

**Universidad de Costa Rica**

**Escuela de Ciencias de la Computación e Informática**

**Ensambladores y Microprocesadores**

**Profesor:  
Carlos Vargas**

**Proyecto: Calculadora Científica**

**Integrantes:  
Katherine Hernández Benavides B63318  
David Retana Fernández B15379  
Fabián Rojas Masís B66236  
Adrián Vargas Martínez B57576**

**Grupo 01**

**I-2018**

## **INTRODUCCIÓN**

---

En el siguiente reporte se podrá observar el proceso con el cual se pretende mostrar una solución al proyecto de programación en lenguaje ensamblador.

El principal objetivo es desarrollar una “calculadora científica” con las funciones básicas de cualquier calculadora y algunas otras funciones especiales que se detallarán más adelante. Dicha calculadora deberá mostrar una interfaz gráfica amigable con el usuario.

Para esto, será necesario llevar a cabo una investigación sobre el desarrollo de programación híbrida, ya que este aspecto será de gran ayuda para llevar a cabo el proyecto. Otro de los puntos importantes a recalcar, es el uso de Visual Studio, esta será el Entorno de Desarrollo Integrado que se utilizará.

Dentro de las principales características de esta calculadora, está que la misma debe trabajar con números de punto flotante, por ello se dispondrá de código y algoritmos utilizados anteriormente en la solución de la tarea programada #1.

## **NOTA TEÓRICA**

---

Ya que esta herramienta será parte importante en el desarrollo de este proyecto, se debe entender el significado de “programación híbrida”.

Como se sabe, algunas de las ventajas de programar en lenguaje ensamblador son que este lenguaje brinda al programador mayor control sobre el hardware de la máquina en la que está trabajando, le permite controlar exactamente las instrucciones que se van a ejecutar, por lo que suena ideal para utilizar en casos críticos donde se necesite una rápida ejecución de instrucciones y menos uso de memoria en ejecutables.

Por otro lado, a la hora de programar en este lenguaje, se debe tener cuidado, ya que se está modificando y haciendo uso recursos de manera inmediata. Además, de que al ser un lenguaje de bajo nivel, llevar a cabo procesos requiere la programación de más instrucciones que si se estuviera programando en un lenguaje de alto nivel, lo cual conlleva a un tiempo de programación muy lento y vuelve complicado la programación de grandes códigos o procesos.

Es aquí donde entra en juego el concepto de programación híbrida, mediante el cual se puede escribir la mayor parte del programa en lenguaje de alto nivel, esto con el fin de ahorrar instrucciones, programar de manera más rápida, más comprensible y se puede programar en lenguaje ensamblador la parte del código más crítica o que necesita más detalle.

Otro aspecto importante e indispensable para el desarrollo de este proyecto es la implementación del ciclo de ensamblado-enlazado-ejecución, el cual está seccionado en 4 pasos:

Paso 1: Se utiliza un editor de texto para crear el archivo conocido como código fuente.

Paso 2: un traductor del lenguaje ensamblador lee el archivo de código fuente y produce un archivo con la traducción del programa a lenguaje de máquina. Si ocurriera un error se debe regresar al paso 1 y corregirlo.

Paso 3: Se verifica si el archivo creado en el paso dos contiene funciones de alguna biblioteca y lo adjunta al código para crear el ejecutable

Paso 4: El sistema operativo lee el archivo ejecutable y lo carga en memoria para que este empiece a ejecutarse.

## **DESCRIPCIÓN DEL PROBLEMA**

---

Como se mencionó anteriormente, se requiere desarrollar mediante programación híbrida (lenguaje ensamblador y c++), “una calculadora”, la cual debe operar con números racionales.

Además de las funciones básicas que presenta cualquier calculadora, tales como suma, resta, multiplicación y división, se solicita el desarrollo de algunas otras funciones no tan convencionales. Por lo que se implementarán las siguientes funciones:

- Potencia
- Raíz
- Factorial
- Logaritmo

Es importante recalcar que el código o algoritmo de al menos 2 de estas funciones se desarrollará en lenguaje ensamblador.

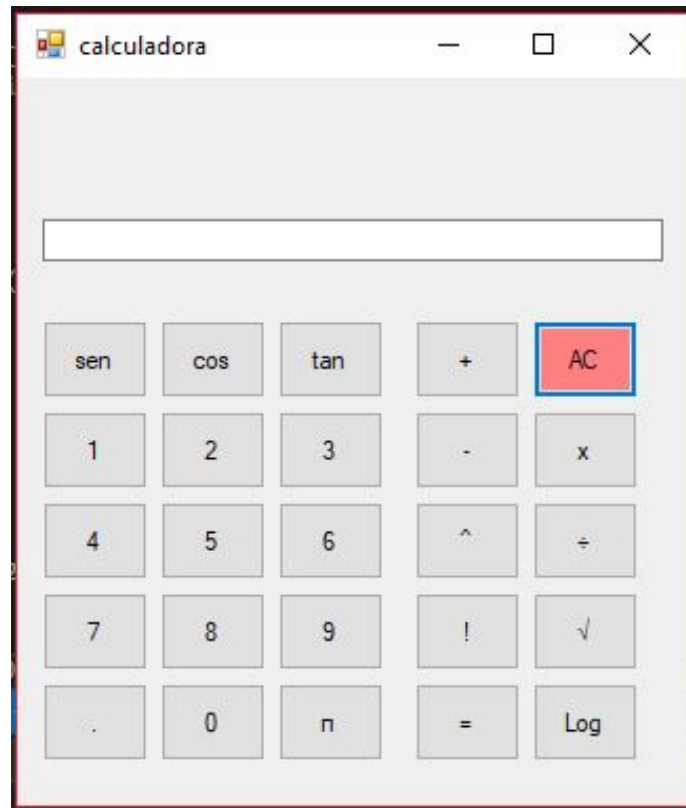
Se debe realizar la interfaz de la calculadora, utilizando lenguaje c++, con Visual Studio como IDE.

## **DESCRIPCIÓN DE LA SOLUCIÓN**

---

Para la implementación de la calculadora se seguirá el ciclo de ensamblado-enlazado-ejecución, basado en los cuatro pasos descritos de este ciclo y en lo que respecta a la interfaz gráfica de la calculadora, se implementará una similar a la que se observa a continuación, basándose en el lenguaje de programación C++, que contará con un recuadro en donde se ingresa la expresión a operar, es decir, la

pantalla de la calculadora. Además, contará con los 10 botones que contienen los dígitos del 0 al 9 junto con los botones que permitirán seleccionar alguna de las operaciones posibles que se mencionaron anteriormente en la descripción del problema.



Por otra parte, en la implementación de las funciones que tendrá la calculadora, se desarrollarán 2 de las 7 funciones, como mínimo, basándose en lenguaje ensamblador (potencia y factorial) esperando mejor eficiencia al hacer los cálculos cuando se decida utilizar dicha función en la calculadora. Si fuera necesario las demás operaciones serán implementadas en el lenguaje C++ .

Para ejecutar cualquier función de esta calculadora se implementará una secuencia o un algoritmo similar en cada caso, primeramente se lee la expresión que el usuario digita en la calculadora y se analiza para determinar la prioridad en los símbolos o caracteres digitados, de esta manera se evaluará de manera correcta la expresión matemática, una vez establecido el orden de prioridad, se hace el llamado al método correspondiente y así sucesivamente hasta obtener el resultado final que será mostrado en la pantalla de la calculadora.

El programa permitirá al usuario reiniciar la calculadora o “limpiarla” por medio del botón <AC> para ejecutar alguna otra operación si así lo desea. Una vez presionado el botón mencionado, el programa volverá a ejecutar el algoritmo descrito

anteriormente; lectura de datos, análisis de expresión, definición de prioridad, ejecución de operaciones, devolución de resultado.

## **ESTRUCTURA DEL PROGRAMA**

---

En lo que respecta a constantes y variables, se utilizaron variables locales en cada uno de los métodos de las funciones que se implementaron, para poder retornar el valor calculado por cada función. A esa variable se le dio el nombre de “rel” (abreviatura de la palabra “resultado”) en todos los métodos. Además, en los métodos usados para la interpretación de la expresión ingresada se usan pilas, ambas llamadas “pila” (son usadas en métodos diferentes ) una de tipo “char” y otra de tipo “double”. No se usan variables globales. La lista de métodos utilizados se adjunta a continuación (los primeros 3 métodos son usados para la interpretación de la expresión):

- `int prioridad(char op);`
- `string convertir(std::string t);`
- `double evaluar(std::string p);`
- `double suma(const double x, const double y);`
- `double resta(const double x, const double y);`
- `double multiplicacion(const double x, const double y);`
- `double division(const double x, const double y);`
- `double potencia(const double x, const int y);`
- `double raiz(const double x);`
- `double factorial(const int x);`
- `double seno(const double x);`
- `double coseno(const double x);`
- `double tangente(const double x);`
- `double logaritmo(const int x);`

## **ANÁLISIS DE LOS RESULTADOS**

---

En el desarrollo del proyecto, se utilizó el proceso que se mencionó en la descripción de la solución (editar, enlazar y ejecutar) el cual permitió, que al finalizar el proyecto, se logró conseguir los resultados esperados, propuestos al inicio del trabajo. Inclusive, se han incluido a la calculadora más funciones de las que se plantearon en la descripción de problema, tales como seno, coseno, tangente y la posibilidad de indicar la prioridad de las funciones con el uso de paréntesis. Además de lo anterior, se ha conseguido implementar en lenguaje ensamblador 9 de las 11 funciones que tiene la calculadora, donde la mayoría de las funciones soportan números reales, exceptuando el logaritmo, factorial y el exponente utilizado por la función “potencia”.

Cabe recalcar la importancia de la programación híbrida en este proyecto, ya que permitió que el desarrollo fuera más amigable, por ejemplo procesos tales como la lectura de datos o la interfaz gráfica de la calculadora que son un poco más complejos, era ideal implementarlos en C++ por facilidad, legibilidad del código o simplemente para depuración y corrección de errores. Por otro lado la mayoría de operaciones matemáticas implementadas se hicieron en lenguaje ensamblador, ya que su desarrollo no era complicado.

Durante el desarrollo del proyecto se comprobó la funcionalidad de cada operación por separado y en conjunto con otras, primeramente se comprobó con números enteros para verificar su correcta ejecución y luego se implementó para números reales en las operaciones en las que se permite su uso.

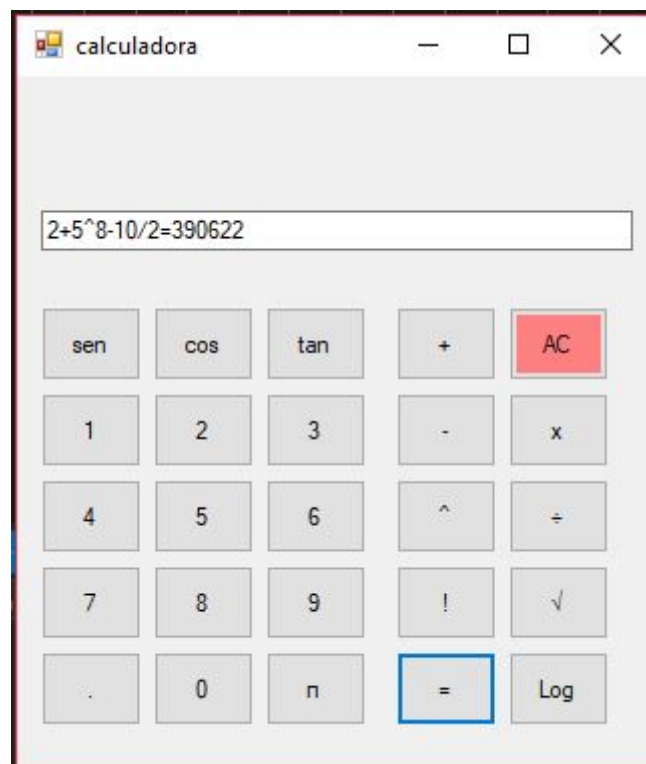
## MANUAL DE USUARIO

---

- **Operaciones +, \*, ^, /, -,**

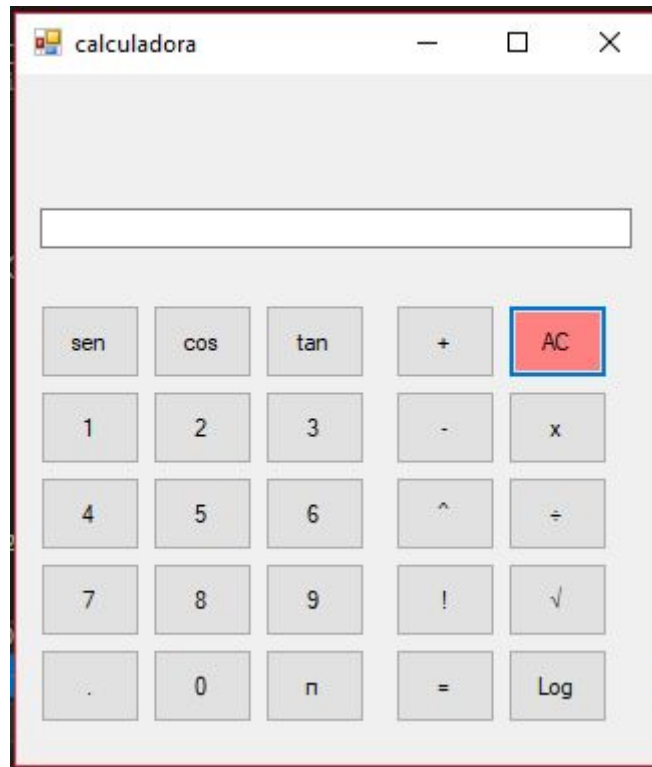
Para calcular una expresión con los operadores anteriormente mencionados, se debe escribir dicha expresión y presionar el botón “=” se calcula el resultado y se presenta en la caja de texto.

NOTA: se pueden introducir las expresiones con los botones de la calculadora o directamente del teclado siempre que al final se presione “=”.



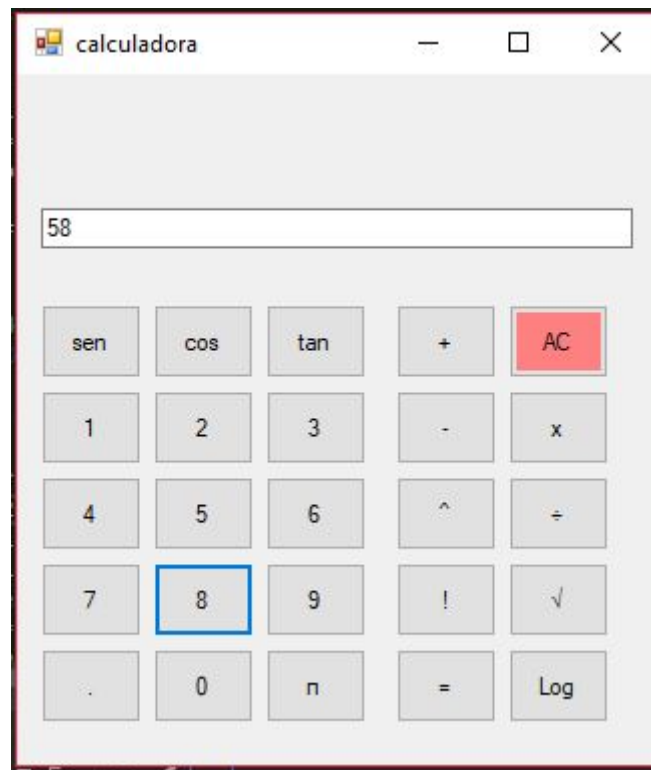
- **Limpiar caja de Texto**

Para limpiar la caja de texto al terminar de calcular una expresión o introducir una de manera errónea, se debe presionar el botón “AC”.

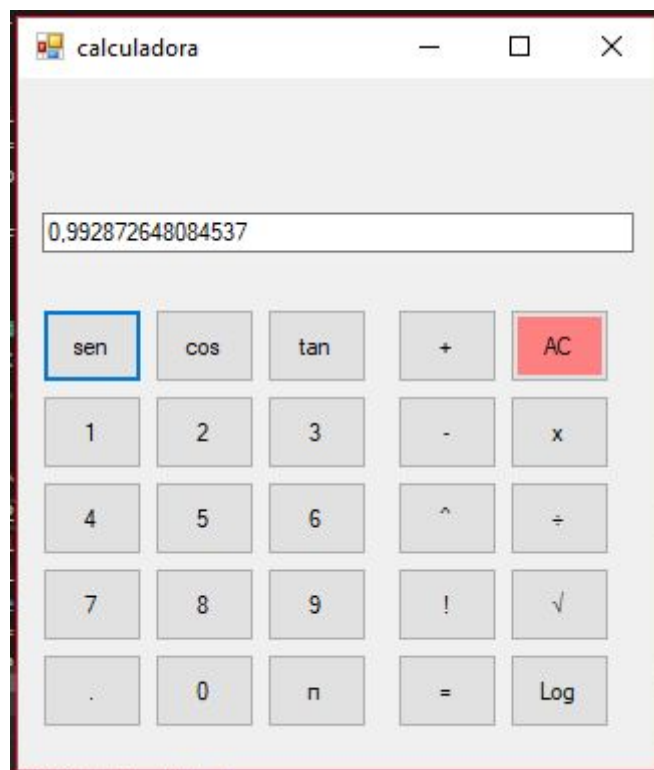


- **SEN, COS, TAN**

Para calcular el seno, coseno o tangente de algún número se debe escribir el número.



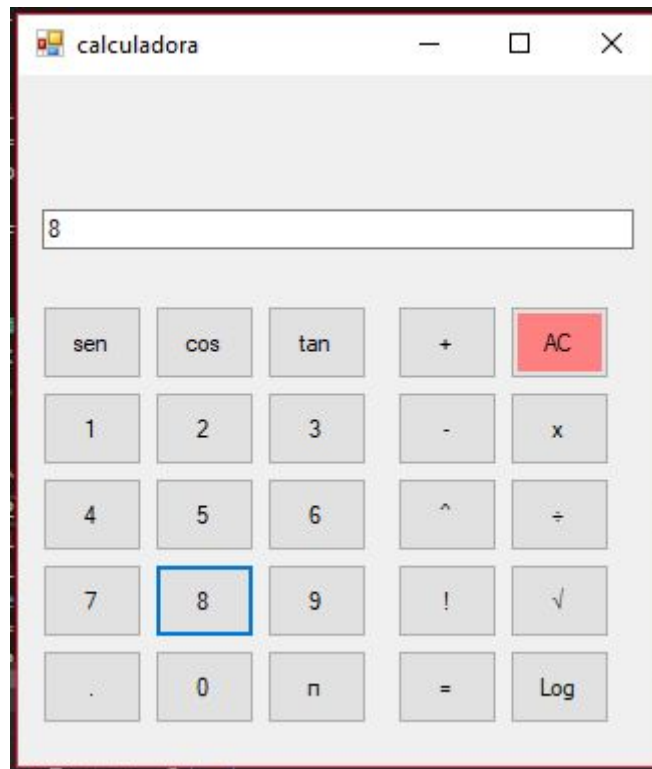
Y luego presionar el botón de sen, cos o tan, y el resultado se mostrará en la caja de texto.



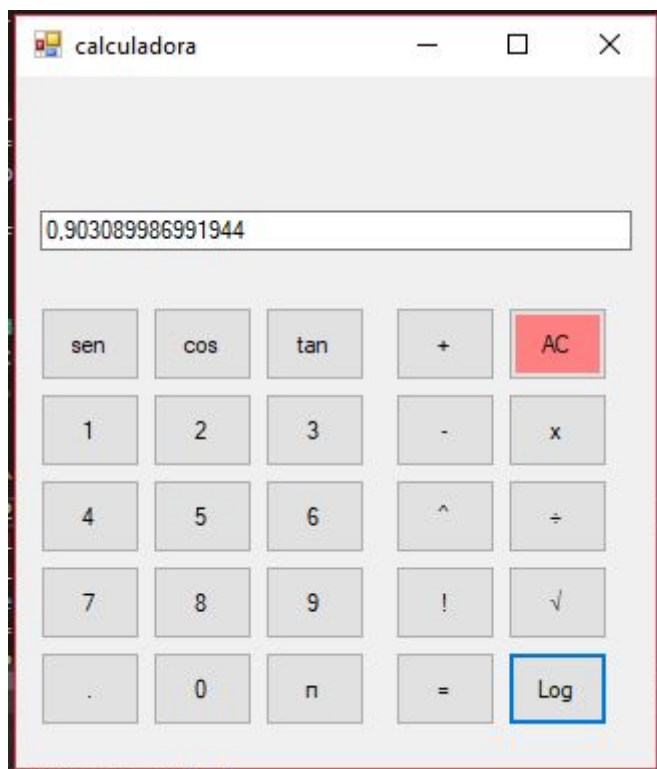


- **LOG**

Para calcular el logaritmo se debe introducir el número que se desea calcular.

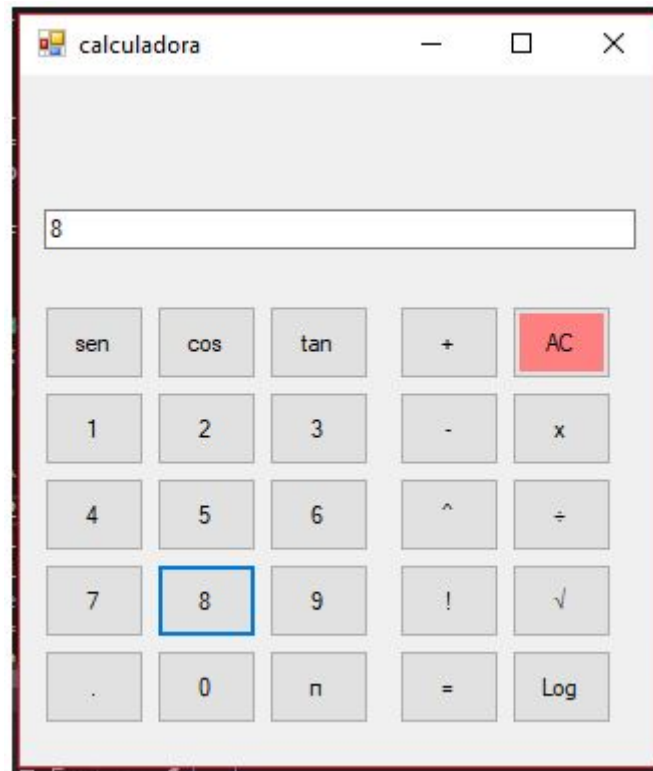


Y luego apretar el botón de logaritmo. El resultado se muestra en la caja de texto.

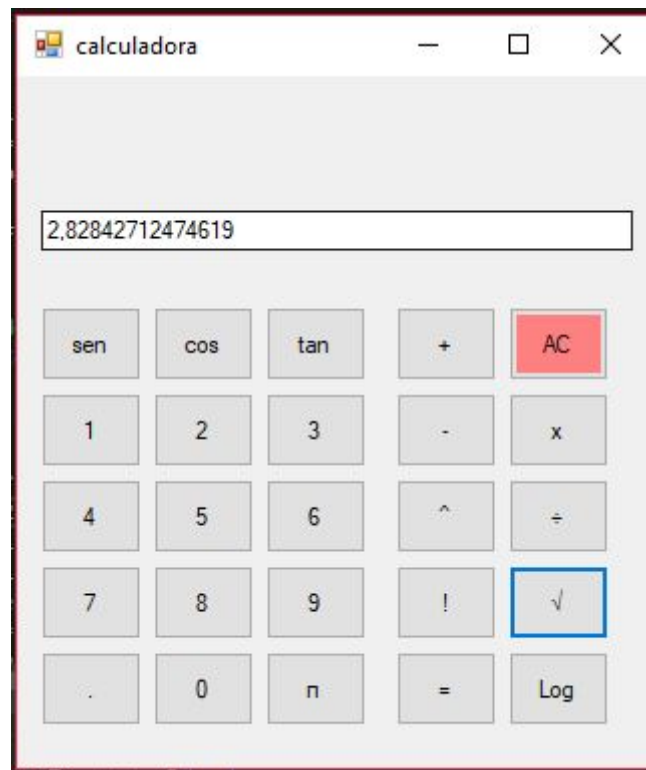


- **RAÍZ CUADRADA**

Para calcular la raíz cuadrada de algún número se debe introducir el número.

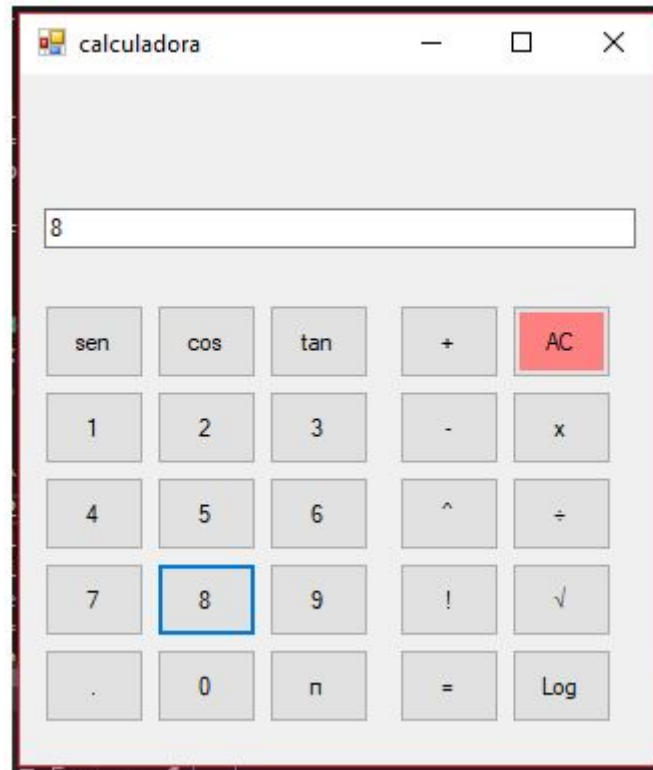


Luego apretar el botón de  $\sqrt{\phantom{x}}$  y el resultado se mostrará en la caja de texto.

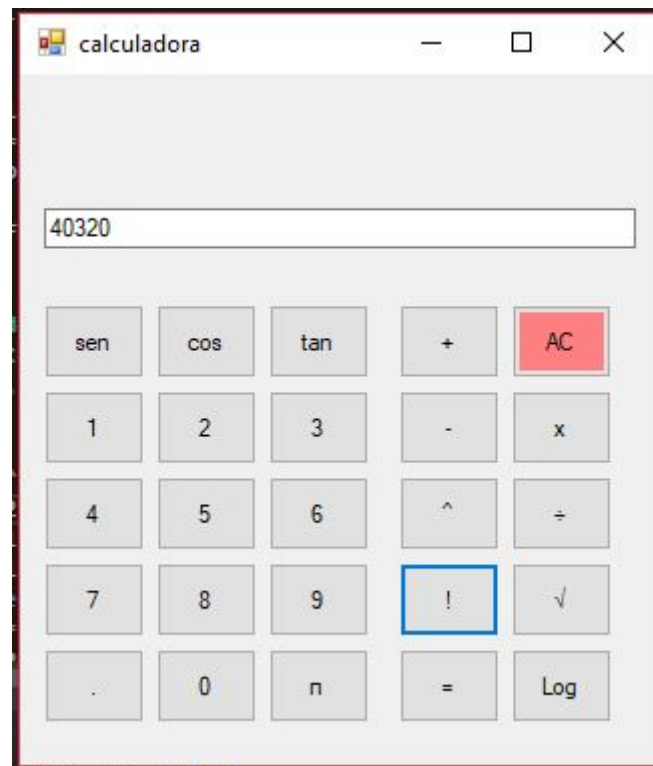


- **FACTORIAL**

Para calcular el factorial de un número se digita el numero en pantalla.



Luego se presiona el botón “!” y el resultado del factorial del número anteriormente digitado se mostrará en la caja de texto.



## **BIBLIOGRAFÍA**

---

- Kip, I. (2008). Lenguaje ensamblador para computadoras basadas en Intel. Pearson Education de México.
- Abel, P. (1996). Lenguaje ensamblador y programación para IBC PC y compatibles. Pearson Education.
- Hernández, H., Hernández, J & Juan, M. (2001). C++ estándar. Editorial Paraninfo.