

PROGRAMACIÓN ORIENTADA A OBJETOS

2o. PARCIAL

Practica #9

Aplicación de un Cronómetro en C#

Daniel Vázquez de la Rosa
d_vazquez@outlook.com

Instituto Politécnico Nacional



Escuela Superior de Cómputo

Septiembre 2018

Datos Personales

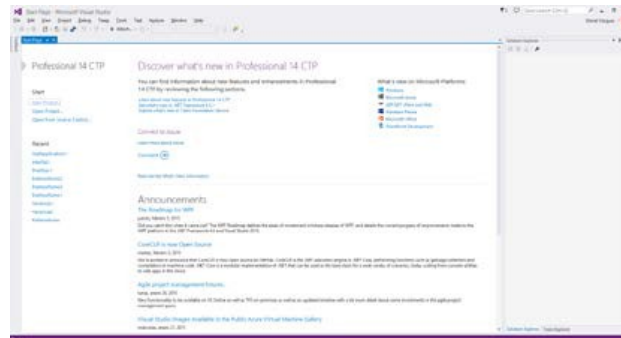
Crear un Documento en Word que incluya una caratula que contenga tus datos. Dicho documento será guardado como: Grupo_ApellidoPaterno_NombreAlumno_Practica9_POO y deberá contener toda la información de la practica creada.

Objetivos de la Práctica

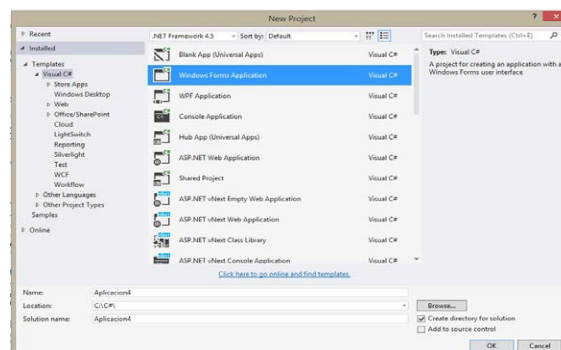
- Comprender los fundamentos de mejorar la interface en C#.
- Entender el funcionamiento de los controles timer.

Desarrollo

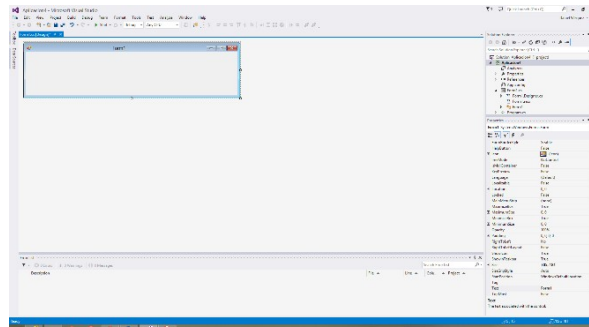
1. Abrir la aplicación de Visual Studio.



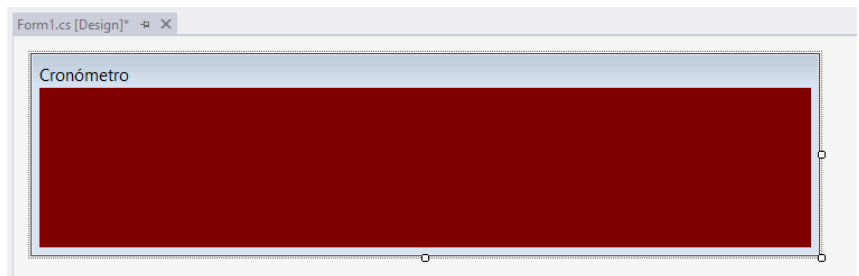
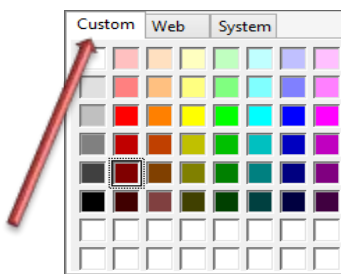
2. Ve al menú File o Archivo, dependiendo del idioma del programa.
3. Crea un Nuevo Proyecto y selecciona la opción de Windows Form Application y Nómbralo como **Cronometro**



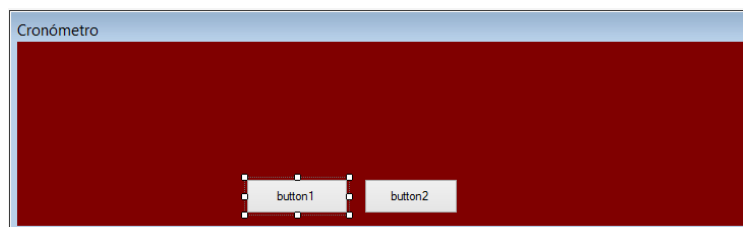
- Se creará el proyecto con una ventana de aplicación y redimensiona tu ventana a un tamaño amplio para poder efectuar el diseño de manera correcta.



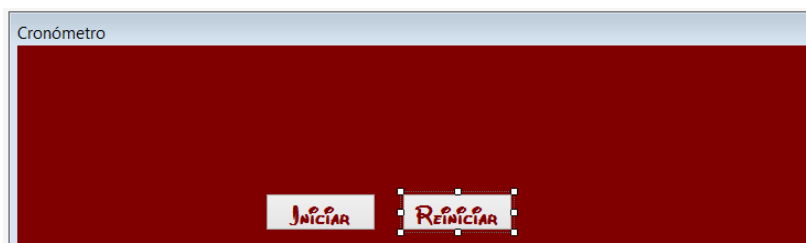
- Dirígete a la ventana de propiedades y localiza la propiedad **Control Box** y cambia su valor que es de **True** a **False**, observa que esto quitara los controles de la pantalla que crearas. Así mismo también localiza la propiedad **Text** y cambia el texto **Form 1** por el texto **Cronómetro**, de igual manera localiza la propiedad **FormBorderStyle** y cambia su valor **sizable** por **FixedToolWindow**.
- Una vez que hemos hecho esos cambios, ahora le daremos un color a la ventana a través de la propiedad **BackColor** en la pestaña **Custom** y selecciona el color de tu agrado.



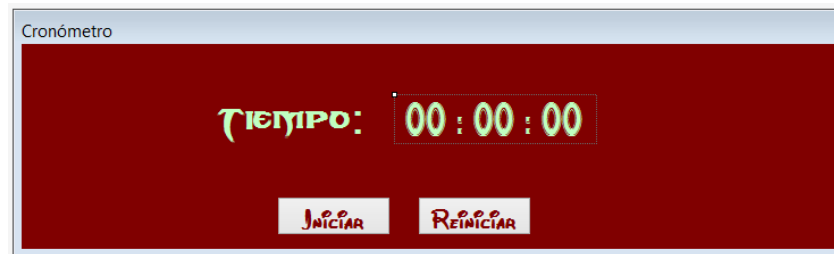
- Posteriormente, iras a la barra de Herramientas (Toolbox) y seleccionaras el control de Botón (Button), dibuja dos botones en tu ventana.



Al seleccionar el botón 1 dirígete a la ventana de propiedades y selecciona la propiedad **name** y cambia su valor a **cmdinicio**, posteriormente seleccionamos la propiedad **Text** y cambiamos su valor por el texto **Iniciar**, así mismo, puedes modificar las características de tu botón con las propiedades **Font** y **ForeColor** a los valores que tu decidas. Haz lo mismo con el segundo Botón, en el cual su propiedad **name** será **cmdreset** y su propiedad **Text** dirá **Reiniciar**.



8. Ahora colocaras dos controles de Etiquetas (**Label**) en la parte central de tu ventana. Al control **Label 1** modifica su propiedad **Text** por **Tiempo:** y a **Label 2** por **00 : 00 : 00**, así mismo, modifica sus propiedades **Font** y **ForeColor** colocando los valores de tu agrado



Ahora colocamos un control **timer**, recuerda que este control solo deberás soltarlo dentro de la ventana y en la parte de abajo se creara el control **timer1**

9. Al tener nuestra ventana y los controles ya diseñados, damos doble click sobre el control **Iniciar**. Esto nos llevara a la ventana de código donde escribiremos las acciones que deseamos que realice el botón **Iniciar**.

Para ello, lo primero que haremos, será crear las variables para llevar en cuenta el tiempo, crearemos variables generales.


```
namespace Aplicacion4
{
    public partial class Form1 : Form
    {
        int segundos = 0; // Variable que llevara la cuenta de los segundos
        string mostrarseg = "00"; // Variable que llevara la cuenta de los segundos
        int minutos = 0; // Variable que llevara la cuenta de los minutos
        string mostrarmin = "00"; // Variable que llevara la cuenta de los segundos
        int horas = 0; // Variable que llevara la cuenta de las horas
        string mostrarhora = "00"; // Variable que llevara la cuenta de los segundos

        public Form1()
        {
            InitializeComponent();
        }

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

Una vez creado las variables, nos colocamos en el método `private void cmdinicio_Click(object sender, EventArgs e)` que se creó al dar doble click sobre el botón Iniciar.

Allí, escribiremos el siguiente código que servirá para la pulsación del botón Iniciar, esto permitirá que el texto del botón se modifique, al igual que su acción.



```
private void cmdinicio_Click(object sender, EventArgs e)
{
    if (cmdinicio.Text == "Iniciar") //Verifica si el botón iniciar ha sido pulsado
    {
        timer1.Start(); //Si la condición se cumple
        cmdinicio.Text = "Pausa"; //El timer1 se activa y comienza la cuenta de tiempo
        //Una vez oprimido el boton Iniciar, el texto cambia a Pausa
    }
    else //Si la condición no se cumple, el botón iniciar no ha sido pulsado
    {
        timer1.Stop(); //El timer1 esta desactivado por lo tanto no hay cuenta de tiempo
        cmdinicio.Text = "Iniciar"; //El texto del botón 1 será Iniciar
    }
}
```

10. Una vez colocado el código para el botón **Iniciar**, volvemos a la pantalla de diseño y damos doble click sobre el botón **Reiniciar**, y escribiremos el código sobre su método correspondiente

```
private void cmdreset_Click(object sender, EventArgs e)
{
    timer1.Stop(); //Al dar click sobre el boton 2 la cuenta del tiempo se detiene
    cmdinicio.Text = "Iniciar"; //Cambia el texto del botón 1 a Iniciar
    label2.Text = "00 : 00 : 00"; // Se reinicializa el cronometro a cero en la Label 2
    segundos = 0; // la variable segundos se resetea a su valor cero
    minutos = 0; // la variable minutos se resetea a su valor cero
    horas = 0; // la variable horas se resetea a su valor cero
}
```

11. Ahora que hemos escrito el código de los botones, nos posicionamos en la ventana de diseño y damos doble click sobre el **timer1** puesto que ahora programaremos los eventos que ocurrirán al iniciar la cuenta de tiempo o al detener dicha cuenta.

```
private void timer1_Tick(object sender, EventArgs e)
{
    if (segundos == 59) //Verifica si la variable segundos = 59
    {
        //Si la condición se cumple
        segundos = 0; //La variable Segundos = 0

        if (minutos == 59) //Verifica si la variable minutos = 59
        {
            minutos = 0; //La variable minutos = 0
            horas = horas + 1; //Los minutos al volver al valor de cero, las horas se incrementan
        }
        else //Si la condición no se cumple
        {
            minutos = minutos + 1; //Los segundos vuelven a cero y los minutos se incrementan
        }
    }
    else
    {
        segundos = segundos + 1; //La variable segundos se incrementan en 1
    }
}
```

Ahora hacemos uso de las variables declaradas al inicio, el manejo de estas variables determinara el funcionamiento inicial de nuestro cronómetro.

12. Siguiendo con el código, creamos las condiciones para comenzar a realizar los cambios en la etiqueta 2 (**Label12**) a través del siguiente código, que continuará después de la última línea anterior.

```
if (segundos == 0 || segundos == 1 || segundos == 2 || segundos == 3 || segundos == 4 || segundos == 5 || segundos == 6 || segundos == 7 || segundos == 8 || segundos == 9)
{
    mostrarseg = "0" + segundos;
}
else
{
    mostrarseg = Convert.ToString(segundos);
}

if (minutos == 0 || minutos == 1 || minutos == 2 || minutos == 3 || minutos == 4 || minutos == 5 || minutos == 6 || minutos == 7 || minutos == 8 || minutos == 9)
{
    mostrarmin = "0" + minutos;
}
else
{
    mostrarmin = Convert.ToString(minutos);
}

if (horas == 0 || horas == 1 || horas == 2 || horas == 3 || horas == 4 || horas == 5 || horas == 6 || horas == 7 || horas == 8 || horas == 9)
{
    mostrarhora = "0" + horas;
}
else
{
    mostrarhora = Convert.ToString(horas);
}

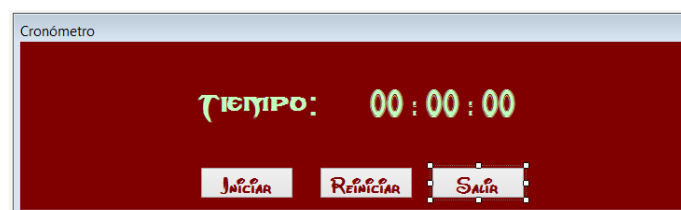
label2.Text = mostrarhora + " : " + mostrarmin + " : " + mostrarseg;
```

Ahora que ya tenemos el código escrito, es hora de probar la aplicación, así que da click en el botón de iniciar (**Start**).



Observaras que el cronómetro cuenta de manera correcta, recuerda haber configurado el **timer1** con el **valor 1000** en su propiedad **Interval**. Puedes modificar este valor hasta 1, para que observes el diferente funcionamiento del cronómetro.

13. Por último, crea un botón que nos permita salir de la aplicación



Recuerda modificar las propiedades del 3er. Botón, **Name = cmdstop**, Text= **Salir**. Y

una vez creado el botón da doble click sobre el para darle la acción de salir.

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

Ejecuta el código y haz diferentes pruebas, cambiando el intervalo del **timer1**

Responde:

- 1) ¿Cuál es la diferencia de este programa con relación a los anteriores?
- 2) ¿Qué tan complicado ha sido seguir la cuenta de tiempo?
- 3) Diseña un cronómetro que contenga: decimas de segundos, segundos, minutos y Horas y tenga los botones de Iniciar, Reset y Pausa.
- 4) Diseña un cronómetro que contenga: decimas de segundos, segundos, minutos, horas, días, semana, el timer deberá estar ajustado a un valor de intervalo = 1segundo. Este diseño deberá contar con un elemento visual de alarma el cual, al ser puesto por un tiempo determinado por el usuario, se active y muestre a través de un control, una imagen o un color o un texto, deberá de contar con los controles adecuados que el programador desee incluir (start, reset, stop, set, etc.).

Una vez creado el programa, copia el código en el documento de Word que creaste con la caratula y tus datos, el documento deberá de contener los códigos realizados, así, como las pantallas de captura de tu programa ejecutándose.

Al final anota un breve resumen de la creación de en aplicaciones visuales y posteriormente después del resumen anota tus conclusiones.

Esta Práctica deberá de entregarse a más tardar el día jueves 11/10/2018 antes de las 10:00 PM, toda práctica que no sea enviada dentro de este horario será evaluada con cero.