

INSTITUTO POLITÉCNICO NACIONAL
 Centro de Estudios Científicos y Tecnológicos N° 1
 “Gonzalo Vázquez Vela”
 Carrera de Técnico en Sistemas Digitales
 Prácticas de Programación Visual



NOMBRE DEL ALUMNO: _____
 Apellido Paterno

_____ Apellido Materno Nombre

N° DE BOLETA: _____ GRUPO: _____

HOJA	DE	FECHA			EVALUACION
1	6	DIA	MES	AÑO	

PROFESOR: _____

Práctica 4

Estructura de un programa en lenguaje de programación visual

UNIDAD TEMATICA 2 DEL PROGRAMA DE ESTUDIOS FUNDAMENTOS DE LA PROGRAMACION VISUAL
RAP RELACIONADO CON LA PRÁCTICA: Explica los componentes que constituyen a un programa en ambiente visual

Objetivos De La Práctica:

1.- El alumno contrastará la diferencia entre la programación estructurada y la programación orientada a eventos.

Equipo Necesario Una Computadora con Microsoft Visual Basic instalado	Material Bata de laboratorio
---	--

MARCO TEORICO.

Investigar los eventos que suceden en un objeto de Visual Basic .Net.

Investigar los métodos que pueden utilizarse en Visual Basic .Net.

Describe los siguientes controles básicos de Visual Basic .net:

- GroupBox
- RadioButton

DESARROLLO

1. Enciende la computadora y ubica en el escritorio el icono de acceso directo a Visual Basic que este instalado. Haz click en Él.
2. Crea un nuevo proyecto del tipo **Aplicación de Windows Forms**.
3. Visual Basic tiene una gran variedad de controles como se muestran a continuación:

Controles Comunes

Como se van a dar cuenta, cada control tiene su icono, nombre, prefijo y su descripción.

Icono	Nombre	Prefijo	Descripción
	Button	btn	Se utiliza para iniciar, detener o interrumpir un proceso.
	CheckBox	chk	Muestra una casilla de verificación y una etiqueta para texto. Se utiliza n general para establecer opciones.
	CheckedListBox	clb	Muestra una lista desplazable de elementos, cada uno acompañado por una casilla de verificación.
	ComboBox	cbo	Muestra una lista desplegable de elementos.
	DateTimePicker	dtp	Muestra un calendario gráfico desplegable, que permite que los usuarios seleccionen una fecha o una hora.
	Label	lbl	Muestra texto que los usuarios no pueden modificar directamente.
	LinkLabel	llbl	Muestra texto en forma de vínculo de estilo Web y desencadena un evento cuando el usuario hace clic en el texto.
	ListBox	lst	Muestra una lista de textos, también llamados elementos.
	ListView	lstv	Muestra los elementos ya sean texto solo o con gráficos (iconos).
	MaskedTextBox	msk	Captura texto escrito por el usuario, ya sea teniendo un formato de entrada.
	MonthCalender	mnc	Muestra un calendario gráfico que permite que los usuarios seleccionen un intervalo de fecha

	NotifyIcon	nti	Muestra un icono, en el área de notificación de estado de la barra de tareas, que representa una aplicación que se ejecuta en segundo plano.
	NumericUpDown	nud	Muestra una lista de número a través de la cual los usuarios se pueden desplazar mediante botones hacia arriba y hacia abajo.
	PictureBox	pic	Muestra archivos de imágenes, tales como mapas bits e iconos, gifs, etc.
	ProgressBar	pgb	Muestra una barra de progreso, mientras realiza una instrucción.
	RadioButton	rdb	Muestra un botón que puede activarse o desactivarse.
	RichTextBox	rtb	Habilita la presentación del texto con formato de texto sencillo o de texto enriquecido (RTF).
	TextBox	txt	Muestra texto escrito en tiempo de diseño que puede ser editado por los usuarios en tiempo de ejecución.
	ToolTip	ttp	Este control activa la propiedad ToolTip en los demás controles, su función es brindar una pequeña información al pasar el Mouse por el control que queramos.
	TreeView	trv	Muestra una colección jerárquica de objetos de nodo que puede constar de texto con casilla de verificación o iconos opcionales.
	WebBrowser	wbr	Permite crear un explorador de Internet con fácil navegación, también se puede abrir múltiples archivos.

Nota: Los prefijos son una referencia, se utiliza para saber de que control se habla en tiempo de diseño, no es obligatorio usarlo, pero es recomendable que se familiaricen, ya que la mayoría de programadores lo usan.

Componentes

Solo les muestro dos controles, que son los más usados.

Icono	Nombre	Prefijo	Descripción
	ImageList	img	Es un contenedor de imágenes, para que después se use con otros controles.
	Timer	tmr	Sirve para realizar un conteo de tiempo, en tiempo de ejecución.

Contenedores

Esta categoría se encuentran todos aquellos controles que sirven para contener otros controles.

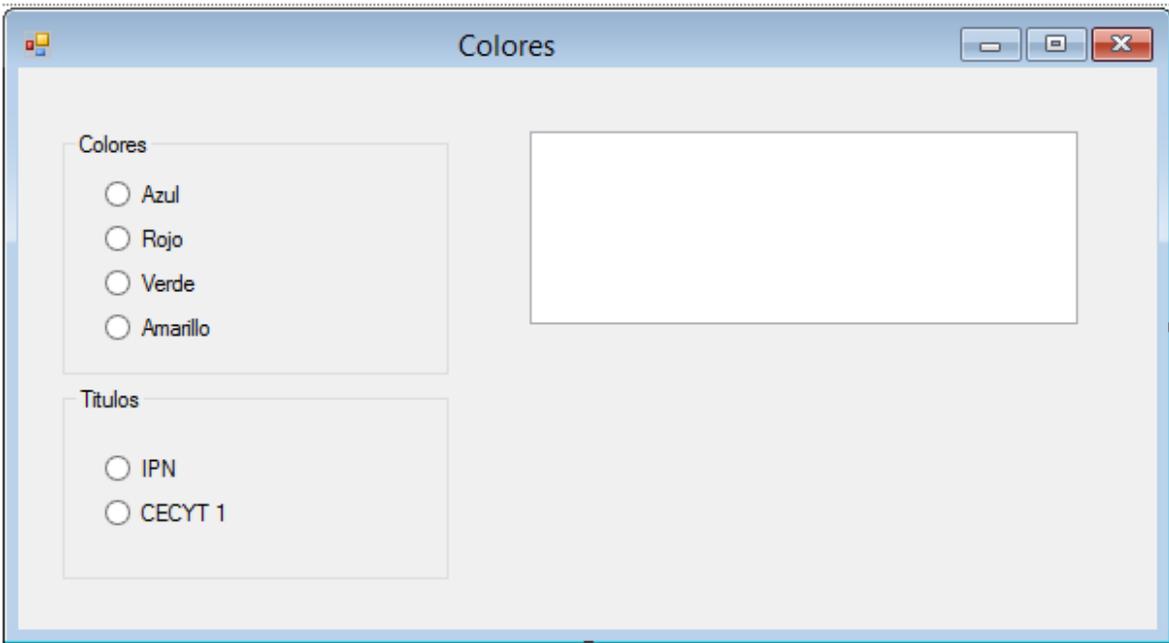
Icono	Nombre	Prefijo	Descripción
	FlowLayoutPanel	flp	Es un contenedor de controles que no permite la modificación de la posición del control.
	GroupBox	gpb	Agrupar un conjunto de controles (tales como botones de opción) en un marco con etiqueta.
	Panel	pnl	Agrupar un conjunto de controles en un marco sin etiqueta que permite el desplazamiento.
	SplitContainer	spc	Es un contenedor de controles con dos paneles, en donde se puede insertar cualquier tipo de control.
	TabControl	tbc	Proporciona una página con fichas para organizar y tener acceso a controles agrupados de forma eficiente.
	TableLayoutPanel	tlp	Es un organizador de controles, ya que podemos añadir filas o columnas para control que queramos.

Menús y Barras de Herramientas

Como se van a dar cuenta, cada control tiene su icono, nombre, prefijo y su descripción.

Icono	Nombre	Prefijo	Descripción
	ContextMenuStrip	cms	Implementa un menú que aparece cuando el usuario hace clic en un objeto con el botón secundario del Mouse.
	MenuStrip	mnu	Proporciona una interfaz en tiempo de diseño para la creación de menús.
	StatusStrip	sts	Muestra información acerca del estado actual de la aplicación mediante una ventana con marco, habitualmente en parte inferior de un formulario.
	ToolStrip	tls	Contiene una colección de botones, combos, etiquetas, etc.
	ToolStripContainer	tsc	Muestra una interfaz de botones con menú.

4. Ahora crea un nuevo proyecto y realiza el diseño de la siguiente ventana:



5. La siguiente tabla muestra los controles y propiedades que se utilizaran en el ejemplo, modifica las propiedades con los valores que se proporcionan para que el código proporcionado funcione:

Control	Propiedad	Valor	Control	Propiedad	Valor
Form1	Name	frmColores	RadioButton4	Name	rbtVerde
	Text	Colores		Text	Verde
GroupBox1	Name	grpColores	GroupBox2	Name	grpTitulos
	Text	Colores		Text	Titulos
RadioButton1	Name	rbtAzul	RadioButton5	Name	rbtIpn
	Text	Azul		Text	IPN
RadioButton2	Name	rbtRojo	RadioButton6	Name	rbtCecyt1
	Text	Rojo		Text	Voca 1
RadioButton3	Name	rbtAmarillo	TextBox	Name	txtCaja
	Text	Amarillo		Text	

6. Ahora captura el código para cada objeto, recuerda dar doble click al control específico para abrir su ventana de código y captúralo.

```
Private Sub rbtAzul1_CheckedChanged(ByVal sender As System.Object, ByVal e As System.EventArgs)
    Handles rbtAzul1.CheckedChanged
        txtCaja.BackColor = Color.Blue
End Sub
```

```
Private Sub rbtIpn_CheckedChanged(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles rbtIpn.CheckedChanged
    txtCaja.Text = "Instituto Politecnico Nacional"
End Sub
```

Adapta el código para los controles de color y el control CECYT 1

Sobre este primer programa en **Visual Basic** se pueden hacer algunos comentarios:

1. Cada una de las partes de código que empieza con un **Private Sub** y termina con un **End Sub** es un **procedimiento**, esto es, una parte de código independiente y reutilizable. El nombre de uno de estos procedimientos, por ejemplo **rbtAzul_Click()**, es típico de **Visual Basic**. La primera parte es el nombre de un objeto (control); después va un separador que es el carácter de subrayado “_”; a continuación el nombre de un evento **-click**, en este caso-, y finalmente unos paréntesis entre los que irían los argumentos, en caso de que los hubiera.
2. Es también interesante ver cómo se accede desde programa a la propiedad **BackColor** de la caja de texto que se llama **txtCaja**: se hace utilizando el punto en la forma **txtCaja.BackColor**. Los colores se podrían también introducir con notación hexadecimal (comenzando con &H, seguidos por dos dígitos entre 00 y FF (es decir, entre 0 y 255 en base 10) para los tres colores fundamentales, es decir para el **Red**, **Green** y **Blue** (RGB), de derecha a izquierda.
3. **Es muy importante** crear primero el control **GroupBox** y después, estando seleccionado, colocar los **botones de opción (RadioButton)** en su interior. No sirve hacerlo a la inversa. **Visual Basic** supone que todos los botones de opción que están dentro del mismo **GroupBox** forman parte del mismo grupo y sólo permite que uno esté seleccionado.
7. El control **Textbox**, permite imprimir texto con diferentes tipos de formato, colores y tamaño. Investiga las propiedades **ForeColor**, **TextAlign** y **Font** y practicalos con el ejercicio anterior.
8. Modifica el programa para que imprima tus datos de Nombre completo, Boleta, Unidad de Aprendizaje y Grupo, Utilizando los Controles **RadioButton** y aplicando estilo y colores a los textos que imprimas.
9. Salva tu proyecto.

Valoración del estudiante.

1. ¿Qué ventajas observas en la programación orientada a eventos?
2. ¿Qué complicaciones encuentras al programar en visual basic?
3. ¿Qué hay que hacer en un programa de visual basic para que se produzca un evento?
4. ¿Qué es un proyecto en visual basic?

CONCLUSIONES: En su cuaderno o portafolio de evidencias.