


Trabajo práctico no. 3

Visual C#. Acceso a bases de datos

1 – Abre un proyecto en C# el cual tendrá una forma que sirva para mostrar el material de construcción que vende una tienda comercial. La forma deberá tener la siguiente apariencia:



a) Para elaborar esta forma deberás agregar al proyecto la clase *DefineDatos* que está en el archivo *DefineDatos.cs*. En la clase se usa el arreglo *arts* para guardar datos de artículos que se usarán para completar la forma.

b) Al seleccionar un *Tipo de artículo* se deberá mostrar, primeramente, la imagen que corresponda a ese tipo (utilizando un control **pictureBox**).

Posteriormente, se deberá mostrar en el **comboBox** *Artículos* el nombre de los artículos que pertenecen **sólo** al tipo seleccionado.

Al seleccionar un artículo, éste deberá agregarse al **listBox Artículos seleccionados**, junto con su *Precio* que deberá aparecer en el **listBox Precio**. Un mismo artículo puede seleccionarse más de una vez.

c) Cuando se oprima el botón *Calcular totales*, se deberán mostrar la *Cantidad de artículos* comprados y el *Monto* de los mismos (suma de los precios); hay que aplicar un descuento de 10%, si el monto supera los \$10,000 y de sólo un 5%, si el monto es mayor a \$5,000 (mostrando el resultado en *Pago neto*). Después hay que calcular el *IVA* (con respecto al resultado anterior) y el *Total a pagar*.

d) Cuando se dé clic al botón *Borrar*, se deberá limpiar el contenido del **comboBox**, de los **listBox** y de las cajas de texto.

Establece valores para las propiedades de los siguientes controles:

Objeto	Propiedad	Valor
pictureBox1	(Name)	picGeneral
comboBox1	(Name)	cboArtículos
	Text	(vacío)
listBox1	(Name)	lstArtículos
listBox2	(Name)	lstPrecio
button1	(Name)	btnTotales
	Text	Calcular totales
button2	(Name)	btnBorrar
	Text	Borrar
textBox1	(Name)	txtCant
	Text	(vacío)
textBox2	(Name)	txtMonto
	Text	(vacío)
textBox3	(Name)	txtNeto
	Text	(vacío)
textBox4	(Name)	txtIva
	Text	(vacío)
textBox5	(Name)	txtTotal
	Text	(vacío)

Tarea no. 4

Aplicación de escritorio

2 - La presente tarea servirá para trabajar con el acceso a base de datos desde Visual C# usando aplicaciones de escritorio (tipo cliente-servidor).

La tarea está dividida en varias partes las cuales se desarrollarán paulatinamente hasta tener la aplicación completa. La aplicación tiene como objetivo el manejar varios clientes los cuales piden préstamos monetarios a un banco. Se debe conservar la información de estos clientes, de los préstamos que solicitaron y de los pagos que han realizado.

Nota: en toda la aplicación se deben escribir comentarios adecuados al inicio del código de cada forma, de cada clase, de cada método y a lo largo de estos últimos.

Inicialmente se debe elaborar una forma que sirva para calcular la mensualidad y el total a pagar por el préstamo pedido por un cliente. Los datos del préstamo (plazo, tasa de interés, etc.) serán dados por el gerente del banco. En la aplicación hay que definir una clase que contenga métodos para hacer determinadas operaciones relacionadas con el préstamo. También hay que utilizar una función predefinida del ambiente de Visual Basic que se empleará para calcular el pago mensual.

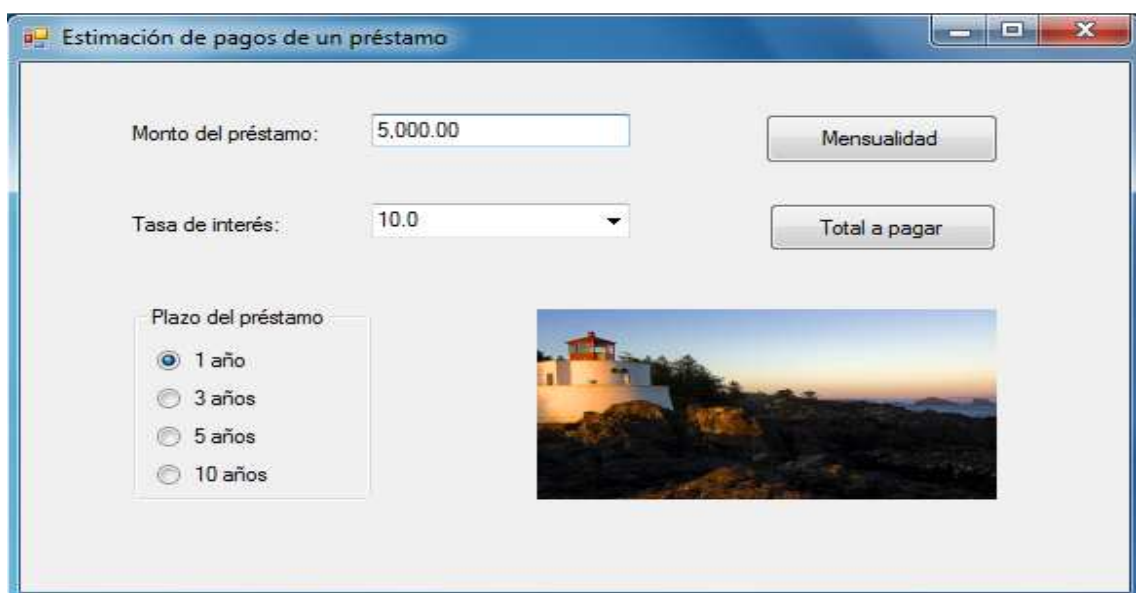
Ejercicio 1: Crear el proyecto

Nota: establecer la ruta: C:\DAI\Programas, en Herramientas, Opciones, Proyectos y soluciones, Ubicación de proyectos.

Crea un proyecto nuevo en Visual C#. En el árbol del proyecto elige a Form1 y cambia su nombre en la propiedad: nombre de archivo, a *FrmCalculaPagos* (la extensión debe seguir como .cs). Selecciona a la forma y en su propiedad (Name) escribe: *FrmCalculaPagos*. Guarda el proyecto.

Ejercicio 2: Interfaz de usuario

Elabora una forma de Windows parecida a la siguiente (con colores e imagen al gusto).



Establece valores para las propiedades de los siguientes controles:

Objeto	Propiedad	Valor
textBox1	(Name)	txtMonto
	Text	(vacío)
comboBox1	(Name)	cboTasa
	Text	(vacío)
groupBox1	(Name)	grbPlazo
	Text	Plazo del préstamo
button1	(Name)	btnMensual
	Text	Mensualidad
button2	(Name)	btnTotal
	Text	Total a pagar

Establece la propiedad TabIndex de txtMonto en 1 para que sea la primera en tener el foco en la forma.

En el ComboBox escribe los valores: 10.0, 12.0, 15.0 y 20.0 como tasas de interés.

Ejercicio 3: Agregar una clase al proyecto

Añadir una clase con el nombre *Fórmulas* (colocarse en el nombre del proyecto, en el Explorador de soluciones, menú breve, Agregar, Clase...).

Crear una función para calcular el pago mensual del préstamo.

1. Agrega un método público con el nombre *mensualidad*, de tipo decimal.
 2. El método tendrá como parámetros: la tasa de interés, el plazo del préstamo y el monto a prestar. Todos los parámetros deben ser de tipo double.
 3. Utiliza la función predefinida de Visual Basic: **Pmt**, para calcular la mensualidad.
- La función Pmt calcula el pago mensual que se debe hacer por un préstamo de dinero. La función requiere tres parámetros:
- a) La tasa de interés mensual que se va a aplicar al préstamo, dada con decimales.
 - b) El plazo total del préstamo, dado en cantidad de meses.
 - c) El monto del préstamo, dado en valor negativo. Por ejemplo, si el préstamo fue de \$10,000.00, este parámetro se da como: -10000.00.

Para poder llamar a esta función dentro de Visual C# hay que hacer lo siguiente:

- a) Agregar al proyecto de C# la biblioteca Microsoft.VisualBasic.dll:
Colocarse en el nombre del proyecto (en el Explorador de soluciones), menú breve, Agregar referencia..., buscarla en la lista de las bibliotecas que aparecen en la entrada Ensamblados/Framework, marcarla, <Aceptar>.
- b) Agregar al inicio del código de la clase (o forma): `using Microsoft.VisualBasic;`
(using tiene una función similar a import en Java).
- c) Llamar a la función como: `Financial.Pmt(par1, par2, par3);`

Ejercicio 4: Usando la función *mensualidad*

1. En el evento Load de la forma *FrmCalculaPagos*, inicializa los controles correspondientes para que: el monto del préstamo aparezca con “5,000.00”, la tasa de interés que se muestre sea 10.0 y el botón de 1 año sea el que aparezca habilitado.
2. Programa los botones de radio de tal manera que al elegir uno, se obtenga la cantidad de años del plazo del préstamo.
3. En el evento Click de *btnMensual* llama al método *mensualidad*, pasándole como parámetros los valores que correspondan según lo indicado por el usuario en los controles de tasa de interés, plazo y monto.
4. Despliega el resultado con un MessageBox¹, cuidando que la mensualidad tenga un formato de pesos y centavos.

Ejercicio 5: Más funciones

1. Crea en la clase *Fórmulas* un método público *pagoTotal* que calcule el total a pagar por el préstamo.
2. Por medio del botón *btnTotal* llama a este método con los parámetros que necesite; despliega el resultado con un MessageBox con el formato de pesos y centavos.

Nota: verifica que los cálculos sean correctos tomando como base la tabla que se muestra a continuación:

Préstamo	Tasa (%)	Plazo (Años)	Mensualidad	Total a pagar
\$5,000.00	10.0	1	\$439.58	\$5,274.95
\$15,000.00	20.0	5	\$397.41	\$23,844.50
\$10,000.00	15.0	10	\$161.33	\$19,360.19
\$20,000.00	12.0	3	\$664.29	\$23,914.30

Ejercicio 6: Biblioteca para acceder a la base de datos

1. Crea un nuevo proyecto para almacenar una biblioteca dll la cual contendrá una clase y varios métodos para hacer la conexión a una base de datos de SQL Server, y para consultar, agregar, borrar y cambiar información en la misma.
Para crear esta biblioteca sigue el procedimiento 4 indicado en las notas del archivo C5ConexiónCS_BD.pdf.
2. Agrega la clase y los métodos de acceso a la base de datos según lo visto en clase.

Ejercicio 7: Construcción de la base de datos: Préstamos

1. La base de datos contiene información de los clientes, de sus préstamos y de los pagos realizados para amortizar los préstamos. Un cliente puede tener varios préstamos y un préstamo puede tener varios pagos registrados.
2. Construye la base de datos correspondiente utilizando las instrucciones de SQL que están en los archivos PréstamosTablas.txt, PréstamosTuplas.txt (o PréstamosTuplasB.txt).

¹ Para poder usar la clase MessageBox, hay que teclear al inicio del archivo: `using System.Windows.Forms;`

Ejercicio 8: Elaboración de la aplicación completa de Préstamos.

1. Agrega al proyecto una nueva forma (*FrmIngreso*) la cual servirá para validar que el usuario y la contraseña dados existan en la base de datos. La forma tendrá una apariencia similar a la siguiente (colores e imagen al gusto).

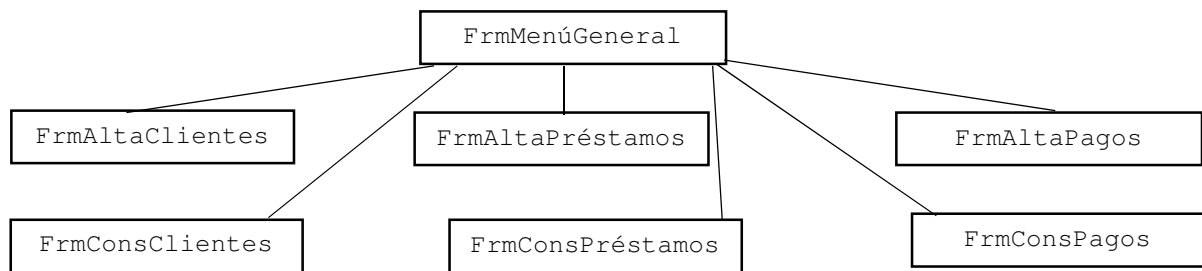


Establece valores para las propiedades de los siguientes controles:

Objeto	Propiedad	Valor
Form1	(Name)	FrmIngreso
	StartPosition	CenterScreen
	BackgroundImage	(Una imagen)
	BackgroundImageLayout	Stretch
textBox1	(Name)	txtUsuario
	Text	(vacío)
textBox2	(Name)	txtContra
	Text	(vacío)
	PasswordChar	*
button1	(Name)	btnIngresar
	Text	Ingresar

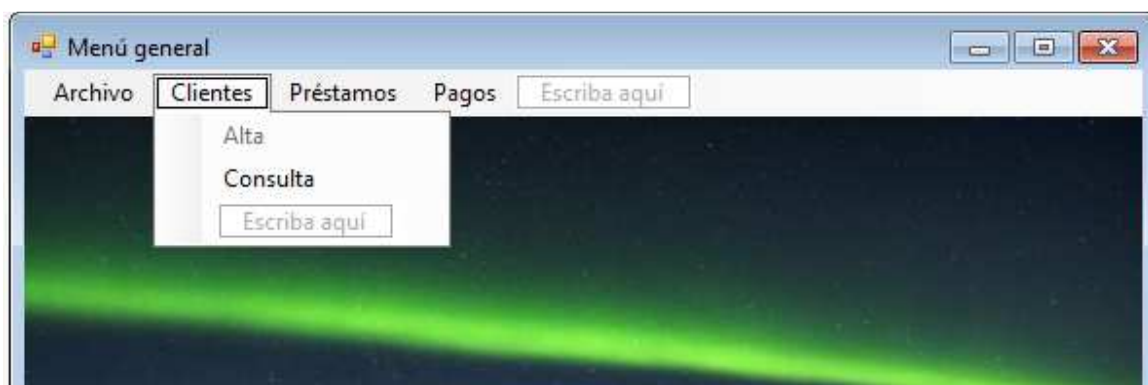
Para que ésta sea la primera forma en mostrarse cuando se ejecute el proyecto, haz lo siguiente: en el Explorador de soluciones: abrir Program.cs, en Application.Run cambiar la forma que ahí aparece por *FrmIngreso*, cerrar Program.cs, guardar el cambio.

2. La forma de ingreso permitirá acceder al resto de la aplicación, la cual consistirá de una serie de formas enlazadas de acuerdo a como se muestra en el siguiente diagrama. La explicación de las formas se hace a partir del punto 4.



Desde la forma del menú general se podrá acceder a las demás formas, aunque el gerente es el único que puede hacer altas, sean de clientes, de préstamos o de pagos. Los clientes sólo pueden consultar sus datos, sean personales, de préstamos o de pagos.

3. Varias de estas formas emplean un método que se llama *cargaCombo* el cual hay que definir en una clase *Comunes*, la cual debes agregar al proyecto de manera similar a como lo hiciste para la clase *Fórmulas* en el ejercicio 3.
4. La forma *FrmMenúGeneral* contendrá una barra de menús desde la cual se tendrán opciones para usar las demás formas. Su aspecto será parecido al siguiente (colores e imagen al gusto):



El menú *Archivo* tendrá un submenú *Salir* para terminar la aplicación.

5. Las formas de altas permitirán incorporar a la base de datos a nuevos clientes, préstamos y pagos, y sólo podrán ser usadas por el gerente del banco.
Para dar de alta un préstamo, primeramente el gerente seleccionará por medio de un combo al cliente al cual se le dará el préstamo. Después deberá usarse la forma *FrmCalculaPagos*, definida al inicio del proyecto, para obtener los diversos valores requeridos para hacer el alta.
Para dar de alta un pago, primeramente el gerente seleccionará al cliente con un combo y luego, por medio de otro combo, a un préstamo de dicho cliente para asignarle el pago. Nota que al registrar un pago a un préstamo, el saldo de éste debe decrementarse.
6. Las formas de consultas podrán ser usadas tanto por el gerente como por los clientes del banco. En el caso del gerente, este seleccionará por medio de un combo al cliente cuyos datos desea consultar. Los clientes sólo podrán ver sus datos: personales, préstamos y pagos (usar combos).

La interfaz gráfica de cada forma se debe elaborar de tal manera que tenga una presentación "agradable" al usuario, tanto en colores como en imágenes mostradas, considerando que los datos deben distinguirse de inmediato al ver la forma.