



INSTITUTO TECNOLÓGICO SUPERIOR PROGRESO

Carrera:

Ingeniería en sistemas computacionales

Materia:

Graficación

Actividad:

Reporte de practica 2-1

Docente:

Dr. Holzen Atocha Martínez García

Alumno:

David Ezequiel Caballero González

Fecha de entrega:

Lunes 8 de septiembre de 2025



Práctica 2-1

Introducción

Processing permite representar fenómenos matemáticos de manera visual e interactiva, y en esta práctica se desarrolló un programa cuyo objetivo es graficar las funciones seno y coseno en un plano cartesiano, representando sus oscilaciones dentro de una ventana digital, con ello, se busca comprender cómo se pueden transformar valores matemáticos en coordenadas gráficas mediante programación.

Processing facilita este proceso gracias a su sencillez y a su integración con funciones matemáticas como ***sin()*** y ***cos()***, además de utilidades como ***map()*** para escalar valores a un rango deseado y la práctica contribuye a reforzar la relación entre matemáticas y computación, mostrando cómo conceptos trigonométricos se aplican directamente en el trazado de gráficos computacionales.

- El código puede verse en el repositorio en GitHub: [Graficación Práctica 2 Práctica 2-1](#)

Objetivo

Desarrollar un programa que grafique las funciones seno y coseno en un plano cartesiano.

Marco teórico

En Processing, las funciones trigonométricas se implementan directamente con ***sin()*** y ***cos()***, que reciben un valor en radianes y devuelven un número entre **-1** y **1**. Para graficarlas, es necesario transformar dichos valores en coordenadas dentro de la ventana gráfica y la función ***map()*** es clave en este proceso, ya que permite convertir un rango de valores (como el ancho de la pantalla) a otro rango (de **0** a **2 π** , que representa un ciclo completo de las funciones trigonométricas).

El programa utiliza bucles ***for*** para recorrer la ventana horizontalmente y calcular, en cada posición, la altura correspondiente de seno o coseno. Posteriormente, se conectan los puntos mediante ***line(x1, y1, x2, y2)***, generando una representación visual continua.

Procedimiento

- Se definen las variables **w** y **h** para almacenar el ancho y alto de la ventana.

```
int w, h;
```

- En la función ***setup()***, se configura la ventana a 480x190 píxeles, se establece el color de fondo en negro y se dibujan los ejes **X** y **Y** en blanco.

```
int w, h;

void setup() {
  size(480, 190);
  w = width;
  h = height;
  background(0); // Fondo blanco

  //Línea del eje X
  stroke(255);
  line(0,h/2,w,h/2);

  //Línea del eje Y
  stroke(255);
  line(0,0,0,h);
}
```

3. Se crean dos funciones: **dibujarSeno()** y **dibujarCoseno()**, que reciben parámetros de color, grosor y tamaño de ventana.

```
void dibujarSeno(color c, float grosor, int w, int h) {
  stroke(c);
  strokeWeight(grosor);
  float prevX = 0;
  float prevY = h/2 - sin(0) * (h/2);
  for (int x = 1; x <= w; x++) {
    float angulo = map(x, 0, w, 0, TWO_PI);
    float y = h/2 - sin(angulo) * (h/2);
    line(prevX, prevY, x, y); // Conectar puntos con líneas
    prevX = x;
    prevY = y;
  }
}
```

```
void dibujarCoseno(color c, float grosor, int w, int h) {
  stroke(c);
  strokeWeight(grosor);
  float prevX = 0;
  float prevY = h/2 - cos(0) * (h/2);
  for (int x = 1; x <= w; x++) {
    float angulo = map(x, 0, w, 0, TWO_PI);
    float y = h/2 - cos(angulo) * (h/2);
    line(prevX, prevY, x, y); // Conectar puntos con líneas
    prevX = x;
    prevY = y;
  }
}
```

4. Dentro de cada función, se recorre el eje **X** de la ventana mediante un bucle **for**, para cada valor de **x**, luego se calcula el ángulo correspondiente en radianes con **map()**, después se obtiene el valor de **sin(angulo)** o **cos(angulo)** y se ajusta al centro de la ventana multiplicando por **h/2**, para al final trazar las líneas entre puntos consecutivos usando **line()**, logrando una curva continua.
5. Por último, se llamaron ambas funciones desde **setup()**, asignando color rojo para la función seno y verde para la función coseno.

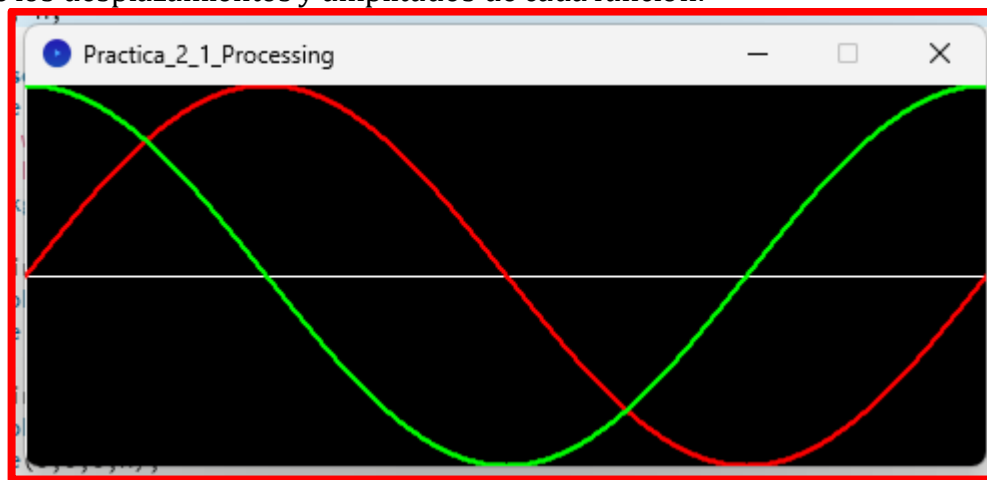
```
// Dibujar funciones
dibujarSeno(color(255, 0, 0), 2, w, h);
dibujarCoseno(color(0, 255, 0), 2, w, h);
}
```

Materiales y equipos utilizados

- Laptop
- Processing
- Página oficial de Processing
- Video de apoyo en YouTube

Resultados

El programa generó correctamente la gráfica de las funciones seno y coseno dentro de la ventana. La curva del seno apareció en color rojo, mientras que la del coseno se representó en color verde y ambas funciones oscilaron alrededor del eje **X** (mitad de la ventana), mostrando un ciclo completo en el ancho total de la pantalla. Los ejes cartesianos permitieron visualizar claramente los desplazamientos y amplitudes de cada función.



Análisis

La práctica permitió visualizar cómo los valores matemáticos se traducen en coordenadas gráficas. La función seno comenzó en el origen **(0,0)** oscilando hacia arriba y abajo, mientras que la función coseno inició en su valor máximo **(1)**, representando la fase de diferencia entre ambas. El uso de **map()** fue esencial para recorrer el rango de **0** a **2π** a lo largo del ancho de la pantalla, asegurando que se dibujara un ciclo completo.

El resultado evidencia que con Processing no solo se pueden representar datos, sino que también ayuda a comprender conceptos matemáticos como periodicidad, fase y amplitud.

Discusión

A través de esta práctica se confirmó la utilidad de Processing para representar funciones matemáticas de manera interactiva, visual y con claridad se mostraron las diferencias entre seno y coseno, lo que ayudó a comprender su relación y la importancia de la fase inicial en cada curva. Asimismo, la práctica permitió reflexionar sobre la simplicidad de los comandos de Processing para realizar representaciones que, en otros entornos, requieren mayor complejidad.



Conclusión

El programa logró representar gráficamente las funciones seno y coseno, mostrando la aplicación de las matemáticas en el programa. Processing resultó ser una herramienta eficiente y sencilla para experimentar con funciones trigonométricas y visualizar sus características principales, como amplitud, frecuencia y fase y esta práctica refuerza la conexión entre teoría y aplicación práctica facilitando la comprensión de conceptos matemáticos mediante representaciones gráficas.

Referencias

- Processing Foundation. (n.d.). cos(). Processing.
https://processing.org/reference/cos_.html
- Processing Foundation. (n.d.). sin(). Processing.
https://processing.org/reference/sin_.html
- Air Room. (2019, April 7). PROCESSING: 7.7 Trigonometría [En español] [Video]. YouTube.
<https://youtu.be/bBGj7R1SbL8>

