



Universidad Autónoma de Querétaro

Facultad de Informática

EVALUACION UNIDAD 1: MINI PROYECTO

Alumnos: Favio Jardinez Montiel
Profesor: Becerril Hurtado Roberto

5 de marzo de 2023

Índice

1	Introducción	2
2	Explicación del código	3



1. Introducción

Para realizar este proyecto se utilizó lo que es una extensión de visual studio llamada CppCLRWinformsProjekt, la cuál necesitaba de algunas otras cosas mas instaladas, con esta extensión se trabajó en lo que es c++/cli con windowsform, con lo que se pudo realizar la interfaz gráfica para el usuario, además de la funcionalidad con c++ como se mostrara a continuación

The screenshot shows a Windows Forms application window titled "Form1". The interface is for a "Juego de adivinanza de números" (Number Guessing Game). It features a title bar with standard Windows window controls. The main content area has a light gray background. At the top, the title "Juego de adivinanza de números" is displayed in a bold, italicized, black font. Below the title, the instruction "Elige el rango del número a adivinar" (Choose the range of the number to guess) is shown. There are two input fields for the range, preceded by the labels "Entre:" and "Y:". Below these, there is a single input field for the "Número para adivinar:" (Number to guess:). At the bottom, there are two buttons: "Reiniciar" (Restart) on the left and "Adivinar" (Guess) on the right. The word "Pista:" (Hint:) is positioned above the "Adivinar" button.

2. Explicación del código

1. Código 1

```
#pragma once
#include <cstdlib>

namespace CppCLRWinFormsProjekt {

    using namespace System;
    using namespace System::ComponentModel;
    using namespace System::Collections;
    using namespace System::Windows::Forms;
    using namespace System::Data;
    using namespace System::Drawing;

    /// <summary>
    /// Zusammenfassung für Form1
    /// </summary>
    public ref class Form1 : public System::Windows::Forms::Form
    {
    public:
        int num_aleatorio, NumeroAdivinar, contador=0;
        int* ptr_num_aleatorio = nullptr;
        bool adivinar = false;
        Form1(void)
        {
            InitializeComponent();
            //
            //TODO: Konstruktorcode hier hinzufügen.
            //
        }

    protected:
        /// <summary>
        /// Verwendete Ressourcen bereinigen.
        /// </summary>
        ~Form1()
        {
            if (components)
            {
                components->Dispose();
            }
        }
    };
}
```

En este código lo que se puede ver lo que sería la plantilla para nuestro formulario de windows form con c++, donde también podemos destacar la declaración de variables tipo publicas las cuales ocuparemos mas adelante, asi como tambien la declaración del puntero, el cuál de momento apunta a nullptr.

2. Código 2

```
/// </summary>
~Form1()
{
    if (components)
    {
        delete components;
    }
}

private: System::Windows::Forms::TextBox^ TxtNum1;
protected:
private: System::Windows::Forms::Label^ label1;
private: System::Windows::Forms::Label^ label2;
private: System::Windows::Forms::Label^ label3;
private: System::Windows::Forms::Label^ label4;
private: System::Windows::Forms::TextBox^ TxtNum2;
private: System::Windows::Forms::Button^ ButtonAdivinar;
private: System::Windows::Forms::Button^ button1;
private: System::Windows::Forms::Label^ label5;
private: System::Windows::Forms::TextBox^ TxtAdivinar;
private: System::Windows::Forms::Label^ LblPistas;

private: System::Windows::Forms::Label^ LblAdivinar;

private:
    /// <summary>
    /// Erforderliche Designervariable.
    /// </summary>
    System::ComponentModel::Container ^components;

#pragma region Windows Form Designer generated code
    /// <summary>
    /// Erforderliche Methode für die Designerunterstützung.
    /// Der Inhalt der Methode darf nicht mit dem Code-Editor geändert werden.
    /// </summary>
    void InitializeComponent(void)
```

En esta parte del código, igual podemos ver parte de la polántilla que nos pone la extensión cada que ponemos cosas en la vista de diseño en el formulario de windows form, pues podemos ver cómo es que nos pone un private para el windows form, para cada elemento que tenemos en el formulario.

3. Código 3

```
/// </summary>
void InitializeComponent(void)
{
    this->TxtNum1 = (gcnew System::Windows::Forms::TextBox());
    this->label1 = (gcnew System::Windows::Forms::Label());
    this->label2 = (gcnew System::Windows::Forms::Label());
    this->label3 = (gcnew System::Windows::Forms::Label());
    this->label4 = (gcnew System::Windows::Forms::Label());
    this->TxtNum2 = (gcnew System::Windows::Forms::TextBox());
    this->ButtonAdivinar = (gcnew System::Windows::Forms::Button());
    this->button1 = (gcnew System::Windows::Forms::Button());
    this->label5 = (gcnew System::Windows::Forms::Label());
    this->TxtAdivinar = (gcnew System::Windows::Forms::TextBox());
    this->LblPistas = (gcnew System::Windows::Forms::Label());
    this->LblAdivinar = (gcnew System::Windows::Forms::Label());
    this->SuspendLayout();

    //
    // TxtNum1
    //
    this->TxtNum1->Location = System::Drawing::Point(101, 128);
    this->TxtNum1->Name = L"TxtNum1";
    this->TxtNum1->Size = System::Drawing::Size(100, 20);
    this->TxtNum1->TabIndex = 0;

    //
    // label1
    //
    this->label1->AutoSize = true;
    this->label1->Font = (gcnew System::Drawing::Font(L"Tahoma", 14.25F, static_cast<System::Drawing::FontStyle>((System::Draw
        System::Drawing::GraphicsUnit::Point, static_cast<System::Byte>(0))));
    this->label1->Location = System::Drawing::Point(88, 18);
    this->label1->Name = L"label1";
    this->label1->Size = System::Drawing::Size(322, 23);
    this->label1->TabIndex = 1;
    this->label1->Text = L"Juego de adivinanza de números";

    //
    // label2
    //
    this->label2->AutoSize = true;
```

En esta parte del código podemos ver cómo es que se inicializan las cosas que tenemos en nuestro formulario, cómo podemos ver el código después cada comentario nos pone las propiedades que tiene cada uno.

4. Código 4

```
//
this->label2->AutoSize = true;
this->label2->Font = (gcnew System::Drawing::Font(L"Tahoma", 11.25F, System::Drawing::FontStyle::Regular, System::Drawing:
    static_cast<System::Byte>(0)));
this->label2->Location = System::Drawing::Point(12, 70);
this->label2->Name = L"label2";
this->label2->Size = System::Drawing::Size(238, 18);
this->label2->TabIndex = 2;
this->label2->Text = L"Elige el rango del número a adivinar";
//
// label3
//
this->label3->AutoSize = true;
this->label3->Font = (gcnew System::Drawing::Font(L"Tahoma", 11.25F, System::Drawing::FontStyle::Regular, System::Drawing:
    static_cast<System::Byte>(0)));
this->label3->Location = System::Drawing::Point(43, 130);
this->label3->Name = L"label3";
this->label3->Size = System::Drawing::Size(52, 18);
this->label3->TabIndex = 3;
this->label3->Text = L"Entre: ";
//
// label4
//
this->label4->AutoSize = true;
this->label4->Font = (gcnew System::Drawing::Font(L"Tahoma", 11.25F, System::Drawing::FontStyle::Regular, System::Drawing:
    static_cast<System::Byte>(0)));
this->label4->Location = System::Drawing::Point(222, 127);
this->label4->Name = L"label4";
this->label4->Size = System::Drawing::Size(28, 18);
this->label4->TabIndex = 5;
this->label4->Text = L"Y: ";
//
// TxtNum2
//
this->TxtNum2->Location = System::Drawing::Point(264, 128);
this->TxtNum2->Name = L"TxtNum2";
this->TxtNum2->Size = System::Drawing::Size(100, 20);
this->TxtNum2->TabIndex = 4;
```

Esta parte del código es similar a lo de arriba pues son las propiedades de los objetos que tenemos en nuestro formulario de windows.

5. Código 5

```
// ButtonAdivinar
//
this->ButtonAdivinar->Location = System::Drawing::Point(429, 272);
this->ButtonAdivinar->Name = L"ButtonAdivinar";
this->ButtonAdivinar->Size = System::Drawing::Size(97, 52);
this->ButtonAdivinar->TabIndex = 6;
this->ButtonAdivinar->Text = L"Adivinar";
this->ButtonAdivinar->UseVisualStyleBackColor = true;
this->ButtonAdivinar->Click += gcnew System::EventHandler(this, &Form1::ButtonAdivinar_Click);
//
// button1
//
this->button1->Location = System::Drawing::Point(12, 272);
this->button1->Name = L"button1";
this->button1->Size = System::Drawing::Size(97, 52);
this->button1->TabIndex = 7;
this->button1->Text = L"Reiniciar";
this->button1->UseVisualStyleBackColor = true;
this->button1->Click += gcnew System::EventHandler(this, &Form1::button1_Click);
//
// label5
//
this->label5->AutoSize = true;
this->label5->Font = (gcnew System::Drawing::Font(L"Tahoma", 11.25F, System::Drawing::FontStyle::Regular, System::Drawing::
    static_cast<System::Byte>(0)));
this->label5->Location = System::Drawing::Point(61, 199);
this->label5->Name = L"label5";
this->label5->Size = System::Drawing::Size(158, 18);
this->label5->TabIndex = 9;
this->label5->Text = L"Número para adivinar: ";
//
// TxtAdivinar
//
this->TxtAdivinar->Location = System::Drawing::Point(225, 197);
this->TxtAdivinar->Name = L"TxtAdivinar";
this->TxtAdivinar->Size = System::Drawing::Size(100, 20);
this->TxtAdivinar->TabIndex = 8;
//
```

Esta parte del código es similar a lo de arriba pues siguen siendo las propiedades de los objetos que tenemos en nuestro formulario de windows.

6. Código 6

```
this->TxtAdivinar->TabIndex = 8;
//
// LblPistas
//
this->LblPistas->AutoSize = true;
this->LblPistas->Font = (gcnew System::Drawing::Font(L"Tahoma", 11.25F, System::Drawing::FontStyle::Regular, System::Drawing::FontFlags::Default));
this->LblPistas->Location = System::Drawing::Point(186, 257);
this->LblPistas->Name = L"LblPistas";
this->LblPistas->Size = System::Drawing::Size(48, 18);
this->LblPistas->TabIndex = 10;
this->LblPistas->Text = L"Pista: ";
//
// LblAdivinar
//
this->LblAdivinar->AutoSize = true;
this->LblAdivinar->Font = (gcnew System::Drawing::Font(L"Tahoma", 11.25F, System::Drawing::FontStyle::Regular, System::Drawing::FontFlags::Default));
this->LblAdivinar->Location = System::Drawing::Point(129, 288);
this->LblAdivinar->Name = L"LblAdivinar";
this->LblAdivinar->Size = System::Drawing::Size(0, 18);
this->LblAdivinar->TabIndex = 11;
//
// Form1
//
this->AutoScaleDimensions = System::Drawing::SizeF(6, 13);
this->AutoScaleMode = System::Windows::Forms::AutoScaleMode::Font;
this->ClientSize = System::Drawing::Size(548, 364);
this->Controls->Add(this->LblAdivinar);
this->Controls->Add(this->LblPistas);
this->Controls->Add(this->label5);
this->Controls->Add(this->TxtAdivinar);
this->Controls->Add(this->button1);
this->Controls->Add(this->ButtonAdivinar);
this->Controls->Add(this->label4);
this->Controls->Add(this->TxtNum2);
this->Controls->Add(this->label3);
this->Controls->Add(this->label2);
```

Esta parte del código es similar a lo de arriba pues siguen siendo las propiedades de los objetos que tenemos en nuestro formulario de windows. Así como también las propiedades del formulario llamado form1 que es dónde trabajamos.

7. Código 7

```
        this->Controls->Add(this->label3);
        this->Controls->Add(this->label2);
        this->Controls->Add(this->label1);
        this->Controls->Add(this->TxtNum1);
        this->Name = L"Form1";
        this->StartPosition = System::Windows::Forms::FormStartPosition::CenterScreen;
        this->Text = L"Form1";
        this->Load += gcnew System::EventHandler(this, &Form1::Form1_Load);
        this->ResumeLayout(false);
        this->PerformLayout();
    }

#pragma endregion

private: System::Void Form1_Load(System::Object^ sender, System::EventArgs^ e) {
    //instrucciones del juego cuando carga
    MessageBox::Show(L"Para jugar este juego, elige dos numeros que sirvan como el rango para generar un numero aleatorio, RECUERDA");
}

private: System::Void ButtonAdivinar_Click(System::Object^ sender, System::EventArgs^ e) {
    //validacion de datos que no se hayan ingresado
    if (TxtNum1->Text == "" || TxtNum2->Text == "" || TxtAdivinar->Text == "") {
        MessageBox::Show(L"Quedan campos sin llenar, Checa de nuevo");
    }
    //validacion si el numero aleatorio esta vacio, es decir es la primera vez que se pulsa el boton para generar el numero aleatorio
    else if (num_aleatorio == NULL) {
        //conversion de textbox a int
        int Rango1 = System::Int32::Parse(TxtNum1->Text);
        int Rango2 = System::Int32::Parse(TxtNum2->Text);
        int NumeroAdivinar = System::Int32::Parse(TxtAdivinar->Text);
        //validacion de rango valido
        if (Rango1 >= Rango2 || Rango2==Rango1+1 ) {
            MessageBox::Show(L"Ingresa un rango valido para iniciar, siendo el primer numero el menor y el segundo el mayor");
            this->TxtNum1->Text = "";
            this->TxtNum2->Text = "";
            this->TxtNum1->Focus();
        }
        else {
            //realizar los textbox no modificables y generar el numero aleatorio si el rango es valido
        }
    }
}
```

Aquí ya es cuando empezamos con las instrucciones mas específicas, trabajamos con c++, y de primer instancia tenemos el evento de load, el cual se ejecuta cuando se carga el formulario por primera vez. en ese evento ponemos un messagebox el cual nos pone las instrucciones para saber cómo jugar y usar la aplicacion. Después de eso tenemos nuestro evento cuando se le da click al botón de adivinar, el cual como podemos ver en el primer comentario nos valida si ya insertaron los datos necesarios para poder ingresar, de no ser así, se envía un mensaje al usuario y no se permite iniciar. Después tenemos que si una condición que entra si no tenemos un número aleatorio generado ya. En el cual se hace la conversión de lo que tenemos en los textbox a enteros. Después podemos ver la validación que tenemos si el rango que se ingresó para iniciar es válido de no ser así, se manda un mensaje al usuario y se limpian los txt y no se deja iniciar hasta que se ingrese un número válido.

8. Código 8

```
else {
    //realizar los textbox no modificables y generar el numero aleatorio si el rango es valido
    this->TxtNum1->ReadOnly = true;
    this->TxtNum2->ReadOnly = true;
    num_aleatorio = rand() % (Rango1 - Rango2 + 1) + Rango1;
    interior_ptr<int> ptr_num_aleatorio = &num_aleatorio;
    //this->LblAdivinar->Text = "El numero es: " + *ptr_num_aleatorio;
    //verificar si se adivina el numero
    if (NumeroAdivinar == *ptr_num_aleatorio) {
        this->LblAdivinar->Text = "Haz acertado el numero, Felicidades!!";
        contador++;
        this->LblPistas->Text = "Pista: " + contador + " Num ingresado: " + NumeroAdivinar;
        this->TxtAdivinar->ReadOnly = true;
        adivinar = true;
    }
    else if (NumeroAdivinar > *ptr_num_aleatorio) {
        contador++;
        this->LblAdivinar->Text = "¡El numero es menor!";
        this->TxtAdivinar->Text = "";
        this->LblPistas->Text = "Pista: " + contador + " Num ingresado: " + NumeroAdivinar;
        this->TxtAdivinar->Focus();
    }
    else if (NumeroAdivinar < *ptr_num_aleatorio) {
        contador++;
        this->LblAdivinar->Text = "¡El numero es mayor!";
        this->TxtAdivinar->Text = "";
        this->LblPistas->Text = "Pista: " + contador + " Num ingresado: " + NumeroAdivinar;
        this->TxtAdivinar->Focus();
    }
}
}
else
{
    //declarar la variable cada que ingrese el usuario un nuevo intento, así como el puntero a el numero random
    int NumeroAdivinar = System::Int32::Parse(TxtAdivinar->Text);
    interior_ptr<int> ptr_num_aleatorio = &num_aleatorio;
    //verificar si adivina el numero
    if (NumeroAdivinar == *ptr_num_aleatorio) {
```

En caso de que el rango ingresado sea válido se procede a realizar los textbox donde se insertó el rango para que no se pueda modificar. También se genera el número aleatorio así como también se declara el puntero que apunta hacia la variable num-aleatorio. Después se hace una validación para saber si el usuario adivinó el número aleatorio, o es menor o mayor, y en caso de que entre alguna validación y dependiendo el número ingresado para adivinar se cambia la pista que se da al usuario, se borra el número ingresado para insertar otro, se pone el número de pista que es así como el número puesto anteriormente y se aumenta el contador para saber cuántos intentos lleva. y se pone al usuario para que pueda volver a escribir y en caso de acertar se vuelve ineditable los textbox para poder dar reiniciar e iniciar de nuevo. El último else, entra cuándo ya tenemos un número aleatorio generado, y se vuelve a parsear el textbox del número a adivinar a el nuevo número adivinar, y el puntero de la misma manera al numero aleatorio.

9. Código 9

```
if (NumeroAdivinar == *ptr_num_aleatorio) {
    this->LblAdivinar->Text = "Haz acertado el numero, Felicidades!!";
    this->TxtAdivinar->ReadOnly = true;
    this->LblPistas->Text = "Pista: " + contador + " Num ingresado: " + NumeroAdivinar;
    adivinar = true;
}

//si ya tiene mas de 5 intentos (contador inicia en 0) y no ha adivinado el numero te dice cual era
else if (contador >= 4 && adivinar == false) {
    MessageBox::Show("Ya has hecho 5 intentos, si quieres volver a intentarlo da click en reiniciar");
    this->TxtAdivinar->ReadOnly = true;
    this->LblPistas->Text = "Revelacion del numero: ";
    this->LblAdivinar->Text = "El numero era: " + *ptr_num_aleatorio;
}

//aun tiene intentos restantes
else
{
    //verificar si adivina el numero
    if (NumeroAdivinar > *ptr_num_aleatorio) {
        this->LblAdivinar->Text = ";El numero es menor!";
        this->TxtAdivinar->Text = "";
        contador++;
        this->LblPistas->Text = "Pista: " + contador + " Num ingresado: " + NumeroAdivinar;
        this->TxtAdivinar->Focus();
    }
    else if (NumeroAdivinar < *ptr_num_aleatorio) {
        this->LblAdivinar->Text = ";El numero es mayor!";
        this->TxtAdivinar->Text = "";
        contador++;
        this->LblPistas->Text = "Pista: " + contador + " Num ingresado: " + NumeroAdivinar;
        this->TxtAdivinar->Focus();
    }
}
}

private: System::Void button1_Click(System::Object^ sender, System::EventArgs^ e) {
    //reiniciar tanto los txt como los labels y los datos a como estaban a inicio
    this->TxtNum1->ReadOnly = false;
    this->TxtNum2->ReadOnly = false;
}
```

En esta parte podemos ver cómo es que tenemos una validación la cuál si adivina el número hace los cambios necesarios y te menciona que has ganado, si no se cumple se tiene otra validación la cuál entra si ya se hicieron 5 intentos, pues el contador inicio en 0, y si no se ha adivinado te dice cual era el número secreto y y que se ha quedado sin intentos y ya no se podra ingresar un nuevo número. Si no pasa ninguna de esas dos cosas y aún tiene intentos se volverá a validar y dar pistas dependiendo el número que se ingrese. Luego se tiene el último evento el cuál pertenece al button1 que es el de reiniciar.

10. Código 10

```
        contador++;
        this->LblPistas->Text = "Pista: " + contador + " Num ingresado: " + NumeroAdivinar;
        this->TxtAdivinar->Focus();
    }
    else if (NumeroAdivinar < "ptr_num_aleatorio") {
        this->LblAdivinar->Text = ";El numero es mayor!";
        this->TxtAdivinar->Text = "";
        contador++;
        this->LblPistas->Text = "Pista: " + contador + " Num ingresado: " + NumeroAdivinar;
        this->TxtAdivinar->Focus();
    }
}

private: void button1_Click(System::Object^ sender, System::EventArgs^ e) {
    //reiniciar tanto los txt como los labels y los datos a como estaban a inicio
    this->TxtNum1->ReadOnly = false;
    this->TxtNum2->ReadOnly = false;
    this->TxtAdivinar->ReadOnly = false;
    this->TxtNum1->Text = "";
    this->TxtNum2->Text = "";
    this->TxtAdivinar->Text = "";
    this->LblAdivinar->Text = "";
    this->LblPistas->Text = "Pista: ";
    num_aleatorio = NULL;
    NumeroAdivinar = NULL;
    contador = 0;
    adivinar = false;
    ptr_num_aleatorio = nullptr;
}
};
```

Esta parte del evento del boton 1, lo que hace es regresar a editables los textbox, así como ponerlos en blancos, y los label donde se dan las pistas de igual manera así como poner las variables y el puntero a valores que tenían de inicio, como nullo, o igualarlo a 0 y poner el puntero como nullptr.

Referencias

[Microsoft, 2021] Microsoft (2021). .net programming with c++/cli (visual c++). Microsoft Learn.

[Tzunun, 2020] Tzunun, O. (2019-2020). Interfaz grafica con c++ — c++ de oliver tzunun. YouTube video series.

[Microsoft, 2021] [Tzunun, 2020]