

Universidad de San Carlos de Guatemala

Facultad de Ingeniería

Arquitectura de Computadores y Ensambladores 1

Sección N

Fecha: 23/12/2022



MANUAL TÉCNICO

Nombre:

Gerson Rubén Quiroa del Cid

Carné:

2020 00166

Índice

Introducción	3
.model.....	4
.data.....	4
.data.....	6
MACRO printLinea.....	7
MACRO clearPantalla	7
MACRO printMenu.....	7
MACRO getChar	8
MACRO saveCoeficiente.....	8
MACRO getCoeficiente.....	9
MACRO opcion1	10
MACRO opcion2	10
MACRO getDerivada.....	11
MACRO opcion3	11
MACRO getIntegral	12
MACRO opcion4	13
MACRO opcion5	13
MACRO modoGrafico	14
MACRO opGraph1	15
MACRO opGraph2	15
MACRO opGraph2	16
MACRO graficando	16
MACRO quintaM	17
MACRO cuartaM	17
MACRO terceraM	18
MACRO segundaM	18
MACRO primeraM	19
MACRO sinCero	19

Introducción

El manual técnico tiene como finalidad de explicar el funcionamiento de cada función utilizado en el presente programa, con el objetivo de que no haya confusión al leer el código del programa y cualquier persona con conocimiento básico de programación en Assembler pueda comprender lo que se hizo en cada línea de código. Por lo cual se recomienda que toda persona que quiera leer o modificar el código vea este manual para ahorrarse tiempo y así sea más fácil su comprensión.

.model

En esta ocasión se utilizó el modelo small para realizar este programa.

.data

Aquí tenemos todas nuestras variables que utilizamos en este programa.

Primero tenemos las cadenas predefinidas y constantes, los cuales son los siguientes:

```
; variables que usaremos para mostrar texto en la consola
ln      db 0ah, '$' ;salto de línea
opErr   db "No se ha reconocido la opcion ingresada, intente de nuevo.", "$"
; menú principal
intro   db "===== CALCULADORA =====", "$"
intro2   db "Gerson Ruben Quiroa del Cid - Carnet: 202000166 ", "$"
op1      db "      (1) Ingresar funcion", "$"
op2      db "      (2) Imprimir la funcion almacenada", "$"
op3      db "      (3) Imprimir derivada de la funcion", "$"
op4      db "      (4) Imprimir integral de la funcion", "$"
op5      db "      (5) Graficar funcion", "$"
op6      db "      (6) Ceros por metodo de Newton", "$"
op7      db "      (7) Ceros por metodo de Steffensen", "$"
op8      db "      (8) Salir de la aplicacion", "$"
intro3   db "Ingrese el numero de la opcion que desea.", "$"
intro4   db "      >", "$"
```

```
graph   db "===== GRAFICAR FUNCION =====", "$"
        db "      (5.1) Funcion original", 10,13,"$"
        db "      (5.2) Funcion derivada", 10,13,"$"
        db "      (5.3) Funcion integral", 10,13,"$"

op1Intro db "Ingrese la funcion que desee guardar por coeficientes. Ingresar solo numero enteros", "$"
op1x6    db "X^6 ", "$"
op1x5    db "X^5 ", "$"
op1x4    db "X^4 ", "$"
op1x3    db "X^3 ", "$"
op1x2    db "X^2 ", "$"
op1x1    db "X^1 ", "$"
op1x0    db "X^0 ", "$"
dosPuntos db ":", "$"

op2Intro db "La funcion almacenada es la siguiente f(x) = ", "$"
mas      db "+", "$"
menos    db "-", "$"

op3Intro db "La derivada de la funcion almacenada es la siguiente (d/dx)f(x) = ", "$"
```

Mensajes de error y de función en construcción:

```
msgError    db "ERROR. Se esperaba que ingrese un número de 0 a 9, por favor intente de nuevo.", "$"
enProceso   db "Estamos en proceso de construcción :D", "$"
```

Se tiene 2 variables, una llamada u y la otra llamada d, la primera será para manejar la unidad de un número de 2 dígitos y el otro para manejar la decena de dicha unidad, respectivamente.

También se tiene una variable negativo, la cual se usa como flag o boolean, para saber cuando un número ingresado es negativo o positivo.

El multiplicador en este caso, no servía a la hora de derivar e integrar, para saber por qué número habrá que multiplicar.

```
u          db 0 ; unidad del número
d          db 0 ; decena del número
negativo    db 0 ; 0 = número positivo, 1 = número negativo
multiplicador db 0
```

Las variables xn, dxn y sxn, donde n es el número del exponente de 6 a 0, donde guardaremos los coeficientes correspondientes para cada caso.

```
; funcion normal
x5          db 0 ; coeficiente para x^5
x4          db 0 ; coeficiente para x^4
x3          db 0 ; coeficiente para x^3
x2          db 0 ; coeficiente para x^2
x1          db 0 ; coeficiente para x^1
x0          db 0 ; coeficiente para x^0
; derivada
dx4         db 0 ; coeficiente para x^5
dx3         db 0 ; coeficiente para x^4
dx2         db 0 ; coeficiente para x^3
dx1         db 0 ; coeficiente para x^2
dx0         db 0 ; coeficiente para x^1
; integral
sx6         db 0 ; coeficiente para x^5
sx5         db 0 ; coeficiente para x^4
sx4         db 0 ; coeficiente para x^3
sx3         db 0 ; coeficiente para x^2
sx2         db 0 ; coeficiente para x^1
sx1         db 0 ; coeficiente para x^0
```

.data

Pasamos al segmento de código, aquí estará toda nuestra lógica para que el programa se ejecute de manera correcta.

Se tiene el procedimiento main, donde estará nuestro menú con las diferentes opciones.

```
main PROC ; proceso main
clearPantalla ; limpiamos la consola
MENU:
    printMenu
    getChar
    cmp al, '1' ; comparamos si se seleccionó la op 1
    je UNO
    cmp al, '2' ; comparamos si se seleccionó la op 2
    je DOS
    cmp al, '3' ; comparamos si se seleccionó la op 3
    je TRES
    cmp al, '4' ; comparamos si se seleccionó la op 4
    je CUATRO
    cmp al, '5' ; comparamos si se seleccionó la op 5
    je CINCO
    cmp al, '6' ; comparamos si se seleccionó la op 6
    je SEIS
    cmp al, '7' ; comparamos si se seleccionó la op 7
    je SIETE
    cmp al, '8' ; comparamos si se seleccionó la op 8
    je SALIDA
    jne NOOP
```

```
UNO: ; nos vamos a la opcion uno
    opcion1

DOS: ; nos vamos a la opcion dos
    opcion2

TRES: ; nos vamos a la opcion tres
    opcion3

CUATRO: ; nos vamos a la opcion cuatro
    opcion4

CINCO: ; nos vamos a la opcion cinco
    clearPantalla
    printLinea ln, 0d
    printLinea enProceso, 10d
    jmp MENU

SEIS: ; nos vamos a la opcion seis
    clearPantalla
    printLinea ln, 0d
    printLinea enProceso, 10d
    jmp MENU
```

MACRO printLinea

Esta macro se utiliza para imprimir líneas en pantalla, se pasaran 2 parámetros los cuales son: txt que es el mensaje a imprimir, y color que es el color en que se imprimirá el mensaje.

```
printLinea MACRO txt, color ; mac
; cargamos en memoria las var
mov ax, @data
mov ds, ax
; imprimimos en pantalla
mov ah, 09h
mov bl, color
mov cx, lengthof txt - 1
int 10h
lea dx, txt
int 21h
11 references
ENDM
```

MACRO clearPantalla

Este macro se utiliza para limpiar la pantalla mientras se vaya navegando por el menú.

```
10 references
clearPantalla MACRO
mov ax, 03h
int 10h
11 references
ENDM
```

MACRO printMenu

Esta macro simplemente imprime el menú, con el título del programa y las opciones a elegir.

```

printMenu MACRO ; macro para imprimir el menú principal
    printlinea ln, 0d ; imprimimos un salto de línea por cada línea
    printlinea ln, 0d
    printlinea ln, 0d
    printlinea ln, 0d
    printlinea intro, 5d
    printlinea ln, 0d
    printlinea intro2, 10d
    printlinea ln, 0d
    printlinea op1, 10d
    printlinea ln, 0d
    printlinea op2, 10d
    printlinea ln, 0d
    printlinea op3, 10d
    printlinea ln, 0d
    printlinea op4, 10d
    printlinea ln, 0d
    printlinea op5, 10d
    printlinea ln, 0d
    printlinea op6, 10d
    printlinea ln, 0d
    printlinea op7, 10d
    printlinea ln, 0d
    printlinea op8, 10d
    printlinea ln, 0d

```

MACRO getChar

Esta macro es la encargada de obtener un char desde el teclado e interrumpir el sistema.

```

1 reference
getChar MACRO ;macro
    mov ah, 01h
    int 21h
11 references
ENDM

```

MACRO saveCoeficiente

Esta macro es la encargada de guardar en una variable pasada por parámetro, el coeficiente de cada x, también sabrá si el usuario ingresó un – o +, para controlar su signo y también controlará los errores en caso el usuario ingresa algún carácter que no sea número. Consta de 2 parámetros, xn es la x de quien se quiere el coeficiente y var es la variable donde se guardará dicho coeficiente.

6 references

```
saveCoeficiente MACRO xn, var ; aquí leeremos los coeficientes de
    LOCAL LNEGATIVO, LCAPTURAR, LPOSITIVO, RESTA, SALIR, LERROR
    mov ax, @data
    mov ds, ax
    ; imprimimos el coeficiente de quien se quiere
    printLinea xn, 7d
    printLinea dosPuntos, 7d
    ; capturamos el primer dígito del número (la unidad)
    mov ah, 01h
    int 21h
    ; primero miramos si se trata de un número negativo
    cmp al, '-'
    je LNEGATIVO
    ; miramos si pone el signo positivo
    cmp al, '+'
    je LPOSITIVO
    jmp LCAPTURAR

LPOSITIVO:
    ; capturamos el primer dígito del número (la unidad)
    mov ah, 01h
    int 21h
    jmp LCAPTURAR
```

MACRO getCoeficiente

Esta macro es la encargada de obtener los coeficientes a la hora de imprimir las funciones, se pasan 2 parámetros uno es el var, que es la variable donde se guardó el coeficiente, y el otro xn que es el la x al exponente que le corresponde.

17 references

```
getCoeficiente MACRO var, xn ; macro para obtener los números g
    LOCAL MOSTRAR, POSITIVO, LNEGATIVO, PONERX, NEGAR, SALIDA
    mov al, var
    cmp al, 0
    jg POSITIVO
    jl LNEGATIVO
    jmp SALIDA

LNEGATIVO:
    printLinea menos, 7d
    ; negamos los negativos para poder mostrarlos en pantalla
    mov al, var
    neg al
    mov var, al
    ; y ponemos negativo en 1, para saber que son negativos
    mov al, negativo
    add al, 1
    mov negativo, al
    jmp MOSTRAR

POSITIVO:
    printLinea mas, 7d
    jmp MOSTRAR
```

MACRO opcion1

En esta macro se encuentra la opción 1 del programa que sería el de ingresar 1 función por coeficientes, primero se imprime la x que se solicita y luego el usuario puede ingresar un número de 2 dígitos con o sin signo. Luego salta al menú nuevamente.

```
1 reference
opcion1 MACRO ; opcion 1 que debe most
    clearPantalla
    printLinea ln, 0d
    printLinea op1Intro, 10d
    printLinea ln, 0d
    ; ingresamos x5
    saveCoeficiente op1x5, x5
    ; ingresamos x4
    saveCoeficiente op1x4, x4
    ; ingresamos x3
    saveCoeficiente op1x3, x3
    ; ingresamos x2
    saveCoeficiente op1x2, x2
    ; ingresamos x1
    saveCoeficiente op1x1, x1
    ; ingresamos x0
    saveCoeficiente op1x0, x0
    jmp MENU
11 references
ENDM
```

MACRO opcion2

En esta macro se encuentra la opción 2 del programa en donde se mostrará la función ingresada por el usuario. Primero habrá un mensaje donde le diga al usuario que esta es la función que ingresó, seguido por dicha función.

```
1 reference
opcion2 MACRO ; opción para mo
    clearPantalla
    printLinea ln, 0d
    printLinea op2Intro, 10d
    printLinea ln, 0d
    getCoeficiente x5, op1x5 ;
    getCoeficiente x4, op1x4 ;
    getCoeficiente x3, op1x3 ;
    getCoeficiente x2, op1x2 ;
    getCoeficiente x1, op1x1 ;
    getCoeficiente x0, menos ;
    jmp MENU
11 references
ENDM
```

MACRO getDerivada

En esta macro se realizará la derivada de la función por cada coeficiente. Esta macro recibirá 3 parámetros, la primera llamada var es la variable del coeficiente a derivar, xn es el exponente por el cual se multiplicará, y por último dvar que será donde se guardará el coeficiente derivado.

```
getDerivada MACRO var, xn, dvar ; macro para obt
    LOCAL EXP4, EXP3, EXP2, EXP1, EXP0, SALIDA
    ; var = número a multiplicar, xn = número mu
    ; realizamos la multiplicacion
    mov al, var
    mov bl, xn
    mul bl
    mov dvar, al
    ; restamos 1 al multiplicador xn
    mov al, xn
    sub al, 1
    ; comparamos para saber que x^n le toca
    cmp al, 4
    je EXP4
    cmp al, 3
    je EXP3
    cmp al, 2
    je EXP2
    cmp al, 1
    je EXP1
    cmp al, 0
    je EXP0
    jmp SALIDA
    ; mostramos el coeficiente correspondiente
EXP4:
```

MACRO opcion3

En esta macro se encuentra la opción 3 del programa en donde se mostrará la función derivada de la función original. Primero habrá un mensaje donde le diga al usuario que esta es la función derivada de la función que ingresó, seguido por dicha función derivada.

```

1 reference
opcion3 MACRO ; macro para mostrar la de
    clearPantalla
    printLinea ln, 0d
    printLinea op3Intro, 10d
    printLinea ln, 0d
    mov multiplicador, 5
    getDerivada x5, multiplicador, dx4 ;
    mov multiplicador, 4
    getDerivada x4, multiplicador, dx3 ;
    mov multiplicador, 3
    getDerivada x3, multiplicador, dx2 ;
    mov multiplicador, 2
    getDerivada x2, multiplicador, dx1 ;
    mov multiplicador, 1
    getDerivada x1, multiplicador, dx0 ;
    jmp MENU
11 references
ENDM

```

MACRO getIntegral

En esta macro se realiza la integral de la función original. Se reciben 3 parámetros, var es la variable donde se encuentra el coeficiente a integrar, xn es el exponente al cual se va a dividir y svar es la variable donde se guardará el coeficiente ya integrado.

```

6 references
getIntegral MACRO var, xn, svar ; macro para obtener la integral
    LOCAL LOG, LOG2, EXP6, EXP5, EXP4, EXP3, EXP2, EXP1, NEGAR, NEGAR2, SALIDA
    ; revisamos si se trata de un número negativo, para pasarlo a positivo para
    mov al, var
    cmp al, 0
    jl NEGAR
    jmp LOG

    NEGAR:
        ; ponemos en 1 el negativo para saber que es un número negativo
        mov al, negativo
        add al, 1
        mov negativo, al
        ; negamos el número para tener su positivo
        mov al, var
        neg al
        mov var, al
        jmp LOG

    NEGAR2:
        ; ponemos en 1 el negativo para saber que es un número negativo
        mov al, negativo
        sub al, 1
        mov negativo, al

```

MACRO opcion4

En esta macro se encuentra la opción 4 del programa en donde se mostrará la función integrada de la función original. Primero habrá un mensaje donde le diga al usuario que esta es la función integrada de la función que ingresó, seguido por dicha función integrada.

```
1 reference
✓ opcion4 MACRO ; macro para la función 4
    clearPantalla
    printLinea ln, 0d
    printLinea op4Intro, 10d
    printLinea ln, 0d
    mov multiplicador, 5
    getIntegral x5, multiplicador, sx6 ; imp
    mov multiplicador, 4
    getIntegral x4, multiplicador, sx5 ; imp
    mov multiplicador, 3
    getIntegral x3, multiplicador, sx4 ; imp
    mov multiplicador, 2
    getIntegral x2, multiplicador, sx3 ; imp
    mov multiplicador, 1
    getIntegral x1, multiplicador, sx2 ; imp
    mov multiplicador, 0
    getIntegral x0, multiplicador, sx1 ; imp
    jmp MENU
11 references
ENDM
```

MACRO opcion5

En esta macro se encuentra la opción 4 del programa en donde se mostrarán las gráficas de las funciones (original, derivada e integral). Se mostrará otro menú donde podrá elegir la opción de la función que desee graficar.

```
opcion5 MACRO ; macro para la función 5
    LOCAL GUNO, GDOS, GTRES, GNOOP, GMENU, SALIDA
    clearPantalla
    GMENU:
        printlinea ln, 0d
        printlinea graph, 10d
        printlinea ln, 0d
        printlinea graph2, 10d
        printlinea ln, 0d
        printlinea graph3, 10d
        printlinea ln, 0d
        printlinea graph4, 10d
        printlinea ln, 0d
        printlinea graph5, 10d
        printlinea ln, 0d
        getChar
        cmp al, '1' ; grafica 1
        je GUNO
        cmp al, '2' ; grafica 2
        je GDOS
        cmp al, '3' ; grafica 3
        je GTRES
        cmp al, '4' ; grafica 4
        je SALIDA
        jne GNOOP
ENDMACRO
```

MACRO modoGrafico

En esta macro se dibujará el plano cartesiano de color amarillo. Y dependiendo de qué opción eligió el usuario, entrará a otra macro que es la encargada de graficar la función.

```
modoGrafico MACRO var
    LOCAL HORIZONTAL, VERTICAL, GUNO, GDOS, GTRES, SALIDA
    clearPantalla
    ; RESETEAMOS CONTADORES
    mov xmax, 0
    mov ymax, 0
    mov color, 14
    ; cambiamos a modo gráfico
    mov ah, 00
    mov al, 12h
    int 10h ; interrupcion bios

    ; pintar un pixel (prueba)
    mov cx, 0 ; columna del pixel x
    HORIZONTAL:
        mov ah, 0ch ; ppinta un pixel
        mov al, color ; color
        mov bh, 0h ; pagina 0
        mov dx, y ; renglon y
        int 10h
        inc cx
        inc xmax
        cmp xmax, 640
        jne HORIZONTAL

    mov dx, 0
    VERTICAL:
        mov ah, 0ch ; ppinta un pixel
        mov al, color ; color
        mov bh, 0h ; pagina 0
        mov cx, x ; renglon y
        int 10h
    ENDMACRO
```

MACRO opGraph1

En esta macro se graficará la función original. Dependiendo de coeficiente mayor que se encuentre diferente a cero, empezará a graficar la forma y curvas características de la función.

```
opGraph1 MACRO ; aqui graficamos la funcion normal
LOCAL GCINCO, GCUATRO, GTRES, GDOS, GUNO, GCERO, SALIDA
mov al, x5
cmp al, 0
jne GCINCO
mov al, x4
cmp al, 0
jne GCUATRO
mov al, x3
cmp al, 0
jne GTRES
mov al, x2
cmp al, 0
jne GDOS
mov al, x1
cmp al, 0
jne GUNO
jmp GCERO

GCINCO:
    graficando 5, x0
    jmp SALIDA

GCUATRO:
    graficando 4, x0
    jmp SALIDA
```

MACRO opGraph2

En esta macro se graficará la función derivada. Dependiendo de coeficiente mayor que se encuentre diferente a cero, empezará a graficar la forma y curvas características de la función.

```
opGraph2 MACRO
LOCAL GCUATRO, GTRES, GDOS, GUNO, GCERO, SALIDA
mov al, dx4
cmp al, 0
jne GCUATRO
mov al, dx3
cmp al, 0
jne GTRES
mov al, dx2
cmp al, 0
jne GDOS
mov al, dx1
cmp al, 0
jne GUNO
jmp GCERO

GCUATRO:
    graficando 4, dx0
    jmp SALIDA

GTRES:
    graficando 3, dx0
    jmp SALIDA

GDOS:
    graficando 2, dx0
    jmp SALIDA
```

MACRO opGraph2

En esta macro se graficará la función integral. Dependiendo de coeficiente mayor que se encuentre diferente a cero, empezará a graficar la forma y curvas características de la función.

```
opGraph3 MACRO
    LOCAL GSEIS, GCINCO, GCUATRO, GTRES, GDOS, GUNO, SALIDA
    mov al, sx6
    cmp al, 0
    jne GSEIS
    mov al, sx5
    cmp al, 0
    jne GCINCO
    mov al, sx4
    cmp al, 0
    jne GCUATRO
    mov al, sx3
    cmp al, 0
    jne GTRES
    mov al, sx2
    cmp al, 0
    jne GDOS
    mov al, sx1
    cmp al, 0
    jne GUNO

    GSEIS:
        graficando 6, 0
        jmp SALIDA

    GCINCO:
        graficando 5, 0
        jmp SALIDA
```

MACRO graficando

En esta macro se encarga solamente de ir a la macro correspondiente de cada grado polinomial correcto para graficarlo.

```
grafiando MACRO var, nx0 ; aquí graficamos coeficiente por coeficiente
    LOCAL QUINTA, CUARTA, TERCERA, SEGUNDA, PRIMERA, SIN, SALIDA
    ; RESETEAMOS CONTADORES
    mov xmax, 0
    mov ymax, 0
    ; cambiamos color
    mov color, 5
    ; vemos de que grado es
    mov al, var
    cmp al, 6
    je CUARTA
    mov al, var
    cmp al, 5
    je QUINTA
    cmp al, 4
    je CUARTA
    cmp al, 3
    je TERCERA
    cmp al, 2
    je SEGUNDA
    cmp al, 1
    je PRIMERA
    cmp al, 0
    je SIN
    jmp SALIDA
```


MACRO quintaM

En esta macro se encarga de un polinomio de grado 5. Dependiendo de la constante estará en el origen, arriba o debajo del origen.

```
quintaM MACRO const
    LOCAL CICLO1, CICLO2, PASO1, CICLO3, CICLO4, PASO2, CICLO5, CICLO6, SALIDA
    ; pasamos const a ax
    mov al, const
    cbw
    mov constAux, ax
    ; pintar un pixel (prueba)
    mov cx, 260 ; x
    mov dx, 480 ; y
    mov stopG, 252 ; stop para saber cuando parar
    mov ax, stopG
    sub ax, constAux ; aquí hacemos el ajuste de la constante para saber si arriba o abajo del origen
    mov stopG, ax
    mov aux, 470

    CICLO1:
        mov ah, 0ch ; ppinta un pixel
        mov al, color ; color
        mov bh, 0h ; pagina 0
        int 10h ; aquí pintamos
        dec dx ; incrementamo en y el siguiente pixel
        cmp dx, aux ; comparamos para subir en x
        je CICLO2
        cmp dx, stopG ; comparamos para salir del cilo
        jne CICLO1
        jmp PASO1
```

MACRO cuartaM

En esta macro se encarga de un polinomio de grado 4. Su forma es parecida de una parábola un poco más ancha del lado izquierdo. Dependiendo de la constante estará en el origen, arriba o debajo del origen.

```
cuartaM MACRO const
    LOCAL CICLO1, CICLO2, PASO1, CICLO3, CICLO4, PASO2, CICLO5, CICLO6, SALIDA
    ; pasamos const a ax
    mov al, const
    cbw
    mov constAux, ax
    ; pintar un pixel (prueba)
    mov cx, 260 ; x
    mov dx, 0 ; y
    mov stopG, 257 ; stop para saber cuando parar
    mov ax, stopG
    sub ax, constAux ; aquí hacemos el ajuste de la constante para saber si arriba o abajo del origen
    mov stopG, ax
    mov aux, 10

    CICLO1:
        mov ah, 0ch ; ppinta un pixel
        mov al, color ; color
        mov bh, 0h ; pagina 0
        int 10h ; aquí pintamos
        inc dx ; incrementamo en y el siguiente pixel
        cmp dx, aux ; comparamos para subir en x
        je CICLO2
        cmp dx, stopG ; comparamos para salir del cilo
        jne CICLO1
        jmp PASO1
```

MACRO terceraM

En esta macro se encarga de un polinomio de grado 3. Dependiendo de la constante estará en el origen, arriba o debajo del origen.

```
terceraM MACRO const
    LOCAL CICLO1, CICLO2, PASO1, CICLO3, CICLO4, PASO2, CICLO5, CICLO6, SALIDA
    ; pasamos const a ax
    mov al, const
    cbw
    mov constAux, ax
    ; pintar un pixel (prueba)
    mov cx, 290 ; x
    mov dx, 480 ; y
    mov stopG, 248 ; stop para saber cuando parar
    mov ax, stopG
    sub ax, constAux ; aquí hacemos el ajuste de la constante para saber si arriba o abajo del origen
    mov stopG, ax
    mov aux, 470

    CICLO1:
        mov ah, 0ch ; ppinta un pixel
        mov al, color ; color
        mov bh, 0h ; pagina 0
        int 10h ; aquí pintamos
        dec dx ; incrementamo en y el siguiente pixel
        cmp dx, aux ; comparamos para subir en x
        je CICLO2
        cmp dx, stopG ; comparamos para salir del cilo
        jne CICLO1
        jmp PASO1
```

MACRO segundaM

En esta macro se encarga de dibujar una parábola, que es la forma que caracteriza a un polinomio de grado 2. Dependiendo de la constante estará en el origen, arriba o debajo del origen.

```
segundaM MACRO const
    LOCAL CICLO1, CICLO2, PASO1, CICLO3, CICLO4, PASO2, CICLO5, CICLO6, SALIDA
    ; pasamos const a ax
    mov al, const
    cbw
    mov constAux, ax
    ; pintar un pixel (prueba)
    mov cx, 290 ; x
    mov dx, 0 ; y
    mov stopG, 240 ; stop para saber cuando parar
    mov ax, stopG
    sub ax, constAux ; aquí hacemos el ajuste de la constante para saber si arriba o abajo del origen
    mov stopG, ax
    mov aux, 10

    CICLO1:
        mov ah, 0ch ; ppinta un pixel
        mov al, color ; color
        mov bh, 0h ; pagina 0
        int 10h ; aquí pintamos
        inc dx ; incrementamo en y el siguiente pixel
        cmp dx, aux ; comparamos para subir en x
        je CICLO2
        cmp dx, stopG ; comparamos para salir del cilo
        jne CICLO1
        jmp PASO1
```

MACRO primeraM

En esta macro se encarga de dibujar una línea recta a 45 grados, que es la forma característica de una función de grado 1. Dependiendo de la constante estará en el origen, arriba o debajo del origen.

```
primeraM MACRO const
    LOCAL CICLO1, CICLO2, VERIFICAR, SALIDA
    ; pasamos const a ax
    mov al, const
    cbw
    mov constAux, ax
    ; pintar un pixel (prueba)
    mov cx, 70 ; x
    mov dx, 480 ; y
    mov stopG, 640 ; stop para saber cuando parar
    sub cx, constAux ; aquí hacemos el ajuste de la constante para saber si arriba o abajo del origen
    mov aux, 80
    mov ax, aux
    sub ax, constAux
    mov aux, ax
```

MACRO sinCero

En esta macro se encarga de dibujar una línea recta horizontal, ya que solo se cuenta con una constante.

```
sinCeroM MACRO const
    LOCAL CICLO1
    ; pasamos const a ax
    mov al, const
    cbw
    mov constAux, ax
    ; pintar un pixel (prueba)
    mov cx, 0 ; x
    mov dx, 240 ; y
    sub dx, constAux ; ajuste para saber si está arriba o abajo del origen
    mov stopG, 640 ; stop para saber cuando parar

    CICLO1:
        mov ah, 0ch ; ppinta un pixel
        mov al, color ; color
        mov bh, 0h ; pagina 0
        int 10h ; aquí pintamos
        inc cx ; incrementamos en x el siguiente pixel
        cmp cx, stopG ; comparamos para salir del ciclo
        jne CICLO1
29 references
ENDM
```