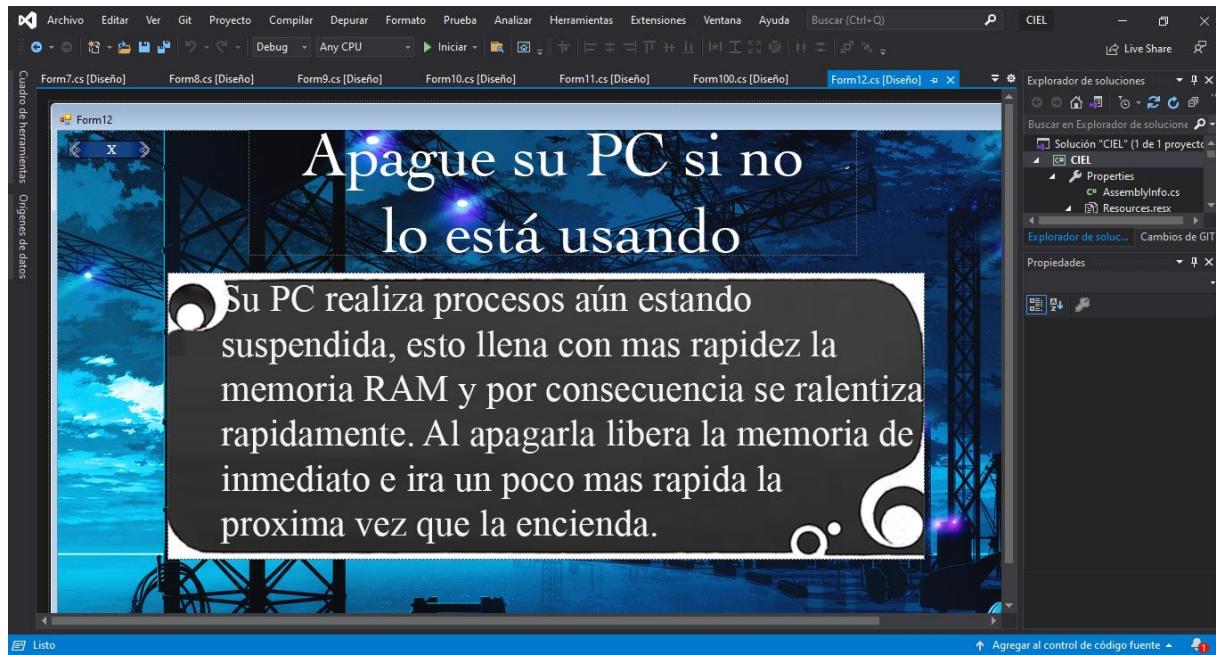
Los botones inferiores al ser presionados, mostrarán los siguiente:

Para el segundo botón con consejo:

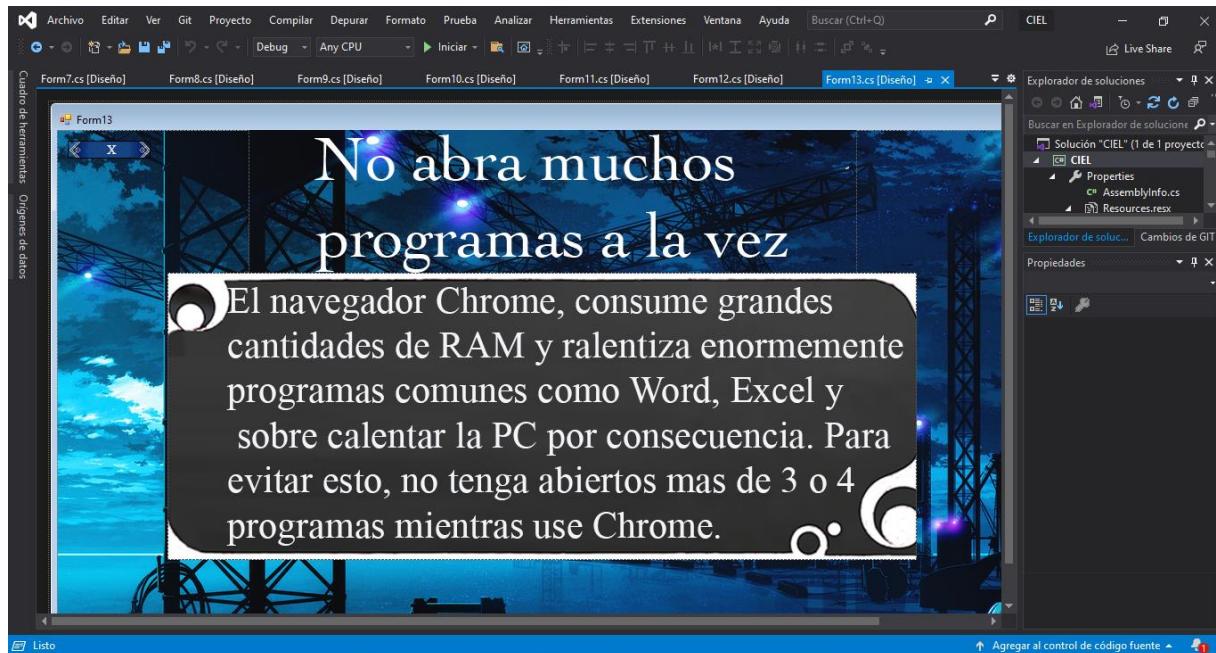




Para el tercer botón con consejo:



Para el cuarto botón con consejo:





Para el quinto botón con consejo:





Codificación

Con el prototipo se establecieron las funciones principales que tendrá el programa original para su adecuado funcionamiento. Estos son las funciones agregadas:

Primero se crea un método sin retorno llamado “CustomizeDesing”. En este método simplemente se ocultan los paneles del sub menú. Para ello establecemos la propiedad a visible a falso en los paneles llamados “Funciones” e “Información de botones”, estos tienen el nombre de “panelfun” y “panelop”.

Por último, se debe de invocar el método en el constructor antes del “InitializeComponent”.

```
1 referencia
public panel()
{
    InitializeComponent();
    customizeDesing();
}
1 referencia
private void customizeDesing()
{
    panelfun.Visible = false;
    panelop.Visible = false;
    //...
}
```

Se crea otro método de tipo vacío para ocultar el sub menú. En este método se ocultará el sub menú mostrado previamente, se debe de añadir la condición:

“Si el panel sub menú está visible, lo ocultamos estableciendo la propiedad visible a falso”. Se deberá hacer lo mismo en ambos sub menús.

```
10 referencias
private void hideSubMenu()
{
    if (panelfun.Visible == true)
        panelfun.Visible = false;
    if (panelop.Visible == true)
        panelop.Visible = false;
}
```



Se crea otro método de tipo vacío para mostrar el sub menú. Ya que todos los sub menús son paneles, como parámetro, indicamos que es de tipo Panel llamado sub menú, se crea una condición:

“Si panel sub menú está oculto, mostramos el sub menú”, pero antes se debe de invocar el método de ocultar el sub menú si es que hay un sub menú abierto, luego mostrar el sub menú. En caso contrario, es decir, si el sub menú esta visible o abierto, se vuelve a ocultar el sub menú. De esta manera se alternará la visibilidad del sub menú actual o el sub menú que se mostró anteriormente, finalmente se invoca todos los métodos en los respectivos botones.

```
10 referencias
private void hideSubMenu()
{
    if (panelfun.Visible == true)
        panelfun.Visible = false;
    if (panelop.Visible == true)
        panelop.Visible = false;
}

2 referencias
private void ShowSubMenu(Panel subMenu)
{
    if (subMenu.Visible == false){
        hideSubMenu();
        subMenu.Visible = true;
    }
    else
        subMenu.Visible = false;
}
```

Para ello se debe de ir directamente al botón e invocar el método de mostrar el sub menú. Como parámetro enviamos el panel sub menú en este caso “*panelfun*”.

```
// Menu funciones
1 referencia
private void btnfunciones_Click(object sender, EventArgs e)
{
    ShowSubMenu(panelfun);
}
```



Se crea el método click de cada botón del sub menú.

Se invoca el código ocultar el sub menú obviamente después de seleccionar cualquier botón del sub menú. Se debe de realizar lo mismo con los demás botones.

```
    ...
    //Codigo y mas codigo
    //
    hideSubMenu();
}
```

Todo el código anterior, será utilizado para casi todas las pantallas a mostrar, por lo que se mostrará de manera completa todo el código utilizado en todas las pantallas:

Para la pantalla principal:

```
C# CIEL
dri de herramientas Orígenes de datos
  # CIEL
  Programa_CIEL.panel
  Using System;
  using System.Collections.Generic;
  using System.ComponentModel;
  using System.Data;
  using System.Drawing;
  using System.Linq;
  using System.Text;
  using System.Threading.Tasks;
  using System.Windows.Forms;

  namespace Programa_CIEL
  {
    4 referencias
    public partial class panel : Form
    {
      1 referencia
      public panel()
      {
        InitializeComponent();
        customizeDesing();
      }
      1 referencia
      private void customizeDesing()
      {
        panelfun.Visible = false;
        panelop.Visible = false;
        ...
      }
    }
  }
```



En esta parte del código, tiene la función de llamar y almacenar a todas las pantallas llamadas, estas pantallas vendrán de los botones antes mencionados:

The image shows two side-by-side code editors in Microsoft Visual Studio. Both editors have the title bar "CIEL" and the tab "Programa_CIEL.panel".

Editor 1 (Top):

```
10 referencias
private void hideSubMenu()
{
    if (panelfun.Visible == true)
        panelfun.Visible = false;
    if (panelop.Visible == true)
        panelop.Visible = false;
}

2 referencias
private void ShowSubMenu(Panel subMenu)
{
    if (subMenu.Visible == false){
        hideSubMenu();
        subMenu.Visible = true;
    }
    else
        subMenu.Visible = false;
}

// Menu funciones
1 referencia
private void btnfunciones_Click(object sender, EventArgs e)
{
    ShowSubMenu(panelfun);
}
```

Editor 2 (Bottom):

```
1 referencia
private void button2_Click(object sender, EventArgs e)
{
    openChildForm(new Form2());
    ...
    //Codigo y mas codigo
    //
    hideSubMenu();
}

1 referencia
private void button3_Click(object sender, EventArgs e)
{
    openChildForm(new Form3());
    ...
    //Codigo y mas codigo
    //
    hideSubMenu();
}

1 referencia
private void button4_Click(object sender, EventArgs e)
{
    openChildForm(new Form4());
    ...
    //Codigo y mas codigo
    //
    hideSubMenu();
}
```



```
//Menu informacion
referencia
private void btninfo_Click(object sender, EventArgs e)
{
    ShowSubMenu(panelop);
}

referencia
private void button7_Click(object sender, EventArgs e)
{
    openChildForm(new Form5());
    ...
    //Codigo y mas codigo
    //
    hideSubMenu();
}

referencia
private void button6_Click(object sender, EventArgs e)
{
    openChildForm(new Form6());
    ...
    //Codigo y mas codigo
    //
    hideSubMenu();
}

referencia
private void button5_Click(object sender, EventArgs e)
{
    openChildForm(new Form7());
    ...
    //Codigo y mas codigo
    //
    hideSubMenu();
}
```

Este código se encarga de mostrar el contenido de las pantallas de los botones, ojo, no es lo mismo la función llamar a las pantallas, a la de mostrar el contenido de las mismas:

```
private Form activeForm = null;
referencias
private void openChildForm(Form childForm)
{
    if (activeForm != null)
        activeForm.Close();
    activeForm = childForm;
    childForm.TopLevel = false;
    childForm.FormBorderStyle = FormBorderStyle.None;
    childForm.Dock = DockStyle.Fill;
    panelChildForm.Controls.Add(childForm);
    panelChildForm.Tag = childForm;
    childForm.BringToFront();
    childForm.Show();
}
```



Después se continua a llamar el código anterior de llamar a las pantallas:

The screenshot shows a Windows application window with two code snippets in a code editor. The left snippet is for button1_Click and the right snippet is for button9_Click. Both snippets include code to open a child form and hide a submenu. The code is as follows:

```
1 referencia
private void pictureBox2_Click(object sender, EventArgs e)
{
}

1 referencia
private void panelChildForm_Paint(object sender, PaintEventArgs e)
{
}

1 referencia
private void button1_Click(object sender, EventArgs e)
{
    openChildForm(new Form10());
    //...
    //Código y mas código
    //
    hideSubMenu();
}

1 referencia
private void button8_Click(object sender, EventArgs e)
{
    openChildForm(new Form8());
    //...
    //Código y mas código
    //
    hideSubMenu();
}

1 referencia
private void button9_Click(object sender, EventArgs e)
{
    openChildForm(new Form8());
    //...
    //Código y mas código
    //
    hideSubMenu();
}
```

En las pantallas con funciones tienen el elemento Script seleccionado que se ejecutará al momento de dar click en el botón. Esto es gracias al using y a la clase Process del programa como se muestra a continuación.

Para el primer botón con función:

The screenshot shows the code for the first button's click event. It includes a using statement for System.Diagnostics and a call to System.Diagnostics.Process.Start with a command to run a script. The code is as follows:

```
Using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.Diagnostics;

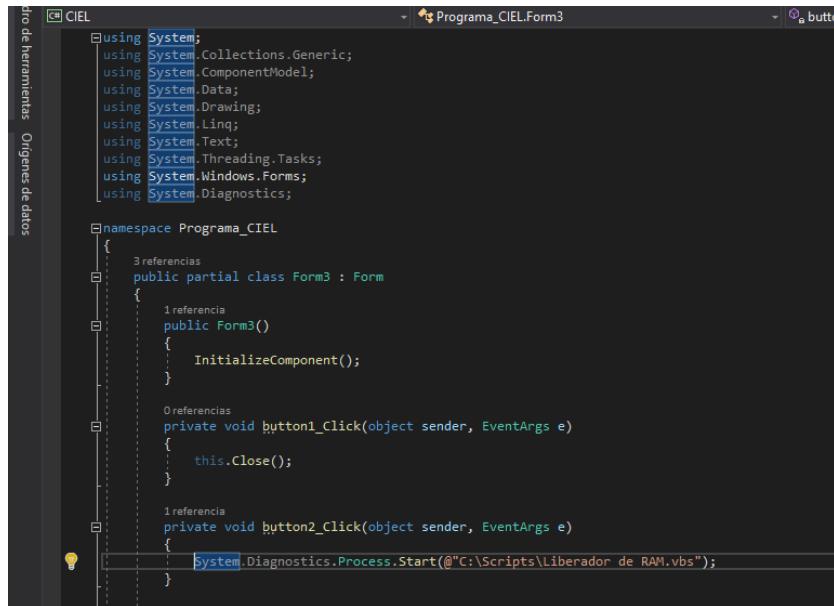
namespace Programa_CIEL
{
    3 Referencias
    public partial class Form2 : Form
    {
        1 referencia
        public Form2()
        {
            InitializeComponent();
        }

        1 referencia
        private void button1_Click(object sender, EventArgs e)
        {
            this.Close();
        }

        1 referencia
        private void button2_Click(object sender, EventArgs e)
        {
            System.Diagnostics.Process.Start(@"C:\Scripts\Temp.cmd");
        }
    }
}
```



Para el segundo botón con función:



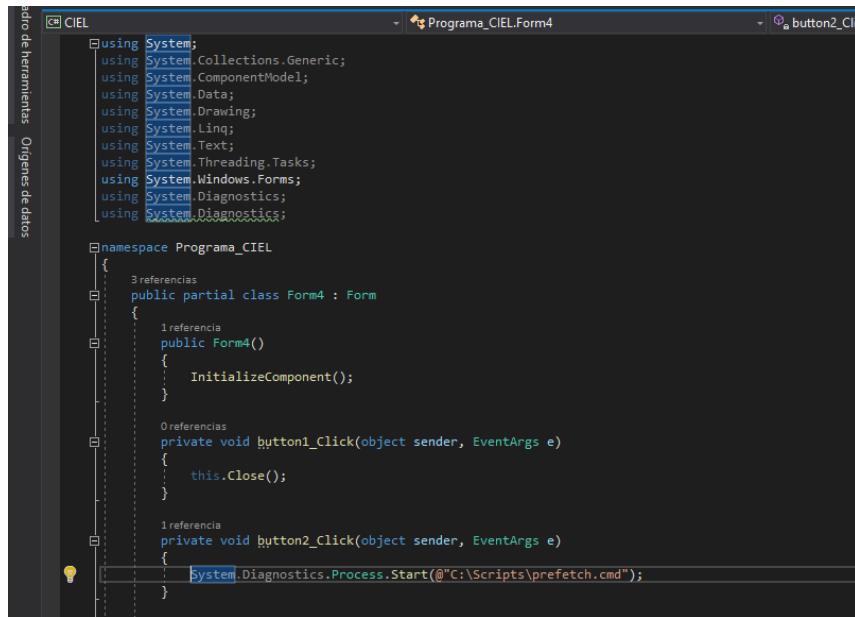
```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.Diagnostics;

namespace Programa_CIEL
{
    public partial class Form3 : Form
    {
        public Form3()
        {
            InitializeComponent();
        }

        private void button1_Click(object sender, EventArgs e)
        {
            this.Close();
        }

        private void button2_Click(object sender, EventArgs e)
        {
            System.Diagnostics.Process.Start(@"C:\Scripts\Liberador de RAM.vbs");
        }
    }
}
```

Para el tercer botón con función:



```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.Diagnostics;

namespace Programa_CIEL
{
    public partial class Form4 : Form
    {
        public Form4()
        {
            InitializeComponent();
        }

        private void button1_Click(object sender, EventArgs e)
        {
            this.Close();
        }

        private void button2_Click(object sender, EventArgs e)
        {
            System.Diagnostics.Process.Start(@"C:\Scripts\prefetch.cmd");
        }
    }
}
```



Para el cuarto botón con función:

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.Diagnostics;

namespace Programa_CIEL
{
    public partial class Form9 : Form
    {
        public Form9()
        {
            InitializeComponent();
        }

        private void button2_Click(object sender, EventArgs e)
        {
            System.Diagnostics.Process.Start(@"C:\Scripts\Mejorar Internet.cmd");
        }

        private void button1_Click(object sender, EventArgs e)
        {
            this.Close();
        }

        private void button1_Click_1(object sender, EventArgs e)
        {
        }
    }
}
```

En los botones con solo información, estos son más “sencillos”, ya que no llevan código complejo, solamente tiene el código necesario para mostrar la información:

Para el primer botón informativo:

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace Programa_CIEL
{
    public partial class Form5 : Form
    {
        public Form5()
        {
            InitializeComponent();
        }

        private void button1_Click(object sender, EventArgs e)
        {
            this.Close();
        }

        private void button1_Click_1(object sender, EventArgs e)
        {
            this.Close();
        }
    }
}
```



Para el segundo botón informativo:

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace Programa_CIEL
{
    public partial class Form6 : Form
    {
        public Form6()
        {
            InitializeComponent();
        }

        private void button1_Click(object sender, EventArgs e)
        {
            this.Close();
        }

        private void button1_Click_1(object sender, EventArgs e)
        {
            this.Close();
        }
    }
}
```

Para el tercer botón informativo:

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace Programa_CIEL
{
    public partial class Form7 : Form
    {
        public Form7()
        {
            InitializeComponent();
        }

        private void button1_Click(object sender, EventArgs e)
        {
            this.Close();
        }

        private void button1_Click_1(object sender, EventArgs e)
        {
            this.Close();
        }
    }
}
```



Para el cuarto botón informativo:

```
do de herramientas Orígenes de datos
C# CIEL Programa_CIEL.Form8
using System;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace Programa_CIEL
{
    3 referencias
    public partial class Form8 : Form
    {
        1 referencia
        public Form8()
        {
            InitializeComponent();
        }

        1 referencia
        private void Form8_Load(object sender, EventArgs e)
        {

        }

        0 referencias
        private void label1_Click(object sender, EventArgs e)
        {

        }

        0 referencias
        private void button1_Click(object sender, EventArgs e)
        {
            this.Close();
        }
    }
}
```

En el caso del botón “Click aquí”, se tuvo que usar el código de la pantalla principal, ya que este botón llama y almacena todos los botones pertenecientes al botón “Click aquí”, por lo que básicamente, es el mismo código, sin ningún cambio. A continuación, se muestra todo el código empleado en el último botón:

Para la pantalla principal o el primer botón con consejo:

```
Formulas Formulas Formulas Formulas Formulas[ Diseño] Formulas[ Diseño]
radio de herramientas Orígenes de datos
C# CIEL Programa_CIEL.Form10
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace Programa_CIEL
{
    3 referencias
    public partial class Form10 : Form
    {
        1 referencia
        public Form10()
        {
            InitializeComponent();
        }
    }
}
```



```
private Form ActiveForm = null;
5 referencias
private void openChildForm(Form childForm)
{
    if (ActiveForm != null)
        ActiveForm.Close();
    ActiveForm = childForm;
    childForm.TopLevel = false;
    childForm.FormBorderStyle = FormBorderStyle.None;
    childForm.Dock = DockStyle.Fill;
    panelChildForm.Controls.Add(childForm);
    panelChildForm.Tag = childForm;
    childForm.BringToFront();
    childForm.Show();
}

1 referencia
private void button1_Click(object sender, EventArgs e)
{
    openChildForm(new Form1());
    //...
    //Código y más código
    //
}

1 referencia
private void button2_Click(object sender, EventArgs e)
{
    openChildForm(new Form12());
    //...
    //Código y más código
    //
}
```

```
1 referencia
private void button3_Click(object sender, EventArgs e)
{
    openChildForm(new Form13());
    //...
    //Código y más código
    //
}

1 referencia
private void button4_Click(object sender, EventArgs e)
{
    openChildForm(new Form14());
    //...
    //Código y más código
    //
}

1 referencia
private void button5_Click(object sender, EventArgs e)
{
    openChildForm(new Form100());
    //...
    //Código y más código
    //
}

0 referencias
private void button6_Click(object sender, EventArgs e)
{
    this.Close();
}

1 referencia
private void button6_Click_1(object sender, EventArgs e)
{
    this.Close();
}
```



Al igual que los botones anteriores de información, estos son similares en código, pero se muestran para evitar faltas de información:

Para el segundo botón con consejo:

```
using System;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace Programa_CIEM
{
    public partial class Form11 : Form
    {
        public Form11()
        {
            InitializeComponent();
        }

        private void panelChildForm_Paint(object sender, PaintEventArgs e)
        {
        }

        private void button1_Click(object sender, EventArgs e)
        {
            this.Close();
        }

        private void button1_Click_1(object sender, EventArgs e)
        {
            this.Close();
        }
    }
}
```

Para el tercer botón con consejo:

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace Programa_CIEM
{
    public partial class Form12 : Form
    {
        public Form12()
        {
            InitializeComponent();
        }

        private void button1_Click(object sender, EventArgs e)
        {
            this.Close();
        }

        private void button1_Click_1(object sender, EventArgs e)
        {
            this.Close();
        }
    }
}
```



Para el cuarto botón con consejo:

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace Programa_CIEM
{
    public partial class Form13 : Form
    {
        public Form13()
        {
            InitializeComponent();
        }

        private void button1_Click(object sender, EventArgs e)
        {
            this.Close();
        }

        private void button1_Click_1(object sender, EventArgs e)
        {
            this.Close();
        }
    }
}
```

Para el quinto botón con consejo:

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace Programa_CIEM
{
    public partial class Form14 : Form
    {
        public Form14()
        {
            InitializeComponent();
        }

        private void button1_Click(object sender, EventArgs e)
        {
            this.Close();
        }

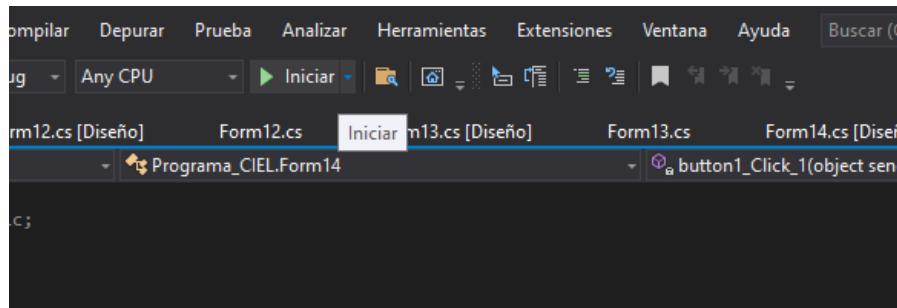
        private void button1_Click_1(object sender, EventArgs e)
        {
            this.Close();
        }
    }
}
```



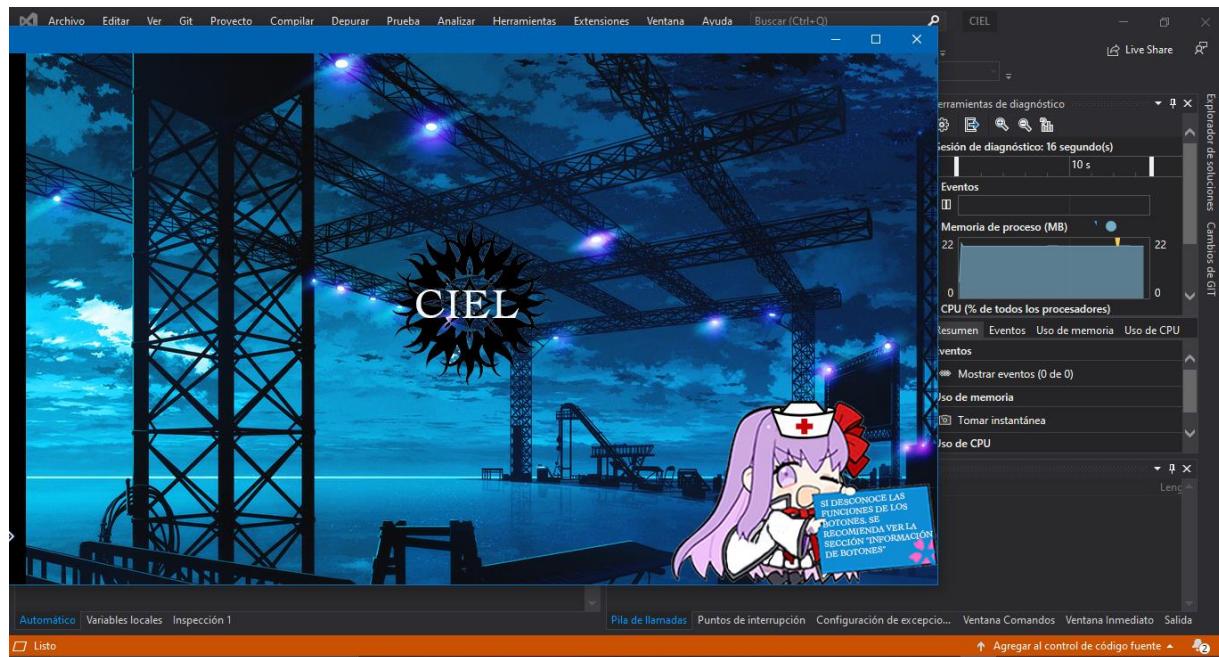
3.9.4 Prueba del programa

Por medio del Visual Studio

Si el programa no presenta ningún problema de compilación, este podrá ser ejecutado de manera sencilla presionando el ícono de **Iniciar**.



El programa será ejecutado directamente, y en forma de ayuda, a lado aparecerá una sección de **Memoria de proceso (MB)**, su función es mostrar el impacto que tiene en el CPU al ser ejecutado. Si estuviera realizando un proceso muy pesado, la barra azul tendría un "grano" naranja o rojo.



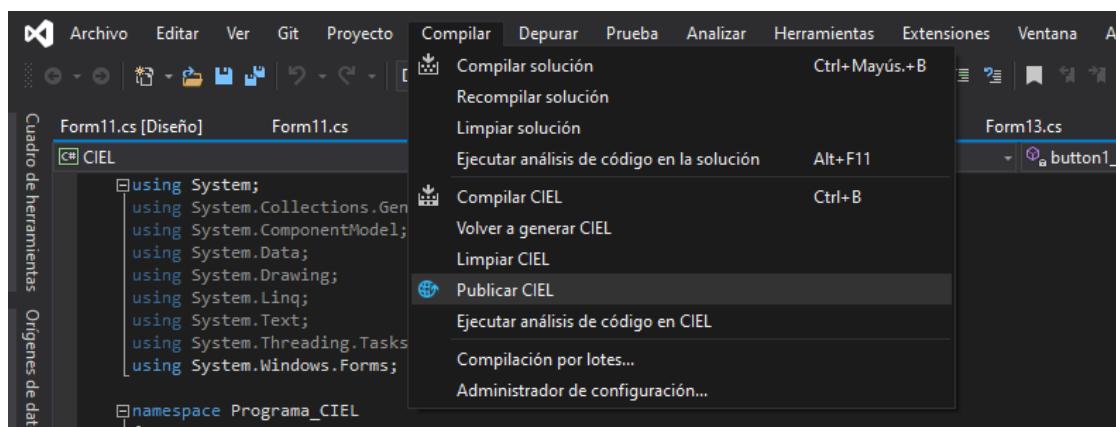


Y al ser cerrado, mostrará todos los procesos realizados mientras se ejecutaba el programa.

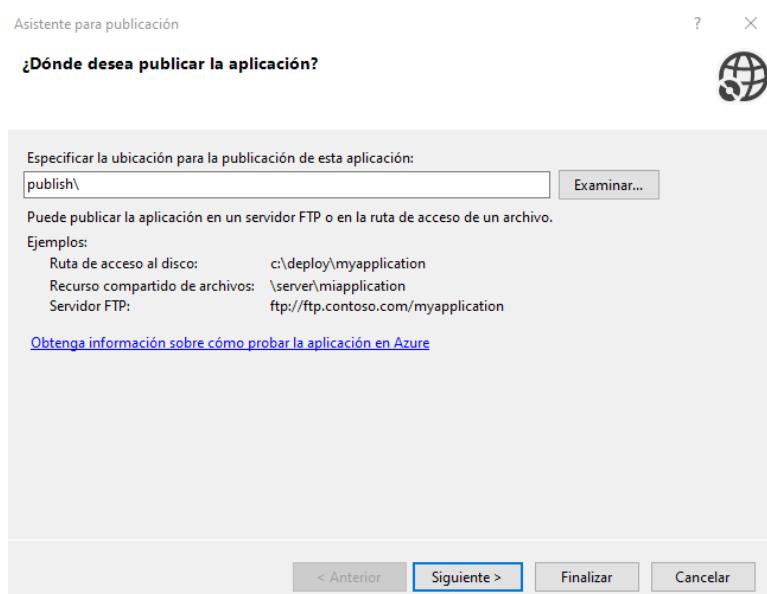
```
'CIEL.exe' (CLR v4.0.30319: CIEL.exe): 'C:\WINDOWS\Microsoft.NET\Assembly\GAC_MSIL\System\v4.0.4.0.0.0_b77a5c561934e089\System.dll' cargado. Se omitió 1 archivo.
'CIEL.exe' (CLR v4.0.30319: CIEL.exe): 'C:\WINDOWS\Microsoft.NET\Assembly\GAC_MSIL\System.Drawing\v4.0.4.0.0.0_b03f57f1ld50a3a\System.Drawing.dll' cargado. Se omitió 1 archivo.
'CIEL.exe' (CLR v4.0.30319: CIEL.exe): 'C:\WINDOWS\Microsoft.NET\Assembly\GAC_MSIL\System.Configuration\v4.0.4.0.0.0_b03f57f1ld50a3a\System.Configuration.dll' cargado. Se omitió 1 archivo.
'CIEL.exe' (CLR v4.0.30319: CIEL.exe): 'C:\WINDOWS\Microsoft.NET\Assembly\GAC_MSIL\System.Core\v4.0.4.0.0.0_b77a5c561934e089\System.Core.dll' cargado. Se omitió 1 archivo.
'CIEL.exe' (CLR v4.0.30319: CIEL.exe): 'C:\WINDOWS\Microsoft.NET\Assembly\GAC_MSIL\System.Xml\v4.0.4.0.0.0_b77a5c561934e089\System.Xml.dll' cargado. Se omitió 1 archivo.
El programa '[2468] CIEL.exe' terminó con código 0 (0x0).
```

Por medio del paquete de instalación

Cuando se desarrolló el prototipo y darlo por “terminado” para ser probado en los sistemas operativos del hospital, se entró a la ventana **Compilar**, y se seleccionó la opción de **Publicar CIEL**.

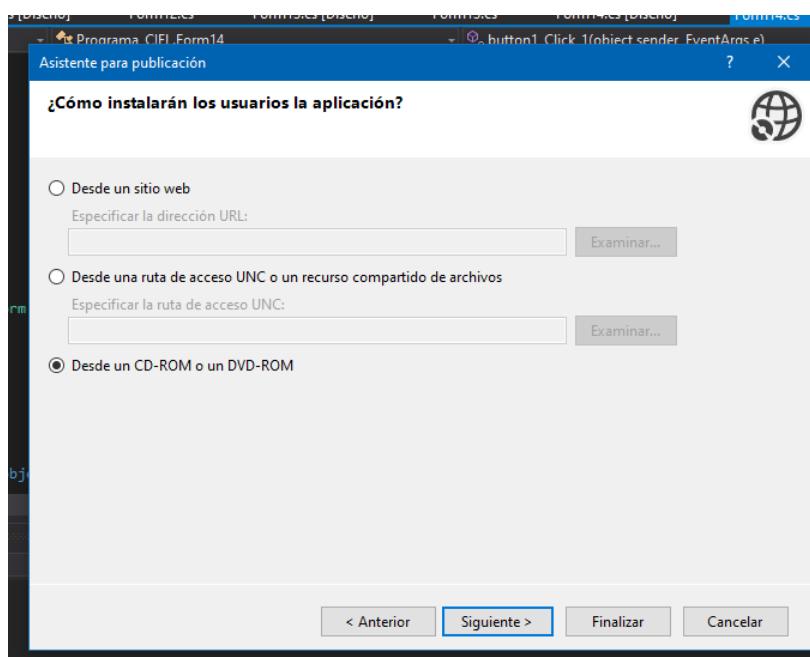


Al ser presionado, manda la siguiente ventana y lo primero que pide es la ubicación del en donde se publicará el programa.

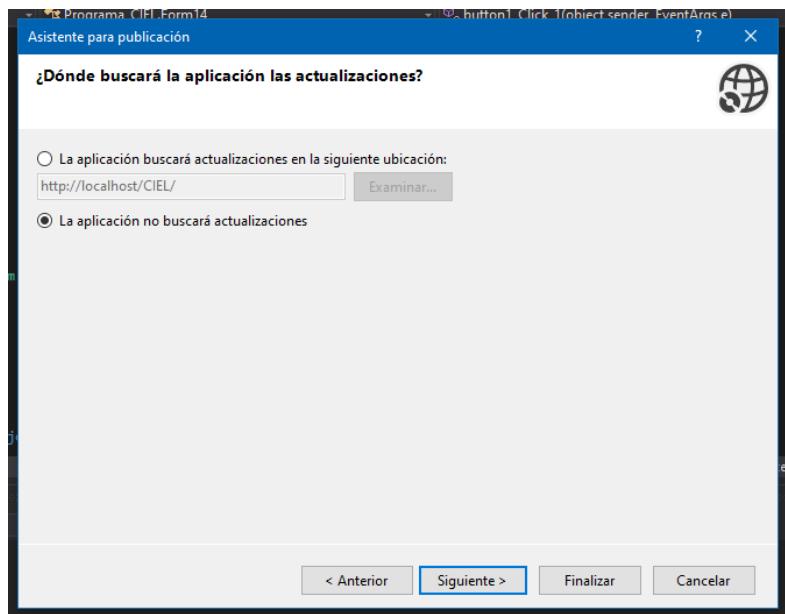




Luego se pide de cómo se instalará. Al ser por instalación directa, se le dará la opción de **CD-ROM** o un **DVD-ROM**.

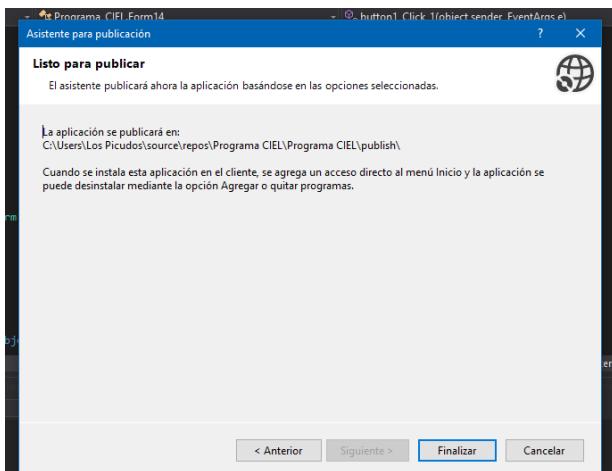


Si existieran actualizaciones constantes, se le daría a la primera opción, pero como no es así, se le dará que no buscará actualizaciones.

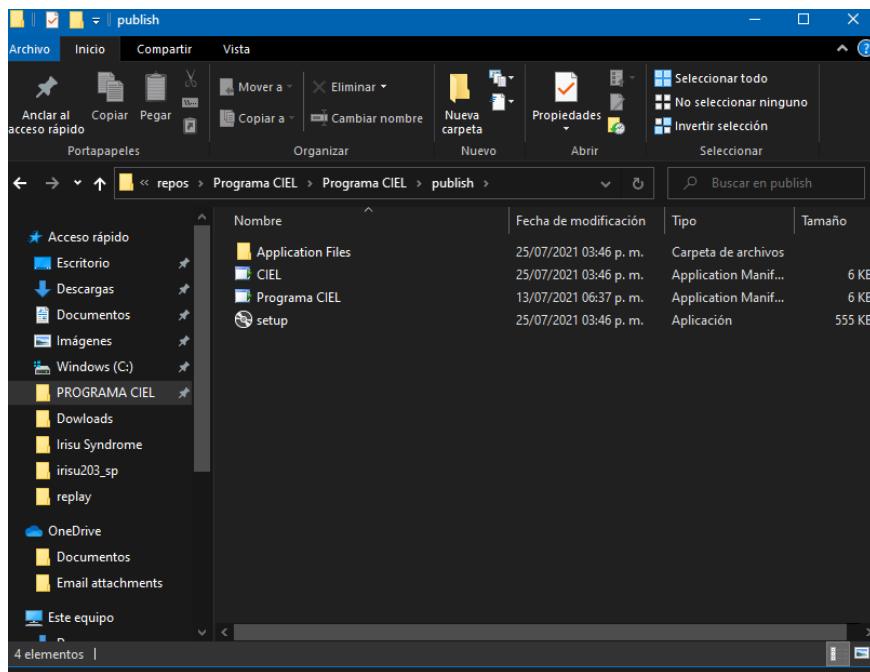




Finalmente, al darle **Finalizar**, este creará el paquete de instalación para poder ser instalado en la PC seleccionada.

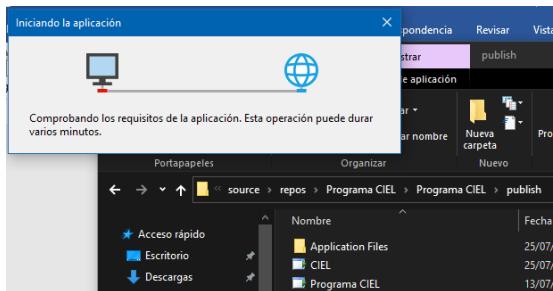


Al meterse a la ubicación mostrada, el elemento importante, es aquel con forma de disco y el nombre **setup**.





Al ser seleccionado, en este deberá de aparecer la siguiente ventana a manera de señal para saber que el programa se instalará correctamente.



3.9.5 Ejecución del programa

Después de todo el código mostrado, se mostrará el aspecto y funcionalidad que tiene dentro del programa.

Además, que se debe de mencionar que existen funciones que no son posibles de agregar por medio del código, sino que se deben de realizar por medio del diseño de cada pantalla.

Por lo que a continuación se mostrará las funciones y el aspecto del programa estando terminado y ejecutado.

Panel negro

La primera pantalla al mostrarse al momento de abrir por primera vez el programa, muestra del lado izquierdo un gran panel color negro, en este vienen dos botones que, dentro de ellos, hay cinco botones en cada uno. Ambos botones se complementan, ya que viene información que ayuda a entender mejor al otro.

En la parte inferior del mismo panel negro, hay un texto que dice lo siguiente: “¿No confía en los botones? ¡Sepa cómo mejorar su PC usted mismo!”, después de ello, debajo hay un botón con el texto “Click aquí”.

Enfermera de información

En la parte inferior derecha del programa, se puso a una “enfermera” que su función es solamente en guiar al usuario si es que este mismo no sabe con qué botón empezar. Tiene escrito lo siguiente: “SI DESCONOCE LAS FUNCIONES DE BOTONES, SE RECOMIENDA VER LA SECCIÓN “INFORMACIÓN DE BOTONES””.



Logo

Por último, pero no menos importante, viene el Logo del programa **CIEL** exactamente centrado en el programa.

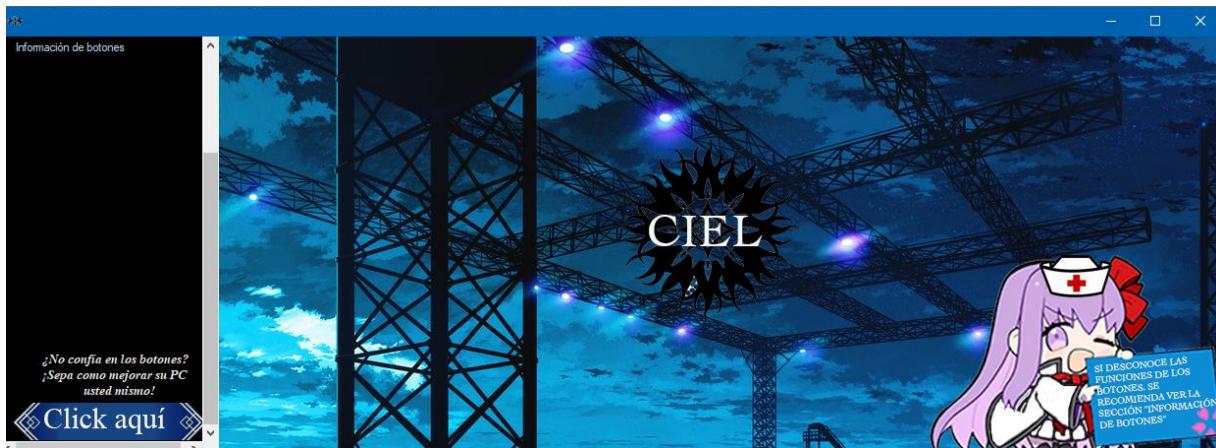


Funciones de la pantalla

AutoScroll

En el panel negro de la izquierda, gracias a la propiedad AutoScroll cambiada de falso a verdadero, se puede navegar en todos sus componentes en caso de que el formulario o "pantalla" del programa quede chico como en el siguiente caso:

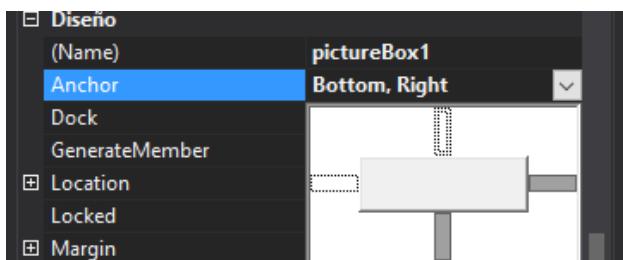




Formulario responsive o adaptable

Gracias a la propiedad Anchor, se pueden quitar o agregar las “anclas” de los elementos disponibles. Con esto se puede manejar la adaptabilidad de cada elemento, con esto cualquier logo o botón con anclas definidas anteriormente, se ajustará automáticamente al tamaño actual del formulario.

Estas son las anclas vistas desde el Visual Studio.





Se puede ver que la enfermera y el logo se adaptan a cualquier tamaño que se tenga presente en el programa, ya sea muy amplio o muy estrecho. Debe de mencionarse que no es recomendable usar siempre estas anclas, ya que puede ocasionar que demás elementos se encimen uno tras otro si es que el tamaño del formulario es demasiado estrecho. Para ello es recomendable verlo con el tamaño más amplio posible para que sea más agradable verlo.





Tamaño del formulario/pantalla

El tamaño mínimo del formulario es de 1263 de ancho por 635 de largo, esto es debido a que se viera a primera vista todos los componentes disponibles, pero sobre todo es para darle más visibilidad a la información de cada botón. Esto último se verá más tarde.



Botón Funciones

Funciones del botón

Método customizeDesing

La función de este método es la de ocultar el panel en donde se guardan los sub botones que incluye este botón.

Método hideSubMenu

La función de este método es la de ocultar el panel que se ha mostrado anteriormente, para ello se realiza la siguiente condición:

“Si el panel sub menú está visible, lo ocultamos estableciendo la propiedad visible a falso”.

Método showSubMenu

La función de este método es mostrar el panel seleccionado, para esto tiene la siguiente condición:

“Si el panel está oculto, mostramos el panel”, tiene invocado el método de ocultar el panel si es que hay uno abierto, luego mostrarlo. En caso contrario, es decir, si el panel está visible o abierto, se vuelve a ocultar. De esta manera se alternará la visibilidad del panel actual o el panel que se mostró anteriormente.

En consecuencia, de todo esto, al dar click en el botón, este mostrará todos los sub botones que tiene disponible. Estos mismos sub botones se cerrarán si se presionar el otro botón llamado “Información de botones”. De este botón se hablará más adelante.



Sub botón Liberar Basura

Variable privada Form

La función de esta variable es únicamente abrir solamente un solo formulario y si un formulario está abierto, se cierra. Esto se realizó para evitar que el programa abriera otro proceso y así acumular más pantallas del programa y por consecuencia reducir el rendimiento del programa.

Método OpenChildForm

Su función consiste en la de llamar un formulario al momento de dar click al sub botón. Esto se traduce a que se mostrará la pantalla donde incluya la función o la información del sub botón correspondiente.

Las funciones anteriores, darán paso a mostrar el contenido del sub botón.

Botón X

Su única función es la de cerrar el contenido del botón y regresar desde el inicio. No es necesario presionar este botón cada vez que se quiera entrar a otro sub botón, solamente SI se desea regresar a la pantalla principal.



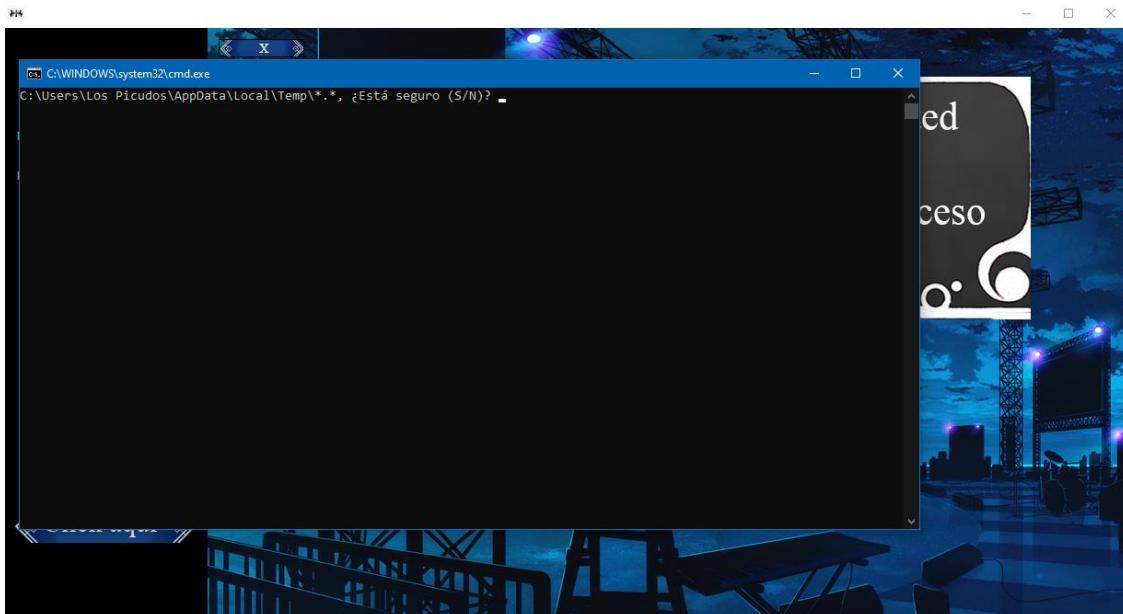
Información

En la parte superior del botón, se encuentra información breve o instrucciones que se deben de seguir si se quiere ejecutar el botón.

Hay scripts que deben de ser ejecutados por el usuario, esto es debido por seguridad y evitar que el Script sea identificado como un elemento malicioso. Esto último es por mera experiencia del programador.



Al ser un Script ejecutado por .cmd, este aparecerá pidiendo la confirmación, si es confirmado, el CMD se cerrará inmediatamente.





Sub botón Liberar RAM

Variable privada Form

La función de esta variable es únicamente abrir solamente un solo formulario y si un formulario está abierto, se cierra. Esto se realizó para evitar que el programa abriera otro proceso y así acumular más pantallas del programa y por consecuencia reducir el rendimiento del programa.

Método OpenChildForm

Su función consiste en la de llamar un formulario al momento de dar click al sub botón. Esto se traduce a que se mostrará la pantalla donde incluya la función o la información del sub botón correspondiente.

Las funciones anteriores, darán paso a mostrar el contenido del sub botón.

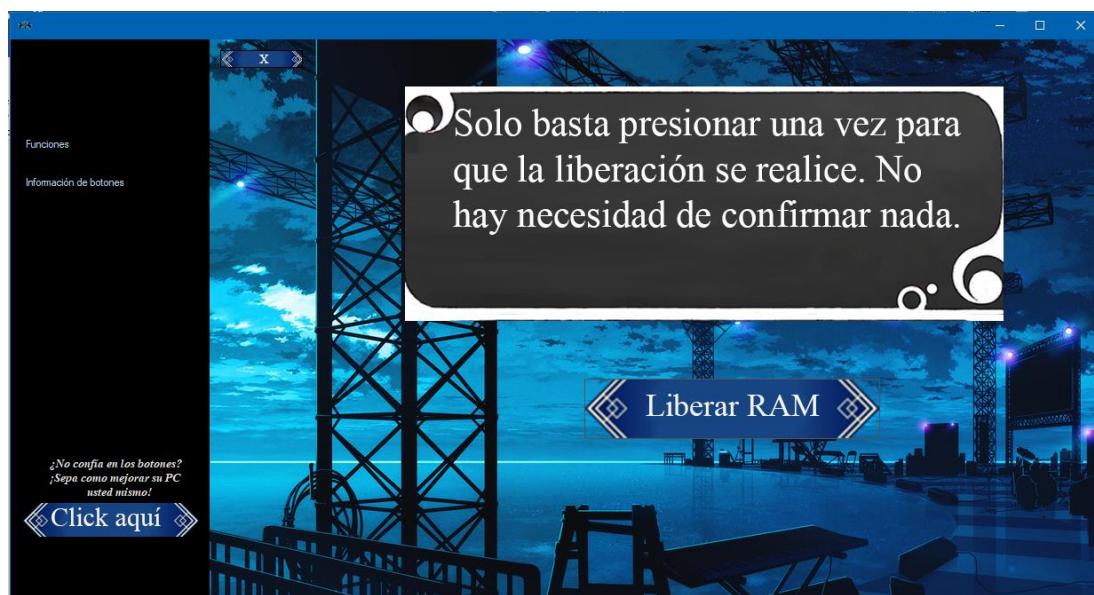
Botón X

Su única función es la de cerrar el contenido del botón y regresar desde el inicio. No es necesario presionar este botón cada vez que se quiera entrar a otro sub botón, solamente SI se desea regresar a la pantalla principal.

Información

En la parte superior del botón, se encuentra información breve o instrucciones que se deben de seguir si se quiere ejecutar el botón.

Al ser un Script ejecutado por .vbs, este se ejecutará en segundo plano por lo que no será visto a simple vista al usuario.





Sub botón Borrar Sub Carpetas

Variable privada Form

La función de esta variable es únicamente abrir solamente un solo formulario y si un formulario está abierto, se cierra. Esto se realizó para evitar que el programa abriera otro proceso y así acumular más pantallas del programa y por consecuencia reducir el rendimiento del programa.

Método OpenChildForm

Su función consiste en la de llamar un formulario al momento de dar click al sub botón. Esto se traduce a que se mostrará la pantalla donde incluya la función o la información del sub botón correspondiente.

Las funciones anteriores, darán paso a mostrar el contenido del sub botón.

Botón X

Su única función es la de cerrar el contenido del botón y regresar desde el inicio. No es necesario presionar este botón cada vez que se quiera entrar a otro sub botón, solamente SI se desea regresar a la pantalla principal.

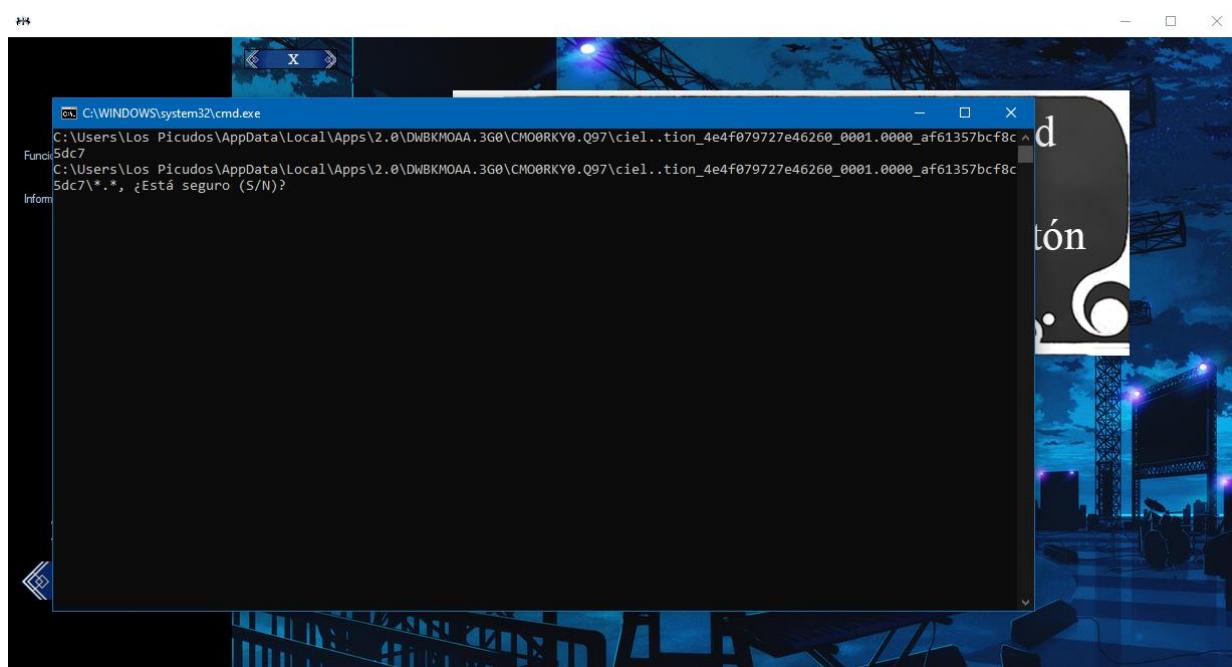
Información

En la parte superior del botón, se encuentra información breve o instrucciones que se deben de seguir si se quiere ejecutar el botón.





Al ser un Script ejecutado por .cmd, este aparecerá pidiendo la confirmación, si es confirmado, el CMD se cerrará inmediatamente.





Sub botón Optimizar Internet

Variable privada Form

La función de esta variable es únicamente abrir solamente un solo formulario y si un formulario está abierto, se cierra. Esto se realizó para evitar que el programa abriera otro proceso y así acumular más pantallas del programa y por consecuencia reducir el rendimiento del programa.

Método OpenChildForm

Su función consiste en la de llamar un formulario al momento de dar click al sub botón. Esto se traduce a que se mostrará la pantalla donde incluya la función o la información del sub botón correspondiente.

Las funciones anteriores, darán paso a mostrar el contenido del sub botón.

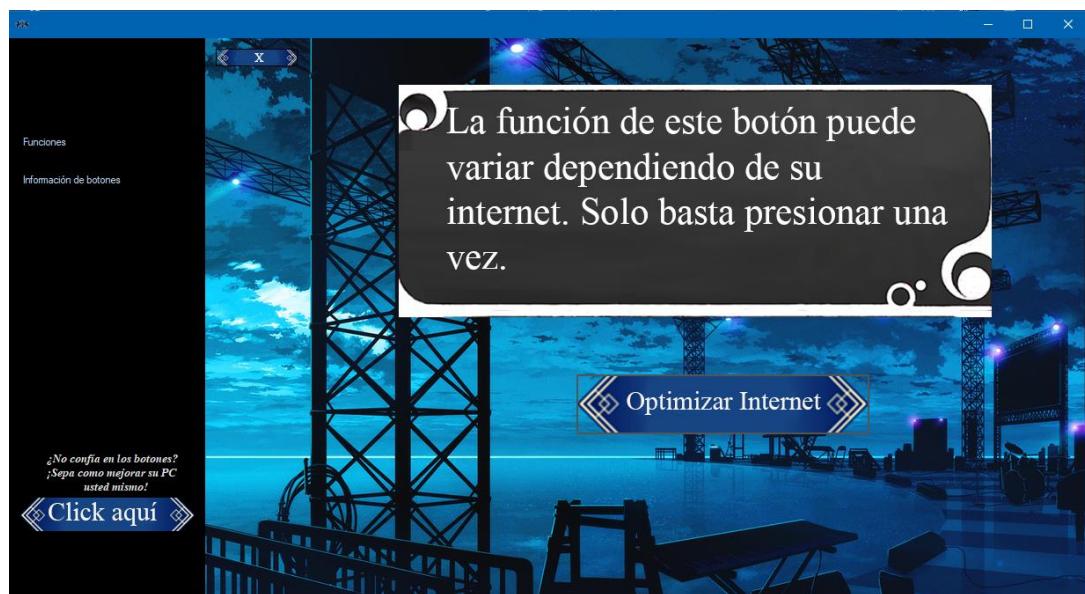
Botón X

Su única función es la de cerrar el contenido del botón y regresar desde el inicio. No es necesario presionar este botón cada vez que se quiera entrar a otro sub botón, solamente SI se desea regresar a la pantalla principal.

Información

En la parte superior del botón, se encuentra información breve o instrucciones que se deben de seguir si se quiere ejecutar el botón.

Al ser un Script ejecutado por .cmd, pero que no necesita ningún tipo de confirmación, este se notará por muy poco tiempo y se cerrará inmediatamente.





Botón Información de botones

Funciones del botón

Método customizeDesing

La función de este método es la de ocultar el panel en donde se guardan los sub botones que incluye este botón.

Método hideSubMenu

La función de este método es la de ocultar el panel que se ha mostrado anteriormente, para ello se realiza la siguiente condición:

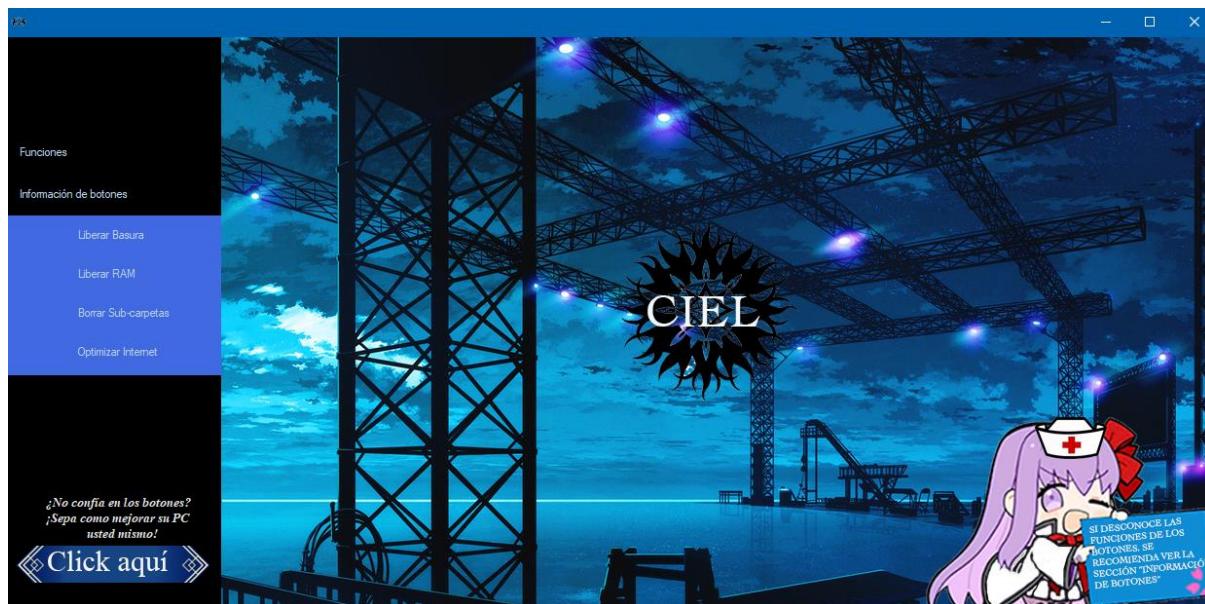
“Si el panel sub menú está visible, lo ocultamos estableciendo la propiedad visible a falso”.

Método showSubMenu

La función de este método es mostrar el panel seleccionado, para esto tiene la siguiente condición:

“Si el panel está oculto, mostramos el panel”, tiene invocado el método de ocultar el panel si es que hay uno abierto, luego mostrarlo. En caso contrario, es decir, si el panel está visible o abierto, se vuelve a ocultar. De esta manera se alternará la visibilidad del panel actual o el panel que se mostró anteriormente.

En consecuencia, de todo esto, al dar click en el botón, este mostrará todos los sub botones que tiene disponible. Estos mismos sub botones se cerrarán si se presionar el otro botón llamado “Funciones”.





Sub botón Liberar Basura

Variable privada Form

La función de esta variable es únicamente abrir solamente un solo formulario y si un formulario está abierto, se cierra. Esto se realizó para evitar que el programa abriera otro proceso y así acumular más pantallas del programa y por consecuencia reducir el rendimiento del programa.

Método OpenChildForm

Su función consiste en la de llamar un formulario al momento de dar click al sub botón. Esto se traduce a que se mostrará la pantalla donde incluya la función o la información del sub botón correspondiente.

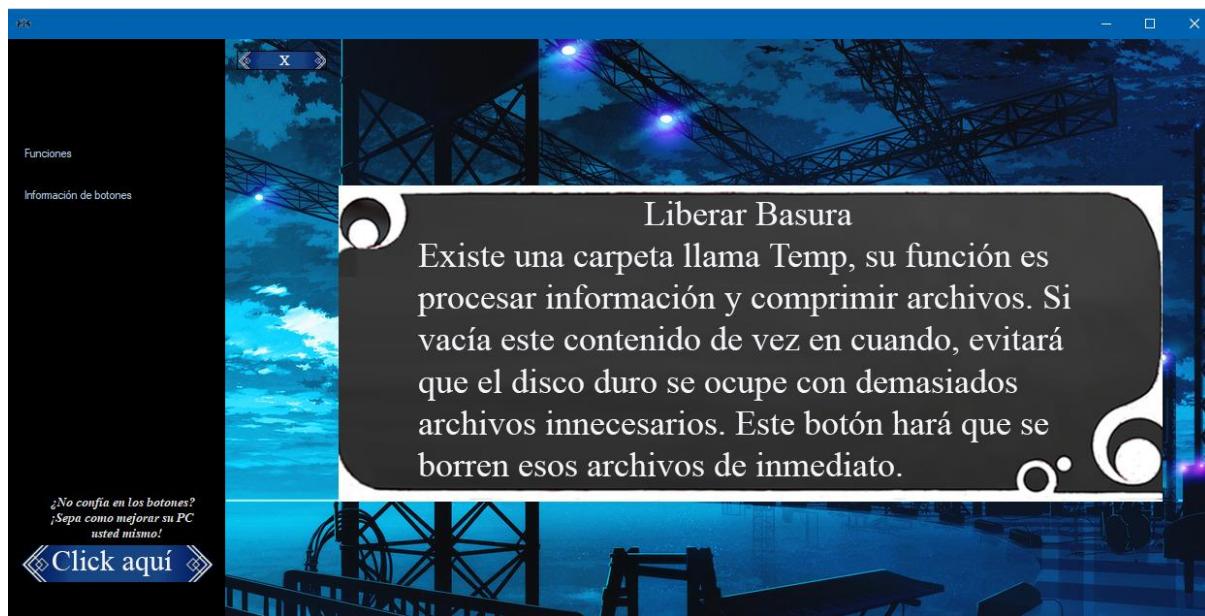
Las funciones anteriores, darán paso a mostrar el contenido del sub botón.

Botón X

Su única función es la de cerrar el contenido del botón y regresar desde el inicio. No es necesario presionar este botón cada vez que se quiera entrar a otro sub botón, solamente SI se desea regresar a la pantalla principal.

Información

A diferencia del sub botón con su mismo nombre, este solamente contiene información acerca de su definición o la función que realiza al momento de ser ejecutado. Se hizo lo más breve y entendible posible, ya que se desea que este botón sea lo más entendible y sea usado por todo aquel que use el programa.





Sub botón Liberar RAM

Variable privada Form

La función de esta variable es únicamente abrir solamente un solo formulario y si un formulario está abierto, se cierra. Esto se realizó para evitar que el programa abriera otro proceso y así acumular más pantallas del programa y por consecuencia reducir el rendimiento del programa.

Método OpenChildForm

Su función consiste en la de llamar un formulario al momento de dar click al sub botón. Esto se traduce a que se mostrará la pantalla donde incluya la función o la información del sub botón correspondiente.

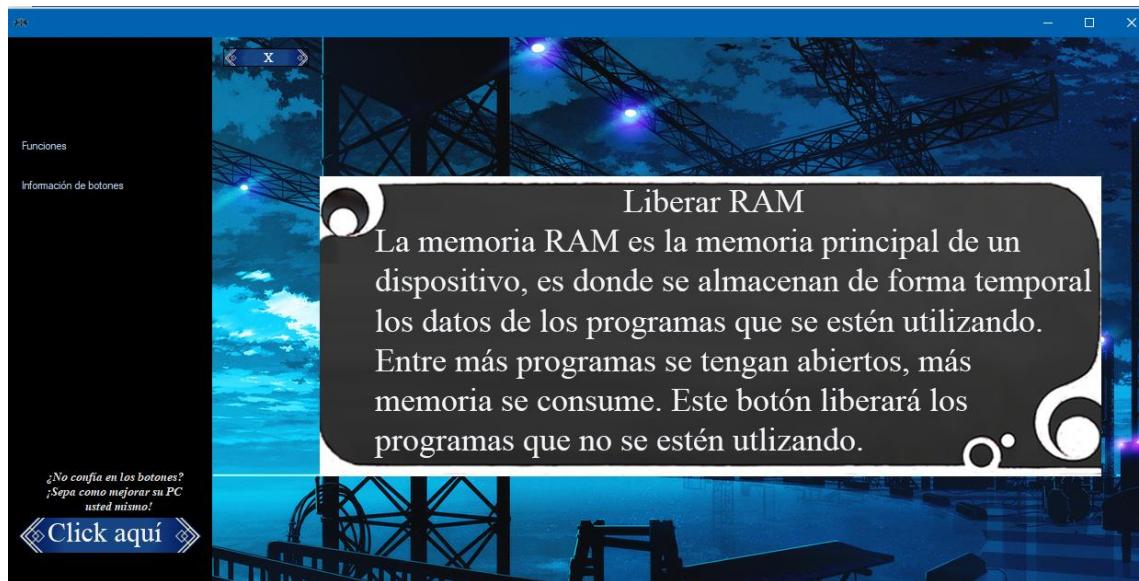
Las funciones anteriores, darán paso a mostrar el contenido del sub botón.

Botón X

Su única función es la de cerrar el contenido del botón y regresar desde el inicio. No es necesario presionar este botón cada vez que se quiera entrar a otro sub botón, solamente SI se desea regresar a la pantalla principal.

Información

A diferencia del sub botón con su mismo nombre, este solamente contiene información acerca de su definición o la función que realiza al momento de ser ejecutado. Se hizo lo más breve y entendible posible, ya que se desea que este botón sea lo más entendible y sea usado por todo aquel que use el programa.





Sub botón Borrar Sub-carpetas

Variable privada Form

La función de esta variable es únicamente abrir solamente un solo formulario y si un formulario está abierto, se cierra. Esto se realizó para evitar que el programa abriera otro proceso y así acumular más pantallas del programa y por consecuencia reducir el rendimiento del programa.

Método OpenChildForm

Su función consiste en la de llamar un formulario al momento de dar click al sub botón. Esto se traduce a que se mostrará la pantalla donde incluya la función o la información del sub botón correspondiente.

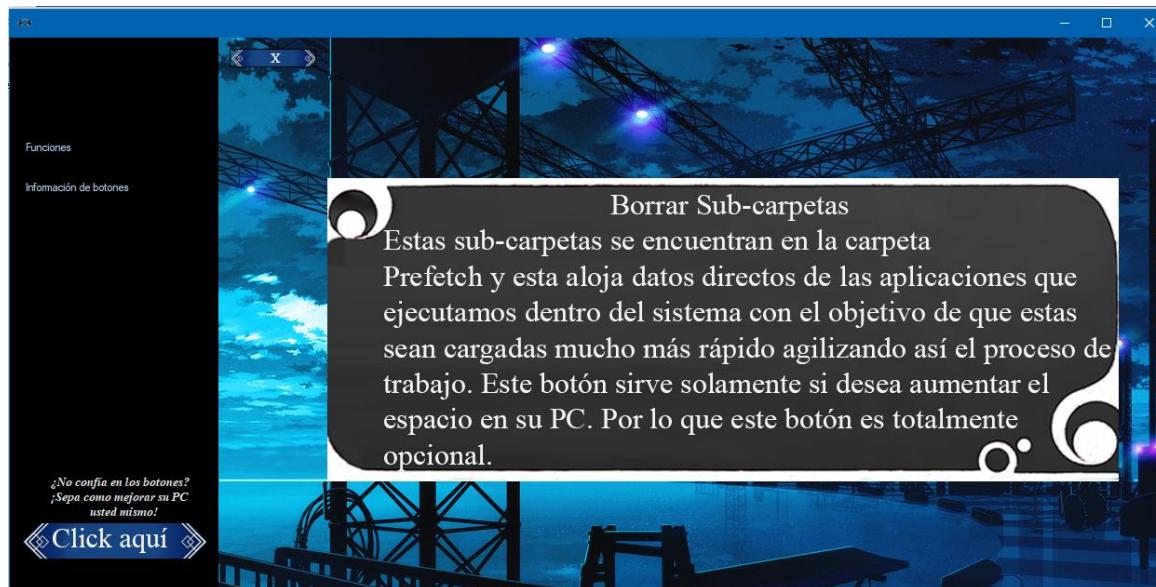
Las funciones anteriores, darán paso a mostrar el contenido del sub botón.

Botón X

Su única función es la de cerrar el contenido del botón y regresar desde el inicio. No es necesario presionar este botón cada vez que se quiera entrar a otro sub botón, solamente SI se desea regresar a la pantalla principal.

Información

A diferencia del sub botón con su mismo nombre, este solamente contiene información acerca de su definición o la función que realiza al momento de ser ejecutado. Se hizo lo más breve y entendible posible, ya que se desea que este botón sea lo más entendible y sea usado por todo aquel que use el programa.





Sub botón Optimizar Internet

Variable privada Form

La función de esta variable es únicamente abrir solamente un solo formulario y si un formulario está abierto, se cierra. Esto se realizó para evitar que el programa abriera otro proceso y así acumular más pantallas del programa y por consecuencia reducir el rendimiento del programa.

Método OpenChildForm

Su función consiste en la de llamar un formulario al momento de dar click al sub botón. Esto se traduce a que se mostrará la pantalla donde incluya la función o la información del sub botón correspondiente.

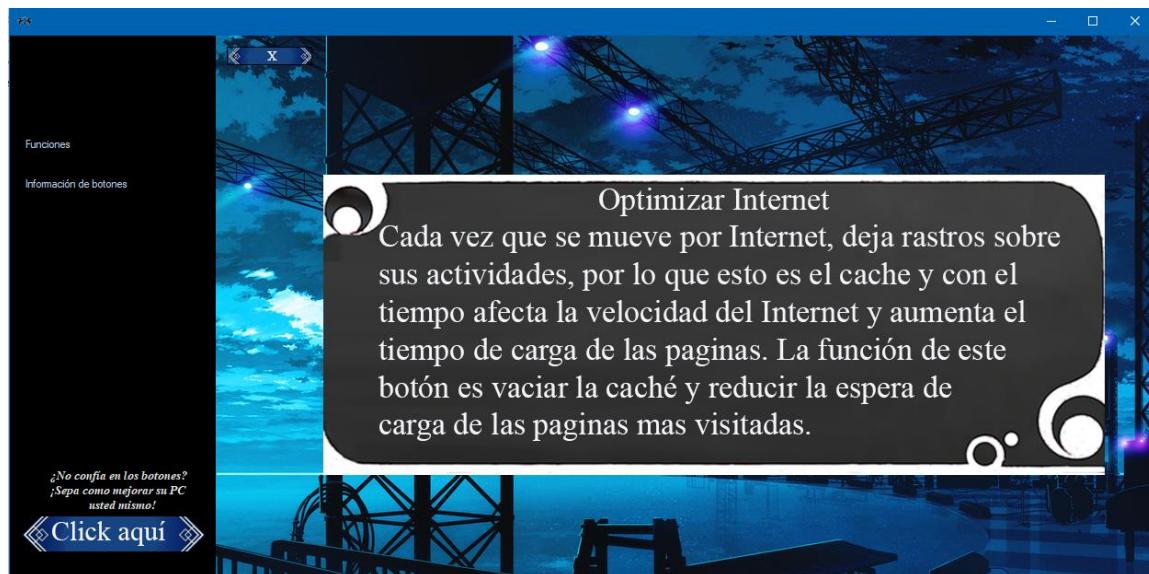
Las funciones anteriores, darán paso a mostrar el contenido del sub botón.

Botón X

Su única función es la de cerrar el contenido del botón y regresar desde el inicio. No es necesario presionar este botón cada vez que se quiera entrar a otro sub botón, solamente SI se desea regresar a la pantalla principal.

Información

A diferencia del sub botón con su mismo nombre, este solamente contiene información acerca de su definición o la función que realiza al momento de ser ejecutado. Se hizo lo más breve y entendible posible, ya que se desea que este botón sea lo más entendible y sea usado por todo aquel que use el programa.





Botón Click aquí

Elementos

Panel negro

Al presionar el botón, se muestra en la parte inferior un gran panel color negro, en este vienen cinco botones que contienen información, estos botones se describen más a detalle a continuación.

Botones consejo

Estos botones contienen información similar a la del primer formulario mostrado, cada botón tiene un consejo distinto y no hay necesidad de presionar la X para abrir otro botón.

Botón X

Su única función es la de cerrar el contenido del botón y regresar desde el inicio. No es necesario presionar este botón cada vez que se quiera entrar a otro sub botón, solamente SI se desea regresar a la pantalla principal.

Información

Se contiene un solo consejo por cada botón, estos consejos son muy sencillos y hechos para que cualquiera pueda realizarlos sin tantos problemas.

Esta sección fue creada por que se planteó la posibilidad de que no todos los usuarios que usen el programa, tengan la confianza total hacia los botones, pero quieren mejorar su PC de alguna manera; para esto fue creada esta sección.

A continuación, se muestran todos los consejos:



Consejo 1

The screenshot shows a Windows desktop with a dark blue background featuring a bridge at night. A tooltip window is open in the center, containing the following text:

Limpie su escritorio

Tener lleno el escritorio de su PC, consume demasiados recursos y es una de las causas principales de que se ralentice al momento de realizar sus trabajos. Guarde todo en carpetas o borrelos si ya no le sirven. Esto hará que inicie su PC con más rapidez.

On the left side of the desktop, there is a sidebar with the following options:

- Funciones
- Información de botones

At the bottom left, there is a link labeled "Click aquí". At the bottom right, there is a navigation bar with five links: "Consejo 1", "Consejo 2", "Consejo 3", "Consejo 4", and "Consejo 5".

Consejo 2

The screenshot shows a Windows desktop with a dark blue background featuring a bridge at night. A tooltip window is open in the center, containing the following text:

Borre documentos innecesarios

Aunque se tengan solamente documentos Word y Excel, estos se acumulan y ocupan bastante espacio en su disco, incluso puede llegar a contagiarse por tomar tal documento en una PC infectada. Borre constantemente.

On the left side of the desktop, there is a sidebar with the following options:

- Funciones
- Información de botones

At the bottom left, there is a link labeled "Click aquí". At the bottom right, there is a navigation bar with five links: "Consejo 1", "Consejo 2", "Consejo 3", "Consejo 4", and "Consejo 5".



Consejo 3

Apague su PC si no lo está usando

Su PC realiza procesos aún estando suspendida, esto llena con mas rapidez la memoria RAM y por consecuencia se ralentiza rápidamente. Al apagarla libera la memoria de inmediato e irá un poco más rápida la próxima vez que la encienda.

«Click aquí»

«Consejo 1» «Consejo 2» «Consejo 3» «Consejo 4» «Consejo 5»

Consejo 4

No abra muchos programas a la vez

El navegador Chrome, consume grandes cantidades de RAM y ralentiza enormemente programas comunes como Word, Excel y sobre calentar la PC por consecuencia. Para evitar esto, no tenga abiertos más de 3 o 4 programas mientras use Chrome.

«Click aquí»

«Consejo 1» «Consejo 2» «Consejo 3» «Consejo 4» «Consejo 5»



Consejo 5

Desinstale programas que ya no use

Para desinstalar profundamente, presione el botón Windows y seleccione “Panel de Control” vea la parte de “Programas” y presione “Desinstalar programas”. Ahí estarán todos los programas instalados. Se recomienda borrar programas que solamente usted instaló.

«Click aquí»

«Consejo 1» «Consejo 2» «Consejo 3» «Consejo 4» «Consejo 5»

Funciones de la pantalla

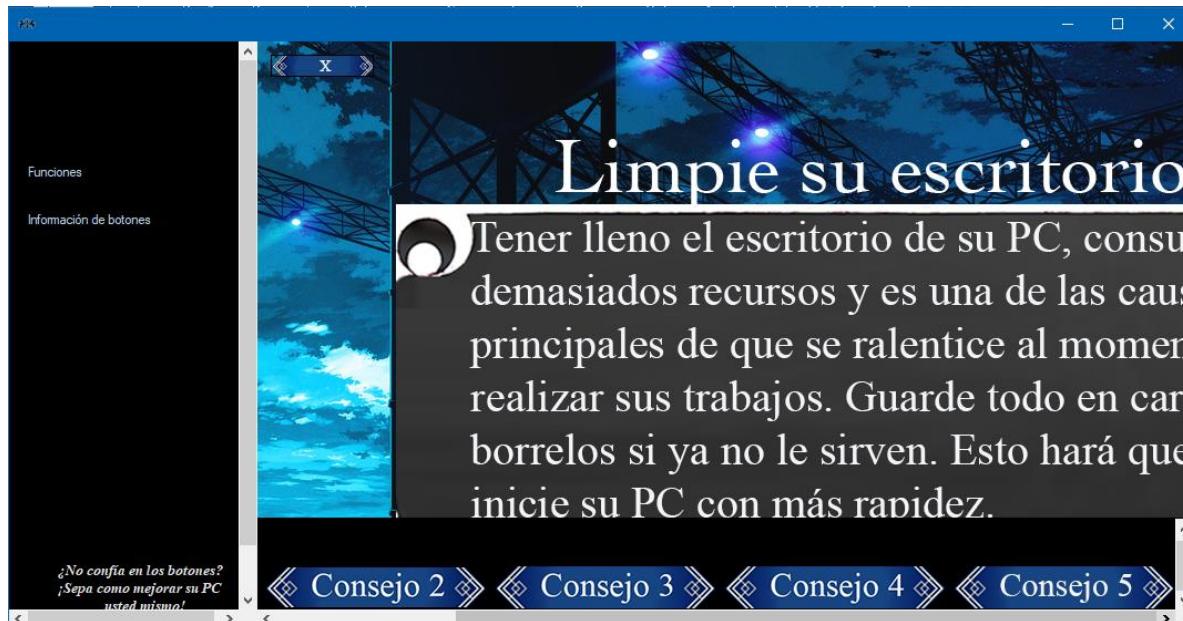
AutoScroll

En el panel negro de la izquierda, gracias a la propiedad AutoScroll cambiada de falso a verdadero, se puede navegar en todos sus componentes en caso de que el formulario o “pantalla” del programa quede chico como en el siguiente caso:

Limpie su escritorio

Tener lleno el escritorio de su PC, consume demasiados recursos y es una de las causas principales de que se ralentice al momento de realizar sus trabajos. Guarde todo en carpetas si ya no le sirven. Esto hará que inicie su PC con más rapidez.

«Consejo 1» «Consejo 2» «Consejo 3» «Consejo 4»



3.9.6 Planificación del siguiente ciclo

Si se producen errores, se buscan soluciones, y si una alternativa es una mejor solución, se prefiere en el siguiente ciclo.

Riesgos

- Si el sistema operativo no se le actualiza este componente, marcará errores de instalación y será imposible que el programa sea instalado de manera correcta.
- Si se cambia a una plataforma de destino menos reciente idéntica a la del sistema de todo el hospital (**.NET Framework 3.0 o 3.5** por ejemplo), el programa no alcanzará toda su potencia y por consecuencia puede generar errores continuos en el programa y ser vulnerable a posibles ataques.

Solución

Actualizar este componente a la versión establecida por el programa (**.NET Framework 4.7.2**). El riesgo queda completamente solucionado con tan solo actualizar esta plataforma en el sistema operativo de la PC a la cual se le desea instalar el programa.



Conclusiones

Los resultados que mostró el proyecto, fueron lo suficiente satisfactorios como para clasificar el proyecto como un éxito; el proyecto logró cumplir con su función, que era la de optimizar el sistema operativo de los equipos del hospital.

- Se logró liberar la memoria RAM instalada gracias al script creado.
- Se logró borrar la cache del sistema que ocupaba suficiente espacio en el disco duro.
- El botón opcional llamado Prefetch, es funcional, por lo que el uso que se le daba a este botón, dependía de cada usuario.
- Se logró borrar el cache del DNS del internet gracias al script, ocasionando que se optimizara el internet utilizado.

Se quiere también abarcar los problemas que no se llegaron a resolver:

- No todo el hardware tenía la misma antigüedad, al igual que el mismo uso que se le daba y características, por lo que la optimización fue muy notable o muy poco en varios sistemas operativos.
- Se debía de actualizar forzosamente la plataforma .NET Framework para que el programa fuera instalado.