



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

Facultad de Ingeniería



Proyecto de fundamentos de programación

Nombre del proyecto:

Calculadora simple de aritmética, geometría, ángulos y matrices (TAGAM).

Nombre del alumno:

Hernández González Braulio.

Semestres:

2021-1.

Nombre del profesor:

Marco Antonio Martínez Quintana.

Asignatura:

Fundamentos de programación.

Fecha de elaboración del proyecto:

15 de diciembre 2020 – 28 de enero 2021.

OBJETIVOS DEL PROYECTO:

- Aplicar todos y cada uno de los temas aprendidos en la asignatura Fundamentos de Programación a través de la propuesta de un proyecto final construido durante el transcurso del semestre 2021-1.

INDICE DEL DOCUMENTO:

Objetivos del proyecto	2
Contenido del proyecto	3
Matemáticas en la vida cotidiana	4
Enfoque de emprendimiento	5
Algoritmo de la calculadora	5
Algoritmo parcial de un tema de la calculadora actual	6
Diagrama de flujo parcial de un tema de la calculadora	7
Pseudocódigo parcial de un tema de la calculadora	8
Código fuente	9
Evidencias	35
Tabla comparativa hardware/software	40
Tabla comparativa de costos	40
Diagrama de Gantt	40
Canal de youtube	41
Repositorio de Github	41
Conclusiones	41
Conclusión personal	42
Referencias	43
Glosario de términos independientes	45
Acrónimos o abreviaturas del proyecto	45

CONTENIDO DEL PROYECTO

El proyecto es una calculadora sencilla, enfocada en la parte numérica meramente que posteriormente se irá alargando hasta hacerla una calculadora que se pueden encontrar en internet, que te ayuda a resolver todo tipo de problemas, desde números simples hasta cálculo y números complejos.

La calculadora es un primer intento de resolución a ciertas incógnitas del mundo actual, enfocado en la parte matemática mas que nada; la calculadora presentó nuevos retos para hacerla, y por ende, fue nombrada SHARPE TAGAM (consultar glosario al final para mejor entendimiento) como muestra de lo que se puede llegar a hacer en corto tiempo y posteriormente una calculadora muy avanzada.

Al inicio de la calculadora presenta un menú sobre los contenidos que se pueden encontrar; está dividida en 4 secciones primordiales que son la parte aritmética, la parte geométrica, la parte trigonométrica (no con demasiada profundidad) y la parte compleja del álgebra, las matrices (no con demasiada profundidad); dentro de cada sección del menú se encuentran diferentes subtemas que componen a la calculadora, como suma de números y matrices, resta de números y matrices, productos entre números y matrices, división de números, corte de una circunferencia, valor de ángulos de las diferentes razones trigonométricas que componen al mundo actual en función de pi y de radianes así como su conversión, cálculo de áreas y perímetros de algunas figuras mas comunes (próximamente se agregará volumen) y los principales componentes de matrices como son la traza, determinante, operaciones entre ellas y la transpuesta.

La parte mas grande y la que se lleva la mayor cantidad de memoria de SHARPE es la parte de las matrices; esa parte permiten hacer las operaciones ya mencionadas, pero de una matriz de 3x3 con cierto grado de dificultad.

Eso es todo el contenido del proyecto actual, aunque no es mucho y quizá pudo haber sido más, la calculadora requiere de más cosas que se irán agregando conforme pase el tiempo y la experiencia.

TEMA: MATEMÁTICAS EN LA VIDA COTIDIANA.

La calculadora permite realizar pequeños cálculos que dan solución a las matemáticas que se manejan actualmente en todos los niveles y en todos los lugares necesarios donde se puedan desempeñar.

Las matemáticas son una ciencia exacta y precisa, que se basan en ideologías, teoremas y leyes que se han hecho a lo largo de la vida para dar solución a problemas; fueron muchos los responsables de traer una de las mejores ciencias que conforman a la naturaleza y muchas cosas, desde grandes pensadores griegos tales como Pitágoras, Arquímedes, Thales y Euclides, quienes dieron importantes contribuciones.

De ahí podemos pasar a otros grandes matemáticos que han hecho de las matemáticas una cosa increíble, como Issac Newton, René Descartes, Gauss, Leonhard Euler, Bernhard Riemann, George Boole, entre muchos otros, quien a su método y su pensamiento dieron respuesta a muchos enigmas que ha tenido que lidiar la humanidad y dieron grandes contribuciones.

Las matemáticas asimismo, ayudan a otras materias que requieren dar soluciones, como es la física, la química, la estadística, astronomía, agronomía, entre muchas otros campos laborales o del conocimientos; es decir, las matemáticas son una herramienta que ayuda a las demás ya sea para calcular un área requerida para cultivar (agronomía), calcular velocidades o rapidez a la que debe ir un objeto para romper cierta tensión (física), la cantidad de calor que debe tener un objeto para que se pueda fundir (química), la distancia en la que se encuentra una galaxia de otra (astrología) y más ejemplos existentes.

Las matemáticas se pueden encontrar en cualquier lado y sin la necesidad de mucho conocimiento o de profundizar a temas complejos; algunos ejemplos es para hacer cuentas bancarias, el total de tus gastos mensuales, el total de dinero que necesitas para comprar el mandando, medir con una regla la cantidad de centímetros que debes cotar de un hilo, y muchos ejemplos que se pueden aplicar en la vida cotidiana, y, en muchas ocasiones, es necesario a veces de ciertos dispositivos electrónicos como celular o calculadora para hacer cuentas que pueden llegar a ser demasiado grandes o muchas cuentas al mismo tiempo.

Podemos concluir que las matemáticas es una de las ciencias naturales más importantes, exactas y temidas en cierto punto, indispensables para realizar actividades o trabajos necesarios y que no se hubieran podido tenerlas si no fueran por los pensadores y desarrolladores que tuvieron cierta curiosidad o necesidad de resolver problemas que beneficiarían a las personas.

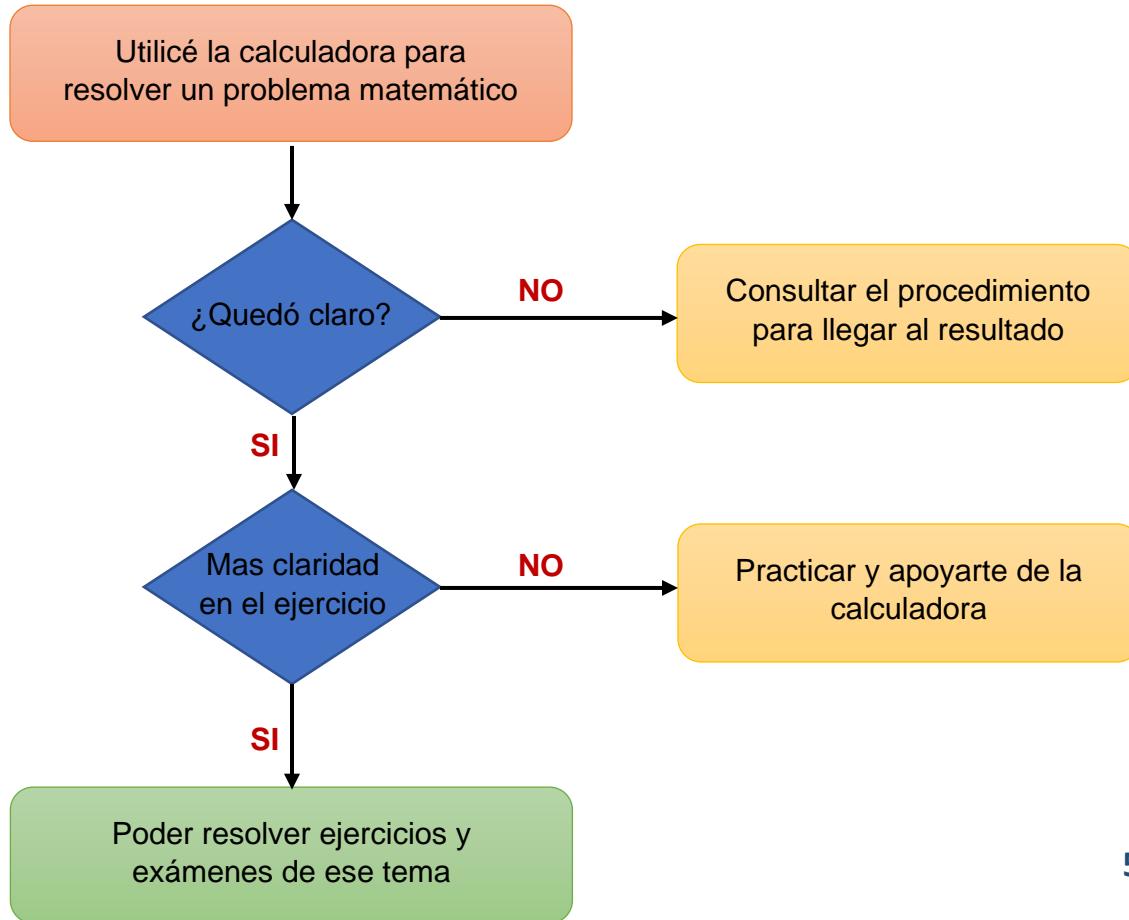
ENFOQUE DE EMPRENDIMIENTO

Con esta calculadora lo que se pretende es ayudar a las personas que tienen una cierta dificultad en cuanto al aprendizaje de matemáticas ya sea por falta de concentración, entendimiento, por falta de ayuda de parte de un profesor o de un orientador que pueda orientar a la persona.

La calculadora tiene como objetivos principales el ser capaz de resolver ecuaciones de las tres principales ciencias que son matemáticas, física y química, con un buen nivel de complejidad y, el segundo objetivo, es que pueda enseñar el procedimiento necesario para llegar a la respuesta correcta, muy parecido a una enseñanza paso a paso sobre la incógnita y si fuera necesario poder explicarlo muy detallado lo que se realizó; y como se ha estado mencionando, se debe ir perfeccionando para que pueda ser un proyecto muy grande.

La calculadora actualmente será de uso personal, pero en un futuro se pretende que sea prestada/rentada o incluso comprada por alguna institución ya sea de investigación, de enseñanza, de programación a la educación u otras parecidas que le den un correcto uso y que el conocimiento de matemáticas, física y química lleguen a todos los rincones del planeta y se vea un poco de todo.

ALGORITMO PRINCIPAL DE LA CALCULADORA:



Cabe recalcar que todos los diagramas de flujo y pseudocódigos que hicieron a la calculadora van a ser parciales, pues todos los temas llevarían mucho tiempo de explicar y no entrarían dentro de uno solo.

ALGORITMO PARCIAL DE UN TEMA DE LA CALCULADORA ACTUAL

El algoritmo está en base en la determinante de una matriz 3x3 que incluye la calculadora dentro de las opciones de “Matrices”:

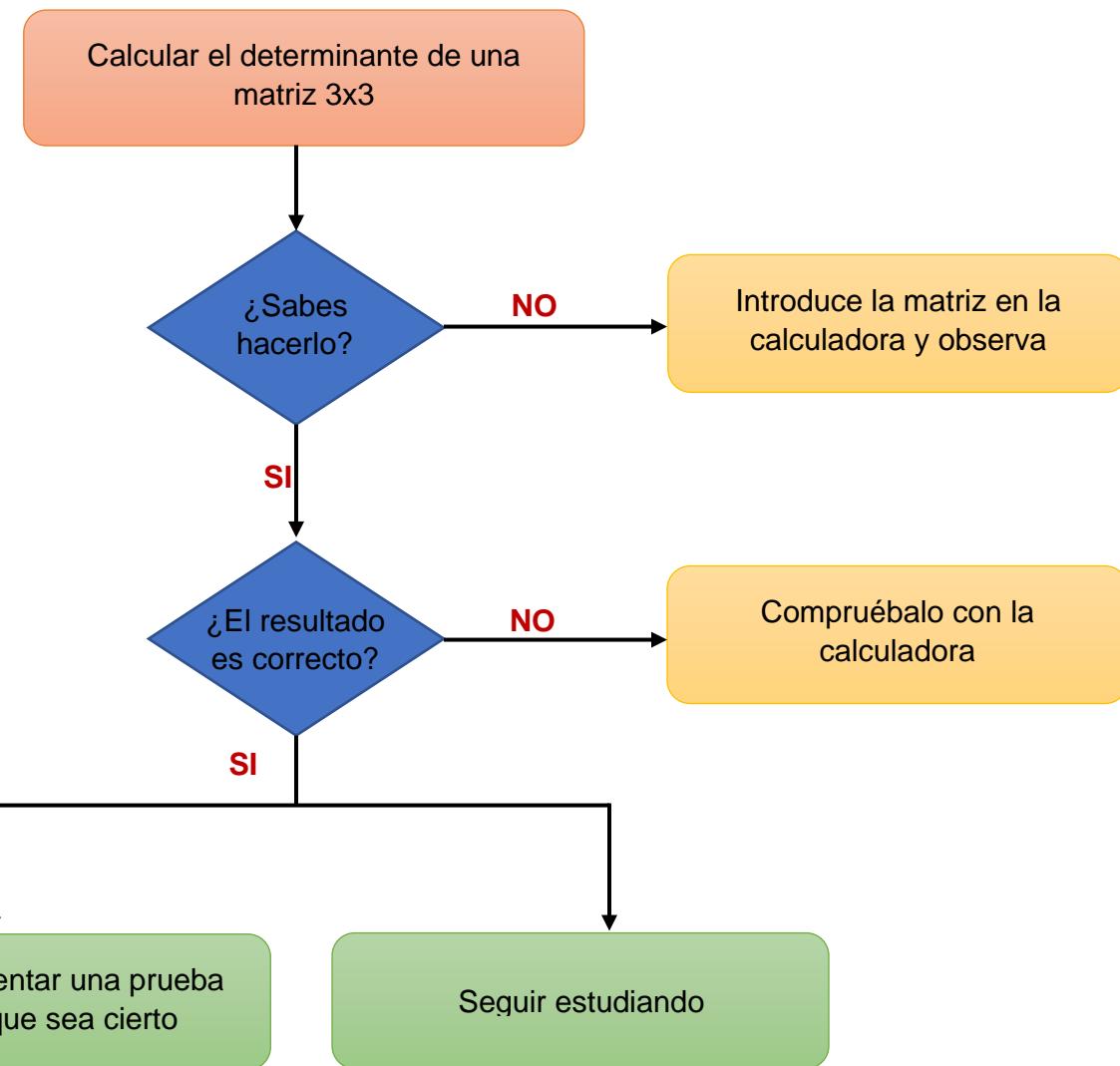
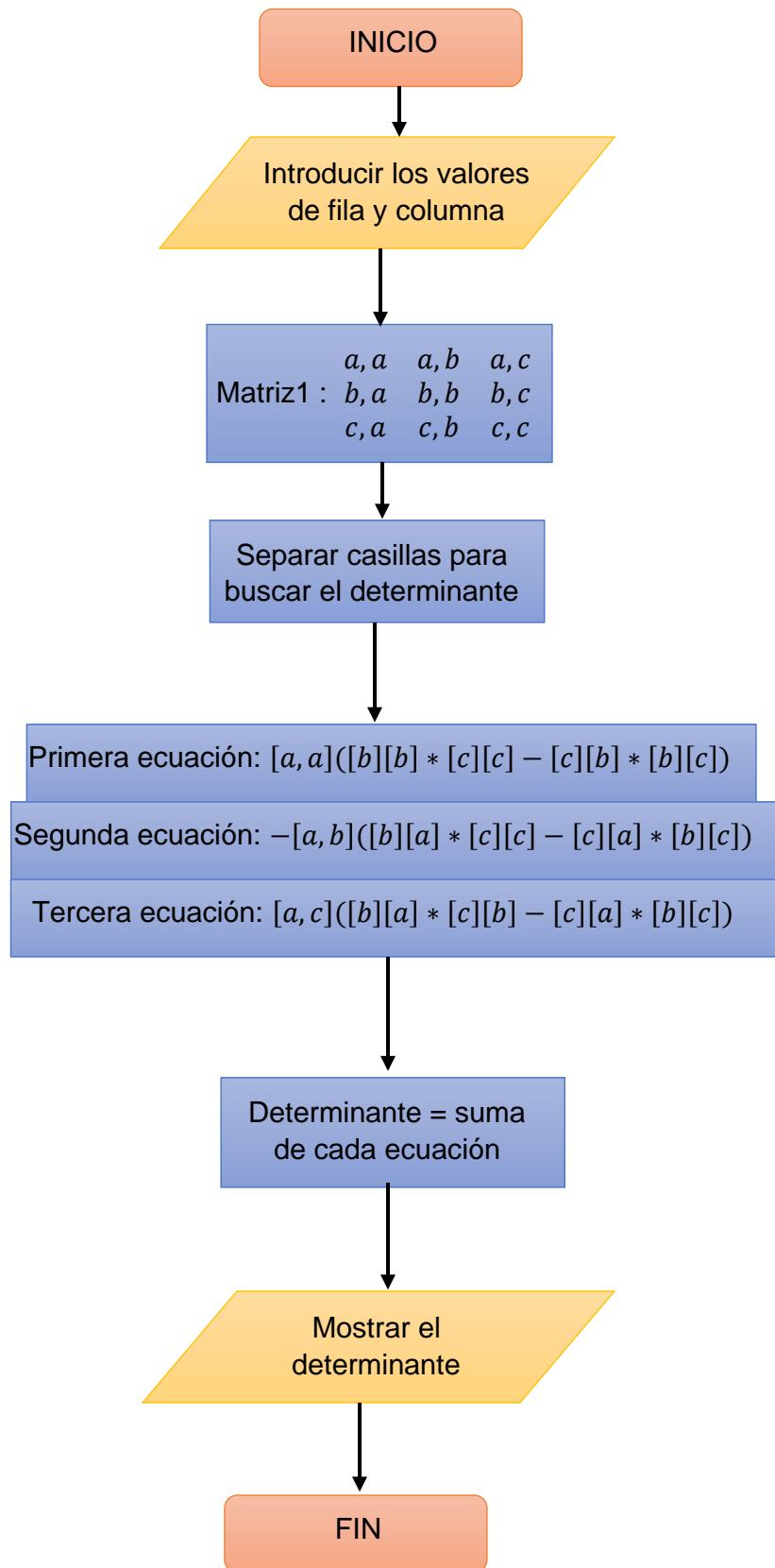


DIAGRAMA DE FLUJO PARCIAL DE UN TEMA DE LA CALCULADORA

El diagrama de flujo está basado en el determinante de una matriz 3x3, el proceso que se sigue para calcular el determinante que se encuentra de los subtemas de “Matrices”:



PSEUDOCÓDIGO PARCIAL DE UN TEMA DE LA CALCULADORA

El pseudocódigo está basado en el determinante de una matriz 3x3, el proceso que se sigue para calcular el determinante que se encuentra de los subtemas de “Matrices”:

```
INICIO
det,D1,D2,D3: ENTEROS
matriz1[3][3]: ENTEROS
//inicio del planteamiento:
PARA matriz1:
    HACER CICLO DE 3x3
        ESCRIBIR valores
        FIN ESCRIBIR
        ESCANEAR valores
            Fila 1: [a][a] [a][b] [a][c]
            Fila 2: [b][a] [b][b] [b][c]
            Fila 3: [c][a] [c][b] [c][c]
        FIN ESCANEAR
    FIN CICLO
FIN PARA
ESCRIBIR "Matriz resultante: "
FIN ESCRIBIR
IMPRIMIR matriz1[i][j]"
FIN IMPRIMIR
PARA D1: //ecuaciones para el determinante
    D1=[a][a] ([b][b]*[c][c] – [c][b][b][c])
FIN PARA
PARA D2:
    D2=-[a][b] ([b][a]*[c][c] – [c][a]*[b][c])
FIN PARA
PARA D3:
    D3=[a][c] ([b][a]*[c][b] – [c][a]*[b][b])
FIN PARA
HACER
det=D1+D2+D3
FIN HACER
ESCRIBIR "El determinante de la matriz es:"
FIN ESCRIBIR
IMPRIMIR det
FIN IMPRIMIR
FIN
```

CÓDIGO FUENTE

A continuación, se presenta el código fuente completo comentado por secciones y todo lo que se tuvo que pensar para realizar la calculadora TAGAM. (Cabe recalcar que se usaron 728 líneas de código para realizar el proyecto).

```
5 int main()
6 {
7 //Variables a usar durante todo el proceso:
8 char aa=160, ae=130, ai=161, ao=162, au=163, sa=168, sc=63, aA=181, g=167; //acentos y demás cosas gramaticales.
9 int opI,opA,opGm,GA,GP,opT,AT,opANG,opMt,opAn; //Respectivos separadores de menús fuera de matrices
10 short n3,v1=360; //Variable para la parte intermedia del proyecto
11 float res,n1,n2,n4,n5,ang,n, x, pi=3.14159265,p; //valores para trigonometría y geometría
12 int i, j, l; //Variables auxiliares
13 int mat1[3][3], mat2[3][3],mat3[3][3],matf[3][3]; //matrices
14 int SRM, D1,D2,D3,det,tr; //variables de matrices (menús switch-case) y valores dentro de estos.
```

Como primera muestra del código se presenta el inicio del proyecto mediante la función int main (la responsable de todo el proyecto) y las variables que se usaron a lo largo del proyecto, entre las que destacan signos gramaticales, variables numéricas, variables para los menús de cada caso y los valores de las matrices.

Nótese que hay comentarios junto a cada variable.

```
16 //Inicio del menú e interfáz
17 printf("\n\n");
18 printf("\t\t Calculadora de diferentes temas\n\n");
19 printf("La calculadora tiene como propósito poder calcular números reales y sistemas de ecuaciones.\n",ao,au);
20 //INICIO MENU PRINCIPAL
21 do
22 {
23     printf("Menú actualizado: \n",au);
24     printf("1)Aritmética.\n2)Geometría.\n3)Trigonometría.\n4)Matrices.\n5)Salir\n",ae,ai,al);
25     scanf("%i",&opI);
26     printf("\n\n");
```

Posteriormente se puede observar el menú principal de la calculadora donde destacan 5 opciones principales: aritmética, geometría, trigonometría, matrices y la opción de salir de la calculadora (fin del programa).

Posteriormente se puede observar un escáner que permite elegir la opción que se desea visualizar; asimismo se puede ver antes de las opciones un do que cierra con su respectivo while:

```
725 }while(opI!=5);
726 //FIN MENU PRINCIPAL
```

La opción 5 es la encargada de dejar de correr la calculadora.

```
27     switch(opI)
28     {
29         case 1:
30             //INICIO DE ARITMÉTICA
```

Para poder elegir la opción que se quiere explorar mediante el escáner, se usó la sentencia switch-case para poder separar cada opción con sus respectivos contenidos y break para evitar que se cree un bucle infinito.

El primer caso que presenta el programa corresponde al de la aritmética; los demás casos se presentan a continuación.

```
93         break;
94     //FIN ARITMÉTICA
95     //-----
96     //INICIO GEOMETRÍA
97     case 2:
280         break;
281     //FIN DE GEOMETRÍA
282     //-----
283     //INICIO DE TRIGONOMETRÍA
284     case 3:
402         break;
403     //FIN TRIGONOMETRÍA
404     //-----
405     //INICIO MATRICES
406     case 4:
```

En el último case de la distribución del menú inicial, se marca el fin de matrices, puesto que código llegaba hasta ahí, pero por el uso de las sentencias y e algunos ciclos que se verán en este documento se redujo considerablemente. El case 5 es el encargado de darle fin al programa, por eso se le agrega ese texto. Nótese la carita feliz.

```

716           break;
717   case 5:
718       printf("Saliendo de la calculadora...\n");
719       printf("Vuelve pronto :)\n");
720       break;
721   default:
722       printf("Null");
723   //FIN MATRICES
724 }
```

Regresando al primer caso del primer menú de Aritmética la calculadora presenta los siguientes códigos:

```

30 //INICIO DE ARITMÉTICA
31 printf("Temas de aritm%ctica\n\n",ae);
32 do
33 {
34     printf("1) Suma.\n");
35     printf("2) Resta.\n");
36     printf("3) Multiplicaci%cn.\n",ao);
37     printf("4) Divisi%cn.\n",ao);
38     printf("5) Regresar.\n\n");
39     scanf("%i",&opA);
40     printf("\n\n");
41     switch(opA)
42     {
43         }
44     }while(opA!=5);
45 }
```

Para introducirse al menú de aritmética se presentan 5 opciones con las operaciones básicas que son la suma, resta, multiplicación y división de números reales; este menú se inicia con un do y cierra con su respectivo while.

Dentro del while se puede observar que la opción 5 es la encargada de cerrar y el menú de aritmética para poder regresar al menú inicial de la TAGAM.

También se puede observar un escáner que selecciona la opción que se desea consultar en base a un switch-case.

```

43     case 1: //SUMA
44         printf("Inserta el primer n%cmero: \n",au);
45         scanf("%ld",&n1);
46         printf("Inserta el segundo n%cmero: \n",au);
47         scanf("%ld",&n2);
48         printf("\n");
49         res=n1+n2;
50         printf("La suma de ambos n%cmeros es: %ld.\n\n",au,res);
51         break;

```

El primer caso que se observa es la suma donde se requiere de dos variables de tipo float (valores reales cortos) que son n1 y n2 respectivamente. Lo que hace esta parte del código es sumar ambos números y mostrar el resultado que daría la adición final mediante otra variable de tipo float, la variable res.

```

52     case 2: //RESTA
53         printf("Inserta el primer n%cmero: \n",au);
54         scanf("%ld",&n1);
55         printf("Inserta el segundo n%cmero: \n",au);
56         scanf("%ld",&n2);
57         printf("\n");
58         res=n1-n2;
59         printf("La resta de ambos n%cmeros es: %ld.\n\n",au,res);
60         break;

```

El segundo caso del menú de aritmética es la resta, donde se requiere nuevamente de las variables n1, n2 y res (variables reutilizadas) para realizar la operación que se solicita.

Esta parte del código lo que hace es simplemente la resta de n1 y n2 que puede dar resultados positivos, el 0 o negativos y el resultado se presenta mediante la variable res.

```

61     case 3: //MULTIPLICACIÓN
62         printf("Inserta el primer n%cmero: \n",au);
63         scanf("%ld",&n1);
64         printf("Inserta el segundo n%cmero: \n",au);
65         scanf("%ld",&n2);
66         printf("\n");
67         res=n1*n2;
68         printf("El producto de ambos n%cmeros es: %ld.\n\n",au,res);
69         break;

```

El tercer caso del menú de aritmética es la multiplicación, donde se volvieron a reutilizar las variables n1, n2 y res.

Esta parte del código lo que hace es efectuar la multiplicación entre n1 y n2 que pueden tomar valores enteros, negativos o decimales; el resultado se presenta y se imprime mediante la variable res.

```

70      case 4: //DIVISIÓN
71          printf("Inserta el primer número: \n", au);
72          scanf("%d", &n1);
73          printf("Inserta el segundo número: \n", au);
74          scanf("%d", &n2);
75          printf("\n");
76          res=n1/n2;
77          if(n2!=0)
78          {
79              printf("La división entre ambos números es: %d.\n\n", ao, au, res);
80          }
81          else
82          {
83              printf("La división entre 0 no se puede efectuar.\n\n", ao);
84          }
85          break;

```

El cuarto y penúltimo caso del menú de aritmética es la división, donde nuevamente se vuelven a usar las variables n1, n2 y res.

Esta parte del código lo que hace es efectuar la división entre n1 y n2 y el resultado lo presenta la variable res; nótese que aquí se usaron las primeras sentencias de selección de todo el proyecto.

La sentencia de selección usada fue if-else, encarga de detectar si el denominador de la división es 0 ya que se sabe que cualquier número dividido entre 0 es infinito.

```

86      case 5:
87          printf("Regresando al menú anterior...\n\n\n", au);
88          break;
89      default:
90          printf("Null");
91      }
92  }while(opA!=5);
93  break;
//FIN ARITMÉTICA

```

El último y no menos importante caso es el 5, el encargado de cerrar el menú de aritmética y regresarnos al menú principal de la calculadora TAGAM.

Se usa la leyenda “Regresando el menú anterior...” con un triple salto de línea para que al momento de ejecutar el programa y volver no se amontone todas las operaciones y el menú con el menú original.

Con esta última parte del código se finaliza la parte de aritmética y las operaciones básicas que en un futuro se podrán implementar para mejorar la calculadora.

Esta parte del código es la base para futuras operaciones de funciones, números complejos y demás cosas.

Como segundo caso del menú original de la calculadora, se observa el inicio de geometría y todo lo que contiene dentro de esta:

```
98     printf("Temas de geometria\n\n");
99
100    do
101    {
102        printf("1) %creas de figuras.\n",aA);
103        printf("2) Per%cmetros de figuras.\n",ai);
104        printf("3) Regresar.\n\n");
105        scanf("%i",&opGm);
106        printf("\n\n");
107        switch(opGm)
108        {
109            case 1:
110                printf("Ingresa el valor de la arista: \n");
111                scanf("%f",&n1);
112                printf("\n");
113                res=n1*n1;
114                printf("El %crea del cuadrado es de: %f unidades cuadradas.\n\n",aa,res);
115                break;
116            case 2://AREA RECTÁNGULO
117                printf("Ingresa el valor de la base: \n");
118                scanf("%f",&n1);
119                printf("Ingresa el valor de la altura: \n");
120                scanf("%f",&n2);
121                printf("\n");
122                res=n1*n2;
123                printf("El %crea del rect%ngulo es de: %f unidades cuadradas.\n\n",aa,aa,res);
124                break;
125            case 3://AREA TRIÁNGULO
126                printf("Ingresa el valor de la base: \n");
127                scanf("%f",&n1);
128                printf("Ingresa el valor de la altura: \n");
129                scanf("%f",&n2);
130                printf("\n");
131                res=(n1*n2)/2;
132                printf("El %crea del tri%ngulo es de: %f unidades cuadradas.\n\n",aa,aa,res);
133                break;
134        }
135    }while(opGm!=3);
136
137
138
139
140
141
142
143
144
145
146
147
148
```

Para el inicio del menú de geometría se inicia con do y cierra con su respectivo while, donde se puede observar que la opción 3 es la encargada de cerrar el menú de geometría y regresar el menú inicial de la calculadora.

Dentro de este menú se puede observar 3 submenús, el primero que calcula áreas de ciertas figuras, el segundo que calcula el perímetro de dichas figuras y el tercero que ya se mencionó.

Así mismo se uso un switch-case para separar cada caso del menú.

Cada valor de área está dado en unidades cuadradas [m²]

```
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
```

```
case 1://AREA CUADRADO
printf("Ingresa el valor de la arista: \n");
scanf("%f",&n1);
printf("\n");
res=n1*n1;
printf("El %crea del cuadrado es de: %f unidades cuadradas.\n\n",aa,res);
break;
case 2://AREA RECTÁNGULO
printf("Ingresa el valor de la base: \n");
scanf("%f",&n1);
printf("Ingresa el valor de la altura: \n");
scanf("%f",&n2);
printf("\n");
res=n1*n2;
printf("El %crea del rect%ngulo es de: %f unidades cuadradas.\n\n",aa,aa,res);
break;
case 3://AREA TRIÁNGULO
printf("Ingresa el valor de la base: \n");
scanf("%f",&n1);
printf("Ingresa el valor de la altura: \n");
scanf("%f",&n2);
printf("\n");
res=(n1*n2)/2;
printf("El %crea del tri%ngulo es de: %f unidades cuadradas.\n\n",aa,aa,res);
break;
```

Dentro de los 3 primeros casos se puede observar que puedes calcular el área del cuadrado ($l * l$), el área del rectángulo ($b * h$) y el área del triángulo ($b * h/2$). Nótese que aquí si se requiere de muchos acentos gramaticales y sus respectivas variables que al inicio del código se declararon, así como nuevamente el uso de las variables n1, n2 y res (variables reutilizadas).

```

149
150
151
152
153
154
155
case 4: //AREA CÍRCULO
    printf("Ingresa el valor del radio: \n");
    scanf("%f",&n1);
    printf("\n");
    res=n1*n1*pi;
    printf("El %crea del c%crculo es de: %f unidades cuadradas.\n\n",aa,ai,res);
    break;

```

El cuarto caso corresponde al área del círculo ($\pi * r^2$) donde solamente se requiere nuevamente del uso de la variable n1, res y de una variable que no se había usado y estaba esperando su momento, la variable pi que fue declarada con el valor aproximado de pi en ciertos decimales.

Nótese que para poner n1 al cuadrado se empleó la multiplicación de n1 por n1.

```

156
157
158
159
160
161
162
163
164
case 5: //AREA ROMBO
    printf("Ingresa el valor de la diagonal mayor: \n");
    scanf("%f",&n1);
    printf("Ingresa el valor de la diagonal menor: \n");
    scanf("%f",&n2);
    printf("\n");
    res=(n1*n2)/2;
    printf("El %crea del rombo es de: %f unidades cuadradas.\n\n",aa,res);
    break;

```

Para el quinto caso del menú, corresponde el área de un rombo regular ($(D * d)/2$) donde nuevamente se vuelve a usar la variable n1, n2 y res para dar la respuesta correcta, que es mediante la multiplicación de n1 por n2 y dividido entre dos, es por eso indispensable el uso de paréntesis.

```

165
166
167
168
169
170
171
172
173
174
175
case 6: //AREA TRAPECIO
    printf("Ingresa el valor de la base mayor: \n");
    scanf("%f",&n1);
    printf("Ingresa el valor de la base menor: \n");
    scanf("%f",&n2);
    printf("Ingresa el valor de la altura: \n");
    scanf("%f",&n4);
    printf("\n");
    res=((n1+n2)/2)*n4;
    printf("El %crea del trapecio es de: %f unidades cuadradas.\n\n",aa,res);
    break;

```

El sexto caso del menú corresponde al área de un trapecio regular ($\frac{(B+b)}{2} * h$) donde se emplean nuevamente las variables n1, n2 y res aunque en esta ocasión se utiliza una nueva variable, n4, quien en este caso tiene el valor de la altura del trapecio, ya que como sabemos, el trapecio tiene dos bases (mayor y menor) y una altura entre base y base.

```
176  
177  
178  
179  
180  
181  
182  
183  
184  
185  
186  
187  
188  
case 7: //AREA PENTÁGONO  
    printf("Ingresa el valor del lado: \n");  
    scanf("%f",&n1);  
    printf("Ingresa el valor del apotema: \n");  
    scanf("%f",&n2);  
    printf("\n");  
    p=5*n1;  
    res=(p*n2)/2;  
    printf("El %crea del pent%cgonos es de: %f unidades cuadradas.\n\n",aa,aa,res);  
    break;  
case 8: //SALIR  
    printf("Regresando al men%c anterior...\n\n",au);  
    break;
```

En el penúltimo caso, el séptimo, es el encargado de calcular el área de un pentágono regular ($(p * a)/2$) donde se requiere nuevamente de las variables n1, n2 y res y una variable nueva denominada “p”, quien es la encargada primeramente de calcular el perímetro del pentágono multiplicando el lado (n1) por 5.

Posteriormente efectúa la multiplicación del perímetro por apotema y lo divide entre dos y res es la encargada de imprimir el resultado.

```
186 case 8://SALIR  
187     printf("Regresando al men%c anterior...\n\n",au);  
188     break;  
189 default:  
190     printf("Null");
```

Finalmente, el último caso es el octavo, y es el encargado de dejar de correr el menú de cálculo de áreas de figuras con solo teclear el número 8.

Nótese también el default para cerrar correctamente el switch-case y que el programa no presente problemas.

```

193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
272

```

```

case 2:
printf("Elige la figura:\n\n");
do
{
    printf("1) Cuadrado.\n");
    printf("2) Rectángulo.\n",aa);
    printf("3) Triángulo.\n",aa);
    printf("4) Círculo.\n",ai);
    printf("5) Rombo.\n");
    printf("6) Trapecio.\n");
    printf("7) Pentágono regular.\n",aa);
    printf("8) Regresar.\n\n");
    scanf("%i",&GP);
    printf("\n\n");
    switch(GP)
{
}
```

}while(GP!=8);

El segundo submenú del menú de geometría consiste en el cálculo de perímetros de las figuras antes mencionadas, el menú presenta 8 opciones, donde la octava indica que el menú se cierra y regresa el menú inicial de la calculadora.

Se inicia con un do y cierra con su respectivo while en la opción 8; se pueden observar las opciones en un listado y luego un switch-case con 8 casos, asimismo un escáner encargado de seleccionar que opción se desea elegir.

(Cabe mencionar que se volverán a usar las mismas variables).

```

209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231

```

```

case 1: //PERÍMETRO CUADRADO
printf("Ingresa el valor de la arista: \n");
scanf("%f",&n1);
printf("\n");
res=4*n1;
printf("El perímetro del cuadrado es de: %f unidades.\n\n",ai,res);
break;
case 2: //PERÍMETRO RECTÁNGULO
printf("Ingresa el valor de la base: \n");
scanf("%f",&n1);
printf("Ingresa el valor de la altura: \n\n");
scanf("%f",&n2);
printf("\n");
res=n1+n1+n2+n2;
printf("El perímetro del rectángulo es de: %f unidades.\n\n",ai,aa,res);
break;
case 3: //PERÍMETRO TRIÁNGULO
printf("Ingresa el valor de la base: \n");
scanf("%f",&n1);
printf("\n");
res=3*n1;
printf("El perímetro del triángulo es de: %f unidades.\n\n",ai,aa,res);
break;
```

En los primeros 3 casos se puede observar el cálculo del perímetro del cuadrado, rectángulo y triángulo respectivamente, que se obtiene mediante la suma de todos sus lados.

Las unidades de los perímetros están dadas en unidades ([u]).

```

231
232         case 4://PERÍMETRO CÍRCULO
233             printf("Ingresa el valor del radio: \n");
234             scanf("%f",&n1);
235             printf("\n");
236             res=2*n1*pi;
237             printf("El per%cmetro del c%crculo es de: %f unidades.\n\n",ai,ai,res);
238             break;

```

El cuarto caso presenta el cálculo del perímetro del círculo ($2\pi r$) donde se vuelve a hacer uso de la variable n1 que representa al radio, res y la variable pi, que al ser declarada se usaron cuantos decimales posibles que simularan el valor exacto de Pi.

```

239
240         case 5://PERÍMETRO ROMBO
241             printf("Ingresa el valor del lado: \n");
242             scanf("%f",&n1);
243             printf("\n");
244             res=4*n1;
245             printf("El per%cmetro del rombo es de: %f unidades.\n\n",ai,res);
246             break;
247         case 6://PERÍMETRO TRAPECIO
248             printf("Ingresa el valor de la base mayor: \n");
249             scanf("%f",&n1);
250             printf("Ingresa el valor de la base menor:\n");
251             scanf("%f",&n2);
252             printf("Ingresa el valor del lado derecho: \n");
253             scanf("%f",&n4);
254             printf("Ingresa el valor del lado izquierdo: \n");
255             scanf("%f",&n5);
256             printf("\n");
257             res=n1+n2+n4+n5;
258             printf("El per%cmetro del trapecio es de: %f unidades.\n\n",ai,res);
259             break;
260         case 7://PERÍMETRO PENTÁGONO
261             printf("Ingresa el valor del lado: \n");
262             scanf("%f",&n1);
263             printf("\n");
264             res=5*n1;
265             printf("El per%cmetro del pent%cgono es de: %f unidades.\n\n",ai,aa,res);
266             break;

```

Los siguientes tres casos, del 5 al 7 corresponden a los perímetros del rombo regular, trapecio y del pentágono regular respectivamente, notamos que se vuelven a hacer uso de las variables n1, n2 y res (reutilizadas) solo que en el trapecio se hace uso de las variables n4 y n5 (no existe n3 porque es usada en otro entorno); estas dos variables representan un los lados derecho e izquierdo del trapecio, puesto que esta vez se puede tratar de un trapecio irregular.

Nótese como en todos los casos se hizo el uso de las variables gramaticales de acento, puesto que el programa por si solo no detecta acentos.

```
266 //SALIR
267 printf("Regresando al men%c anterior...\n\n",au);
268 break;
269 default:
270     printf("Null");
271 }
```

El último y no menos importante caso es el octavo, el encargado de cerrar el menú de perímetros y poder regresarnos al menú inicial de geometría. Asimismo, se observa el default que se tiene que agregar a cada switch-case para que corra correctamente y no presente fallas.

Hasta esta parte del proyecto fue solamente el “calentamiento” de programación, ya que se usaron las sentencias básicas que se vio a lo largo de todo el periodo como las estructuras de repetición, estructuras de control, estructuras de impresión y de escaneo.

También se incluyen los operadores como suma, resta, división y multiplicación; no se incluyó el módulo puesto que no encontré un lugar adecuado para meterlo, pero en el futuro se agregará en algún proceso algebraico o modular que así lo requiera.

Las horas invertidas a esta parte del proyecto no fueron demasiadas, pero donde mas se fue el tiempo fue en arreglar error, “bugs” que me marcaban error o detenían la calculadora, variables mal declaradas, mal uso del scanf y de las sumas y restas de lados de áreas.

También se encontraron problemas en la parte del cálculo de áreas, puesto que las variables no correspondían, malas sentencias de escritura, mal uso del scanf y el olvido de puntos y comas.

A partir de esta parte del proyecto que se puso mas atención al momento de programarlo y de cuidar cada detalle del scanf y otros “bugs” o fallos, la siguiente parte fue lo mas laborioso y lo que requirió de más tiempo para hacerlo.

Sin más comentarios que agregar se procede a explicar y seguir comentando el código fuente de toda la calculadora.

```

284
285     printf("Temas de trigonometr%ca\n\n",ai);
286     do
287     {
288         printf("1) Circunferencia.\n");
289         printf("2) %cngulos.\n",aA);
290         printf("3) Razones trigonom%ctricas.\n",ae);
291         printf("4) Volver al men%c inicial.\n\n",au);
292         scanf("%i",&opT);
293         printf("\n\n");
294         switch (opT)
295         {
296             }
297         }while(opT!=4);

```

Para la tercera parte de la calculadora se hizo primeramente el uso de un do con su respectivo while que cierra en la opción 4.

Dentro de este menú se pueden observar 4 opciones que son la circunferencia, ángulos (equivalencias), razones trigonométricas (de ángulos a pi) y finalmente la opción que permite cerrar la calculadora de trigonometría y volver al menú principal de la calculadora TAGAM.

Así mismo se usó un switch-case para poder separar cada caso del submenú.

```

296     case 1:
297         printf("Divisor de la circunferencia en partes iguales.\n");
298         printf("Recuerda que la circunferencia es equivalente a 2 pi radianes o 360 grados.\n");
299         printf("Partes en las que quieras dividir la circunferencia: \n");
300         scanf("%f",&n);
301         printf("\n");
302         x=360/n;
303         printf("Cada %f grados habr%oc un punto divisor dentro de la circunferencia.\n\n",x,aa);
304         break;

```

El primer caso consiste en una división de la circunferencia en puntos divisores, es decir, el usuario elige en cuantas partes quiere dividir la circunferencia y el programa en automática te marca el resultado.

Se hizo uso de las variables n y x en este caso, donde n representa las partes en que se desean dividir la circunferencia y x el resultado de dividir los 360 grados entre n para dar un resultado.

Nótese que se agrega una nota de cuantos grados y pi radianes equivale una circunferencia entera para de ahí hacer conversiones rápidas.

```

305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
    case 2:
        printf("Ingresa el valor del %cngulo para encontrar la equivalencia. \n",aa);
        printf("Intervalo aceptado: [-720%c a 720%c]\n",g,g);
        scanf("%f",&ang);
        printf("\n");
        if (ang<0)
        {
            if (ang<-360)
            {
                x=ang+720;
                printf("%f grados es equivalente a %f grados.\n\n",ang,x);
            }
            else
            {
                x=ang+360;
                printf("%f grados es igual a %f grados.\n\n",ang,x);
            }
        }
        else
        {
            x=ang-360;
            printf("%f grados es igual a %f grados.\n\n",ang,x);
        }
        break;

```

Para el segundo caso del menú de trigonometría se puede observar que se realizan equivalencia de ángulos, cuando son menores a 360 o mayores a este dentro de un intervalo de 720 grados como lo indica el programa, ya que normalmente son los grados mas utilizados en ingeniería y cálculos.

Se puede observar que se hace uso de la variable ang, quien es la encargada de registrar el valor del ángulo y nuevamente la variable x (reutilizada), encargada de calcular el ángulo equivalente.

Aquí se vuelve a usar las estructuras de selección if-else para resolver el problema, que en un futuro será intercambiada por do-while y while.

```

329
330
331
332 case 3:
333     printf("Convierte tus grados en radianes.\n");
334     do
335     {
336         printf("Selecciona una opci%cn: \n",ao);
337         printf("1) Seno.\n");
338         printf("2) Coseno.\n");
339         printf("3) Tangente.\n");
340         printf("4) Cotangente.\n");
341         printf("5) Secante.\n");
342         printf("6) Cosecante.\n");
343         printf("7) Volver al men%c\n",au);
344         scanf("%i",&opANG);
345         printf("\n\n");
346         switch(opANG)
347         {
348             }
349         }
350     }
351     break;

```

El tercer caso del menú de trigonometría consiste en convertir ángulos a radianes en función de una razón trigonométrica (seno, coseno, tangente, cotangente, cosecante y cosecante).

Para abrir el caso 3 se requirió de un nuevo submenú denominado “opANG” el cual se inicia con un do y su respectivo while que finaliza al teclear la opción 7, encargada de cerrar ese submenú y regresar al usuario al menú de trigonometría.

Dentro se puede observar una lista de todas las razones, donde solamente con teclear el número que corresponde a la razón marcada ahí.

También se observa un escáner que permite detectar la opción.

Antes de seguir avanzando por este menú se recuerda que en el mundo de las matemáticas existen 6 razones trigonométricas muy usadas, de las cuales 3 destacan que son el seno, coseno y la tangente de un ángulo.

Las otras 3 que no destacan tanto son la cotangente, cosecante y secante, puesto no son muy indispensables pero usadas en ciertas ocasiones y son las recíprocas a las funciones principales, es decir, el recíproco de seno es la cosecante, el recíproco del coseno es la secante y de la tangente la cotangente.

Sin más que agregar sobre estos ángulos en la parte matemáticas proseguimos a seguir explicando el submenú de razones trigonométricas y su estructura de funciones para obtener los resultados.

```

345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365

    case 1:
        printf("Ingresa el valor del %cngulo: \n",aa);
        scanf("%f",&ang);
        printf("\n");
        x=sin(ang);
        printf("El seno de %f grados es igual a: %f radianes.\n\n",ang,x);
        break;
    case 2:
        printf("Ingresa el valor del %cngulo: \n",aa);
        scanf("%f",&ang);
        printf("\n");
        x=cos(ang);
        printf("El coseno de %f grados es igual a: %f radianes.\n\n",ang,x);
        break;
    case 3:
        printf("Ingresa el valor del %cngulo: \n",aa);
        scanf("%f",&ang);
        printf("\n");
        x=tan(ang);
        printf("La tangente de %f grados es igual a: %f radianes.\n\n",ang,x);
        break;

```

Dentro del submenú de razones trigonométricas se muestran los 3 primeros casos, del seno, coseno y la tangente respectivamente.

Ahí se puede ver nuevamente el uso de las variables ang, quien es la encargada de detectar el ángulo que se desea obtener, y la variable x, quien se encarga de calcular el seno, coseno o tangente del ángulo (ang) dependiendo el caso.

```

366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386

    case 4:
        printf("Ingresa el valor del %cngulo: \n",aa);
        scanf("%f",&ang);
        printf("\n");
        x=1/(tan(ang));
        printf("La cotangente de %f grados es igual a: %f radianes.\n\n",ang,x);
        break;
    case 5:
        printf("Ingresa el valor del %cngulo: \n",aa);
        scanf("%f",&ang);
        printf("\n");
        x=1/(cos(ang));
        printf("La secante de %f grados es igual a: %f radianes.\n\n",ang,x);
        break;
    case 6:
        printf("Ingresa el valor del %cngulo: \n",aa);
        scanf("%f",&ang);
        printf("\n");
        x=1/(sin(ang));
        printf("La cosecante de %f grados es igual a: %f radianes.\n\n",ang,x);
        break;

```

Los siguientes tres casos, del 4 al 6 corresponden al cálculo de un ángulo de la cotangente, secante y cosecante respectivamente.

Al ser reciprocas de las funciones seno, coseno y tangente lo único que se realizó en esta parte es hacer una división de 1 en el numerador sobre el valor del ángulo en el denominador, de forma que al multiplicar estos valores consu recíproco darán como resultado 1.

Nótese que se vuelven a usar las variables ang y x.

```
387  
388  
389  
390  
391           case 7:  
                  printf("Regresando al men%c anterior...\n\n",au);  
                  break;  
              default:  
                  printf("Null");
```

El último caso del submenú de razones trigonométricas y no menos importante fue el caso séptimo, encargado de cerrar el menú y regresarnos al menú de trigonometría.

Nótese que al finalizar cada caso y agregar el que se encarga de regresar a un menú está dado con un triple espacio, que permite una mejor visualización del proyecto una vez acabada la sentencia.

Se observa el default que acompaña a cada switch-case.

```
395  
396  
397  
398  
399           case 4:  
                  printf("Regresando al men%c principal...\n\n",au);  
                  break;  
              default:  
                  printf("Null");
```

Finalmente, se llega al cuarto y último caso del menú de trigonometría, el encargado de cerrar el menú y regresarnos al menú principal de la calculadora. Se puede observar que está separado por un triple salto de linea que da una mejor estética al programa mientras se está ejecutando.

También se observa el default que acompaña a cada switch-case en su escritura dentro del código.

```

406     case 4:
407         printf("Matrices 3x3\n\n");
408         do
409             {
410                 printf("1) Suma | Resta.\n");
411                 printf("2) Multiplicaci%cn entre matrices.\n",ao);
412                 printf("3) Determinante.\n");
413                 printf("4) Trazo.\n");
414                 printf("5) Transpuesta.\n");
415                 printf("6) Regresar.\n");
416                 scanf("%i",&opMt);
417                 printf("\n\n");
418                 switch(opMt)
419                     {
714             }
715         }while(opMt!=6);
716         break;

```

En esta parte del proyecto estamos en el penúltimo caso y último tema de la calculadora, la parte de las matrices y la mejor parte.

Para este submenú se inicia con un do y su respectivo while con opción 6, encargado de cerrar el menú de matrices y regresarnos al menú inicial de la calculadora TAGAM.

Se observa dentro del menú 6 opciones, donde 5 corresponden al cálculo de los principales componentes de una matriz y el sexto es para regresar al inicio.

Existe también un switch-case que permite separar cada cálculo y un escáner que selecciona la opción que se desea visualizar.

```

433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
case 1: //Suma
printf("Ingrese las dos matrices que desea sumar: \n");
//Primera matriz
printf("\tPRIMER MATRIZ\n\n");
for (int i=0;i<3;i++)
{
    for (int j=0;j<3;j++)
    {
        printf("Valor que deseas agregar al rengl%cn %i de la columna %i: ",ao,i+1,j+1);
        scanf("%i",&mat1[i][j]);
    }
    printf("\n\n");
}
printf("Matriz resultante: \n");
for (int i=0;i<3;i++)
{
    for (int j=0;j<3;j++)
    {
        printf("%i\t",mat1[i][j]);
    }
    printf("\n");
}
printf("\n\n");
//Segunda matriz
printf("\tSEGUNDA MATRIZ\n\n");
for (int i=0;i<3;i++)
{
    for (int j=0;j<3;j++)
    {
        printf("Valor que deseas agregar al rengl%cn fila %i de la columna %i: ",ao,i+1,j+1);
        scanf("%i",&mat2[i][j]);
    }
    printf("\n\n");
}
printf("Matriz resultante: \n");
for (int i=0;i<3;i++)
{
    for (int j=0;j<3;j++)
    {
        printf("%i\t",mat2[i][j]);
    }
    printf("\n");
}
printf("\n\n");

```

El primer caso del menú se trata de suma de dos matrices de 3x3; aquí se hizo el uso por primera vez de la estructura de repetición for, así como el uso de los arreglos bidimensionales para poder hacer las matrices.

Se usan las variables “mat1[3][3]” y “mat2[3][3]”.

Primeramente, existe un for con la variable i, que permite agregar el valor entero de cada renglón 1, 2 y 3; dentro de este for existe otro for con la variable j que permite agregar el valor entero por cada columna 1, 2 y 3, por eso ambas variables están dadas desde el valor 0 y menor a 3.

El escáner nos permite capturar estos valores para formar la matriz.

Posteriormente existe un apartado que imprime la matriz resultante, usándose un tabulador para dar un espacio entre cada columna dentro del for de la variable “j” y luego en el for de la variable “i” un salto de línea entre cada renglón.

Mismo proceso ocurre con la matriz 2:

```

456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
//Segunda matriz
printf("\tSEGUNDA MATRIZ\n\n");
for (int i=0;i<3;i++)
{
    for (int j=0;j<3;j++)
    {
        printf("Valor que deseas agregar al rengl%cn fila %i de la columna %i: ",ao,i+1,j+1);
        scanf("%i",&mat2[i][j]);
    }
    printf("\n\n");
}
printf("Matriz resultante: \n");
for (int i=0;i<3;i++)
{
    for (int j=0;j<3;j++)
    {
        printf("%i\t",mat2[i][j]);
    }
    printf("\n");
}
printf("\n\n");

```

Nótese que entre matriz y matriz existe un doble salto de linea.

```
477     printf("La suma de ambas matrices es de: \n");
478     for (int i=0;i<3;i++)
479     {
480         for (int j=0;j<3;j++)
481         {
482             matf[i][j]=mat1[i][j]+mat2[i][j];
483             printf("%i\t",matf[i][j]);
484             //printf("%i,%i\n",i,j);
485         }
486         printf("\n");
487     }
488     printf("\n\n");
489     break;
490 }
```

Finalmente, mediante el uso de un doble for para la variable i y j, existe la siguiente variable declarada como “matf[3][3]”, la cual es la encargada de realizar la suma de la matriz 1 y la matriz 1, sumando cada valor de renglón y columna de la primera matriz más cada valor de renglón y columna de la segunda matriz. Al final solamente se imprime la matriz final (“matf”) con un tabulador dentro del f de la variable j para diferenciar cada columna y un salto de linea dentro del for de la variable i para diferenciar cada renglón.

El mismo proceso que se realiza en la suma, ocurre para la resta de matrices:

Proceso de la matriz 1.

```
490 case 2://Resta
491     printf("Ingrese las dos matrices que deseé restar: \n");
492     //Primera matriz
493     printf("\tPRIMER MATRIZ\n\n");
494     for (int i=0;i<3;i++)
495     {
496         for (int j=0;j<3;j++)
497         {
498             printf("Valor que deseas agregar al rengl%c fil %i de la columna %i: ",ao,i+1,j+1);
499             scanf("%i",&mat1[i][j]);
500         }
501         printf("\n\n");
502     }
503     printf("Matriz resultante: \n");
504     for (int i=0;i<3;i++)
505     {
506         for (int j=0;j<3;j++)
507         {
508             printf("%i\t",mat1[i][j]);
509         }
510         printf("\n");
511     }
512     printf("\n\n");
513 }
```

Proceso de la matriz 2.

```
513 //Segunda matriz
514 printf("\tSEGUNDA MATRIZ\n\n");
515 for (int i=0;i<3;i++)
516 {
517     for (int j=0;j<3;j++)
518     {
519         printf("Valor que deseas agregar al rengl%c n %i de la columna %i: ",ao,i+1,j+1);
520         scanf("%i",&mat2[i][j]);
521     }
522     printf("\n\n");
523 }
524 printf("Matriz resultante: \n");
525 for (int i=0;i<3;i++)
526 {
527     for (int j=0;j<3;j++)
528     {
529         printf("%i\t",mat2[i][j]);
530     }
531     printf("\n");
532 }
533 printf("\n\n");
```

El mismo proceso ocurre para la matriz final, la cual efectúa la resta de cada valor de cada renglón y columna de la matriz 1 y le resta cada valor de cada renglón y columna de la matriz 2.

```
534     printf("La resta de ambas matrices es de: \n");
535
536     for (int i=0;i<3;i++)
537     {
538         for (int j=0;j<3;j++)
539         {
540             matf[i][j]=mat1[i][j]-mat2[i][j];
541             printf("%i\t",matf[i][j]);
542             //printf("%i,%i\n",i,j);
543         }
544         printf("\n");
545     }
546     printf("\n\n");
547     break;
```



```
548 case 3: //Salir
549     printf("Volviendo al men%c anterior...\n\n",au);
550     break;
551 default:
552     printf("Null");
553 //FIN SUMA Y RESTA MATRICES
```

Finalmente, el tercer caso en el menú de sumas y restas, el encargado de cerrar este menú y regresar al usuario al menú de matrices.

Se observa también el default, sentencia que acompaña a cada switch-case.

```

557     case 2: //multiplicación de matrices
558         //Primera matriz
559         printf("Ingrese las dos matrices que desea multiplicar: \n");
560         printf("\tPRIMER MATRIZ\n\n");
561         for (int i=0;i<3;i++)
562         {
563             for (int j=0;j<3;j++)
564             {
565                 printf("Valor que deseas agregar al renglón %i de la columna %i: ",ao,i+1,j+1);
566                 scanf("%i",&mat1[i][j]);
567             }
568             printf("\n\n");
569         }
570         printf("Matriz resultante: \n");
571         for (int i=0;i<3;i++)
572         {
573             for (int j=0;j<3;j++)
574             {
575                 printf("%i\t",mat1[i][j]);
576             }
577             printf("\n");
578         }
579         printf("\n\n");

```

El segundo caso del menú se trata del producto de dos matrices de 3x3; aquí se hace uso nuevamente de la estructura de repetición for, así como el uso de los arreglos bidimensionales para poder hacer las matrices.

Se usan nuevamente las variables “mat1[3][3]” y “mat2[3][3]”.

Primeramente, existe un for con la variable i, que permite agregar el valor entero de cada renglón 1, 2 y 3; dentro de este for existe otro for con la variable j que permite agregar el valor entero por cada columna 1, 2 y 3, por eso ambas variables están dadas desde el valor 0 y menor a 3.

El escáner nos permite capturar estos valores para formar la matriz.

Posteriormente existe un apartado que imprime la matriz resultante, usándose un tabulador para dar un espacio entre cada columna dentro del for de la variable j y luego en el for de la variable i un salto de línea entre cada renglón.

El mismo proceso ocurre para determinar la segunda matriz:

```
580 //Segunda matriz
581 printf("\tSEGUNDA MATRIZ\n\n");
582 for (int i=0;i<3;i++)
583 {
584     for (int j=0;j<3;j++)
585     {
586         printf("Valor que deseas agregar al renglón %i de la columna %i: ",ao,i+1,j+1);
587         scanf("%i",&mat2[i][j]);
588     }
589     printf("\n\n");
590 }
591 printf("Matriz resultante: \n");
592 for (int i=0;i<3;i++)
593 {
594     for (int j=0;j<3;j++)
595     {
596         printf("%i\t",mat2[i][j]);
597     }
598     printf("\n");
599 }
600 printf("\n\n");
```

Para poder realizar el producto de ambas matrices se tuvo que recurrir a una nueva matriz auxiliar con un nuevo arreglo:

```
601     printf("El producto de ambas matrices es de: \n");
602     for (int i=0;i<3;i++)
603     {
604         for (int j=0;j<3;j++)
605         {
606             mat3[i][j]=0;
607             for (l=0;l<3;l++)
608             {
609                 mat3[i][j]=(mat1[i][l]*mat2[l][j])+mat3[i][j];
610             }
611             matf[i][j]=mat3[i][j];
612             printf("%i\t",matf[i][j]);
613             //printf("%i,%i\n",i,j);
614         }
615         printf("\n");
616     }
617     printf("\n\n");
618     break;
```

En esta ocasión se puede observar que hacemos uso de una nueva variable, la matriz 3 (“mat3[3][3]”) y un tercer ciclor for, el cual es el encargado de darle vida a la matriz 3; lo que hace este ciclo es asignar un valor a la variable “l” partiendo desde 0 y menor a 3, para así poder implementar este nuevo arreglo de tres dimensiones.

Lo que hace el valor de “l” es armar un nuevo orden de matriz que ayudará a que la variable 1 y 2 puedan multiplicarse entre ellas, cada renglón por cada columna y finalmente sumarle la matriz 3 para obtener la multiplicación entre ambas matrices.

Claro ejemplo de un arreglo tridimensional y su implementación.

```

619           case 3://determinante
620               printf("Ingrese la matriz a la que le deseas obtener su determinante: \n");
621               for (int i=0;i<3;i++)
622               {
623                   for (int j=0;j<3;j++)
624                   {
625                       printf("Valor que deseas agregar al renglón %i de la columna %i: ",ao,i+1,j+1);
626                       scanf("%i",&mat1[i][j]);
627                   }
628                   printf("\n");
629               }
630               printf("Matriz resultante: \n");
631               for (int i=0;i<3;i++)
632               {
633                   for (int j=0;j<3;j++)
634                   {
635                       printf("%i\t",mat1[i][j]);
636                   }
637                   printf("\n");
638               }
639               printf("\n\n");

```

El tercer caso del menú de matrices el cálculo del determinante de una matriz 3x3, donde lo primero que hace el programa es todo el proceso antes mencionado para capturar e imprimir la matriz resultante.

Nuevamente se vuelve a usar la variable “mat1[3][3]” y la estructura de repetición for, tanto para capturar los valores de la matriz como para imprimir la matriz.

```

640   D1=mat1[0][0]*((mat1[1][1]*mat1[2][2])-(mat1[2][1]*mat1[1][2]));
641   D2=-1*(mat1[0][1]*((mat1[1][0]*mat1[2][2])-(mat1[2][0]*mat1[1][2])));
642   D3=mat1[0][2]*((mat1[1][0]*mat1[2][1])-(mat1[2][0]*mat1[1][1]));
643   det=D1+D2+D3;
644   printf("Respuesta: \n");
645   printf("El determinante de la matriz es: %i",det);
646   printf("\n\n");
647   break;

```

Para calcular el determinante se recurrió a la suma de 3 determinantes internos, donde los primeros arreglos ([0][0] [0][1] [0][2]) son los coeficientes que van a multiplicar al determinante de los subdeterminantes de los valores internos de la matriz de 3x3, esto de acuerdo con el método de cofactores implementados en matrices y arreglos bidimensionales.

Aquí se usan las variables “D1”, “D2”, “D3” quienes se encarga de encontrar el valor de cada cofactor de la matriz interna y la variable “det”, la cual es la suma de “D1”, “D2” y “D3” y es la misma que va a imprimir el determinante final.

Idea de cómo surgió el método de cofactores mediante los arreglos:

$$A = \begin{pmatrix} a & b & c \\ d & e & f \\ g & h & i \end{pmatrix} \Rightarrow Arreglo = \begin{pmatrix} [0][0] & [0][1] & [0][2] \\ [1][0] & [1][1] & [1][2] \\ [2][0] & [2][1] & [2][2] \end{pmatrix}$$

$$detA = a(ei - hf) - b(di - gf) + c(dh - ge)$$

```

648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
    case 4://Traza
        printf("Ingrese la matriz a la que le desea calcular su traza: \n");
        for (int i=0;i<3;i++)
        {
            for (int j=0;j<3;j++)
            {
                printf("Valor que deseas agregar al rengl%cn %i de la columna %i: ",ao,i+1,j+1);
                scanf("%i",&mat1[i][j]);
            }
            printf("\n");
        }
        printf("\n\n");
        printf("Matriz resultante: \n");
        for (int i=0;i<3;i++)
        {
            for (int j=0;j<3;j++)
            {
                printf("%i\t",mat1[i][j]);
            }
            printf("\n");
        }
        printf("\n\n");

```

El cuarto caso del menú de matrices es el cálculo de la traza de la matriz de 3x3, la cual comienza con todo el proceso antes mencionado para poder capturar los valores de cada renglón y columna de la matriz e imprimir la matriz resultante. Se vuelve a usar la variable “mat1[3][3]”, los arreglos bidimensionales y la estructura de repetición for.

```

670
671
672
673
674
        printf("Respuesta: \n");
        tr=mat1[0][0]+mat1[1][1]+mat1[2][2];
        printf("La traza de la matriz es: %i\n",tr);
        printf("\n\n");
        break;

```

Finalmente, para obtener la traza de la matriz (recordando que la traza de la matriz es la suma de los componentes de la diagonal) se recurre nuevamente a la representación gráfica del arreglo para poder determinar la suma.

Se usa una nueva variable denominada “tr” que se obtiene mediante la suma de cada componente de la diagonal de la matriz capturada.

La idea salió a partir de la teoría de las matrices:

$$A = \begin{pmatrix} a & b & c \\ d & e & f \\ g & h & i \end{pmatrix} \Rightarrow Arreglo = \begin{pmatrix} [0][0] & [0][1] & [0][2] \\ [1][0] & [1][1] & [1][2] \\ [2][0] & [2][1] & [2][2] \end{pmatrix}$$

$$tr(A) = a + e + i$$

```

675     case 5://Transpuesta
676     printf("Ingrese la matriz a la que se le desea obtener su transpuesta: \n");
677     for (int i=0;i<3;i++)
678     {
679         for (int j=0;j<3;j++)
680         {
681             printf("Valor que deseas agregar al renglon %i de la columna %i: ",ao,i+1,j+1);
682             scanf("%i",&mat1[i][j]);
683         }
684         printf("\n\n");
685     }
686     printf("Matriz resultante: \n");
687     for (int i=0;i<3;i++)
688     {
689         for (int j=0;j<3;j++)
690         {
691             printf("%i\t",mat1[i][j]);
692         }
693         printf("\n");
694     }
695     printf("\n\n");

```

El quinto caso del menú de las matrices es el de matriz transpuesta, la cual, primeramente, vuelve a realizar todo el proceso para capturar los valores de cada renglón y fila de la matriz deseada.

Se vuelve a usar la variable “mat1[3][3]” como la matriz original.

```

696     printf("La matriz transpuesta a la original es: \n");
697     for (int i=0;i<3;i++)
698     {
699         for (int j=0;j<3;j++)
700         {
701             matf[j][i]=mat1[j][i];
702             printf("%i\t",matf[j][i]);
703         }
704         printf("\n");
705     }
706     printf("\n\n");
707     break;

```

Para poder calcular la matriz transpuesta, volvemos a hacer uso de la variable “matf[3][3]” para imprimir el resultado final, solo que presenta una diferencia; cuando nosotros imprimimos la matriz original se presenta en renglones y columnas, pero al pasarla a la transpuesta se presenta como columnas y renglones, es decir, la matriz final lo que hace es cambiar de lugar columnas y renglones.

Los renglones pasan al lugar de las columnas y las columnas al lugar de los renglones.

```
708 case 6: //Salir  
709     printf("Volviendo al men%c anterior...\n\n",au);  
710     break;  
711 default:  
712     printf("Null");
```

Como último caso y no menos importante, el caso sexto del menú de matrices es el encargado de cerrar el menú de matrices y regresar al usuario al menú principal de la calculadora AGAM.

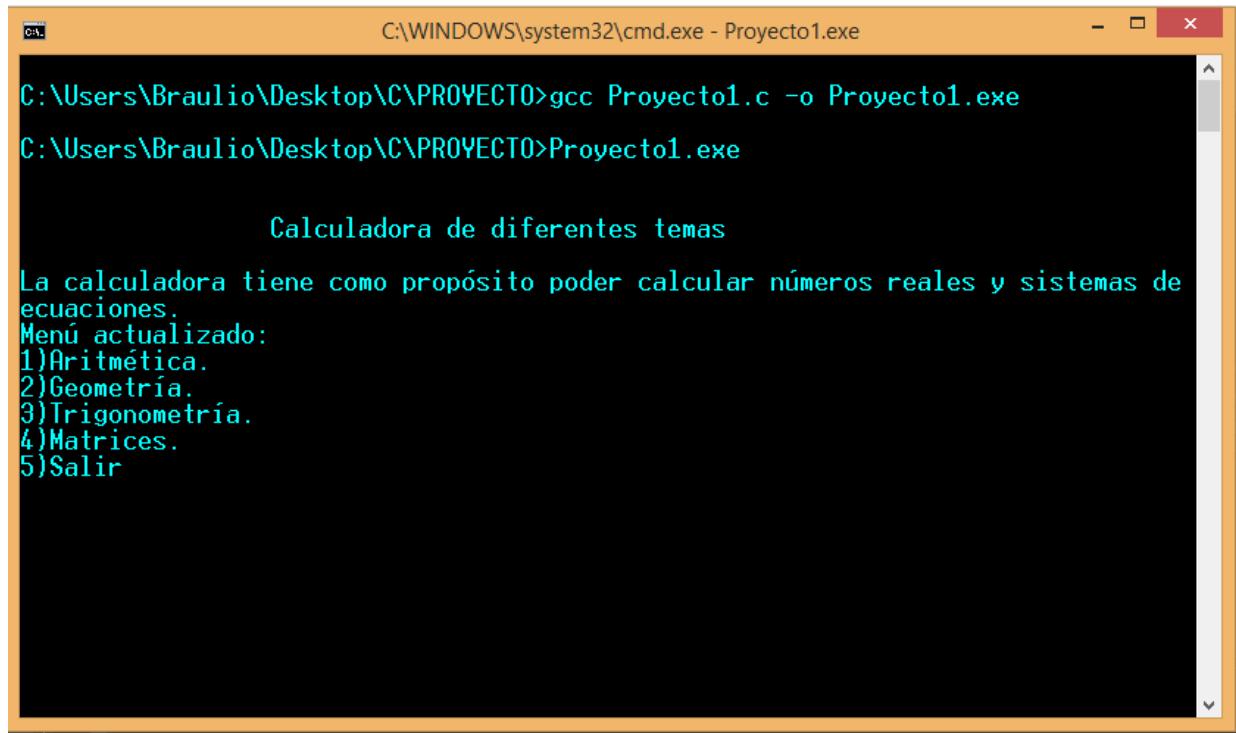
Con esto se cierran todas las opciones del menú de la calculadora TAGAM, todo lo que fue aritmética, geometría, trigonometría y matrices y todo lo que fue el desarrollo.

```
724 }while(opI!=5);  
725 //FIN MENU PRINCIPAL  
726 printf("Espero te haya gustado...\n\n");  
727 return 0;  
728 }
```

Finalmente, el proyecto concluye con el cierre del menú principal una vez que se seleccionó la opción 5 y ya no se repite el ciclo do-while.

Se puede apreciar la leyenda “Espero te haya gustado” en señal de que el proyecto a acabado y se ha cerrado lo necesario del menú y concluido lo que tuvo que haber concluido.

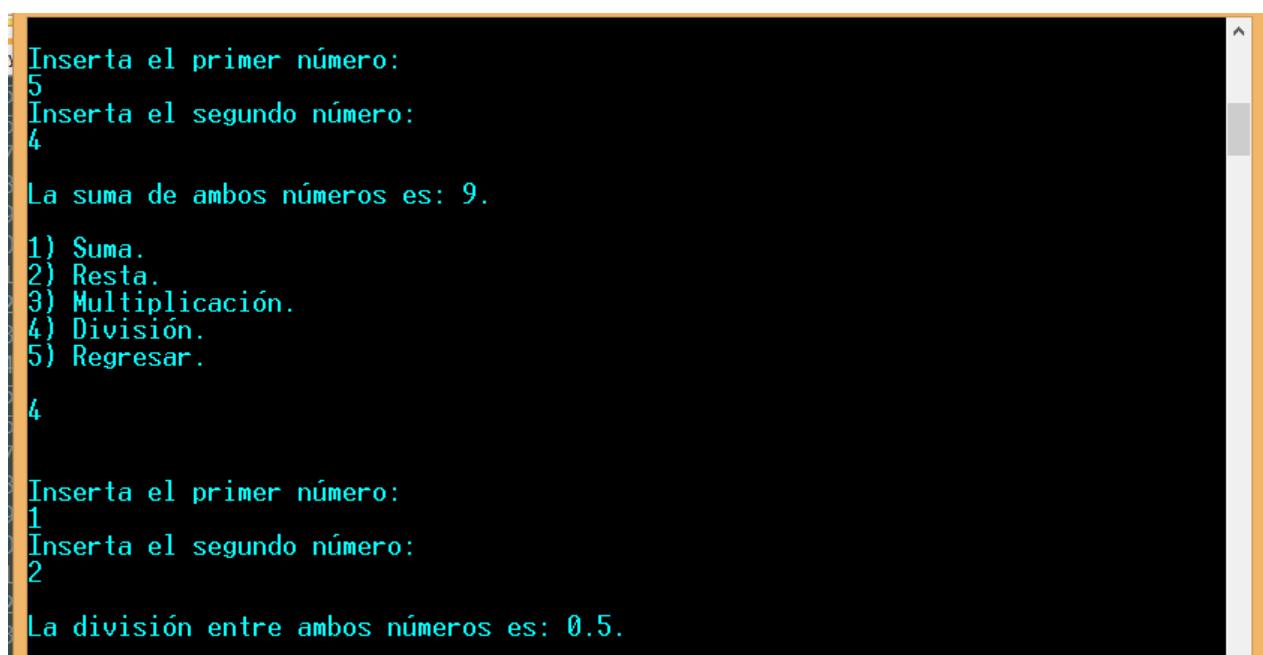
EVIDENCIAS



```
C:\WINDOWS\system32\cmd.exe - Proyecto1.exe
C:\Users\Braulio\Desktop\C\PROYECTO>gcc Proyecto1.c -o Proyecto1.exe
C:\Users\Braulio\Desktop\C\PROYECTO>Proyecto1.exe

        Calculadora de diferentes temas

La calculadora tiene como propósito poder calcular números reales y sistemas de
ecuaciones.
Menú actualizado:
1)Aritmética.
2)Geometría.
3)Trigonometría.
4)Matrices.
5)Salir
```



```
Inserta el primer número:
5
Inserta el segundo número:
4

La suma de ambos números es: 9.

1) Suma.
2) Resta.
3) Multiplicación.
4) División.
5) Regresar.

4

Inserta el primer número:
1
Inserta el segundo número:
2

La división entre ambos números es: 0.5.
```

```
Inserta el primer número:  
5  
Inserta el segundo número:  
0  
La división entre 0 no se puede efectuar.  
1) Suma.  
2) Resta.  
3) Multiplicación.  
4) División.  
5) Regresar.
```

```
2) Rectángulo.  
3) Triángulo.  
4) Círculo.  
5) Rombo.  
6) Trapecio.  
7) Pentágono regular.  
8) Regresar.  
4  
Ingresa el valor del radio:  
2.5  
El área del círculo es de: 19.634954 unidades cuadradas.  
1) Cuadrado.  
2) Rectángulo.  
3) Triángulo.  
4) Círculo.  
5) Rombo.  
6) Trapecio.  
7) Pentágono regular.  
8) Regresar.
```

```
1) Suma  
2) Resta  
3) Regresar  
2
```

Ingrese las dos matrices que deseé restar:
PRIMER MATRIZ

Valor que deseas agregar al renglón fila 1 de la columna 1: -1
Valor que deseas agregar al renglón fila 1 de la columna 2: 6
Valor que deseas agregar al renglón fila 1 de la columna 3: -10

Valor que deseas agregar al renglón fila 2 de la columna 1: 4
Valor que deseas agregar al renglón fila 2 de la columna 2: 7
Valor que deseas agregar al renglón fila 2 de la columna 3: 2

Valor que deseas agregar al renglón fila 3 de la columna 1: -5
Valor que deseas agregar al renglón fila 3 de la columna 2: 0
Valor que deseas agregar al renglón fila 3 de la columna 3: -6

Matriz resultante:

```
-1      6      -10  
4      7       2  
-5      0      -6
```

SEGUNDA MATRIZ

Valor que deseas agregar al renglón 1 de la columna 1: -1
Valor que deseas agregar al renglón 1 de la columna 2: 4
Valor que deseas agregar al renglón 1 de la columna 3: 3

Valor que deseas agregar al renglón 2 de la columna 1: -6
Valor que deseas agregar al renglón 2 de la columna 2: -5
Valor que deseas agregar al renglón 2 de la columna 3: 0

Valor que deseas agregar al renglón 3 de la columna 1: 3
Valor que deseas agregar al renglón 3 de la columna 2: 2
Valor que deseas agregar al renglón 3 de la columna 3: -6

Matriz resultante:

```
-1      4      3  
-6     -5      0  
 3      2      -6
```

La resta de ambas matrices es de:

```
 0      2      -13  
10     12      2  
-8     -2      0
```

```
1) Suma  
2) Resta
```

```
2) Multiplicación entre matrices.
3) Determinante.
4) Traza.
5) Transpuesta.
6) Regresar.
3

Ingrese la matriz a la que le deseé obtener su determinante:
Valor que deseas agregar al renglón 1 de la columna 1: 2
Valor que deseas agregar al renglón 1 de la columna 2: 3
Valor que deseas agregar al renglón 1 de la columna 3: 6

Valor que deseas agregar al renglón 2 de la columna 1: 7
Valor que deseas agregar al renglón 2 de la columna 2: 4
Valor que deseas agregar al renglón 2 de la columna 3: 5

Valor que deseas agregar al renglón 3 de la columna 1: 1
Valor que deseas agregar al renglón 3 de la columna 2: 2
Valor que deseas agregar al renglón 3 de la columna 3: 0

Matriz resultante:
2      3      6
7      4      5
1      2      0

Respuesta:
El determinante de la matriz es: 55

1) Suma | Resta.
2) Multiplicación entre matrices.
3) Determinante.
4) Traza.
5) Transpuesta.
6) Regresar.
4
```

TABLA COMPARATIVA

OCTUBRE 2020		ENERO 2021	
HARDWARE	SOFTWARE	HARDWARE	SOFTWARE
Laptop lenovo	GNU Linux	Mejor procesador para tomar cursos y aprender	Adobe Media Encoder
Mouse inalámbrico sensible al tacto		Touchpad inalámbrico	Adobe Premier Pro
			Grabadora de video online

DIAGRAMA DE GANTT

Actividades	Fecha del proyecto expresado en cuatrimestres											
	Año 1			Año 2			Año 3			Año 4		
	E-A	M-A	S-D	E-A	M-A	S-D	E-A	M-A	S-D	E-A	M-A	S-D
Planear el algoritmo del proyecto	■											
Cursos y asesorías	■	■	■									
Algoritmo para expresar todas las ecuaciones existentes			■	■	■	■						
Plasmado geométrico mediante simuladores						■	■					
Programarlo				■	■	■	■	■	■	■	■	■
Física y Química									■	■	■	■

TABLA COMPARATIVA PRECIO

OCTUBRE 2020	ENERO 2021
Sin número dado	\$20000

CANAL DE YOUTUBE:

<https://www.youtube.com/channel/UCSA6N9Cn1i8c7dCtscAwXEA>

REPOSITORIO DE GITHUB:

<https://github.com/BraulioHzGz/proyecto1.git>

CONCLUSIONES

La calculadora TAGAM es un proyecto de programación con muchos buenos objetivos y en parte puede beneficiar a muchas personas en cuanto al conocimiento de las ciencias básicas, para su fácil entendimiento y refuerzo en cuanto a la parte del cálculo.

Así mismo funcionaría muy bien en servicios de educación ya sean privados o públicos o en servicios de investigación para tratar de explicar fenómenos o ayudar en las ecuaciones, así como plasmar lo que se requiere y llevarlo a muchas cosas futuras.

Beneficia mucho porque se pretende que TAGAM sea una calculadora de software libre, es decir, donde todos puedan acceder a ella sin problemas económicos, puesto que para eso no está formada ni pensada.

En cuanto al ámbito de la IEEE creo que no podría contribuir de una forma directa, puesto que la TAGAM está más orientada a la enseñanza y a la libre adquisición de conocimiento, aunque si en caso de que los trabajadores de ahí o algún participe de la IEEE requiera realizar cálculos o tratar de hacer grandes cosas de investigación en física, matemáticas o químicas pues si le serviría y quizá pensaría en tratar de hacer que TAGAM formara parte de ellos aunque es muy dudoso, pues no estoy enfocado en la industria de sistemas muy complejos o que creen un avance en la humanidad, si no, como contribución y ayuda a la gente mas o una herramienta que creara un impacto como tal.

TAGAM se resguarda sus derechos, sería gran partidaria del software libre para contribuir.

CONCLUSIÓN PERSONAL

La perspectiva de mi proyecto siguió siendo la misma, no presenté muchos cambios de vista sobre lo que en verdad quería hacer y sobre lo que pretendo llegar a hacer en un futuro con esta calculadora.

El nombre lo elegí para un proyecto en construcción, próximamente le cambiaré el nombre a uno mas sofisticado o mas llamativo que llame la atención de todo aquel que quiera usar la calculadora.

Lo que mas me gustó de este pequeño proyecto fue el hecho de tener que haber pasado días enteros y varias noches leyendo libros y buscando en internet como hacer ciertas cosas, que tan complejos podían llegar a ser algunos comandos y el hecho de que al final me haya salido como yo esperaba.

Una de mis secciones favoritas fue solamente la de matrices, aunque por temas de tiempo no pude seguir con lo mío y terminar con la inversa, estoy dispuesto a crear una inversa mediante varios métodos y tener que volver a pensar que hacer y como hacerle, fue muy bueno.

Otra sección que me gustó también fue la trigonométrica, aunque es muy sencilla no pensé que pudiera llegar a hacer eso la librería y enseñar que todo es posible.

Una parte que no me gustó fue la de aritmética y la de geometría, puesto que serán mis bases para en un futuro poder resolver sistemas de ecuaciones y volúmenes o integrales mas complejos que puedan requerir de eso, no me gustó la simpleza con la que lo hice, pero, por algo se empieza.

REFERENCIAS

Bibliografía virtual:

- Jorge. (s.f). Determinante de una matriz. Ejemplos gratis de programación [Foro de ayuda]. Recuperado de: <http://jamper91.blogspot.com/2013/08/determinante-de-una-matriz-c.html>
- Andrade, Bases S.A de C.V (2012, marzo 7). Programa para sumar, restar y multiplicar matrices. Compu-Tec [Foro de ayuda]. Recuperado de: <http://lalo-c.blogspot.com/2012/03/programa-para-sumar-restar-o.html>
- Martínez, Marco. (s.f). Desarrollo de software mediante lenguaje C. Sistema de cursos en linea. Recuperado de: http://solucionesmyl.com/cursos/lenguaje_c/home.php
- (s.f). Soluciones. Symbolab [Calculadora en linea], Recuperado de: <https://es.symbolab.com/>

Bibliografía física:

- López, Norma. (s.f). Propiedades de matrices y determinantes. Facultad de Ingeniería, UNAM, División de ciencias básicas, álgebra lineal.
- CCPM. (2015). Programación con visual net [avanzado]. 1^a Edición. Doctores, Miguel Hidalgo, CDMX. Editorial Mc Graw Hill Education.
- CCPM. (2015). Programación con visual net [experto]. 2^a Edición. Doctores, Miguel Hidalgo, CDMX. Editorial Mc Graw Hill Education.
- Ceballos, Jorge. (2013, marzo). Visual Basic. Net [Curso de programación]. 1^a Edición. México. Editorial Alfaomega Ra-Ma.
- -. (2013). Introducción al lenguaje C y C++. 2^a Edición. México. Editorial Mc Graw Hill Education.

Bibliografía de apuntes:

- Apuntes de CCPM Java y C. (año 2013) [secundaria].
- Apuntes de CCPM Visual Basic. (año 2013). [secundaria].
- Apuntes de matemáticas. [primaria].
- Apuntes de áreas y perímetros de figuras. [primaria].
- Apuntes de círculos y cilindros. [secundaria].
- Apuntes de determinantes. [bachillerato].
- Apuntes de cálculo. [bachillerato]
- Apuntes de matrices y sistemas de ecuaciones lineales. [licenciatura].

Documentación oficial del lenguaje C:

- C11 standar (ISO/IEC 9899:2011):
 - o 7 Library
- C99 standar (ISO/IEC 9899:1999)
 - o 7 Library
- <tgmath.h>
- <math.h>
- <stdio.h>
- <string.h>
- <stdarg.h>
- <errno.h>
- <stdlib.h>
- <ctype.h>

Consultas a profesionales independientes:

- Marco Antonio Martínez Quintana.
 - o Solucionesmyl.
 - o Facultad de Ingeniería de la UNAM.
 - o Preguntas sobre return, funciones.
 - o 14 diciembre
 - o E-mail (Outlook).
- José Luis González.
 - o Centro de Computación Profesional de México (CCPM).
 - o Arreglos en C, metacomandos.
 - o 28 diciembre, 3 enero.
 - o Mensaje virtual (Messenger).
- José Carlos Munguía
 - o Centro de Computación Profesional de México.
 - o Preguntas sobre while y for.
 - o 29 diciembre.
 - o Mensaje virtual (Messenger).
 - o Correo personal (Gmail).

Glosario de términos independientes:

- Calculadora TAGAM: calculadora de trigonometría, álgebra, geometría, ángulos y matrices (TAGAM).
- SHARPE: apodo a un programador cercano al creador de este documento que falleció hace tiempo.
- Bugs: fallo.

Acrónimos o abreviaturas del proyecto:

- aa: acento en la a.
- ae: acento en la e.
- ai: acento en la i.
- ao: acento en la o.
- au: acento en la u.
- aA: acento en la A.
- opl: opción Inicial.
- opA: opción Aritmética.
- opGM: opción Geometría.
- GA: Geometría Áreas.
- GP: Geometría Perímetros.
- opT: opción Trigonometría.
- opANG: opción Ángulos.
- opMT: opción Matriz.
- mat1: matriz 1.
- mat2: matriz 2.
- mat3: matriz 3.
- matf: matriz final.
- SRM: Suma y Resta de Matrices.
- D1: Determinante 1.
- D2: Determinante 2.
- D3: Determinante 3.