

**UNIVERSIDAD TECNOLÓGICA NACIONAL**  
**Facultad Regional Reconquista**

**Programación en Computación**  
Ciclo Lectivo: 2020

Trabajo Práctico N.º 10

**10 – APLICACIONES ESPECIFICAS**

**GRUPO N.º: 12**

**INTEGRANTES:** Fulano, Mengano, Zutano

## Guía Unidad 10: APLICACIONES ESPECIFICAS

### SITUACIONES PROBLEMÁTICAS:

1. Realizar un procedimiento que obtenga la división entera y el resto, utilizando únicamente los operadores suma y resta.
2. Realizar un procedimiento que obtenga el valor en Radianes de un ángulo ingresado en formato Grados, Minutos, Segundos y viceversa.
3. Realizar un procedimiento que permita convertir coordenadas polares (radio, ángulo) en cartesianas (x,y).
4. Realizar una función que permita saber si una fecha ingresada en el formato: dd/mm/aaaa es válida.
5. Realizar una función que permita hallar el valor de  $e^x$ , siendo  $x$  un número real  $e$  y un entero.
- 6, 7 y 8. Realizar una función que permitan hallar el valor de PI mediante una serie matemáticas. (La precisión del cálculo dependerá del número de elementos de la serie (i) que será un parámetro que se debe pasar a la función.)

$$\pi = 4 \cdot \sum_{i=0}^{\infty} \frac{(-1)^i}{2i+1} = 4 * (1 - 1/3 + 1/5 - 1/7 + \dots)$$

$$\pi = \frac{1}{2} \cdot \sum_{i=1}^{\infty} \sqrt{\frac{(24)}{(i)^2}} = \frac{1}{2} \sqrt{24 + \frac{24}{2^2} + \frac{24}{3^2} + \frac{24}{4^2} + \frac{24}{5^2} \dots}$$

$$\pi = 4 \cdot \frac{2}{3} \cdot \frac{4}{3} \cdot \frac{4}{5} \cdot \frac{6}{5} \cdot \frac{6}{7} \cdot \frac{8}{7} \dots$$

9. Implemente una funciones que permitan hallar el valor de  $e^x$  mediante la serie matemática siguiente.

$$e^x = \sum_{i=0}^n \frac{(x)^i}{i!} = 1 + x + x^2/2! + x^3/3! + x^4/4! + \dots$$

10. Implemente una funciones que permitan hallar el valor de **sen(x)** mediante la serie matemática siguiente. (El valor de  $x$  es en Radianes):

$$\text{sen}(x) = \sum_{i=1}^n (-1)^i \frac{x^{2i}}{(2i)!} = - (x^2 / 2!) + (x^4 / 4!) - (x^6 / 6!) + (x^8 / 8!) - \dots$$

Observaciones (para Ej 9 y 10)

El número de elementos de la serie ( $n$ ) será el suficiente para que la diferencia absoluta entre dos valores sucesivos sea menor a  $10^{-4}$

12. Diseñar un algoritmo que permita resolver por el método iterativo de Newton-Raphson la **raíz cuadrada** de un número  $m$ . (El número de elementos de la serie será el suficiente para que la diferencia absoluta entre dos valores sucesivos sea menor a  $10^{-4}$ )

## **ACTIVIDADES:**

Resolver las situaciones problemáticas anteriores, debidamente comentadas y comenzando con:

1. **ANALIZAR** el Problema, *Datos de Entrada, Salida y Auxiliares*, y *¿Que ocurre sí?...*
2. **Diagrama de Flujo** en RAPTOR,
3. Codificación en **VSC# Ventana**. (Windows Form)

## **CONOCIMIENTOS NECESARIOS:**

U\_2) Algoritmos.

U\_3) Tipos de Datos.

U\_4) Diagramación lógica. Diagramas de Flujo.

U\_5) Estructuras Secuenciales. Uso de Asignaciones, Entradas y Salidas de Datos.

U\_6) Estructuras de selección o condicionales, estructuras de selección múltiple. Uso de condicionales para la formulación de algoritmos.

U\_7) Estructuras repetitivas: HACER-PARA, HACER-MIENTRAS, REPETIR-HASTA.

U\_8) Vectores. Utilización del tipo de dato vector.

U\_9) Matrices. Índices. Recorridos por filas y columnas. Casos de utilización de matrices.

U\_10) Introducción al cálculo numérico. Algoritmos de métodos numéricos elementales.

## **OPERATORIA:**

Varia con cada situación problemática presentada, pero se resuelve en base a los conocimientos necesarios considerados.

## **RECOMENDACIÓN:**

Para proceder a la realización de los TPs de esta Guía, es recomendable releer la siguiente bibliografía para un repaso de los conceptos teóricos involucrados en resolución de estos problemas:

**Diseño\_de\_Algoritmos.pdf,**

**Ejercicio 12.** Diseñar un algoritmo que permita resolver por el método iterativo de Newton-Raphson la **raíz<sup>2</sup>** de un número **m**. (El número de elementos de la serie será el suficiente para que la diferencia absoluta entre dos valores sucesivos sea menor a  $10^{-4}$ )

### 1) Análisis del Problema:

El problema nos pide como Datos iniciales cargar un número (**m**), del cual queremos obtener su raíz cuadrada, y obtener la misma por métodos numéricos, aplicando el método iterativo de *Newton-Raphson* y con una precisión (o Error menor) de 4 dígitos decimales. Buscando información en la Web encontramos:

### Raíz Cuadrada de un Numero Utilizando método numérico.

$$x = \sqrt{m}$$

$$x^2 = m$$

$$x^2 - m = 0$$

$$f(x) = x^2 - m$$

$$f'(x) = 2x$$

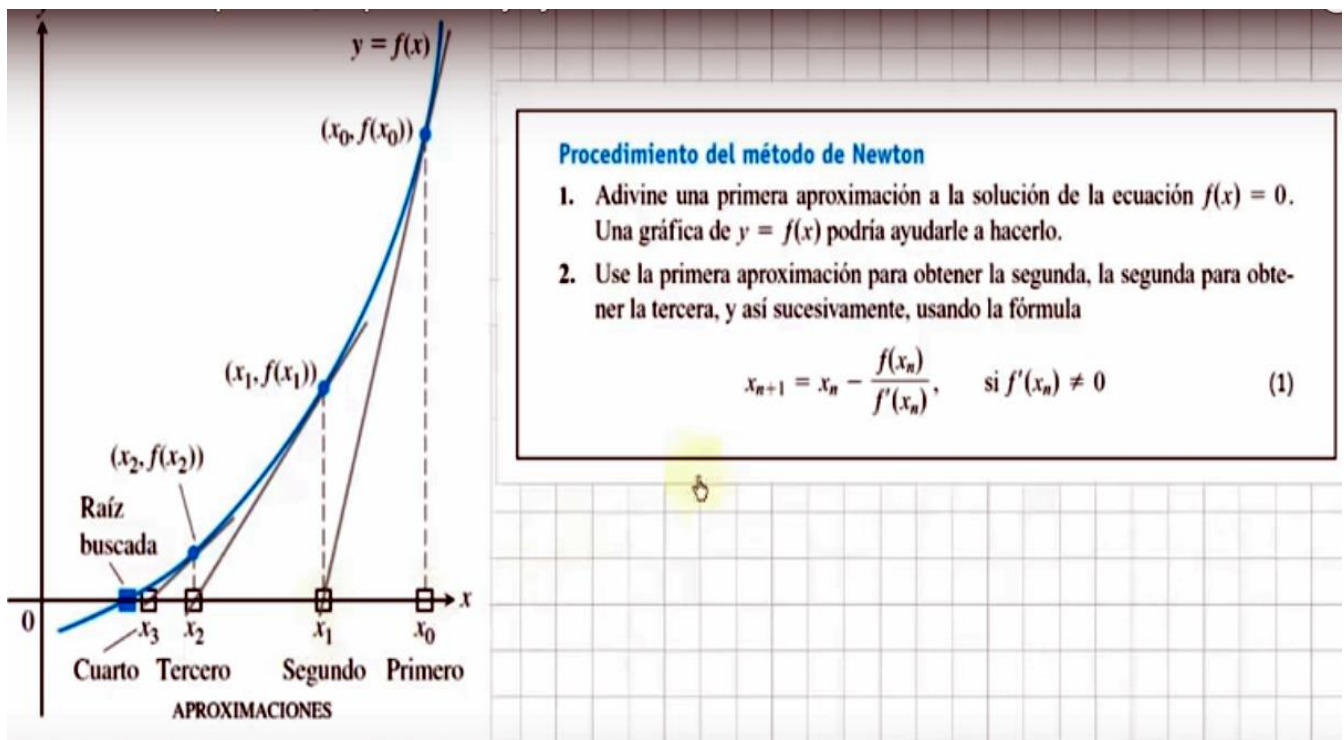
La raíz cuadrada de un número **m** es un número **x** tal que el cuadrado de **x** es igual al valor de **m**

**Definición de la raíz cuadrada de un número, la función y su primera derivada.**

Si pasamos el valor de **m** al lado izquierdo de la ecuación la podemos transformar en una función, donde el **valor** de la **raíz** de la **función** es el **valor de la raíz cuadrada**.

Sí utilizamos el método de **Newton-Raphson** podemos conseguir el valor de la raíz de la función, que correspondería a la raíz cuadrada del número **m**. El procedimiento se puede generalizar para raíces cubicas, cuartas, quintas, sextas, etc....

*Método de Newton-Raphson para hallar una aproximación a la raíz de una función:*



$$x_{n+1} = x_n - \frac{f(x)}{f'(x)}$$

Una aproximación a la raíz de una ecuación de acuerdo con el método de **Newton-Raphson** sería:

$$x_{n+1} = \frac{1}{2} \left[ x_n + \frac{m}{x_n} \right]$$

Sustituyendo el valor de  $f(x)$  y  $f'(x)$

**Aplicación del método Newton-Raphson a la función  $f(x) = x^2 - m$**

Por ejemplo, si queremos hallar la raíz cuadrada de 50, o sea  $m=50$  debemos hallar una raíz de la ecuación:

$$x^2 - m = 0 \quad x^2 - 50 = 0 \quad x_{n+1} = \frac{1}{2} \left[ x_n + \frac{50}{x_n} \right]$$

Una primera aproximación es 7 porque  $7^2 = 49$ , así que:  $x_0 = 7$  y con la fórmula obtenemos  $x_1$ :

$$x_1 = \frac{1}{2} \left[ 7 + \frac{50}{7} \right] = 7.071429$$

Primeras tres iteraciones para hallar la raíz cuadrada de 50.

$$x_2 = \frac{1}{2} \left[ 7.071429 + \frac{50}{7.071429} \right] = 7.071068$$

Como se ve el valor de la raíz converge muy rápidamente a una solución.

$$x_3 = \frac{1}{2} \left[ 7.071068 + \frac{50}{7.071068} \right] = 7.071068$$

Este método que aplicamos a la RAIZ Cuadrada va también para cualquier Raíz. (3, 4, 5, ... 8, 9...)

**DATOS de ENTRADA:**

Número Real: **Num\_m** ; // El número del cual queremos obtener la raíz cuadrada

**DATOS de Proceso o Cálculo:**

Número Real: **Error** =  $10^{-4}$ , **Num\_Xn** = 1, **Num\_Xm**;

// Error permitido y las variables reales para almacenar los valores de las iteraciones.

**DATOS de SALIDA:**

Número Real: **Num\_Xm**;

// El valor de la raíz, cuando:  $| \text{Num}_Xm - \text{Num}_Xn | < \text{Error}$

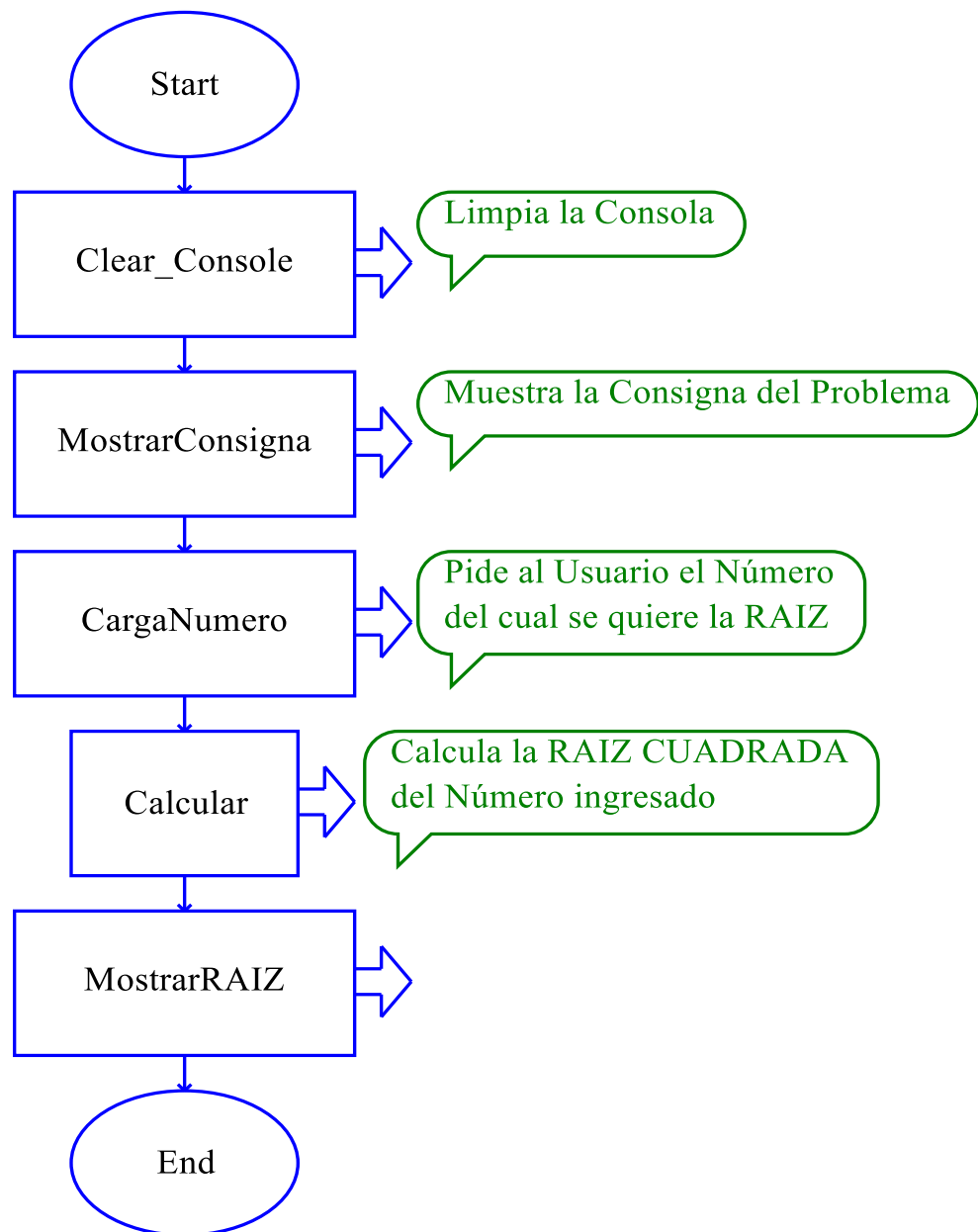
**DATOS Auxiliares:**

De Texto: **Mensaje** ;

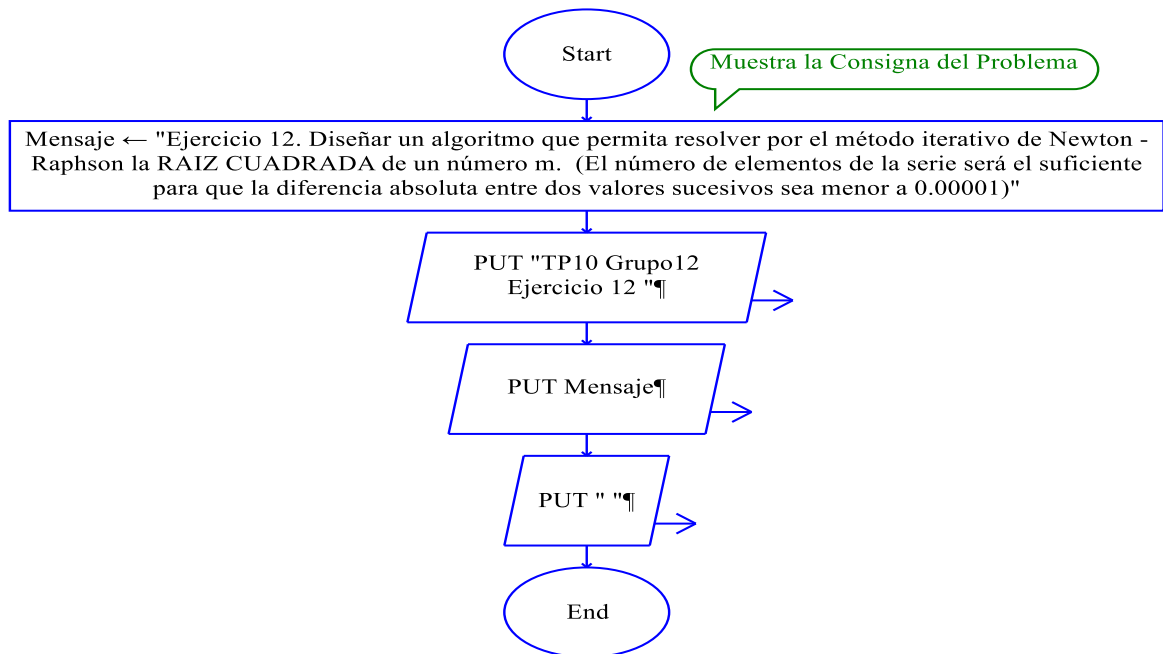
## 2) Diagrama de Flujo en RAPTOR,

Al *Diagrama de Flujo en RAPTOR*, lo construiremos en base a los lineamientos surgidos del ANALISIS del problema. Para concentrarnos en el Manejo del problema, hemos dejado de lado la presentación en Ventana Gráfica y solo usamos la salida de Consola. Emplearemos la técnica de Refinamiento Progresivo con el uso de Módulos. Los Diagramas están auto explicados a través de sus comentarios.

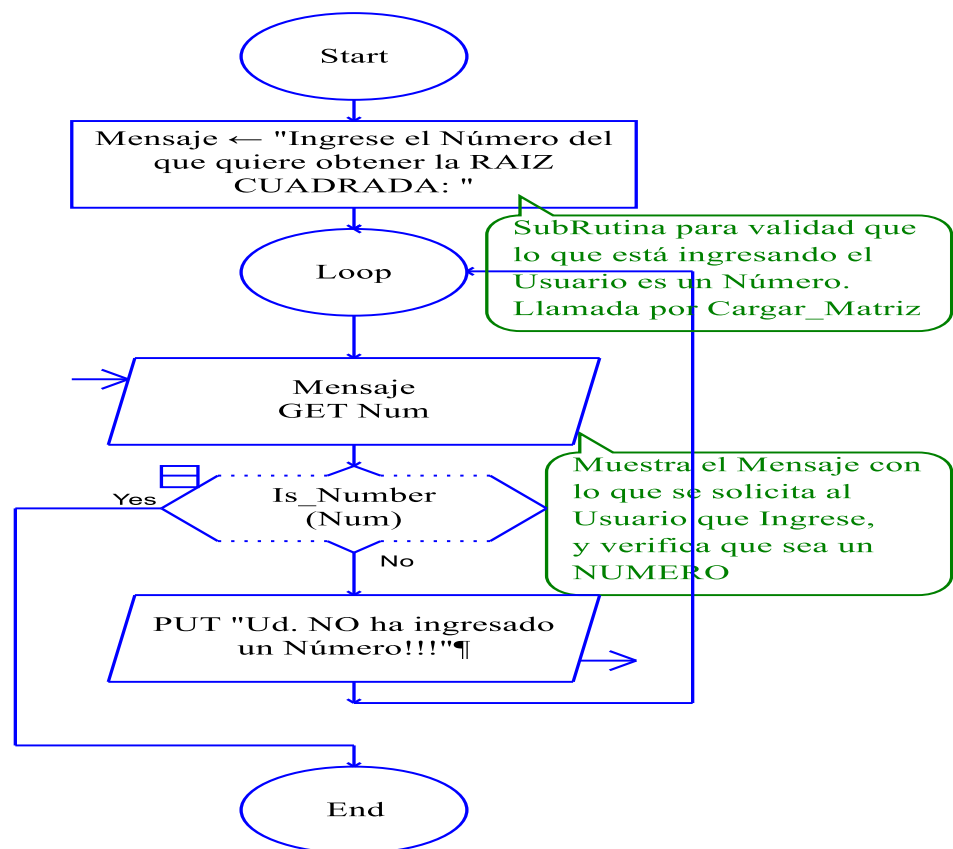
Programa “main”



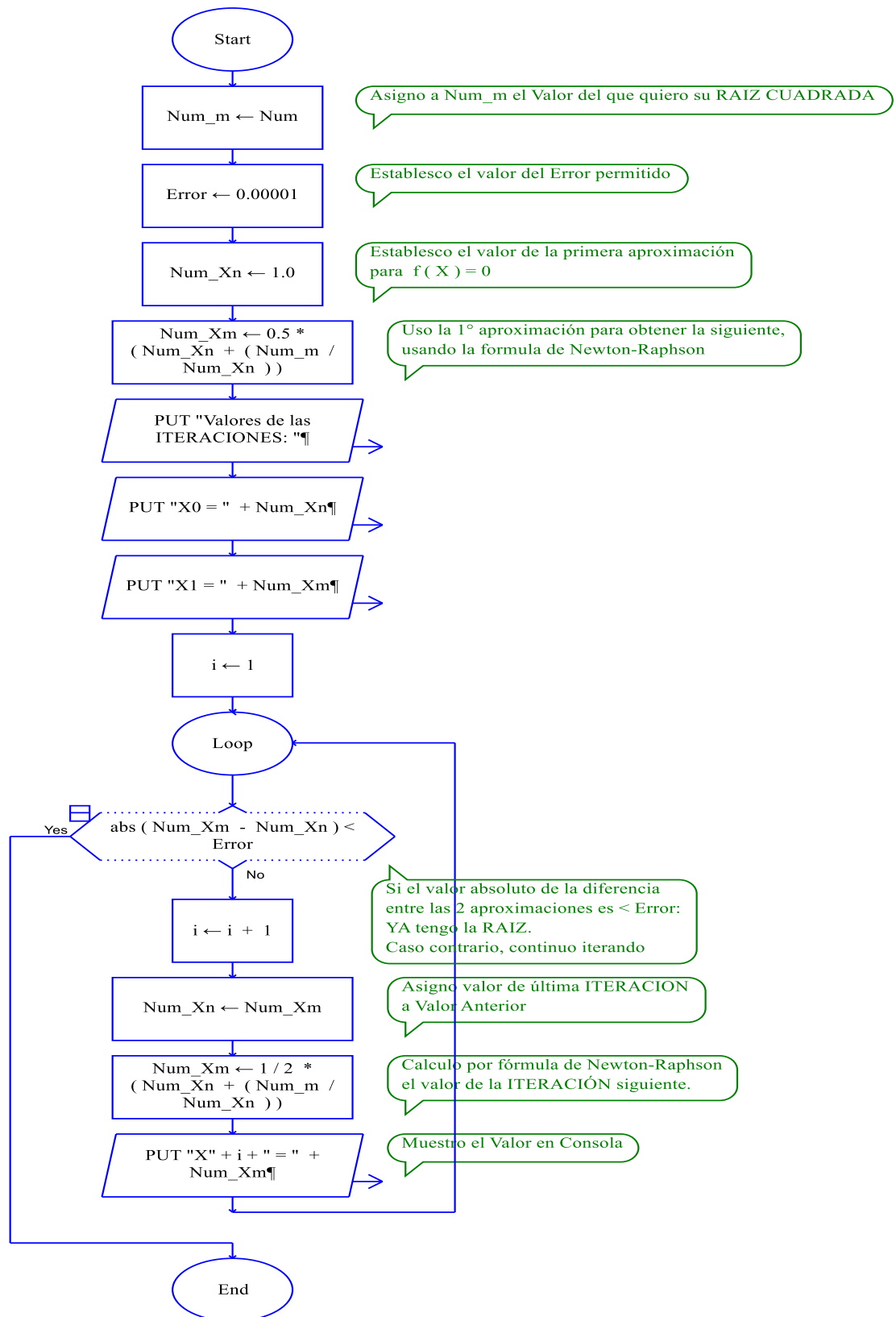
## Sub Programa “MostrarConsigna”



## Sub Programa “CargaNumero”

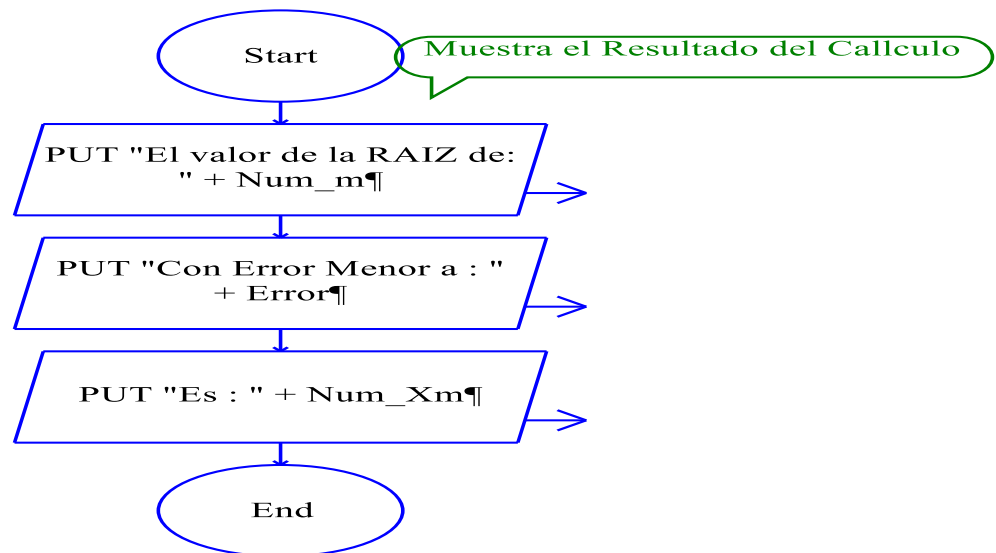


## Sub Programa “Calcular”





## Sub Programa “MostrarRAIZ”



Al EJECUTAR el *D.F. de RAPTOR*, vemos:

TP10 Grupo12 Ejercicio 12  
Ejercicio 12. Diseñar un algoritmo que permita resolver por el método iterativo de Newton-Raphson la RAIZ CUADRADA de un número m. (El número de elementos de la serie será el suficiente para que la diferencia absoluta entre dos valores sucesivos sea menor a 0.00001)

Y la salida por Consola:

```

MasterConsole
Font  Font Size  Edit  Help
TP10 Grupo12 Ejercicio 12
Ejercicio 12. Diseñar un algoritmo que permita
resolver por el método iterativo de Newton-
Raphson la RAIZ CUADRADA de un número m.
(El número de elementos de la serie será el
suficiente para que la diferencia absoluta entre
dos valores sucesivos sea menor a 0.00001)

Valores de las ITERACIONES:
X0 = 1
X1 = 1.5000
X2 = 1.4167
X3 = 1.4142
X4 = 1.4142
El valor de la RAIZ de: 2
Con Error Menor a : 0.000010000
Es : 1.4142
---Run complete. 54 symbols evaluated.---

```

### 3) Codificación en C# (Windows From).

Realizaremos el programa en C# Ventana como una aplicación nueva, a la cual le hemos agregado un plus, ya que calcula Raíces Cuadradas y **Cúbicas**.

Como siempre que debemos programa con Interfaz Gráfica de Usuario (GUI), es conveniente comenzar diseñando ésta.

1. Esta GUI es muy simple, comienza con un GroupBox que contiene una Label con la Consigna del Problema que resuelve.
2. Continúa con un Área para la Carga de Datos formada por un NumericUpDown que permite hasta tres valores decimales, y No permite valores negativos, para cargar el Número del cual queremos obtener su Raíz cuadrada y cúbica. En esta área hemos agregado el botón Calcular
3. Continúa con otro GroupBox que contiene como salida de Datos los valores final calculados para las raíces y los respectivos ComboBox que muestras las Iteraciones realizadas para llegar a los valores finales.
4. Mas abajo tenemos el Área con los Botones: Limpiar y Salir.

**Raices por Método Iterativo de Newton-Raphson**

**Consigna del Problema**

12. Diseñar un algoritmo que permita resolver por el método iterativo de Newton-Raphson la raíz cuadrada de un número  $m$ . El número de elementos de la serie será el suficiente para que la diferencia absoluta entre dos valores sucesivos sea menor a  $1E-10$ .

Ingrese el Número cuyas Raices quiere obtener:

**Salida de Datos**

Raiz Cuadrada =

Raiz Cúbica =

Ventana en tiempo de Diseño y Ventana en tiempo de Ejecución

**Raices por Método Iterativo de Newton-Raphson**

**Consigna del Problema**

12. Diseñar un algoritmo que permita resolver por el método iterativo de Newton-Raphson la raíz cuadrada de un número  $m$ . El número de elementos de la serie será el suficiente para que la diferencia absoluta entre dos valores sucesivos sea menor a  $1E-10$ .

Ingrese el Número cuyas Raices quiere obtener:

**Salida de Datos**

Raiz Cuadrada = 1,41421356237309

Raiz Cúbica = 1,25992104989487

**Ver ITERACIONES**

X[i]	Valor
X[0]	1,0000000000
X[1]	1,5000000000
X[2]	1,4166666667
X[3]	1,4142156863
X[4]	1,4142135624
X[5]	1,4142135624

X[i]	Valor
X[0]	1,0000000000
X[1]	1,3333333333
X[2]	1,2638888889
X[3]	1,2599334934
X[4]	1,2599210500
X[5]	1,2599210499
X[6]	1,2599210499

Para el cálculo de la Raíz Cúbica hemos planteado la función  $f(X) = x^3 - m = 0$ , y  $f'(X) = 3.x^2$

Luego por método de Newton obtenemos:  $X_m = X_n - f(X) / f'(X) = X_n - (X_n^3 - m) / (3 * X_n^2) = (1/3) * (2 * X_n + (m / X_n^2))$ , que es la fórmula aplicada para la Raíz cúbica.

Al finalizar el Cálculo, se deshabilita el botón Calcular y en este punto, los únicos botones habilitados serán los de LIMPIAR para reiniciar el cálculo y la carga y SALIR para finalizar el Programa.

## CODIGO del Programa

Veamos ahora el código de esta aplicación que maneja la entrada del dato numérico a través de un NumericUpDown que permite 3 decimales, valor inicial de uno y final de 10000.

El código está debidamente comentado, y aquí lo presento:

```

16      {
17          InitializeComponent();
18      }
19
20      // Inicializo variables Globales
21      double Num_m, Num_Xn, Num_Xm, Error;
22      Int16 IntCont = 1;
23
24      private void BtnLimpiar_Click(object sender, EventArgs e)
25      {
26          CbxRaizCuadrada.Items.Clear(); // Limpio los valores del ComboBox
27          CbxRaizCuadrada.Text = "Sin Datos"; // Cambio el Título del ComboBox
28          Nud_Num.Value = 2; //Repongo valores iniciales
29          LblSolucion1.Text = "Raiz Cuadrada = ";
30          CbxRaizCubica.Items.Clear(); // Limpio los valores del ComboBox
31          CbxRaizCubica.Text = "Sin Datos"; // Cambio el Título del ComboBox
32          Nud_Num.Value = 2; //Repongo valores iniciales
33          LblSolucion2.Text = "Raiz Cúbica = ";
34          IntCont = 1;
35
36          // Además, controlo el estado de los Botones
37          BtnCalcular.Enabled = true;
38          BtnLimpiar.Enabled = false;
39      }
40
41      private void BtnSalir_Click(object sender, EventArgs e)
42      {

```

Código definición de variables Globales y del Módulo Limpiar.

El Módulo del botón Salir NO tiene novedades, pero lo presento igual.

```

40      {
41      private void BtnSalir_Click(object sender, EventArgs e)
42      {
43          DialogResult DrOpcion; // Declaro la Variable del tipo Resultado de la Opción de un MessageBox
44          DrOpcion = MessageBox.Show("¿Realmente quiere salir de la aplicación?", "Salir del Programa",
45                                     MessageBoxButtons.YesNo, MessageBoxIcon.Question);
46
47          if (DrOpcion == DialogResult.Yes) //Evaluo la Respuesta, Si es SI, muestro mensaje de Agradecimiento
48          {
49              MessageBox.Show("Gracias por usar la aplicación", "Salir del Programa",
50                              MessageBoxButtons.OK, MessageBoxIcon.Asterisk);
51              Close(); // y salgo de la Aplicación
52          }
53      }
54
55      private void BtnCalcular_Click(object sender, EventArgs e)
56      {

```

Las novedades están en el botón Calcular, en el que he repetido el mismo proceso dos veces; una para el cálculo de la RAIZ Cuadrada, y otro para el de la RAIZ Cúbica.

### 1ra. Parte: Calculo de la Raíz Cuadrada:

```

54
55 private void BtnCalcular_Click(object sender, EventArgs e)
56 {
57     // Inicializo variables para el cálculo de la RAIZ_CUADRADA
58     string StrMensaje;
59     Error = 1E-10;    IntCont = 1;    // Inicializo el contador de ITERACIONES
60     Num_m = Convert.ToDouble(Nud_Num.Value);    Num_Xn = 1; // valor inicial primera ITERACIÓN
61     Num_Xm = (1.0/2) * (Num_Xn + (Num_m / Num_Xn));    // Calculo 2da. Iteración
62
63     // Muestra Linea cabecera y el Valor de las ITERACIONES en el ComboBox
64     CbxRaizCuadrada.Items.Add("X[ i ].PadRight(8) + "Valor".PadRight(15)); //Linea de Título
65     CbxRaizCuadrada.Items.Add("X[0].PadRight(7) + String.Format("{0:#0.0000000000}", Num_Xn));
66     CbxRaizCuadrada.Items.Add("X[1].PadRight(7) + String.Format("{0:#0.0000000000}", Num_Xm));
67
68     while (!(Math.Abs(Num_Xm - Num_Xn) < Error))    // Inicio el CICLO para Raiz CUADRADA
69     {
70         IntCont++; // Incremento contador de Iteraciones
71         StrMensaje = "X[" + Convert.ToString(IntCont) + "]";
72         Num_Xn = Num_Xm;    // Paso nueva Iteración a Iteración anterior
73         Num_Xm = 0.5 * (Num_Xn + (Num_m / Num_Xn)); // Calculo Nuevas Iteraciones
74         // En el ciclo Sigo agregando el Valor de las ITERACIONES en el ComboBox
75         CbxRaizCuadrada.Items.Add(StrMensaje.PadRight(7) + String.Format("{0:#0.0000000000}", Num_Xm));
76     }
77
78     LblSolucion1.Text = LblSolucion1.Text + Num_Xm; // Muestro el valor de la Raiz Cuadrada
79     // FINALIZO el cálculo de la RAIZ_CUADRADA
80
81
82     // Inicio el CALCULO para Raiz CUBICA
83     IntCont = 1;    // Inicializo el contador de ITERACIONES

```

**2da. Parte: Calculo de la Raíz Cúbica:**

```

79 // FINALIZO el cálculo de la RAIZ_CUADRADA
80
81
82 // Inicio el CALCULO para Raiz CUBICA
83 IntCont = 1; // Inicializo el contador de ITERACIONES
84 Num_Xn = 1; // valor inicial primera ITERACIÓN
85 Num_Xm = (1.0/3.0) * (2 * Num_Xn + (Num_m / (Num_Xn * Num_Xn))); //Calculo Nueva ITERACION
86
87 // Muestra Linea cabecera y el Valor de las ITERACIONES en el ComboBox
88 CbxRaizCubica.Items.Add("X[ i ].PadRight(8) + "Valor".PadRight(15)); //Linea de Título
89 CbxRaizCubica.Items.Add("X[0].PadRight(7) + String.Format("{0:#0.0000000000}", Num_Xn));
90 CbxRaizCubica.Items.Add("X[1].PadRight(7) + String.Format("{0:#0.0000000000}", Num_Xm));
91
92 while (!(Math.Abs(Num_Xm - Num_Xn) < Error))
93 {
94     IntCont++;
95     StrMensaje = "X[" + Convert.ToString(IntCont) + "]";
96     Num_Xn = Num_Xm;
97     Num_Xm = (1.0 / 3) * (2 * Num_Xn + (Num_m / (Num_Xn * Num_Xn))); //Calculo Nueva ITERACION
98     // En el ciclo Sigue agregando el Valor de las ITERACIONES en el ComboBox
99     CbxRaizCubica.Items.Add(StrMensaje.PadRight(7) + String.Format("{0:#0.0000000000}", Num_Xm));
100 }
101
102 LblSolucion2.Text = LblSolucion2.Text + Num_Xm; // Muestro el valor de la Raiz Cuadrada
103
104 CbxRaizCuadrada.Text = "Ver ITERACIONES"; // Cambio el Título del ComboBox
105 CbxRaizCubica.Text = "Ver ITERACIONES"; // Cambio el Título del ComboBox
106
107 // Además, controlo el estado de los Botones
108 BtnCalcular.Enabled = false;
109 BtnLimpiar.Enabled = true;
110 }
111 }

```

**Experimentar con el programa calculando raíces 4, 5, 6, 7, etc.**

**Fin del Documento.**