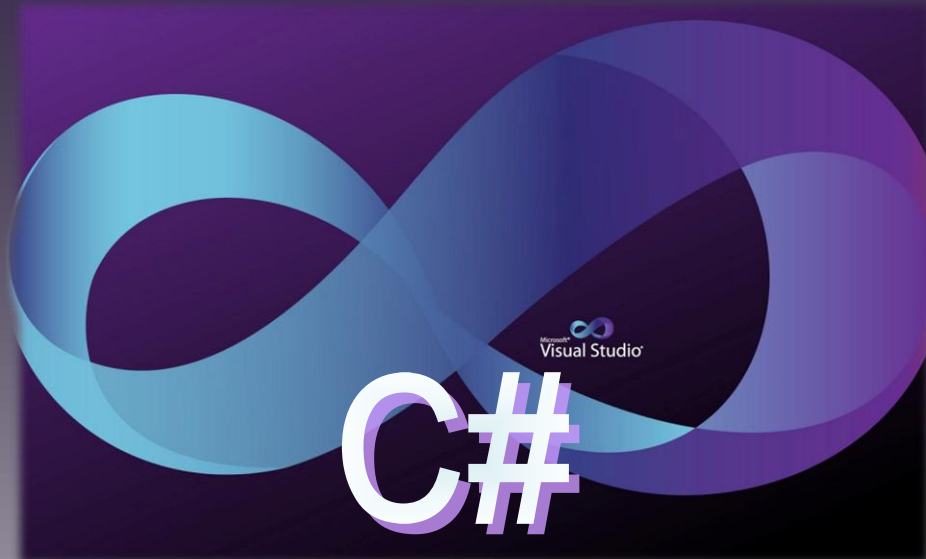


**ISTITUTO TECNICO INDUSTRIALE
G. M. ANGIOY
SASSARI**



CORSO DI PROGRAMMAZIONE

RIFINITURE DEL PROGETTO

DISPENSA 04.06

[04-06_Rifiniture_Progetto_\[ver_11\]_DARIVEDERE](#)



Questa dispensa è rilasciata sotto la licenza Creative Common CC BY-NC-SA. Chiunque può copiare, distribuire, modificare, creare opere derivate dall'originale, ma non a scopi commerciali, a condizione che venga riconosciuta la paternità dell'opera all'autore e che alla nuova opera vengano attribuite le stesse licenze dell'originale.

Versione del: **07/11/2015**

Revisione numero: **15**

Prof. Andrea Zoccheddu
Dipartimento di Informatica

**DIPARTIMENTO
INFORMATICA E TELECOMUNICAZIONI**





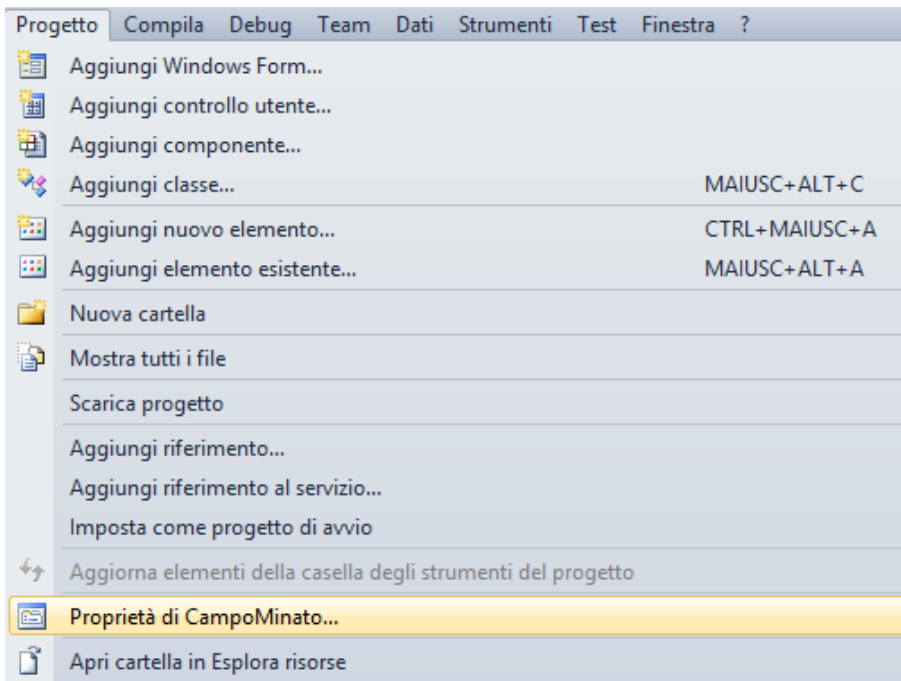
RIFINITURE DELL'APPLICAZIONE

RIFINITURE DELL'APPLICAZIONE

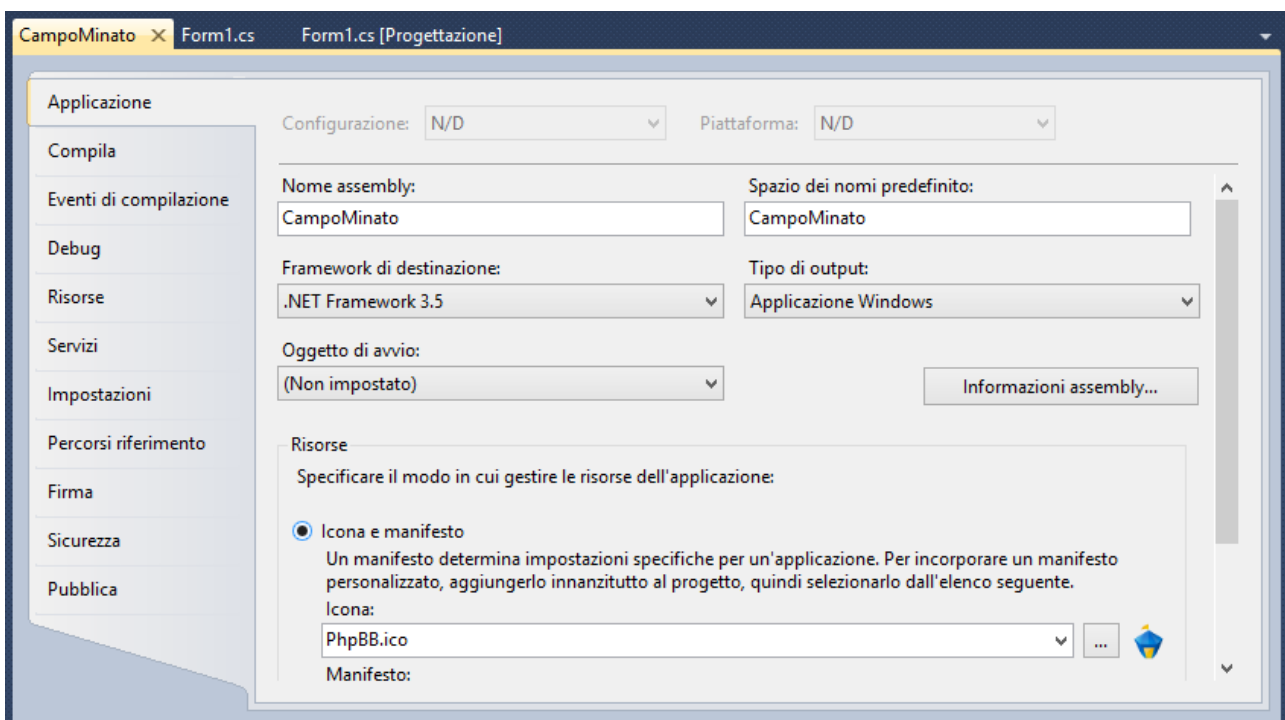
PAGINA PROGETTAZIONE PROGETTI

PROGETTO GUIDATO

- Si apra un Progetto di quelli già realizzati in precedenza; in questo esempio farò riferimento all'applicazione Campo Minato
- Dopo aver aperto il progetto si scelga dal menu Progetto, la voce Proprietà di [progetto]:

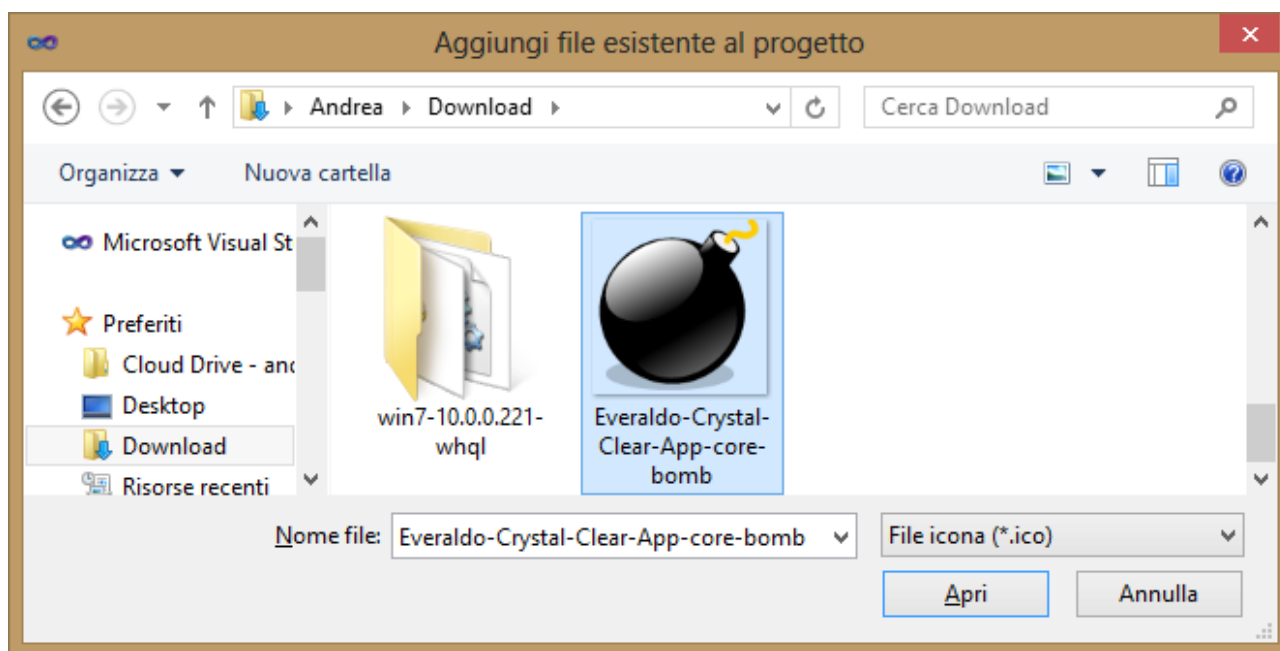


- Appare la finestra seguente che mostra le caratteristiche dell'applicazione:

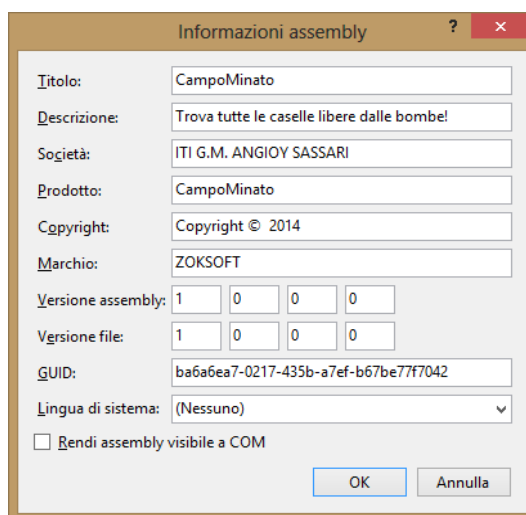




- Nella finestra individuiamo la dicitura «Icona e manifesto» e la selezioniamo
- Possiamo a questo punto modificare l'icona dell'applicazione, facendo clic sul pulsante (ellissi) coi tre puntini evidenziato dal numero 2 della immagine precedente
- Nella finestra di dialogo possiamo cercare l'icona che intendiamo scegliere per la nostra applicazione; se ne cerchi una ti suggerisco di provare qui:
- <http://www.iconarchive.com/commercialfree.html>
- Fai attenzione alla licenza dell'immagine che prelevi e ricorda che, anche quando non è obbligatorio, è sempre eticamente corretto citare la fonte e l'artista di riferimento dell'immagine; prendi quindi nota del link di download e del nominativo dell'autore!
- Per esempio scegliamo: <http://www.iconarchive.com/download/i47780/everaldo/crystal-clear/App-core-bomb.ico> e scarichiamo l'icona in una cartella nota da cui la possiamo scegliere come icona della nostra applicazione:



- Conclusa questa azione ci dedichiamo al pulsante Informazioni Assembly, il numero 3 della nostra finestra iniziale
- Appare quindi una finestra di dialogo simile a quella mostrata qui di lato





In questa finestra è possibile impostare le seguenti informazioni:

Titolo	il titolo della applicazione
Descrizione	utilizzabile anche per i tooltip sulla barra di applicazione
Società	in ambito scolastico impostate sempre il nome dell'istituto, anche per legittimare la licenza del RAD utilizzato
Prodotto	sintesi dell'applicazione realizzata
Copyright	indicare il tipo di licenza con cui rilasciare il software; personalmente proporrei di rilasciarle come Open Source
Marchio	può essere un ombrello di software, come una dinastia di videogiochi realizzata da una software house
Versione assembly e versione file	per catalogare le varie release del software
GUID	identificatore di interfaccia grafica
Lingua di sistema	no comment

Le seguenti figure mostrano degli esempi di informazione di applicazione:

The image shows two side-by-side screenshots of the 'Informazioni assembly' dialog box. The left window has a gold border and contains the following data: Titolo: CampoMinato; Descrizione: Trova tutte le caselle libere dalle bombe!; Società: ITI G.M. ANGIOY SASSARI; Prodotto: CampoMinato; Copyright: Copyright © 2014; Marchio: ZOKSOFT; Versione assembly: 1.0.0.0; Versione file: 1.0.0.0; GUID: ba6a6ea7-0217-435b-a7ef-b67be77f7042; Lingua di sistema: (Nessuno); checkbox 'Rendi assembly visibile a COM' is unchecked. The right window has a blue border and contains: Titolo: Nome Applicazione; Descrizione: Questa è un'applicazione Realizzata con Visual Studio; Società: Marcello Cherchi; Prodotto: Applicazione; Copyright: Copyright © Marcello Cherchi 2011; Marchio: (empty); Versione assembly: 1.0.0.0; Versione file: 1.0.0.0; GUID: a92f76fd-e544-49de-a0c7-7a7359803153; Lingua di sistema: Italiano (Italia); checkbox 'Rendi assembly visibile a COM' is unchecked.

- Questo insieme di informazioni sarà utilizzata in seguito per costruire la finestra Informazioni (nota anche come About)
- Lo studente deve quindi modificare la casella di testo **Copyright** mettendo il proprio nome e mantenere inalterato il testo delle altre informazioni che compaiono. Inserire il proprio nome e le informazioni sull'organizzazione per cui si lavora, se non sono già state inserite. Fare clic su Ok e chiudere la finestra di dialogo informazioni assembly.
- Dopo aver immesso tutti i dati dell'assembly nel riquadro Applicazione della Finestra Progettazione Progetti, è necessario immetterli nel form della **schermata iniziale** e nelle etichette corrette per ottenere informazioni.
- L'idea è quella di rendere generica la schermata iniziale, in modo da poterla riutilizzare in un altro progetto. Vediamo di cosa si tratta.



SPLASH SCREEN

COSA SONO GLI SPLASH SCREEN

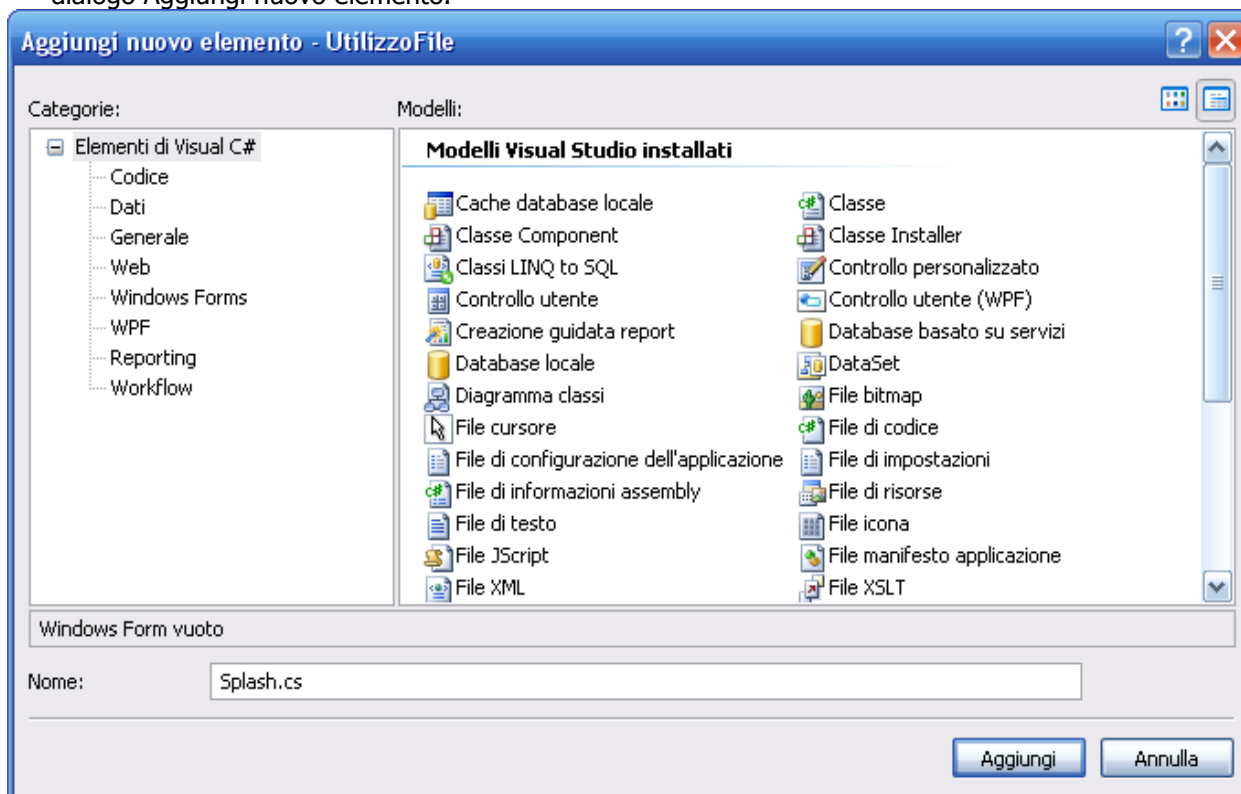
Quando si carica in memoria un'applicazione, viene spesso visualizzata una schermata detta Splash Screen (Schermata Iniziale). Alcuni esempi di schermate iniziali sono le finestre di informazioni iniziali che vengono visualizzate per Microsoft Office e Visual Studio 2008.

Gli Splash Screen non hanno solo lo scopo di mostrare immagini per non annoiare l'utente, hanno anche una funzione specifica. L'avvio di una applicazione comporta una grande quantità di elaborazione. Durante questa elaborazione si può utilizzare la schermata iniziale per visualizzare le informazioni in base al quale l'utente verifica il funzionamento dell'applicazione.

Nella realtà, uno Splash Screen è un form di Windows che non consente l'input da parte dell'utente. Di norma ha un aspetto gradevole ricco di grafica e mostra il nome e la versione dell'applicazione. Uno dei primi interventi che si compiono sull'applicazione è proprio l'aggiunta dello Splash Screen.

PROGETTO GUIDATO - CREAZIONE DI UNA SCHERMATA INIZIALE

- Assegnare al form dell'applicazione un nome significativo. Ricordarsi che tutti i nomi nell'applicazione devono essere significativi per garantire una maggiore leggibilità.
- Aggiungere un form al progetto, nel menu Progetto, selezionare Aggiungi Windows Form. Verrà visualizzata la finestra di dialogo Aggiungi nuovo elemento, che richiede di specificare il tipo di form da aggiungere.
- Selezionare Windows Form e rinominare il file Splash.cs. Osservare la figura che mostra la finestra di dialogo Aggiungi nuovo elemento.



- Una schermata iniziale presenta delle particolari proprietà, perché non è progettata per l'interazione con l'utente. Essa ha una funzione puramente informativa e non dovrebbe essere visualizzata nella barra delle attività, né essere ridimensionabile e altro ancora.



- Impostare le proprietà del form secondo le seguenti indicazioni:

Proprietà	Valore
FormBorderStyle	None
ShowInTaskbar	False
StartPosition	CenterScreen
Size: Width	450
Size: Height	375
BackgroundImage	A scelta
BackgroundImageLayout	Stretch

- Aggiungere tre controlli Label al form e denominarli lblTitoloApplicazione, lblVersione, lblCopyright. Impostare le proprietà delle etichette secondo le seguenti indicazioni.

Nome controllo	Proprietà	Valore
lblTitoloApplicazione	Text	Titolo Applicazione
lblTitoloApplicazione	BackColor	Web: Trasparent
lblTitoloApplicazione	Font	Microsoft Sans Serif, 20pt
lblTitoloApplicazione	ForeColor	Web
lblVersione	Text	Versione
lblVersione	BackColor	Web: Trasparent
lblVersione	Font	Microsoft Sans Serif, 20pt
lblCopyright	Text	Copyright
lblCopyright	BackColor	Web: Trasparent
lblCopyright	Font	Microsoft Sans Serif, 20pt
lblCopyright	ForeColor	Web

- A questo punto è necessario associare la schermata iniziale all'applicazione; per questo motivo dobbiamo modificare il costruttore del form iniziale della nostra applicazione. Aprire il codice del nostro form iniziale e modificare il costruttore in modo che sia come il seguente:

```
public partial class form_iniziale : form
{
    Splash splashScreen = new Splash();
    Public form_iniziale()
    {
        InitializeComponent();
        splashScreen.Show();
        Application.DoEvents();
    }
}
```



- Il punto precedente visualizza lo SplashScreen, ma ovviamente è necessario chiuderlo nel momento che il form iniziale dell'applicazione è pronto per essere visualizzato. L'ultimo evento prima di visualizzare il form è l'evento Activated. E' necessario selezionare il form_iniziale nella visualizzazione Progettazione. Quindi nella finestra proprietà, fare clic sul pulsante Eventi (l'icona con il fulmine giallo) all'inizio della finestra e fare doppio clic sull'evento Activated. Aggiungere il codice seguente:

```
private void form_iniziale_Activated(object sender, EventArgs e)
{
    Thread.Sleep(3000);
    splashScreen.Close();
}
```

- Posizionare il cursore nel testo Thread. Dovrebbe essere visualizzato uno smart tag giallo e rosso. Posizionare il cursore sullo smart tag, fare clic sulla freccia giù, quindi selezionare using System.Threading per aggiungere questa direttiva all'elenco delle direttive using.
- Provare l'applicazione ed al limite intervenire sul tempo di visualizzazione dello SplashScreen.

LA PROPRIETÀ ICON

PROGETTO GUIDATO

 LINK

Scarica immagini da qui:

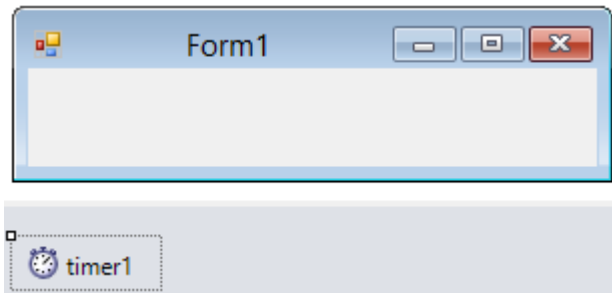
<http://www.iconarchive.com/download/s1933/fasticon/cat/cat-icons.zip>

 LINK

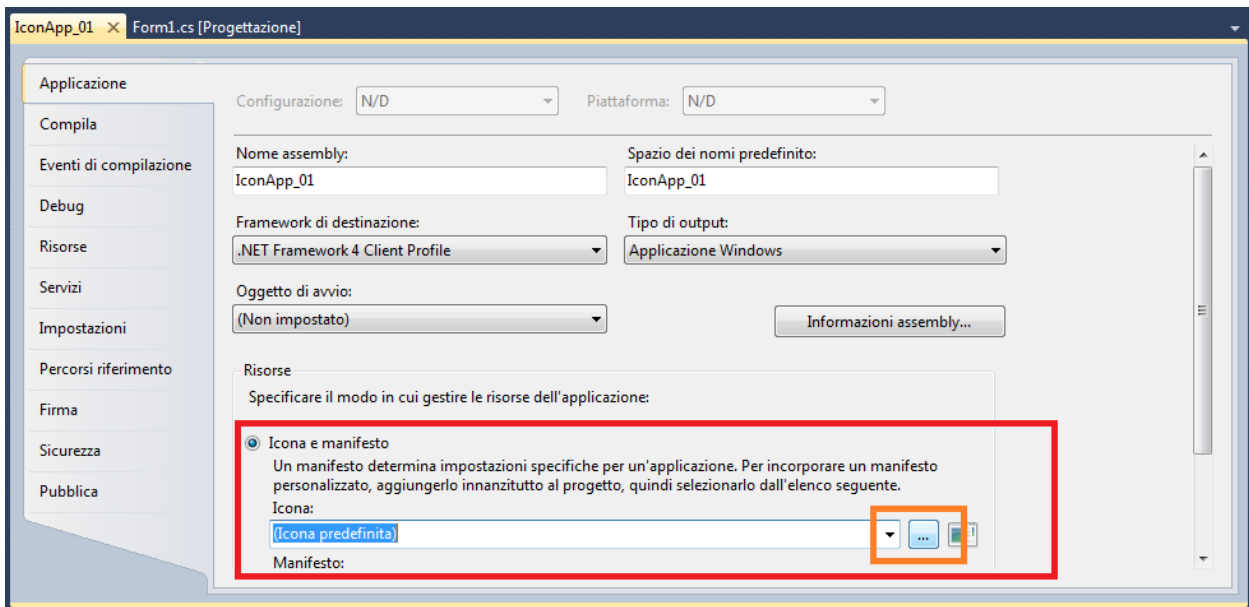
Fonte documentazione:

[http://msdn.microsoft.com/it-it/library/vstudio/339stzf7\(v=vs.100\).aspx](http://msdn.microsoft.com/it-it/library/vstudio/339stzf7(v=vs.100).aspx)

- Creare una nuova applicazione visuale e denominarla IconApp_01
- Costruire un form1 simile alla figura qui illustrata (con un timer1)
- Visualizzare Esplora soluzioni (finestra posta solitamente sulla destra del RAD)



- Visualizzare le Proprietà dell'applicazione (come nella figura illustrata qui sotto)





- Dalla finestra delle proprietà individuare la voce Icona e manifesto e cliccare sul pulsante a destra (ellissi):
- Scegliere dalla finestra di dialogo un'immagine icona (estensione ico).
- Chiudere questa finestra e tornare al progetto; selezionare il button1 e scegliere la sua proprietà Image
- Nelle variabili globali dichiarare le seguenti:

```
Icon[] icons = new Icon[4];  
int index = 0;
```

- Associare al Form_Load il codice seguente:

```
private void Form1_Load(object sender, EventArgs e)  
{  
    string path = @"C:\Users\Andrea\Pictures\cat-icons\cat-icons\icons\";  
    icons[0] = new Icon(path + "Cat-Black.ico");  
    icons[1] = new Icon(path + "Cat-Orange.ico");  
    icons[2] = new Icon(path + "Cat-Brown.ico");  
    icons[3] = new Icon(path + "Cat-Pink.ico");  
    this.Icon = icons[0];  
    timer1.Interval = 500;  
    timer1.Enabled = true;  
}
```

- Associare al tick del timer1 il codice seguente:

```
private void timer1_Tick(object sender, EventArgs e)  
{  
    this.Icon = icons[index];  
    index = (index + 1) % 4;  
}
```

- Lanciare il programma: osservate l'icona sia sulla finestra che sulla barra delle applicazioni di Windows.



POSIZIONE INIZIALE DELLA FINESTRA

PROGETTO GUIDATO

 LINK

Fonte: [http://msdn.microsoft.com/it-it/library/vstudio/339stzf7\(v=vs.100\).aspx](http://msdn.microsoft.com/it-it/library/vstudio/339stzf7(v=vs.100).aspx)

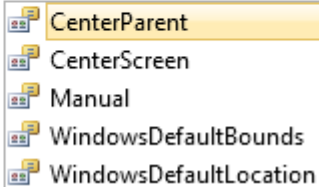
- Riconsiderare il progetto precedente e aprire il codice del programma:
- Cercare il metodo Form1 come appare qui sotto:

```
public Form1 ()
{
    InitializeComponent();
}
```

- E modificarlo come segue:

```
public Form1 ()
{
    InitializeComponent();
    StartPosition = FormStartPosition.CenterScreen;
}
```

```
public Form1()
{
    InitializeComponent();
    StartPosition = FormStartPosition.;
}
```



- Provare l'applicazione e verificare dove appare la finestra al suo avvio (al centro dello schermo).



BORDI DELLA FINESTRA

PROGETTO GUIDATO

LINK

Fonte:

- Riconsiderare il progetto precedente e aprire il codice del programma:

- Cercare il metodo Form1 come appare qui sotto:

```
public Form1()
{
    InitializeComponent()
}
```

- E modificarlo come segue:

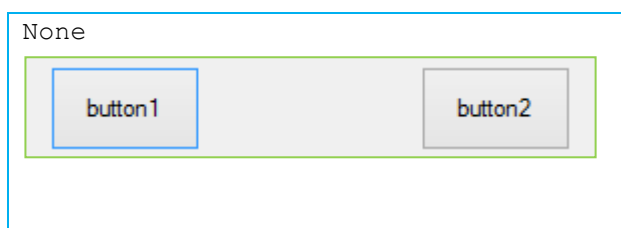
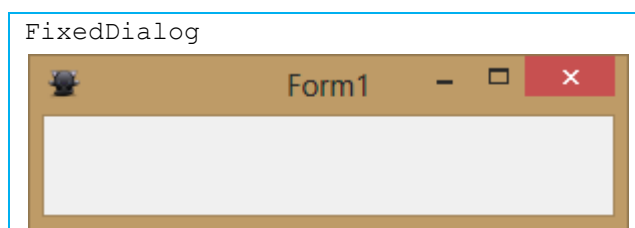
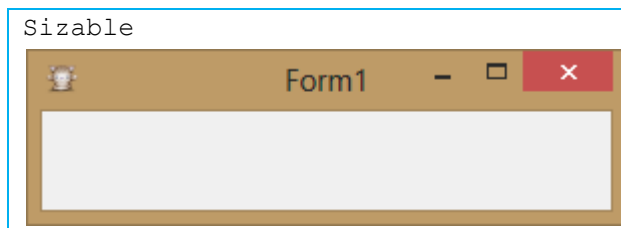
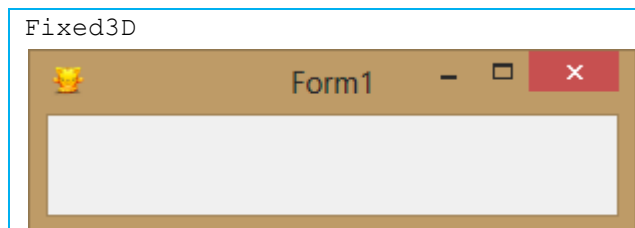
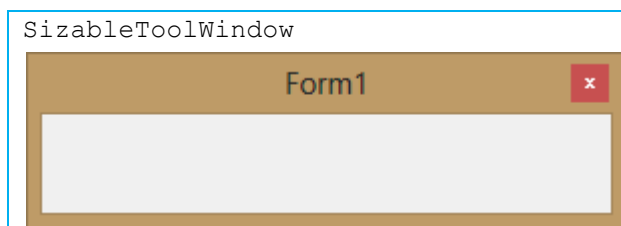
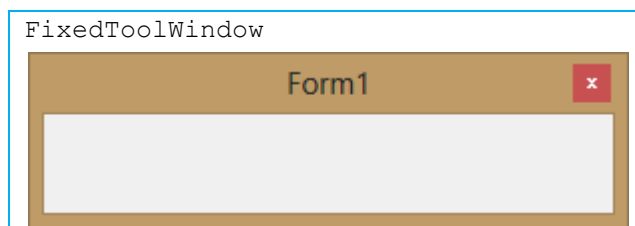
```
public Form1()
{
    InitializeComponent()
    FormBorderStyle = FormBorderStyle.FixedToolWindow;
}
```

```
public Form1()
{
    InitializeComponent();
    StartPosition = FormStartPosition.CenterScreen;
    FormBorderStyle = FormBorderStyle.;
}
```

- FixedDialog
- FixedSingle
- FixedToolWindow
- GetHashCode
- GetType
- GetTypeCode
- None
- Sizable
- SizableToolWindow

- Provare l'applicazione e verificare come appare la finestra al suo avvio (i bordi sono scomparsi e sembrano stretti e non modificabili e i pulsanti della finestra sono diversi).

- Provare anche altre varianti come le seguenti:



- In generale le opzioni che iniziano con **Fixed** non consentono all'utente di modificare le dimensioni della finestra durante la sua attività (come per i dialoghi).

- Viceversa le opzioni che iniziano con **Sizeable** consentono all'utente di modificare le dimensioni della finestra durante la sua attività (come per le ordinarie applicazioni).



LANCIARE APPLICAZIONI DAL FORM1

PROGETTO GUIDATO

- Si crei un nuovo progetto visuale con un Form1 simile alla figura
- Impostare le seguenti proprietà del Form1:

Proprietà	Valore
BackColor	Blue
FormBorderStyle	None
Location X	100
MaximizeBox	False
MinimizeBox	False
Opacity	50
ShowInTaskbar	False
Size Width	294
Size Hight	62
StartPosition	Manual
Text	LaunchTask
WindowState	Normal

- Associare al gestore di evento **Form_Load** il codice seguente:

VC#

```
private void Form1_Load(object sender, EventArgs e)
{
    int dim = Screen.PrimaryScreen.Bounds.Width;
    dim = dim - this.Width;
    this.Left = dim;
    this.Top = 0;
}
```

- Inserire 6 pulsanti di dimensione 40x40 (con immagini 32x32) e predisporre i seguenti gestori di evento:





➤ Associare al pulsante **button1** il codice seguente:

VC#

```
private void button1_Click(object sender, EventArgs e)
{
    try
    {
        System.Diagnostics.Process proc = new
                                                System.Diagnostics.Process();

        proc.EnableRaisingEvents = false;
        proc.StartInfo.FileName = @"explorer";
        proc.StartInfo.Arguments = @"C:\Users\Public\Documents";
        proc.Start();
    }

    catch (Exception)
    {
        MessageBox.Show("Comando fallito");
    }
}
```

➤ Associare al pulsante **button2** il codice seguente:

VC#

```
private void button2_Click(object sender, EventArgs e)
{
    try
    {
        System.Diagnostics.Process proc = new
                                                System.Diagnostics.Process();

        proc.EnableRaisingEvents = false;
        proc.StartInfo.FileName = @"calc";
        proc.Start();
    }

    catch (Exception)
    {
        MessageBox.Show("Comando fallito");
    }
}
```



➤ Associare al pulsante **button3** il codice seguente:

VC#

```
private void button3_Click(object sender, EventArgs e)
{
    try
    {
        System.Diagnostics.Process proc = new
                                                System.Diagnostics.Process ();
        proc.EnableRaisingEvents = false;
        proc.StartInfo.FileName = @"notepad";
        proc.Start();
    }
    catch (Exception)
    {
        MessageBox.Show("Comando fallito");
    }
}
```

➤ Associare al pulsante **button4** il codice seguente:

```
private void button4_Click(object sender, EventArgs e)
{
    try
    {
        System.Diagnostics.Process proc = new
                                                System.Diagnostics.Process ();
        proc.EnableRaisingEvents = false;
        proc.StartInfo.FileName = @"LaunchTM";
        proc.StartInfo.WindowStyle =
                System.Diagnostics.ProcessWindowStyle.Maximized;
        proc.StartInfo.WorkingDirectory = @"C:\Windows\System32";
        proc.StartInfo.Arguments = @"C:\Windows\System32";
        proc.Start();
    }
    catch (Exception)
    {
        MessageBox.Show("Comando fallito");
    }
}
```

➤ Associare al pulsante **button5** il codice seguente:

```
private void button5_Click(object sender, EventArgs e)
{
    try
    {
        System.Diagnostics.Process proc = new
                                                System.Diagnostics.Process ();
        proc.EnableRaisingEvents = false;
        proc.StartInfo.FileName = @"msinfo32";
        proc.StartInfo.WindowStyle =
```




```
        System.Diagnostics.ProcessWindowStyle.Maximized;
        proc.StartInfo.WorkingDirectory = @"C:\Windows\System32";
        proc.Start();
    }
    catch (Exception)
    {
        MessageBox.Show("Comando fallito");
    }
}
```

Associare al pulsante **button6** il codice seguente:

```
private void button6_Click(object sender, EventArgs e)
{
    Close();
}
```

Salvare il progetto e provarlo; ogni pulsante lancia una diversa applicazione esistente

COMMENTO ALLA CLASSE PROCESS E ALLO SPAZIO DEI NOMI SYSTEM.DIAGNOSTICS



[https://msdn.microsoft.com/it-it/library/system.diagnostics\(v=vs.110\).aspx](https://msdn.microsoft.com/it-it/library/system.diagnostics(v=vs.110).aspx)

System.Diagnostics.Process è una classe (un tipo di dato) per costruire nuovi processi di Windows.

La classe **Process** fornisce funzionalità ai processi del sistema di monitoraggio tramite la rete e avviare e arrestare i processi di sistema locale.

Oltre al recupero gli elenchi di processi in esecuzione (specificando il computer, il nome del processo, o l'id processo o a visualizzare le informazioni sul processo che ha attualmente accesso al processore, è possibile ottenere la conoscenza dettagliata dei thread e dei moduli gestiti sia con **Process** classificare e interagisce con **ProcessThread** e **ProcessModule** classi. **ProcessStartInfo** la classe consente di specificare diversi elementi con cui avviare un nuovo processo, come input, output e flussi di errori, directory di lavoro e verbi e argomenti della riga di comando. Questi consentono un controllo accurato sul comportamento dei processi. Altre classi correlate consentono di specificare le priorità degli stili, di processo e thread della finestra e interagiscono con le raccolte di thread e di moduli.

Quando si effettua la dichiarazione:

```
System.Diagnostics.Process proc = new System.Diagnostics.Process();
```

si crea una variabile di tipo processo con cui preparare un nuovo processo che ancora non viene lanciato (e non è attivo nel sistema).

L'istanza processo creata con il costruttore

```
new System.Diagnostics.Process();
```

è un processo da predisporre ed è anche un oggetto con le sue caratteristiche.

In particolare il processo ammette diverse proprietà, tra cui vogliamo osservare le seguenti:

**PROPRIETÀ PROCESS.STARTINFO**

[https://msdn.microsoft.com/it-it/library/system.diagnostics.process.startinfo\(v=vs.110\).aspx](https://msdn.microsoft.com/it-it/library/system.diagnostics.process.startinfo(v=vs.110).aspx)

<https://msdn.microsoft.com/it-it/library/system.diagnostics.processtartinfo.arguments%28v=vs.110%29.aspx>

È un insieme di proprietà che definiscono le caratteristiche iniziali dell'applicazione. È un oggetto che rappresenta i dati con i quali avviare il processo. Tra gli argomenti sono compresi il nome del file eseguibile o il documento utilizzato per avviare il processo.

StartInfo rappresenta l'insieme di parametri da utilizzare per avviare un processo. Quando si chiama Start, viene utilizzata la proprietà **StartInfo** per specificare l'avvio del processo. L'unico membro necessario di **StartInfo** da impostare è la proprietà **FileName**.

L'avvio di un processo specificando la proprietà **FileName** è analogo alla digitazione delle informazioni nella finestra di dialogo Esegui del menu di avvio di Windows. Di conseguenza, la proprietà **FileName** non deve necessariamente rappresentare un file eseguibile. Può essere qualsiasi tipo di file la cui estensione è stata associata a un'applicazione installata sul sistema. Ad esempio, la proprietà **FileName** può avere un'estensione TXT se sono stati associati file di testo a un editor come il Blocco note oppure può avere un'estensione DOC se sono stati associati file DOC a un programma di elaborazione testi come Word.

Analogamente, come la finestra di dialogo Esegui può accettare un nome di file eseguibile con o senza estensione EXE, tale estensione è facoltativa anche nel membro **FileName**. Ad esempio, è possibile impostare la proprietà **FileName** su "Notepad.exe" o su "Notepad".

È possibile modificare i parametri specificati nella proprietà **StartInfo** fino al momento della chiamata al metodo Start sul processo. Dopo l'avvio del processo, la modifica della proprietà **StartInfo** non ha effetti sul processo associato né ne determina il riavvio. Le proprietà importanti sono:

StartInfo.FileName Imposta il nome dell'applicazione che si vuole attivare; per esempio: **explorer**

StartInfo.WindowStyle Imposta lo stile della finestra dell'applicazione da attivare; esempio: **ProcessWindowStyle.Maximized**

StartInfo.WorkingDirectory Imposta la cartella di lavoro dell'applicazione da attivare; è una stringa (meglio preceduta da @)

StartInfo.Arguments Ottiene o imposta l'insieme di argomenti della riga di comando da utilizzare all'avvio dell'applicazione.

StartInfo.UseShellExecute È una proprietà bool, se falsa permette di eseguire qualsiasi processo

StartInfo.CreateNoWindow È una proprietà bool, se true permette di eseguire il processo senza una finestra stile Windows

PROPRIETÀ ENABLERAISINGEVENTS

È un oggetto che indica se il processo deve ricevere notifica della chiusura di un processo da parte del sistema operativo. La proprietà è booleana. Per esempio:

EnableRaisingEvents = false;

**METODO START ();**

Questo metodo avvia l'esecuzione di un processo. In effetti avvia una risorsa di processo e la associa a un componente **Process**.

Si tratta di un membro di overload. Per informazioni complete su questo membro, inclusi la sintassi, l'utilizzo e gli esempi, fare clic su un nome nell'elenco degli overload.

mioProc.Start();

Avvia il processo specificato dalla proprietà **StartInfo** di questo componente **Process** e la associa ad esso

mioProc.Start(ProcessStartInfo);

Avvia il processo indicato nel parametro con le informazioni di avvio del processo

mioProc.Start(String);

Avvia un processo specificando il nome di un documento o un file di applicazione

ALTRI ELEMENTI DI PROCESSO

[https://msdn.microsoft.com/it-it/library/System.Diagnostics.Process_methods\(v=vs.110\).aspx](https://msdn.microsoft.com/it-it/library/System.Diagnostics.Process_methods(v=vs.110).aspx)

L'oggetto processo permette di eseguire diversi metodi, tra cui ricordiamo

Close ();

Chiude il processo in esecuzione

CloseMainWindow ();

Chiude la finestra principale del processo, se esiste

Kill ();

Arresta immediatamente il processo in esecuzione

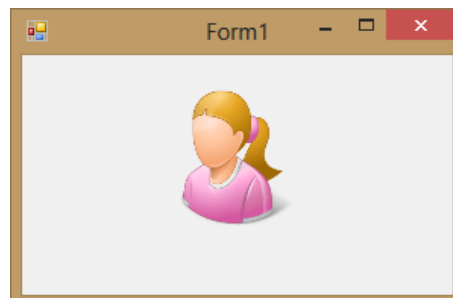


ESERCIZI

ESERCIZI SU: GESTIONE DELLA TASTIERA

ESERCIZIO 1. MOVIMENTO DA TASTIERA I

- Prepara un Form con una PictureBox1 con una immagine a piacere (64x64 pixel).
- Costruisci una applicazione che, quando l'utente preme le frecce della tastiera, l'immagine si muova di 5 pixel nella direzione indicata dalla freccia.



ESERCIZIO 2. MOVIMENTO DA TASTIERA II

- Modificare l'applicazione precedente in modo che quando l'utente preme una freccia insieme al tasto Control, il movimento sia di 15 pixel
- Modificare l'applicazione precedente in modo che quando l'utente preme una freccia insieme al tasto Alt, il movimento sia di 1 solo pixel

ESERCIZIO 3. MOVIMENTO DA TASTIERA III

- Prepara un Form con due PictureBox con delle immagini a piacere (64x64 pixel).
- Costruisci una applicazione che, quando l'utente preme i numeri del tastierino numerico muove opportunamente la prima immagine
- Quando l'utente preme il tasto Alt la seconda immagine si sovrappone alla prima
- Quando l'utente preme il tasto Shift la prima immagine raggiunge la seconda



ESERCIZIO 4. GESTIONE DELLA TASTIERA









- Prepara un Form con quattro immagini
- Quando l'utente preme il tasto Alt+1 una immagine si avvicina alla porta di 10 pixel
- Quando l'utente preme il tasto Alt+2 un'altra immagine si avvicina alla porta di 10 pixel
- Quando l'utente preme il tasto Alt+3 un'altra immagine si avvicina alla porta di 10 pixel

Se una immagine raggiunge la porta la finestra si chiude



SOMMARIO

- RIFINITURE DELL'APPLICAZIONE.....2**
- PAGINA PROGETTAZIONE PROGETTI 2**
 -  Progetto Guidato..... 2
- SPLASH SCREEN 5**
 - Cosa sono gli Splash Screen 5
 -  Progetto Guidato - Creazione di una Schermata iniziale 5
- LA PROPRIETÀ ICON 7**
 -  Progetto Guidato..... 7
- POSIZIONE INIZIALE DELLA FINESTRA 9**
 -  Progetto Guidato..... 9
- BORDI DELLA FINESTRA 10**
 -  Progetto Guidato..... 10
- LANCIARE APPLICAZIONI DAL FORM1 11**
 -  Progetto Guidato..... 11
 - Commento alla classe Process e allo spazio dei nomi System.Diagnostics..... 14
- ESERCIZI SU: GESTIONE DELLA TASTIERA..... 17**
 - Esercizio 1. Movimento da tastiera I 17
 - Esercizio 2. Movimento da tastiera II 17
 - Esercizio 3. Movimento da tastiera III..... 17
 - Esercizio 4. Gestione della tastiera..... 17