La programación hibrida:

La programación hibrida es utilizada en los casos en donde el código en ensamblador dificulta la estructuración del programa. La programación híbrida proporciona un mecanismo por medio del cual podemos aprovechar las ventajas del lenguaje ensamblador y los lenguajes de alto nivel, todo esto con el fin escribir programas más rápidos y eficientes.

Al trabajar con un lenguaje de alto nivel, en ocasiones nos encontramos con el problema de que necesitamos que haga determinada función o trabajo pero desafortunadamente ésta solo existe en otro lenguaje que no es el que necesitamos utilizar, o simplemente, no encontramos esa función en ningún lenguaje de alto nivel.

Ventajas de la Programación Híbrida

• Para mejorar la escalabilidad

• Cuando muchas tareas producen desbalanceo

• Aplicaciones que combinan paralelismo de grano grueso y fino

• Reducción del tiempo de desarrollo de código

• Cuando el número de procesos MPI es fijo

• En caso de mezcla de paralelismo funcional y de datos

En este momento el lenguaje ensamblador constituye una herramienta no solo eficaz, sino simple para producir un parche para el compilador de nuestro lenguaje preferido. Tal vez el mayor problema con el que nos enfrentemos sea el de cómo conectar ambos programas (el de alto y el de bajo niveles) y cómo pasar variables de un programa al otro. Para conseguir nuestro objetivo se utilizan pseudo-operadores, es decir, instrucciones que aparecen en el código fuente del ensamblador pero que no generan ninguna instrucción de máquina, pero proporcionan directivas para que el ensamblador pueda operar con datos, ramificaciones condicionales, generación de listados y con macros durante el proceso de ensamble.

**Turbo Pascal y Ensamblador**

Uses Crt;

Var

Car: Char;

i,j : integer;

;Este procedimiento limpia la pantalla y pone blanco sobre azul

Procedure Limpia\_Pantalla;

Assembler;

Asm

Mov AX,0600h

Mov Bh,17h

Mov CX,0000h

Mov DX,184Fh

Int 10h

End;

;Este procedimiento imprime el carácter en la pantalla

Procedure Imprime\_Car(C: Char);

Assembler;

Asm

Mov Ah,02h

Mov Dl,C

Int 21h

End;

;Este procedimiento tiene la misma función que el procedimiento Goto XY de Turbo Pascal

Procedure Cursor\_XY(X,Y: Byte);

Assembler;

Asm

Mov Ah,02h

Mov Bh,00h

Mov Dh,Y

Mov Dl,X

Int 10h

End;

Begin

Limpia\_Pantalla;

Repeat

Limpia\_Pantalla;

Cursor\_XY(0,0);

Write(‘Introduce un carácter: ‘);

Car:=Read Key;

Imprime\_Car(Car);

Limpia\_Pantalla;

If car #13 then

Begin

For i:=0 to 24 do

For j:=0 to 79 do

Cursor\_XY(j,i);

Imprime\_Car(Car);

End;

Cursor\_XY(30,24);

Write(‘Presiona enter para salir u otro para seguir…’);

Readln;

Until car = #13;

End.

**Java y Ensamblador**

**Parte 1**

public class Operaciones {

public native int sumar(int a, int b);

public native int restar(int a, int b);

public native int multiplicar(int a, int b);

static {

System.loadLibrary("operacionesImpl");

}

public static void main(String[] args) {

int n1 = 2;

int n2 = 3;

Operaciones op = new Operaciones();

int resultado = op.sumar(n1, n2);

System.out.println("El resultado de la suma es: " +

resultado);

resultado = op.restar(n1, n2);

System.out.println("El resultado de la resta es: " +

resultado);

resultado = op.multiplicar(n1, n2);

System.out.println("El resultado de la multiplicacion es: "

+ resultado);

}

}

**Parte 2**

/\* DO NOT EDIT THIS FILE - it is machine generated \*/

#include <jni.h>

/\* Header for class Operaciones \*/

#ifndef \_Included\_Operaciones

#define \_Included\_Operaciones

#ifdef \_\_cplusplus

extern "C" {

#endif

/\*

\* Class: Operaciones

\* Method: sumar\* Signature: (II)I

\*/

JNIEXPORT jint JNICALL Java\_Operaciones\_sumar

(JNIEnv \*, jobject, jint, jint);

/\*

\* Class: Operaciones

\* Method: restar

\* Signature: (II)I

\*/

JNIEXPORT jint JNICALL Java\_Operaciones\_restar

(JNIEnv \*, jobject, jint, jint);

/\*

\* Class: Operaciones

\* Method: multiplicar

\* Signature: (II)I

\*/

JNIEXPORT jint JNICALL Java\_Operaciones\_multiplicar

(JNIEnv \*, jobject, jint, jint);

#ifdef \_\_cplusplus

}

#endif

#endif

**Parte 3**

**.386**

**.model flat,stdcall**

**.code**

**Java\_Operaciones\_sumar proc JNIEnv:DWORD, jobject:DWORD, a:DWORD,**

**b:DWORD**

**mov eax, a ;ponemos el valor de a en el registro eax**

**mov ebx, b ;ponemos el valor de b en el registro ebxadd eax,ebx ;sumamos el contenido del registro eax con el de**

**ebx y**

**;el resultado se guarda en eax**

**ret ;por default se retorna el valor del registro eax**

**Java\_Operaciones\_sumar endp**

**Java\_Operaciones\_restar proc JNIEnv:DWORD, jobject:DWORD, a:DWORD,**

**b:DWORD**

**mov eax, a**

**mov ebx, b**

**sub eax, ebx ;restamos el contenido del registro ebx al de**

**abx**

**ret**

**Java\_Operaciones\_restar endp**

**Java\_Operaciones\_multiplicar proc JNIEnv:DWORD, jobject:DWORD,**

**a:DWORD, b:DWORD**

**mov eax, a**

**mov ebx, b**

**mul ebx ;multiplicamos el contenido del registro ebx con el**

**registro por default eax**

**ret**

**Java\_Operaciones\_multiplicar endp**

**END**

**C y Ensamblador**

#pragma once

namespace calculator {

using namespace System;

using namespace System::ComponentModel;

using namespace System::Collections;

using namespace System::Windows::Forms;

using namespace System::Data;

using namespace System::Drawing;

//método operacion para sumar, restar y multiplicar.

//Recibe ca (cantidad a), cb (cantidad b) y nop (numero de operación)

int operacion(int ca,int cb,int nop)

{

//variable para almacenar el resultado

int resultado=0;

// si nop (numro de operacion es igual a 1 realizara la suma

if(nop==1)

{

//inicio de bloque de código en ensamblador

\_asm{

//transfiere al registro ax extendido ca (cantidad a)

mov eax,ca

//suma al registro ax extendido cb (cantidad b)

add eax,cb

//transfiere a resultado el valor del registro ax extendido

mov resultado,eax

}

//fin de bloque de código en ensamblador

//retorna resultado

return resultado;

}

// si nop número de operación es iguala 2 realizará la resta

else if(nop==2)

{

//inicio de bloque de código en ensamblador

\_asm{

//transfiere al registro ax extendido ca (cantidad a)

mov eax,ca

//resta al registro ax extendido cb (cantidad b)

sub eax,cb

//transfiere a resultado el valor del registro ax extendido

mov resultado,eax

}

//fin de bloque de código en ensamblador

//retorna resultado

return resultado;

}

// si nop (número de operación) es igual a 3 tres realizará la multiplicación

else if(nop==3)

{

//inicia bloque de código ensamblador

\_asm{

//agrega a la pila el registro ax extendido

push eax

//agrega a la pila el registro cx extendido

push ecx

//transfiere al registro cx extendido ca (cantidad a)

mov ecx,ca

//transfiere al registro ax extendido 0h

mov eax,0h

//transfiere al registro ax extendido cb (cantidad b)

mov eax,cb

//multiplica el registro cx extendido por el registro

//ax entendido y almacena el resultado en el registro

//ax extendido

mul ecx

//transfiere a resultado el registro ax extendido

mov resultado,eax

//extrae de la pila el registro cx extendido

pop ecx

//extrae de la pila el registro ex extendido

pop eax

}

//fin de bloque de código en ensamblador

//retorna resultado

return resultado;

}

//si nop(número de operación es 4 entonces

//realizará la división

else if(nop==4)

{

//se declaran dos variables de tipo entero

//llamadas n1 y n2

int n1=0,n2=0;

//inicia bloque de codigo ensamblador donde realizará

//solo transferencias entre los registros pues la operación

//div de ensamblador produce resultados que no pudimos convertir a

//los resultados correctos de la división

\_asm{

//transferimos a los registros ax y dx extendidos el valor de 0000h

mov eax,0000h

mov edx,0000h

//transferimos al registro dx extendido ca(cantidad a)

mov edx,ca

//transferimos al registro ax extendido cb(cantidad b)

mov eax,cb

//transferimos a la variable n1 el valor del registro dx extendido

mov n1,edx

//transferimos a la variable n2 el valor del registro ax extendido

mov n2,eax

}

//fin del bloque de código en ensamblador

//resultado es igual a n1 dividido entre n2

resultado=n1/n2;

//retornamos el valor de resultado

return resultado;

}

}

/// <summary>

/// Resumen de Form1

/// </summary>

public ref class Form1 : public System::Windows::Forms::Form

{

public:

//declaramos variables globales de tipo entero

//cant1(cantidad1),cant2(cantidad2),salida(resultado de la operación),

//numcant(número de cantidad),numoper(número de operación)

int cant1,cant2,salida,numcant,numoper;

//declaramos variables globales de tipo String

// a(cantidad a), b(cantidad b),operador(para indicar la operación)

String ^ a;

String ^ b;

String ^ operador;

String ^ igual;

Form1(void)

{

InitializeComponent();

//

//TODO: agregar código de constructor aquí

//

//inicialización de la variables globales

cant1=0;

cant2=0;

salida=0;

numcant=0;

numoper=0;

a = "";

b = "";

operador="";

igual="=";

}

protected:

/// <summary>

/// Limpiar los recursos que se estén utilizando.

/// </summary>

~Form1()

{

if (components)

{

delete components;

}

}

private: System::Windows::Forms::Button^ btnclear;

protected:

private: System::Windows::Forms::TextBox^ txtdatos;

private: System::Windows::Forms::Button^ btnigual;

private: System::Windows::Forms::Button^ btndiv;

private: System::Windows::Forms::Button^ btnres;

private: System::Windows::Forms::Button^ btnsum;

private: System::Windows::Forms::Button^ btnmul;

private: System::Windows::Forms::Button^ btn1;

private: System::Windows::Forms::Button^ btn3;

private: System::Windows::Forms::Button^ btn4;

private: System::Windows::Forms::Button^ btn7;

private: System::Windows::Forms::Button^ btn5;

private: System::Windows::Forms::Button^ btn8;

private: System::Windows::Forms::Button^ btn9;

private: System::Windows::Forms::Button^ btn6;

private: System::Windows::Forms::Button^ btn0;

private: System::Windows::Forms::Button^ btn2;

private: System::Windows::Forms::GroupBox^ groupBox1;

//public:

private: System::Windows::Forms::GroupBox^ groupBox2;

private:

/// <summary>

/// Variable del diseñador requerida.

/// </summary>

System::ComponentModel::Container ^components;

#pragma region Windows Form Designer generated code

/// <summary>

/// Método necesario para admitir el Diseñador. No se puede modificar

/// el contenido del método con el editor de código.

/// </summary>

void InitializeComponent(void)

{

this->btnclear = (gcnew System::Windows::Forms::Button());

this->txtdatos = (gcnew System::Windows::Forms::TextBox());

this->btnigual = (gcnew System::Windows::Forms::Button());

this->btndiv = (gcnew System::Windows::Forms::Button());

this->btnres = (gcnew System::Windows::Forms::Button());

this->btnsum = (gcnew System::Windows::Forms::Button());

this->btnmul = (gcnew System::Windows::Forms::Button());

this->btn1 = (gcnew System::Windows::Forms::Button());

this->btn3 = (gcnew System::Windows::Forms::Button());

this->btn4 = (gcnew System::Windows::Forms::Button());

this->btn7 = (gcnew System::Windows::Forms::Button());

this->btn5 = (gcnew System::Windows::Forms::Button());

this->btn8 = (gcnew System::Windows::Forms::Button());

this->btn9 = (gcnew System::Windows::Forms::Button());

this->btn6 = (gcnew System::Windows::Forms::Button());

this->btn0 = (gcnew System::Windows::Forms::Button());

this->btn2 = (gcnew System::Windows::Forms::Button());

this->groupBox1 = (gcnew System::Windows::Forms::GroupBox());

this->groupBox2 = (gcnew System::Windows::Forms::GroupBox());

this->groupBox1->SuspendLayout();

this->groupBox2->SuspendLayout();

this->SuspendLayout();

//

// btnclear

//

this->btnclear->Font = (gcnew System::Drawing::Font(L"Arial Black", 12, System::Drawing::FontStyle::Bold, System::Drawing::GraphicsUnit::Point,

static\_cast<System::Byte>(0)));

this->btnclear->Location = System::Drawing::Point(47, 155);

this->btnclear->Name = L"btnclear";

this->btnclear->Size = System::Drawing::Size(37, 37);

this->btnclear->TabIndex = 34;

this->btnclear->Text = L"C";

this->btnclear->UseVisualStyleBackColor = true;

this->btnclear->Click += gcnew System::EventHandler(this, &Form1::btnclear\_Click);

//

// txtdatos

//

this->txtdatos->Location = System::Drawing::Point(18, 15);

this->txtdatos->Name = L"txtdatos";

this->txtdatos->Size = System::Drawing::Size(262, 20);

this->txtdatos->TabIndex = 33;

this->txtdatos->TextAlign = System::Windows::Forms::HorizontalAlignment::Right;

//

// btnigual

//

this->btnigual->Font = (gcnew System::Drawing::Font(L"Arial Black", 12, System::Drawing::FontStyle::Bold, System::Drawing::GraphicsUnit::Point,

static\_cast<System::Byte>(0)));

this->btnigual->Location = System::Drawing::Point(47, 69);

this->btnigual->Name = L"btnigual";

this->btnigual->Size = System::Drawing::Size(37, 37);

this->btnigual->TabIndex = 32;

this->btnigual->Text = L"=";

this->btnigual->UseVisualStyleBackColor = true;

this->btnigual->Click += gcnew System::EventHandler(this, &Form1::btnigual\_Click);

//

// btndiv

//

this->btndiv->Font = (gcnew System::Drawing::Font(L"Arial Black", 12, System::Drawing::FontStyle::Bold, System::Drawing::GraphicsUnit::Point,

static\_cast<System::Byte>(0)));

this->btndiv->Location = System::Drawing::Point(90, 69);

this->btndiv->Name = L"btndiv";

this->btndiv->Size = System::Drawing::Size(37, 37);

this->btndiv->TabIndex = 31;

this->btndiv->Text = L"/";

this->btndiv->UseVisualStyleBackColor = true;

this->btndiv->Click += gcnew System::EventHandler(this, &Form1::btndiv\_Click);

//

// btnres

//

this->btnres->Font = (gcnew System::Drawing::Font(L"Arial Black", 12, System::Drawing::FontStyle::Bold, System::Drawing::GraphicsUnit::Point,

static\_cast<System::Byte>(0)));

this->btnres->Location = System::Drawing::Point(47, 112);

this->btnres->Name = L"btnres";

this->btnres->Size = System::Drawing::Size(37, 37);

this->btnres->TabIndex = 30;

this->btnres->Text = L"-";

this->btnres->UseVisualStyleBackColor = true;

this->btnres->Click += gcnew System::EventHandler(this, &Form1::btnres\_Click);

//

// btnsum

//

this->btnsum->Font = (gcnew System::Drawing::Font(L"Arial Black", 12, System::Drawing::FontStyle::Bold, System::Drawing::GraphicsUnit::Point,

static\_cast<System::Byte>(0)));

this->btnsum->Location = System::Drawing::Point(47, 26);

this->btnsum->Name = L"btnsum";

this->btnsum->Size = System::Drawing::Size(37, 37);

this->btnsum->TabIndex = 29;

this->btnsum->Text = L"+";

this->btnsum->UseVisualStyleBackColor = true;

this->btnsum->Click += gcnew System::EventHandler(this, &Form1::btnsum\_Click);

//

// btnmul

//

this->btnmul->Font = (gcnew System::Drawing::Font(L"Arial Black", 12, System::Drawing::FontStyle::Bold, System::Drawing::GraphicsUnit::Point,

static\_cast<System::Byte>(0)));

this->btnmul->Location = System::Drawing::Point(4, 69);

this->btnmul->Name = L"btnmul";

this->btnmul->Size = System::Drawing::Size(37, 37);

this->btnmul->TabIndex = 28;

this->btnmul->Text = L"X";

this->btnmul->UseVisualStyleBackColor = true;

this->btnmul->Click += gcnew System::EventHandler(this, &Form1::btnmul\_Click);

//

// btn1

//

this->btn1->Font = (gcnew System::Drawing::Font(L"Arial Black", 12, System::Drawing::FontStyle::Bold, System::Drawing::GraphicsUnit::Point,

static\_cast<System::Byte>(0)));

this->btn1->Location = System::Drawing::Point(6, 112);

this->btn1->Name = L"btn1";

this->btn1->Size = System::Drawing::Size(37, 37);

this->btn1->TabIndex = 27;

this->btn1->Text = L"1";

this->btn1->UseVisualStyleBackColor = true;

this->btn1->Click += gcnew System::EventHandler(this, &Form1::btn1\_Click);

//

// btn3

//

this->btn3->Font = (gcnew System::Drawing::Font(L"Arial Black", 12, System::Drawing::FontStyle::Bold, System::Drawing::GraphicsUnit::Point,

static\_cast<System::Byte>(0)));

this->btn3->Location = System::Drawing::Point(92, 112);

this->btn3->Name = L"btn3";

this->btn3->Size = System::Drawing::Size(37, 37);

this->btn3->TabIndex = 26;

this->btn3->Text = L"3";

this->btn3->UseVisualStyleBackColor = true;

this->btn3->Click += gcnew System::EventHandler(this, &Form1::btn3\_Click);

//

// btn4

//

this->btn4->Font = (gcnew System::Drawing::Font(L"Arial Black", 12, System::Drawing::FontStyle::Bold, System::Drawing::GraphicsUnit::Point,

static\_cast<System::Byte>(0)));

this->btn4->Location = System::Drawing::Point(6, 69);

this->btn4->Name = L"btn4";

this->btn4->Size = System::Drawing::Size(37, 37);

this->btn4->TabIndex = 25;

this->btn4->Text = L"4";

this->btn4->UseVisualStyleBackColor = true;

this->btn4->Click += gcnew System::EventHandler(this, &Form1::btn4\_Click);

//

// btn7

//

this->btn7->Font = (gcnew System::Drawing::Font(L"Arial Black", 12, System::Drawing::FontStyle::Bold, System::Drawing::GraphicsUnit::Point,

static\_cast<System::Byte>(0)));

this->btn7->Location = System::Drawing::Point(6, 26);

this->btn7->Name = L"btn7";

this->btn7->Size = System::Drawing::Size(37, 37);

this->btn7->TabIndex = 24;

this->btn7->Text = L"7";

this->btn7->UseVisualStyleBackColor = true;

this->btn7->Click += gcnew System::EventHandler(this, &Form1::btn7\_Click);

//

// btn5

//

this->btn5->Font = (gcnew System::Drawing::Font(L"Arial Black", 12, System::Drawing::FontStyle::Bold, System::Drawing::GraphicsUnit::Point,

static\_cast<System::Byte>(0)));

this->btn5->Location = System::Drawing::Point(49, 69);

this->btn5->Name = L"btn5";

this->btn5->Size = System::Drawing::Size(37, 37);

this->btn5->TabIndex = 23;

this->btn5->Text = L"5";

this->btn5->UseVisualStyleBackColor = true;

this->btn5->Click += gcnew System::EventHandler(this, &Form1::btn5\_Click);

//

// btn8

//

this->btn8->Font = (gcnew System::Drawing::Font(L"Arial Black", 12, System::Drawing::FontStyle::Bold, System::Drawing::GraphicsUnit::Point,

static\_cast<System::Byte>(0)));

this->btn8->Location = System::Drawing::Point(49, 26);

this->btn8->Name = L"btn8";

this->btn8->Size = System::Drawing::Size(37, 37);

this->btn8->TabIndex = 22;

this->btn8->Text = L"8";

this->btn8->UseVisualStyleBackColor = true;

this->btn8->Click += gcnew System::EventHandler(this, &Form1::btn8\_Click);

//

// btn9

//

this->btn9->Font = (gcnew System::Drawing::Font(L"Arial Black", 12, System::Drawing::FontStyle::Bold, System::Drawing::GraphicsUnit::Point,

static\_cast<System::Byte>(0)));

this->btn9->Location = System::Drawing::Point(92, 26);

this->btn9->Name = L"btn9";

this->btn9->Size = System::Drawing::Size(37, 37);

this->btn9->TabIndex = 21;

this->btn9->Text = L"9";

this->btn9->UseVisualStyleBackColor = true;

this->btn9->Click += gcnew System::EventHandler(this, &Form1::btn9\_Click);

//

// btn6

//

this->btn6->Font = (gcnew System::Drawing::Font(L"Arial Black", 12, System::Drawing::FontStyle::Bold, System::Drawing::GraphicsUnit::Point,

static\_cast<System::Byte>(0)));

this->btn6->Location = System::Drawing::Point(92, 69);

this->btn6->Name = L"btn6";

this->btn6->Size = System::Drawing::Size(37, 37);

this->btn6->TabIndex = 20;

this->btn6->Text = L"6";

this->btn6->UseVisualStyleBackColor = true;

this->btn6->Click += gcnew System::EventHandler(this, &Form1::btn6\_Click);

//

// btn0

//

this->btn0->Font = (gcnew System::Drawing::Font(L"Arial Black", 12, System::Drawing::FontStyle::Bold, System::Drawing::GraphicsUnit::Point,

static\_cast<System::Byte>(0)));

this->btn0->Location = System::Drawing::Point(49, 155);

this->btn0->Name = L"btn0";

this->btn0->Size = System::Drawing::Size(37, 37);

this->btn0->TabIndex = 19;

this->btn0->Text = L"0";

this->btn0->UseVisualStyleBackColor = true;

this->btn0->Click += gcnew System::EventHandler(this, &Form1::btn0\_Click);

//

// btn2

//

this->btn2->Font = (gcnew System::Drawing::Font(L"Arial Black", 12, System::Drawing::FontStyle::Bold, System::Drawing::GraphicsUnit::Point,

static\_cast<System::Byte>(0)));

this->btn2->Location = System::Drawing::Point(49, 112);

this->btn2->Name = L"btn2";

this->btn2->Size = System::Drawing::Size(37, 37);

this->btn2->TabIndex = 18;

this->btn2->Text = L"2";

this->btn2->UseVisualStyleBackColor = true;

this->btn2->Click += gcnew System::EventHandler(this, &Form1::btn2\_Click);

//

// groupBox1

//

this->groupBox1->BackColor = System::Drawing::Color::Silver;

this->groupBox1->Controls->Add(this->btnigual);

this->groupBox1->Controls->Add(this->btnclear);

this->groupBox1->Controls->Add(this->btnmul);

this->groupBox1->Controls->Add(this->btnsum);

this->groupBox1->Controls->Add(this->btnres);

this->groupBox1->Controls->Add(this->btndiv);

this->groupBox1->Location = System::Drawing::Point(153, 50);

this->groupBox1->Name = L"groupBox1";

this->groupBox1->Size = System::Drawing::Size(134, 197);

this->groupBox1->TabIndex = 35;

this->groupBox1->TabStop = false;

this->groupBox1->Text = L"Operaciones:";

//

// groupBox2

//

this->groupBox2->BackColor = System::Drawing::Color::Silver;

this->groupBox2->Controls->Add(this->btn5);

this->groupBox2->Controls->Add(this->btn2);

this->groupBox2->Controls->Add(this->btn0);

this->groupBox2->Controls->Add(this->btn1);

this->groupBox2->Controls->Add(this->btn6);

this->groupBox2->Controls->Add(this->btn3);

this->groupBox2->Controls->Add(this->btn9);

this->groupBox2->Controls->Add(this->btn4);

this->groupBox2->Controls->Add(this->btn8);

this->groupBox2->Controls->Add(this->btn7);

this->groupBox2->Location = System::Drawing::Point(12, 50);

this->groupBox2->Name = L"groupBox2";

this->groupBox2->Size = System::Drawing::Size(135, 197);

this->groupBox2->TabIndex = 36;

this->groupBox2->TabStop = false;

this->groupBox2->Text = L"Datos:";

//

// Form1

//

this->AutoScaleDimensions = System::Drawing::SizeF(6, 13);

this->AutoScaleMode = System::Windows::Forms::AutoScaleMode::Font;

this->BackColor = System::Drawing::SystemColors::ControlLight;

this->ClientSize = System::Drawing::Size(300, 255);

this->Controls->Add(this->groupBox2);

this->Controls->Add(this->groupBox1);

this->Controls->Add(this->txtdatos);

this->Name = L"Form1";

this->Text = L"Calculadora";

this->groupBox1->ResumeLayout(false);

this->groupBox2->ResumeLayout(false);

this->ResumeLayout(false);

this->PerformLayout();

}

#pragma endregion

//programamos las acciones de cada botón

//boton de igual

private: System::Void btnigual\_Click(System::Object^ sender, System::EventArgs^ e) {

//si las variables a y b son nulas

if(a==""&b=="")

{

//se mostrará un mensaje de advertencia

MessageBox::Show("No ha insertado cantidades");

}

//si la cantidad b no fue ingresada le asignará un cero

else if(b==""){

b="0";

//asignamos a cant1 y cant2 los valores de a y b convertidos a tipo entero

cant1=System::Int32::Parse(a);

cant2=System::Int32::Parse(b);

//salida será igual al valor retornado por el método operación,

//enviamos por parámetros los valores de cant1(cantidad 1), cant2(cantidad 2)

//y numoper(número de operación)

salida=operacion(cant1,cant2,numoper);

//mostramos en le textBox txtdatos la operación

//la variable salida se convierte a string pues es de tipo entero

txtdatos->Text=a+operador+b+"="+salida.ToString();

}

//si no sucede

else{

//asignamos a cant1 y cant2 los valores de a y b convertidos a tipo entero

cant1=System::Int32::Parse(a);

cant2=System::Int32::Parse(b);

//salida será igual al valor retornado por el método operación,

//enviamos por parámetros los valores de cant1(cantidad 1), cant2(cantidad 2)

//y numoper(número de operación)

salida=operacion(cant1,cant2,numoper);

//mostramos en le textBox txtdatos la operación

//la variable salida se convierte a string pues es de tipo entero

txtdatos->Text=a+operador+b+"="+salida.ToString();

}

//}

}

//boton para sumar

private: System::Void btnsum\_Click(System::Object^ sender, System::EventArgs^ e) {

//si la variable a no fue ingresada asignará un cero

if(a=="")

{

a="0";

}

//si no a tomará su mismo valor

else

{

a=a;

}

//numero de cantidad es 1

numcant=1;

//numero de operacion es 1 para indicar suma

numoper=1;

//operador tomará el valor de +

operador="+";

//el textBox txtdatos mostrará el valor de a y el operador arimético

txtdatos->Text=a+operador;

}

private: System::Void btnres\_Click(System::Object^ sender, System::EventArgs^ e) {

//si la variable a no fue ingresada asignará un cero

if(a=="")

{

a="0";

}

//si no a tomará su mismo valor

else

{

a=a;

}

//numero de cantidad es 1

numcant=1;

//numero de operacion es 2 para indicar resta

numoper=2;

//operador tomará el valor de -

operador="-";

//el textBox txtdatos mostrará el valor de a y el operador arimético

txtdatos->Text=a+operador;

}

//boton para multiplicar

private: System::Void btnmul\_Click(System::Object^ sender, System::EventArgs^ e) {

//si la variable a no fue ingresada asignará un cero

if(a=="")

{

a="0";

}

//si no a tomará su mismo valor

else

{

a=a;

}

//numero de cantidad es 1

numcant=1;

//numero de operacion es 3 para indicar multiplicación

numoper=3;

//operador tomará el valor de x

operador="x";

//el textBox txtdatos mostrará el valor de a y el operador arimético

txtdatos->Text=a+operador;

}

//boton para resta

//boton para division

private: System::Void btndiv\_Click(System::Object^ sender, System::EventArgs^ e) {

//si la variable a no fue ingresada asignará un cero

if(a=="")

{

a="0";

}

//si no a tomará su mismo valor

else

{

a=a;

}

//numero de cantidad es 1

numcant=1;

//numero de operacion es 4 para indicar division

numoper=4;

//operador tomará el valor de /

operador="/";

//el textBox txtdatos mostrará el valor de a y el operador arimético

txtdatos->Text=a+operador;

}

private: System::Void btn0\_Click(System::Object^ sender, System::EventArgs^ e) {

//si numero de cantidad es cero

if(numcant==0)

{

//a será iguala a a mas 0

a=a+"0";

//el textbox txtdatos mostrará el valor de a

txtdatos->Text=a;

}

//si no

else

{

//b es igual a b mas 0

b=b+"0";

//el textbox txtdatos mostrará la variable a, el operador y la variable b

txtdatos->Text=a+operador+b;

}

}

private: System::Void btn1\_Click(System::Object^ sender, System::EventArgs^ e) {

//si numero de cantidad es cero

if(numcant==0)

{

//a será iguala a a mas 1

a=a+"1";

//el textbox txtdatos mostrará el valor de a

txtdatos->Text=a;

}

//si no

else

{

//b es igual a b mas 1

b=b+"1";

//el textbox txtdatos mostrará la variable a, el operador y la variable b

txtdatos->Text=a+operador+b;

}

}

private: System::Void btn2\_Click(System::Object^ sender, System::EventArgs^ e) {

//si numero de cantidad es cero

if(numcant==0)

{

//a será iguala a a mas 2

a=a+"2";

//el textbox txtdatos mostrará el valor de a

txtdatos->Text=a;

}

//si no

else

{

//b es igual a b mas 2

b=b+"2";

//el textbox txtdatos mostrará la variable a, el operador y la variable b

txtdatos->Text=a+operador+b;

}

}

private: System::Void btn3\_Click(System::Object^ sender, System::EventArgs^ e) {

//si numero de cantidad es cero

if(numcant==0)

{

//a será iguala a a mas 3

a=a+"3";

//el textbox txtdatos mostrará el valor de a

txtdatos->Text=a;

}

//si no

else

{

//b es igual a b mas 3

b=b+"3";

//el textbox txtdatos mostrará la variable a, el operador y la variable b

txtdatos->Text=a+operador+b;

}

}

private: System::Void btn4\_Click(System::Object^ sender, System::EventArgs^ e) {

//si numero de cantidad es cero

if(numcant==0)

{

//a será iguala a a mas 4

a=a+"4";

//el textbox txtdatos mostrará el valor de a

txtdatos->Text=a;

}

//si no

else

{

//b es igual a b mas 4

b=b+"4";

//el textbox txtdatos mostrará la variable a, el operador y la variable b

txtdatos->Text=a+operador+b;

}

}

private: System::Void btn5\_Click(System::Object^ sender, System::EventArgs^ e) {

//si numero de cantidad es cero

if(numcant==0)

{

//a será iguala a a mas 5

a=a+"5";

//el textbox txtdatos mostrará el valor de a

txtdatos->Text=a;

}

//si no

else

{

//b es igual a b mas 5

b=b+"5";

//el textbox txtdatos mostrará la variable a, el operador y la variable b

txtdatos->Text=a+operador+b;

}

}

private: System::Void btn6\_Click(System::Object^ sender, System::EventArgs^ e) {

//si numero de cantidad es cero

if(numcant==0)

{

//a será iguala a a mas 6

a=a+"6";

//el textbox txtdatos mostrará el valor de a

txtdatos->Text=a;

}

//si no

else

{

//b es igual a b mas 6

b=b+"6";

//el textbox txtdatos mostrará la variable a, el operador y la variable b

txtdatos->Text=a+operador+b;

}

}

private: System::Void btn7\_Click(System::Object^ sender, System::EventArgs^ e) {

//si numero de cantidad es cero

if(numcant==0)

{

//a será iguala a a mas 7

a=a+"7";

//el textbox txtdatos mostrará el valor de a

txtdatos->Text=a;

}

//si no

else

{

//b es igual a b mas 7

b=b+"7";

//el textbox txtdatos mostrará la variable a, el operador y la variable b

txtdatos->Text=a+operador+b;

}

}

private: System::Void btn8\_Click(System::Object^ sender, System::EventArgs^ e) {

//si numero de cantidad es cero

if(numcant==0)

{

//a será iguala a a mas 8

a=a+"8";

//el textbox txtdatos mostrará el valor de a

txtdatos->Text=a;

}

//si no

else

{

//b es igual a b mas 8

b=b+"8";

//el textbox txtdatos mostrará la variable a, el operador y la variable b

txtdatos->Text=a+operador+b;

}

}

private: System::Void btn9\_Click(System::Object^ sender, System::EventArgs^ e) {

//si numero de cantidad es cero

if(numcant==0)

{

//a será iguala a a mas 9

a=a+"9";

//el textbox txtdatos mostrará el valor de a

txtdatos->Text=a;

}

//si no

else

{

//b es igual a b mas 9

b=b+"9";

//el textbox txtdatos mostrará la variable a, el operador y la variable b

txtdatos->Text=a+operador+b;

}

}

private: System::Void btnclear\_Click(System::Object^ sender, System::EventArgs^ e) {

txtdatos->Text="";

numcant=0;

numoper=0;

a="";

b="";

operador="";

}

};

}

