

Visual Basic

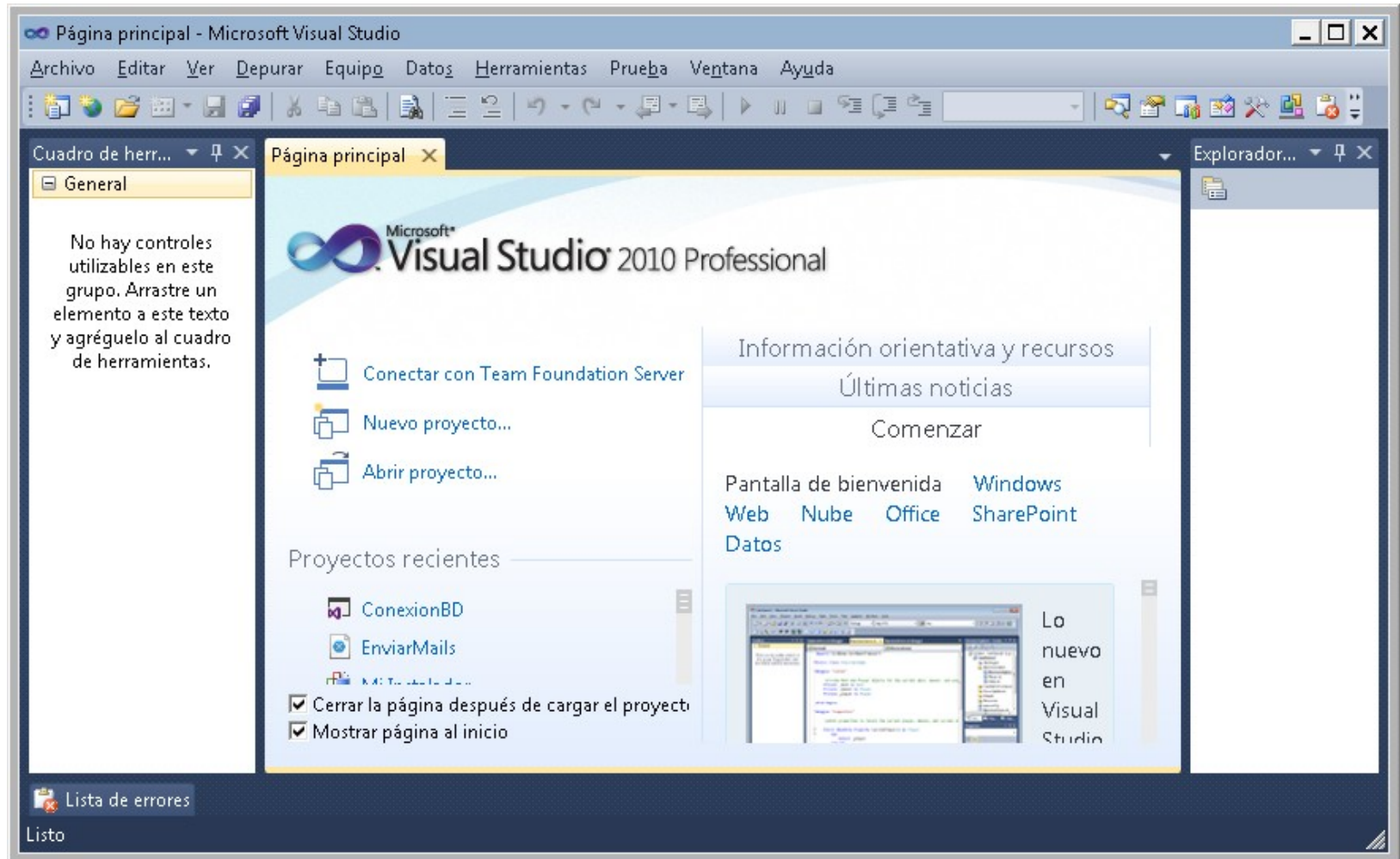
Ejercicio 3

Ejercicio 3

- ❑ **Continuemos programado**
- ❑ **Vamos a desarrollar una aplicación que simule el juego Ta-Te-Ti**
- ❑ **Además veremos el uso de procedimientos y funciones**
- ❑ **Primero ejecutaremos el Visual Studio**

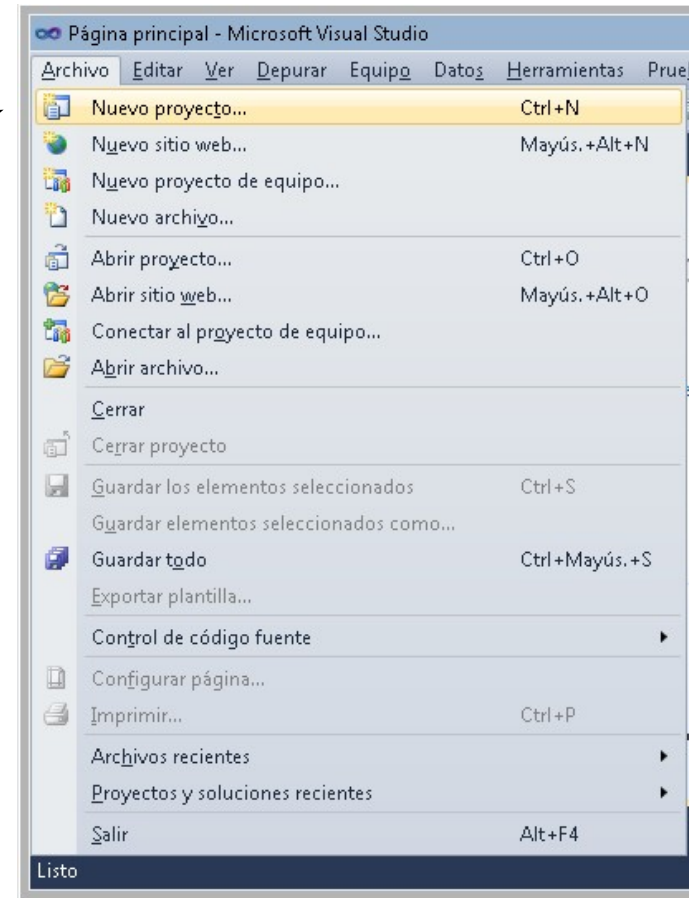


Ejercicio 3

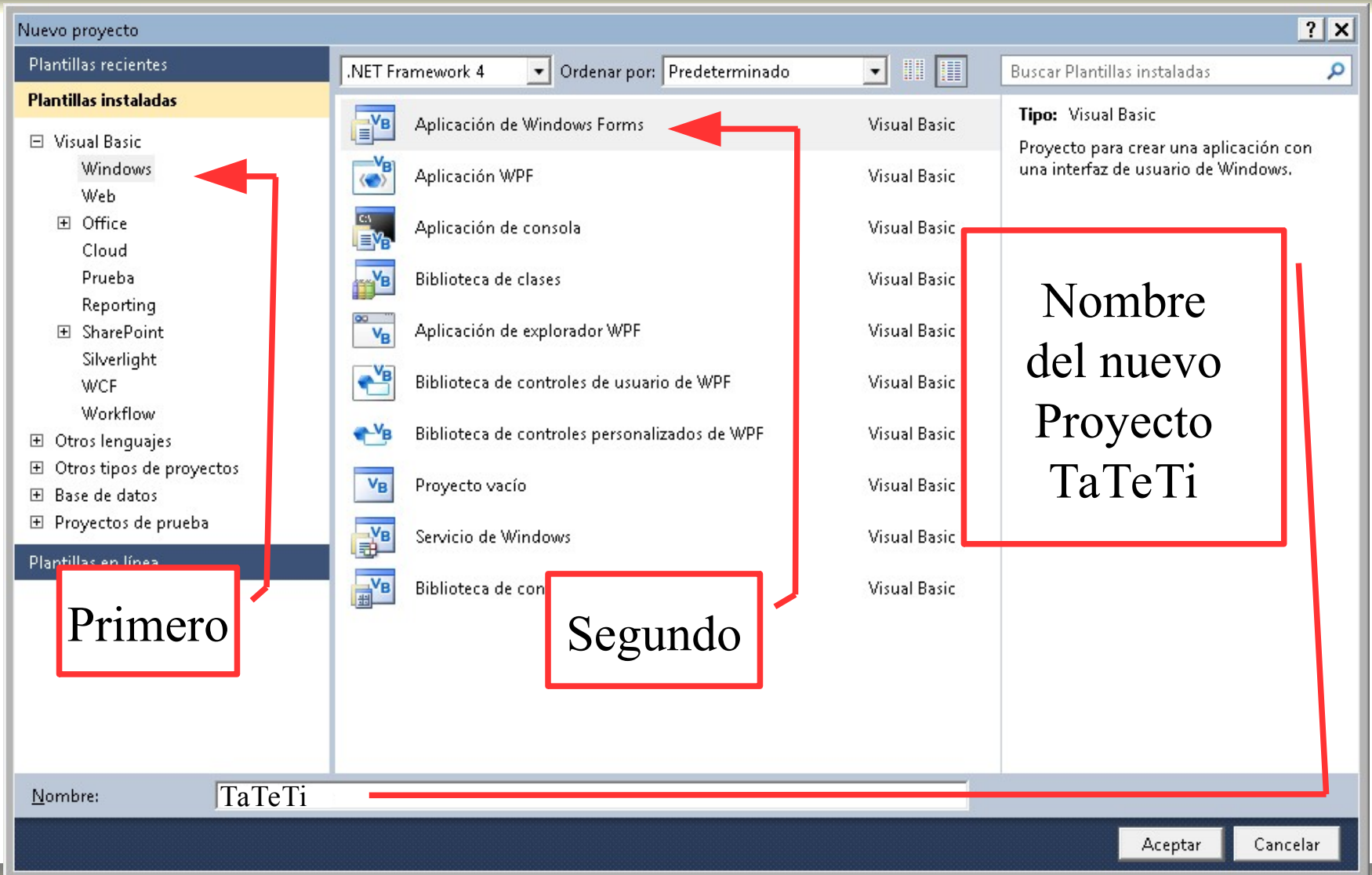


Ejercicio 3

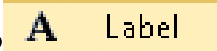
- ❑ **Crear un nuevo proyecto**
 - **Menú Archivo**
 - **Nuevo proyecto...**
- ❑ **Otra opción**
 - **Barra de herramientas**

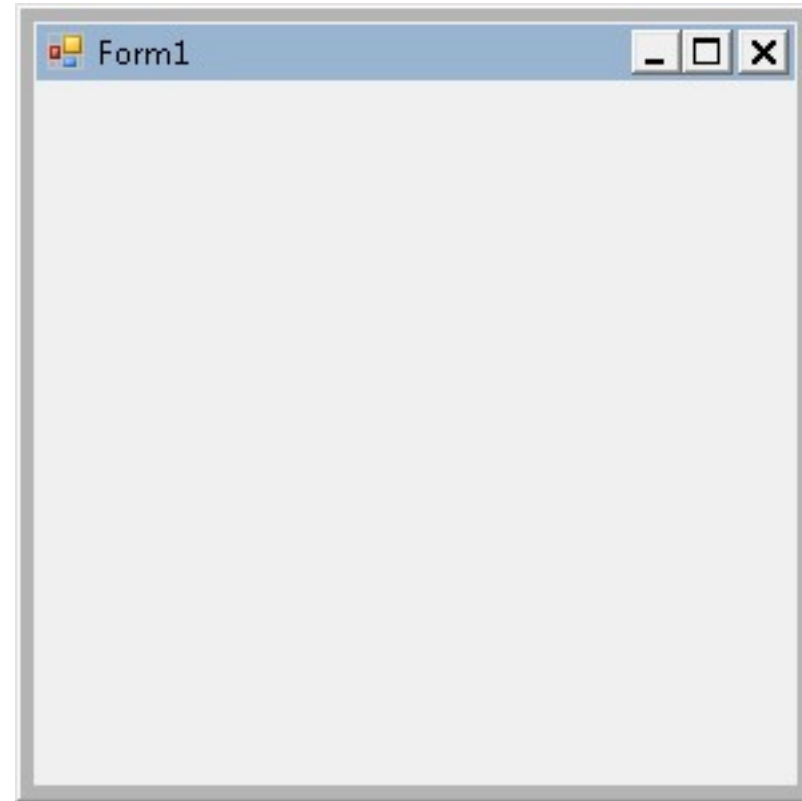


Ejercicio 3

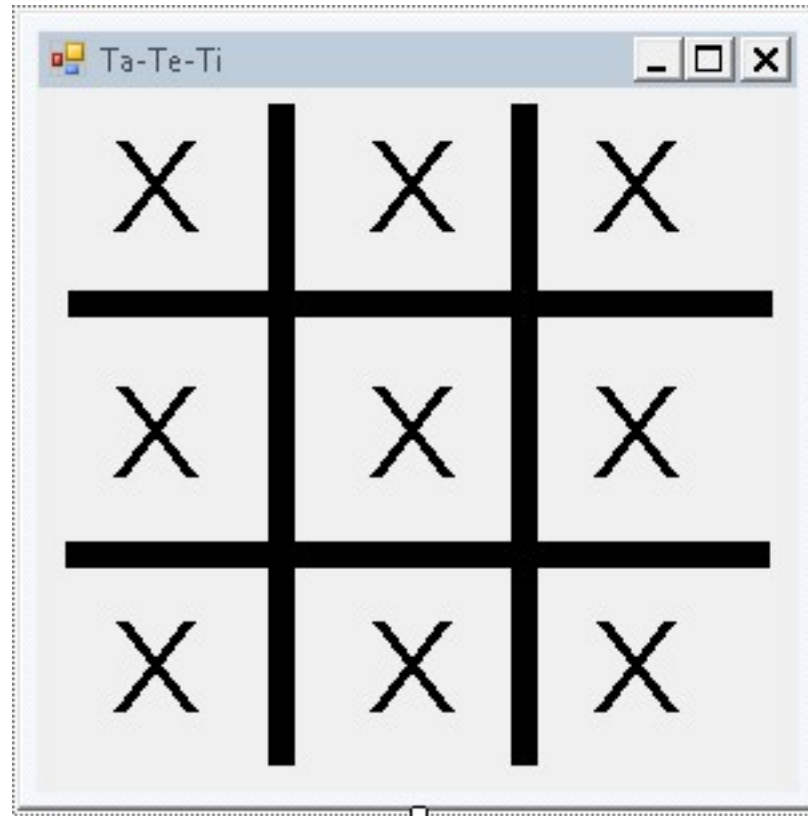


Ejercicio 3

- Al crear un nuevo proyecto del tipo Windows Form se crea automáticamente un formulario por defecto de nombre Form1.
- En este formulario vamos a diseñar nuestra interfaz.
- Del cuadro de herramientas vamos a utilizar los siguientes controles comunes:
 - Trece etiquetas 



Ejercicio 3 - Diseño

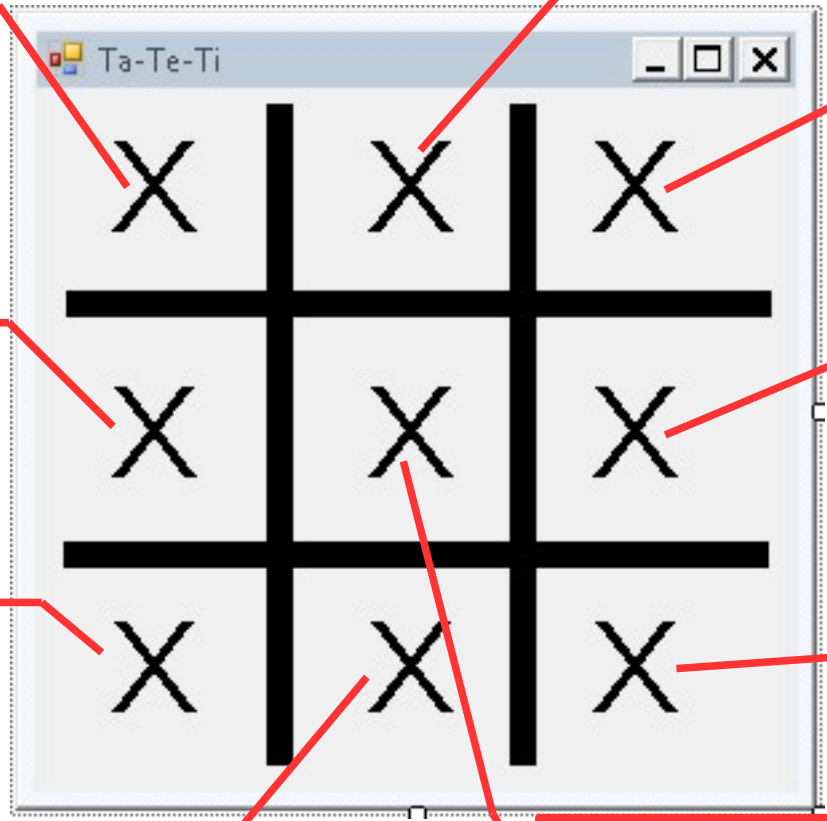


Ejercicio 3 - Diseño

Etiqueta (Label)
Propiedades
(name): lbl_0_1
Text: X

Etiqueta (Label)
Propiedades
(name): lbl_0_0
Text: X

Etiqueta (Label)
Propiedades
(name): lbl_0_2
Text: X



Etiqueta (Label)
Propiedades
(name): lbl_1_0
Text: X

Etiqueta (Label)
Propiedades
(name): lbl_1_2
Text: X

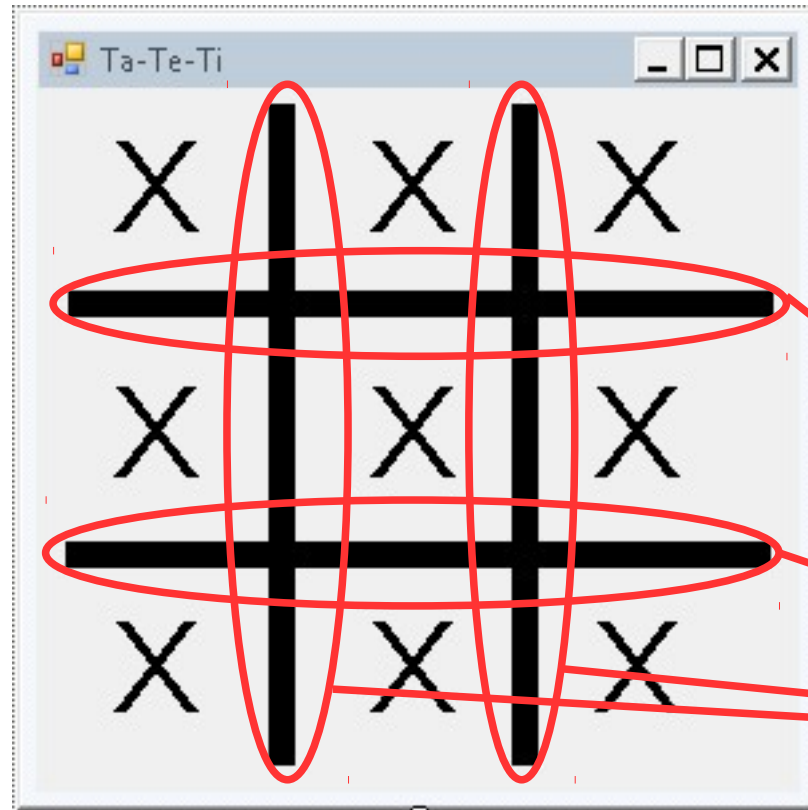
Etiqueta (Label)
Propiedades
(name): lbl_2_0
Text: X

Etiqueta (Label)
Propiedades
(name): lbl_2_2
Text: X

Etiqueta (Label)
Propiedades
(name): lbl_2_1
Text: X

Etiqueta (Label)
Propiedades
(name): lbl_1_1
Text: X

Ejercicio 3 - Diseño



Etiquetas (Label)

El nombre de estas etiquetas no es necesario cambiarlo porque no vamos utilizarlos o mencionarlos en el código, son solo parte del diseño.

Cambiaremos la propiedad BackColor: Negro y AutoSize: False

Ejercicio 3

```
Public Class Form1
```

```
Dim jugador As String = "X"  
Dim turno As Integer = 0
```

Variables globales
Disponibles en todo el formulario

```
Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load
```

```
Lbl_0_0.Text = ""  
Lbl_0_1.Text = ""  
Lbl_0_2.Text = ""  
Lbl_1_0.Text = ""  
Lbl_1_1.Text = ""  
Lbl_1_2.Text = ""  
Lbl_2_0.Text = ""  
Lbl_2_1.Text = ""  
Lbl_2_2.Text = ""
```

```
End Sub
```

```
End Class
```

- A programar el TaTeTi...
 - Al comienzo se muestra el tablero del TaTeTi vacío o limpio. Solo el dibujo.
 - Todas las etiquetas en su propiedad Text tendrán el valor "".
 - Hacemos un doble-clic sobre el formulario y programamos el evento "Load" del mismo.
 - Después cada jugador irá colocando su marca (X o O) en el tablero
 - Haciendo un Click
 - El primer jugador que coloque tres en línea gana (horizontal o verticalmente)

Ejercicio 3

- En VisualBasic, como en la mayoría de los lenguajes de programación, existen las funciones y los procedimientos.
 - La gran diferencia entre ambos es que:
 - Las funciones retornan un valor
 - Los procedimientos no retornan valores
 - Lo que tienen en común:
 - Ambos contienen un conjunto de instrucciones que resuelven parte de un problema mayor
 - Permiten reutilizar código
 - Pueden o no recibir parámetros
 - Tienen un nombre

Ejercicio 3

- Funciones:

Private Function **NombreFuncion** (...) As **TipoDato**

...

NombreFuncion = unValor

End Function

Nombre de la Función

Valor de retorno
de la función

Tipo de dato
Que devuelve
La función

Lista de parámetros.
Opcional

Alcance de la función

Private: Sólo dentro de la clase
Public: Pública para otras clases

Ejercicio 3

- Procedimiento:

```
Private Sub NombreProcedimiento ( ... )
```

```
...
```

```
End Sub
```

Nombre del procedimiento

Lista de parámetros.
Opcional

Alcance del procedimiento
Private: Sólo dentro de la clase
Public: Pública para otras clases

Ejercicio 3

```
Private Sub Lbl_0_0_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Lbl_0_0.Click  
    jugar(sender)  
End Sub
```

- Tenemos que programar el evento Click de cada etiqueta del tablero
 - Para no repetir el mismo código nueve veces vamos a utilizar un procedimiento
 - jugar
 - Recibe como parámetro la etiqueta que corresponda
 - En este caso vamos a utilizar un parámetro que tiene el evento click: sender
 - sender: objeto que envía o dispara el evento

Ejercicio 3

```
Private Function HayGanador() As Boolean
    'Hay que programar esta función!
    HayGanador = False
End Function
```

Definición de la Función HayGanador

```
Private Sub jugar(ByVal sender As System.Object)
    If sender.Text = "" Then
        sender.Text = jugador
        turno += 1
        If HayGanador() Then
            'Ganó alguien
        ElseIf turno = 9 Then
            'Empataron!
            If MsgBox("EMPATEEEEE! Deseas jugar nuevamente?", MsgBoxStyle.YesNo) = MsgBoxResult.Yes Then
                Application.Restart()
            Else
                Application.Exit()
            End If
        End If
        If jugador = "X" Then
            jugador = "O"
        Else
            jugador = "X"
        End If
    Else
        MsgBox("Posición ocupada!", vbCritical)
    End If
End Sub
```

Definición del procedimiento jugar

Ejercicio 3

```
Private Function HayGanador() As Boolean
    'Hay que programar esta función!
    HayGanador = False
End Function
```

```
Private Sub jugar(ByVal sender As System.Object)
    If sender.text = "" Then
        sender.Text = jugador
        turno += 1
        If HayGanador() Then
            'Ganó alguien
        ElseIf turno = 9 Then
            'Empataron!
            If MsgBox("EMPATEEEEE! Deseas jugar nuevamente?", MsgBoxStyle.YesNo) = MsgBoxResult.Yes Then
                Application.Restart()
            Else
                Application.Exit()
            End If
        End If
        If jugador = "X" Then
            jugador = "O"
        Else
            jugador = "X"
        End If
    Else
        MsgBox("Posición ocupada!", vbCritical)
    End If
End Sub
```

Sí la etiqueta está vacía (sin texto)
Colocamos el carácter que corresponda (X o O)
Incrementamos el valor de turno

Sí el jugador es "X" lo cambia por "O"
Sino lo cambia por "X"

Ejercicio 3

```
Private Function HayGanador() As Boolean
    'Hay que programar esta función!
    HayGanador = False
End Function
```



Definición de la Función HayGanador
Debes agregar el código necesario para que se verifique si hay ganador o no.

```
Private Sub jugar(ByVal sender As System.Object)
    If sender.Text = "" Then
        sender.Text = jugador
        turno += 1
        If HayGanador() Then
            'Ganó alguien
        ElseIf turno = 9 Then
            'Empataron!
            If MsgBox("EMPATEEEEE! Deseas jugar nuevamente?", MsgBoxStyle.YesNo) = MsgBoxResult.Yes Then
                Application.Restart()
            Else
                Application.Exit()
            End If
        End If
        If jugador = "X" Then
            jugador = "O"
        Else
            jugador = "X"
        End If
    Else
        MsgBox("Posición ocupada!", vbCritical)
    End If
End Sub
```

Ejercicio 3

- El resto del ejercicio lo deben realizar ustedes
- Sí el jugador “X” o el jugador “O” realizan un TaTeTi o un tres en línea ganan el juego
 - Esto es lo que debe realizar la función HayGanador
 - Devuelve True: Sí hay tres en línea (horizontal o diagonal)
 - Devuelve False: Sí no hay ganador
- La solución de este ejercicio debe enviarse por correo electrónico, al correo leonardocarambula@gmail.com o subirlas a la plataforma Crea 2
 - Asunto: Grupo – Nombre y Apellido – Ejercicio 3
 - Adjuntar la carpeta con la solución de Visual Studio
 - Solo el contenido de la carpeta “TaTeTi”
 - Archivos:
 - Form1.Designer.vb
 - Form1.resx
 - Form1.vb
 - TaTeTi.vbproj
 - TaTeTi.vbproj.user
 - **Por motivos de seguridad Gmail no permite enviar algunos de estos archivos como adjuntos. Por esto debes subirlos a tu Google Drive y agregarlos desde allí.**