Representación gráfica de funciones

Desarrolladores: Esteban Cádiz,

Cristóbal Gómez,

Eduardo Rosas

Asignatura: Taller de Programación I

Sección: 03

**Introducción**

A través de la historia de la matemática las representaciones gráficas han sido de suma importancia, desde figuras que hoy se consideran básicas que estudiaban los griegos en el campo de la geometría, hasta conceptos más abstractos como el cálculo de integrales, la parte visual ha demostrado ser esencial en la comprensión de esta. Uno de los matemáticos claves fue René Descartes, quien mediante la creación del plano cartesiano logró darle forma a expresiones algebraicas y juntarlas con el campo geométrico, creando así la geometría analítica y dando paso a los conceptos de funciones modernas. Es importante para todo estudiante tener algún medio donde pueda guardar y tener a la mano las funciones más importantes y que más se repiten en la matemática.

Hoy en día la gran mayoría de las personas tienen acceso a dispositivos digitales, los cuales trabajan con texto mediante ASCII. Tener representaciones de las funciones más importantes de manera accesible usando estos sería ideal no solo para los estudiantes de la disciplina, sino también para aquellos que trabajan con esta.

**Descripción del Problema:**

El problema a resolver del proyecto fue encontrar una forma de representar las gráficas de las funciones mediante código ASCII en la terminal de Windows, específicamente Powershell, con medidas exactas del tamaño con el cual se debe trabajar.

Las funciones que se deben trabajar en este proyecto son las siguientes:

-Recta ()

-Seno ()

-Coseno ()

-Cuadrática ()

-Cúbica ()

-Logarítmica ()

El proyecto fue realizado con ayuda de la librería math.h dentro del visual studio code y también de la utilización de geogebra para poder lograr ver el como las funciones son dibujadas y como nosotros teníamos que hacerlo

**Objetivos:**

-Representar la gráfica de las funciones en la terminal mediante ASCII.

**Marco teórico**

Para poder graficar los dibujos de las funciones, se utilizó la calculadora gráfica de Geogebra como referencia. Esta existe como un programa gratuito que se ofrece como herramienta auxiliar para la enseñanza de la matemática, tanto planos cartesianos, como planos en R3 y geometría en general. Al trabajar con funciones, es necesario ingresar la expresión en esta, para así desplegar el dibujo en R2 mediante el plano cartesiano. (Geogebra, 2023)

Una vez se tiene el dibujo de referencia a recrear, se trabajó mediante la librería math.h para poder representar y acercarse a estas. Esta librería trabaja con una variedad de funciones matemáticas declarándose como variables “double” (IBM Corporation, 2022). Para el proyecto se aprovecharon ciertas funciones de esta, como por ejemplo “pow(x,y)”, una función donde se ingresa un número x, el cual es base, elevado al exponente y. Además, mencionar que se aprovecharon las funciones trigonométricas y las logarítmicas para ayudar con la representación del gráfico.

Para representar estos dibujos se trabajó con el código de caracteres ASCII. Este es un sistema con el que se pueden mostrar en pantalla y programar distintos caracteres alfanuméricos (ASCII, s.f.). El ASCII original trabaja con caracteres que tienen asignados números del 1 al 127, mientras que el ASCII extendido permite llegar hasta el 255. Dentro del código se usó este último para representar caracteres tales como “┼” o “─” entre otros.

**Metodología:**

Se usó Replit para ver el avance remoto de los códigos que se les asignó a cada integrante, Discord fue usado para mantener una comunicación activa y con retroalimentación continua, por último el mismo trabajo en clases con alguna ayuda y apoyo de los profesores con los problemas que se nos presentaron en el desarrollo de este proyecto

**Extractos del código (cada función):**

**-Recta:**

for (y = -12; y < Y; y++) // coordenada filas

{

for (x = cotainf; x <= cotasup; x++) // coordenada columna

{

if (x == y) // Funcion f(x)=x

{

matriz[12 - y][40 + 2 \* x] = 42; //(0,0)=[12][40]

}

}

}

}

**-Cuadrática**:

for (y = -12; y < 12; y++) // for para los (x,y)

{

for (x = cotainf; x <= cotasup; x++)

{

if (y == pow(x, 2))

{

matriz[12 - y][40 + 2 \* x] = 42;

matriz[11 - y][40 + 2 \* x] = 42;

matriz[10 - y][40 + 3 \* x] = 42;

matriz[9 - y][40 + 3 \* x] = 42;

}

}

}

**-Cúbica:**

for (y = 1; y < Y; y++) // for para los (x,y)

{

for (x = cotainf; x <= cotasup; x++)

{

// if para hacer las lineas

/\*Se usaron puntos especificos para poder completar el dibujo de mejor manera.\*/

// if ((x == 1 && (1 <= y && y < 4)) || (x == 2 && (4 <= y && y < 9)) || (x == 3 && (9 <= y && y < 12)) || (x == 4 && y == 12))

if (pow(x, 2) <= y && y < pow(x, 2) + 2) // esta formula se obtuvo tras generalizar los puntos antes mencionados.

{

matriz[12 - y][40 + 2 \* x] = 42;

matriz[12 + y][40 - 2 \* x] = 42;

}

}

}}

**-Logarítmica:**

for (y = -12; y <= 12; y++) // for para los (x,y)

{

for (x = cotainf; x <= cotasup; x++)

{

// funcion log2

if (y == log2(x))

{

matriz[12 - y][40 + 2 \* x] = '\*';

}

matriz[8][24] = ' ';

}

}

**-Seno:**

for (y = -12; y < 12; y++) // for para los (x,y)

{

for (x = cotainferior; x <= cotasuperior; x++)

{

/\*se pasa de entero a flotante ya que la formula esta echa para

numeros racionales, y despues se vuelve a int para ponerlo

como posicion de matriz.\*/

xf = ((x \* M\_PI) / 36);

yf = sin(xf) \* 10; // se multiplica por 10 para obtener numeros que esten en el grafico.

yi = yf;

xi = 11 \* xf; // valor 11 conseguido por tanteo para ampliar la funcion horizontalmente

/\*xi y yi son versiones enteras de x e y una vez entran en la formula.\*/

matriz[12 - (yi)][40 + (xi)] = 42;

}

}

**-Coseno:**

for (y = -12; y < 12; y++) // for para los (x,y)

{

for (x = cotainferior; x <= cotasuperior; x++)

{

// xf= x en flotante

xf = ((x \* M\_PI) / 36);

yf = cos(xf) \* 10;

yi = yf;

xi = 11 \* xf;

matriz[12 - (yi)][40 + (xi)] = 42;

}

}

**-Función llenarmatrizconlineas:**

**int i, j;**

for (i = 0; i < Y; i++)

{

for (j = 0; j < X; j++)

{

n[i][j] = ' ';

}

}

for (i = 0; i < Y; i++)

{

n[i][40] = 197; // asigna │

}

for (i = 0; i < X; i = i + 2)

{

n[12][i] = 197; // asigna cruces horizontal

}

for (i = 1; i < X; i = i + 2)

{

n[12][i] = 196; // asigna ─ horizontal

}

n[12][40] = 197; // asigna ┼

**-Función printmatriz:**

int i, j;

for (i = 0; i < Y; i++)

{

for (j = 0; j < X; j++)

{

printf("%c", n[i][j]); // caracter para poder usar caracteres al dibujar funciones.

}

printf("\n");

}

}

**Dificultades:**

Las dificultades principales que se encontraron en el trabajo fueron con el tema de poder graficar de buena manera las funciones que tenían curvas en su dibujo en específico: la función seno, logarítmica, cuadrática y cúbica. Para poder solucionarlo, se trabajó con una aproximación al dibujo evitando mostrar valores exactos y restringiendo a rangos específicos por función.

**Conclusión**

En conclusión, el proyecto y su objetivo fueron logrados. Se usó como referencia las gráficas de Geogebra y los dibujos se lograron usando C como lenguaje de programación, ASCII y aprovechando la librería math.h. No fue posible una representación exacta de las funciones por las dimensiones limitantes de la terminal pero se logró una aproximación mediante la limitación de rangos que permite dar el dibujo hasta cierto punto de las funciones. El programa permite ver las gráficas visualmente de las funciones más importantes de las matemáticas.

# Referencias

ASCII. (n.d.). *Codigos ASCII y HTML*. Retrieved from ASCII.cl: https://ascii.cl/es/referencias.htm

Geogebra. (2023). *Calculadora Grafica*. Retrieved from Calculator -Suite Geogebra: https://www.geogebra.org/calculator

IBM Corporation. (2022). *<math.h>*. Retrieved from IBM Documentation: https://www.ibm.com/docs/es/i/7.5?topic=files-mathh