



### Taller 1

Fecha: Septiembre de 2021

**Indicador de logro a medir:** Aplicar los conceptos de la lógica de programación, la orientación a eventos, y los lenguajes de marcado en el desarrollo de una aplicación que funcione en navegadores de Internet.

#### NOTAS:

- Este taller se debe hacer como preparación para evaluaciones. En ningún caso representará una calificación.
- Los primeros ejercicios se entregan resueltos como ejemplo para el desarrollo de los demás

Elaborar la respectiva aplicación en *HTML* y *Javascript* para los siguientes enunciados:

1. Graficar una función matemática de una sola variable en el plano cartesiano, permitiendo especificar el rango en el eje X.

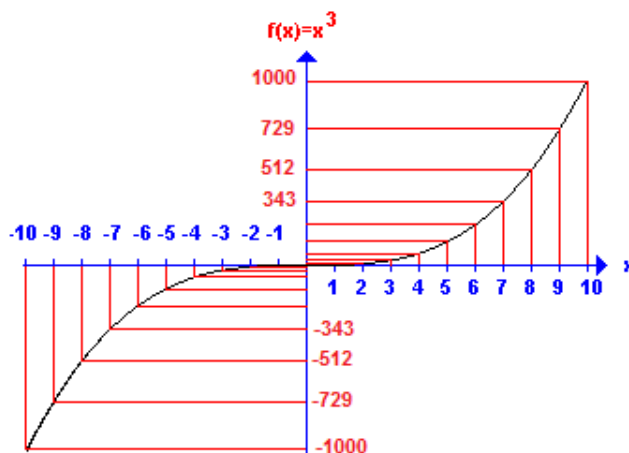
#### Gráfica de una función

Una función matemática permite calcular un valor como resultado de aplicar una expresión a una variable. Por ejemplo, dada la función  $f(x)=x^3$ , para cada valor de  $x$ , donde  $x$  es un número real,  $f(x)$  corresponde al valor de  $x$  elevado al cubo. Con base en esto, se puede obtener la siguiente tabla:

X	-10	-9	-8	-7	-6	-5	-4	-3	-2	-1	0	1	2	3	4	5	6	7	8	9	10
f(x)	-1000	-729	-512	-343	-216	-125	-64	-27	-8	-1	0	1	8	27	64	125	216	343	512	729	1000

Esta tabla permite definir una gráfica en un plano cartesiano de dos ejes, uno para los valores de la variable  $x$  (horizontal) y otro para los valores de la función (vertical). La gráfica se realiza trazando líneas entre dos puntos cada uno correspondiente a la pareja compuesta por un valor de la variable y el respectivo valor de la función  $(x, f(x))$ . Por ejemplo:

Punto 1= $(-10,-1000)$  al Punto 2= $(-9,-729)$   
Punto 2= $(-9,-729)$  al Punto 2= $(-8,-512)$



R/

#### Datos de Entrada:

- Límite Inferior (LI)
- Límite Superior (LS)
- Lista de Funciones

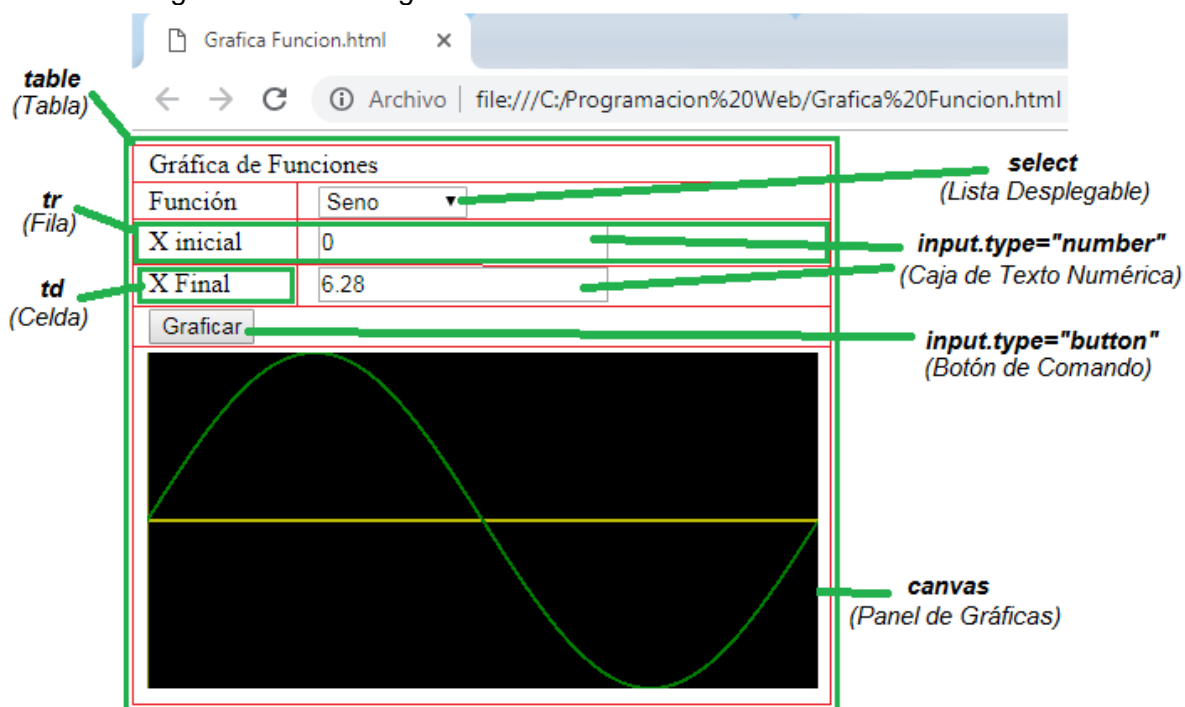
#### Datos de Salida:



<ul style="list-style-type: none"> <li>○ Valor Función para un Valor de <math>X</math> (<math>FX</math>)</li> <li>○ Línea entre un par de coordenadas <math>X</math>, <math>FX</math> y el respectivo par anterior</li> </ul>
<ul style="list-style-type: none"> <li>● <b>Proceso:</b></li> </ul>
<p>Para cada valor de <math>X</math> entre <math>LI</math> y <math>LS</math>:</p> <ul style="list-style-type: none"> <li>○ Calcular el valor de la Función <math>FX</math></li> <li>○ Trazar una línea entre el par de coordenadas <math>X</math>, <math>FX</math> y el respectivo par anterior</li> </ul>

### Programa

Para la implementación del aplicativo en *HTML* se debe comenzar con el diseño de una interfaz gráfica como la siguiente:



Este es una página *web*. La siguiente tabla relaciona los objetos a añadir con las propiedades cuyos valores deben ser cambiados:

Clase	Nombre	Otras Propiedades
<b>html</b> (Página web)		title = "Gráfica de Funciones"
<b>table</b> (Tabla)		Columnas = 2 Filas = 5
<b>select</b> (Lista desplegable)	<i>cmbFuncion</i>	Opciones = "Seno", "Coseno", "Cuadratica", "Cúbica"
<b>Input.type="number"</b> (Caja de texto)	<i>txtLI</i>	
	<i>txtLS</i>	



<i>numérica)</i>		
<b>Input.type="button"</b> (Botón de Comando)	<i>btnGraficar</i>	value = "Graficar" onClick = "Graficar()"
<b>canvas</b> (Panel de Gráficas)	<i>cnvGrafica</i>	width = 400 height = 300

El siguiente es el código en *HTML* para esta interfaz gráfica:

```
<html>
<head>
  <title>Gráfica de Funciones</title>
</head>
<body>
  <table>
    <tr>
      <td colspan="2">Gráfica de Funciones</td>
    </tr>
    <tr>
      <td>Función</td>
      <td>
        <select id="cmbFuncion">
          <option>Seno</option>
          <option>Coseno</option>
          <option>Cuadrática</option>
          <option>Cúbica</option>
        </select>
      </td>
    </tr>
    <tr>
      <td>X inicial</td>
      <td><input type="number" id="txtLI"></td>
    </tr>
    <tr>
      <td>X Final</td>
      <td><input type="number" id="txtLS" /></td>
    </tr>
    <tr>
      <td colspan="2">
        <input
          type="button"
          value="Graficar"
          onclick="Graficar()" />
      </td>
    </tr>
    <tr>
      <td colspan="2">
        <canvas
          id="cnvGrafica"
          width="400"
          height="400"></canvas>
      </td>
    </tr>
  </table>
</body>
</html>
```



Para comprender este código es necesario dominar lo siguiente:

- El código *HTML* se caracteriza por estar compuesto de unos elementos llamados **tag**. Existen varios tipos y permiten definir desde objetos hasta zonas de la página web. El siguiente es el significado de cada tag utilizado:

Tag	Descripción
<b>&lt;html&gt;...&lt;/html&gt;</b>	Demarca el documento web en su totalidad
<b>&lt;head&gt;...&lt;/head&gt;</b>	Define una zona de encabezado que no es visible (con excepción del título) pero en la cual se incluyen tags de configuración
<b>&lt;title&gt;...&lt;/title&gt;</b>	Define el texto para el título de la página
<b>&lt;body&gt;...&lt;/body&gt;</b>	Define el contenido que será mostrado por el navegador
<b>&lt;table&gt;...&lt;/table&gt;</b>	Define una tabla que contendrá objetos o texto
<b>&lt;tr&gt;...&lt;/tr&gt;</b>	Define una fila de la tabla
<b>&lt;td&gt;...&lt;/td&gt;</b>	Define una celda en una fila de la tabla
<b>&lt;select&gt;...&lt;/select&gt;</b>	Define una lista desplegable
<b>&lt;option&gt;...&lt;/option&gt;</b>	Define una opción de una lista desplegable
<b>&lt;input&gt;</b>	Define un objeto para entrada de datos o ejecutar acciones
<b>&lt;canvas&gt;...&lt;/canvas&gt;</b>	Define un área para despliegue de gráficas

- Muchos *tags* incluyen atributos que permiten parametrizarlos:

Tag	Atributo	Descripción
<b>td</b>	<b>colspan</b>	Define si una celda se extiende sobre otras celdas
Cualquiera	<b>id</b>	Define el nombre con el que se identificará un objeto (debe ser único)
<b>input</b>	<b>type</b>	Define la clase de objeto de entrada. Estos son algunos de sus posibles valores: <ul style="list-style-type: none"><li><b>text</b></li><li><b>number</b></li><li><b>button</b></li><li><b>time</b></li><li><b>date</b></li><li><b>checkbox</b></li><li><b>email</b></li></ul>
Objetos de entrada	<b>value</b>	Valor que contiene el elemento de entrada
Elemento desplegado	<b>onClick</b>	Define la acción que se ejecutará cuando se haga click en el objeto

- Los *tags* que incluyen apertura y cierre (donde el cierre se distingue de la apertura por comenzar con el carácter "/") es porque permiten contener a otros *tags* en su interior.



Teniendo definida la interfaz de usuario, se debe programar el código necesario para que el aplicativo funcione. Este código se escribe dentro del tag **<script>...</script>** en el lenguaje de programación **JavaScript**. Los siguientes son los módulos necesarios:

Módulo o función	Descripción
Inicio del aplicativo	Es el que se ejecuta una vez se despliega la interfaz. Este código simplemente se escribe dentro del tag de código
Función <b>Limpiar()</b>	Deja en negro toda el área del <i>canvas</i> para evitar que se superpongan gráficas cada vez que se ejecute
Función <b>ObtenerFuncion()</b>	Permite calcular el valor de la función para un valor de la variable <i>x</i> y de acuerdo al tipo de función escogida mediante la lista desplegable
Función <b>Graficar()</b>	Realiza todos los trazos necesarios para mostrar la gráfica de la función escogida en el área de despliegue

El siguiente es el código **JavaScript** para los anteriores módulos:

```
//Obtener el objeto para despliegue
var canvas = document.getElementById("cnvGrafica");
var ctx = canvas.getContext("2d");
//Obtener las dimensiones del área de despliegue
ancho = canvas.width;
alto = canvas.height;
//Limpiar el área de despliegue
Limpiar();

//Función que permite limpiar el área de despliegue
function Limpiar() {
    ctx.beginPath();
    ctx.rect(0, 0, ancho, alto);
    ctx.fillStyle = "black";
    ctx.fill();
    ctx.closePath();
}

//Función que calcula la función matemática a graficar
function ObtenerFuncion(x) {
    switch (cmbFuncion.selectedIndex) {
        case 0:
            return Math.sin(x);
        case 1:
            return Math.cos(x);
        case 2:
            return Math.pow(x, 2);
        case 3:
            return Math.pow(x, 3);
    }
}
```



```
}

//Función que construye la gráfica
function Graficar() {
    //Limpiar el área de despliegue
    Limpiar();

    //Obtener los datos de entrada
    xi = eval(txtLI.value);
    xf = eval(txtLS.value);
    if (xi != null && xf != null) {
        //Incremento eje X
        dX = (xf - xi) / ancho;

        //Valores minimo y maximo de la Funcion
        x = xi
        yMax = ObtenerFuncion(x);
        yMin = ObtenerFuncion(x);
        for (xP = 0; xP <= ancho; xP++) {
            x = x + dX;
            fX = ObtenerFuncion(x);
            if (fX > yMax)
                yMax = fX;
            if (fX < yMin)
                yMin = fX;
        }
        //Incremento Eje Y
        dY = (yMax - yMin) / alto

        ctx.beginPath();
        ctx.strokeStyle = "yellow";
        //Mostrar eje X
        if (yMin <= 0 && 0 <= yMax) {
            yP = yMax / dY;
            ctx.moveTo(0, yP);
            ctx.lineTo(ancho, yP);
            ctx.stroke();
        }

        //Mostar eje Y
        if(xi<=0 && 0<=xf){
            xP = Math.abs(xi) / dX;

            ctx.moveTo(xP, 0);
            ctx.lineTo(xP, alto);
            ctx.stroke();
        }

        ctx.closePath();
        ctx.beginPath();
        ctx.strokeStyle = "green";
        x = xi;
```

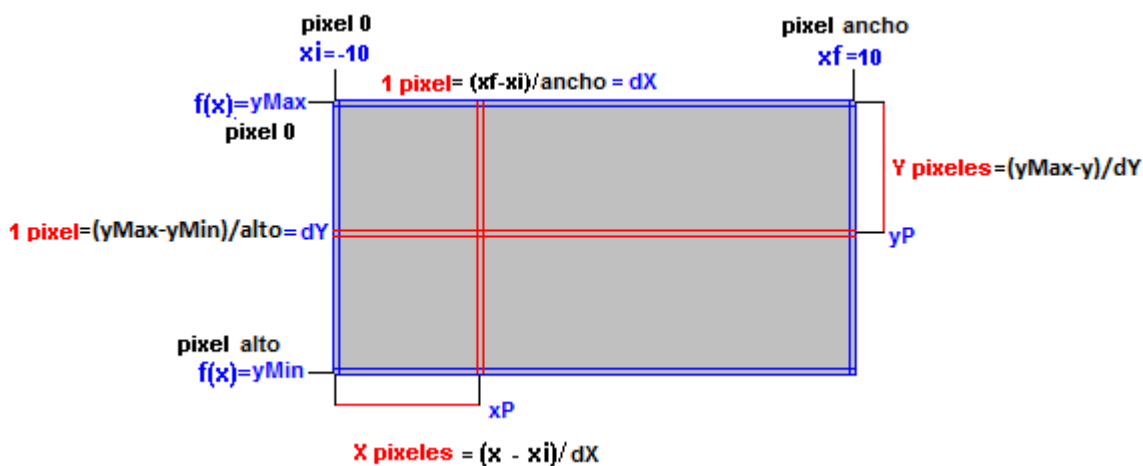


```
//recorrer los puntos del eje X
for (xP = 0; xP <= ancho; xP++) {
    y = ObtenerFuncion(x);

    yP = (yMax - y) / dY;
    if (xP > 0) {
        ctx.moveTo(xP - 1, antyP);
        ctx.lineTo(xP, yP);
        ctx.stroke();
    }
    antyP = yP;
    //incrementar x
    x = x + dX
}
ctx.closePath();
}
else
{
    window.alert("Faltan datos");
}
}
```

Para entender este código es necesario comprender lo siguiente:

- **Coordenadas de computadora vs Coordenadas cartesianas:** Para poder realizar el trazado de líneas por computadora, las respectivas coordenadas del objeto de despliegue (el objeto tipo *canvas*), no corresponden a las coordenadas del plano cartesiano, por lo que hay que hacer los respectivos cálculos para hacer la adaptación. En el caso del objeto *canvas* las coordenadas siempre inician en 0 en la esquina superior izquierda y aumentan horizontalmente hacia la derecha y verticalmente hacia abajo



Estos cálculos comprenden el manejo de las siguientes variables:



- **ancho:** Total de pixeles que comprende horizontalmente el área de despliegue. Se obtiene mediante la propiedad **width**
- **alto:** Total de pixeles que comprende verticalmente el área de despliegue. Se obtiene mediante la propiedad **height**
- **xi:** Valor inicial de x. Corresponde al pixel 0
- **xf:** Valor final de x. Corresponde al pixel del tamaño del objeto de despliegue.
- **yMax:** Valor máximo de la función. Corresponde al pixel 0
- **yMin:** Valor mínimo de la función. Corresponde al pixel del tamaño del alto del objeto de despliegue
- **dX:** Valor de incremento de x para un pixel. Equivalente a  $(xf - xi)/ancho$
- **dY:** Valor de incremento de la función para un pixel. Equivalente a  $(yMax - yMin)/alto$

Para un valor de x determinado, el pixel correspondiente sería:  $(x - xi)/dX$

Para un valor de y determinado, el pixel correspondiente sería:  $(yi - y)/dY$

- **Objeto para gráficas:** Para poder realizar trazos sobre el área de despliegue (*canvas*) se tiene el objeto **Context** que ofrece la funcionalidad para trazar gráficas tales como líneas, rectángulos, óvalos, arcos, textos, entre otros.

Este objeto se obtiene mediante la instrucción **getContext()**.

La siguiente lista describe las instrucciones de este objeto utilizadas en el programa:

Instrucción	Descripción
<b>beginPath()</b>	Función que define el inicio de un trazo
<b>closePath()</b>	Función que define el fin de un trazo
<b>fillStyle</b>	Atributo que define el color de relleno
<b>rect(x, y, ancho, alto)</b>	Función que agrega un rectángulo a un trazo
<b>fill()</b>	Función que llena los trazos definidos
<b>strokeStyle</b>	Atributo que define el color de trazo
<b>moveTo(x, y)</b>	Indica el punto de inicio de un nuevo trazo
<b>lineTo(x, y)</b>	Indica el punto de llegada de una línea
<b>stroke()</b>	Indica que se realice el trazo

- **Trazado de líneas:** El objeto *Context* ofrece varias instrucciones para el trazado de gráficas, entre ellas para el trazado de líneas. Para ello se deben utilizar varias instrucciones que definen el trazo final:
  - Iniciar el trazado con *beginPath()*
  - Indicar el color del trazo con *strokeStyle*
  - Indicar la coordenada inicial con *.moveTo()*
  - Indicar la coordenada siguiente con *lineTo()*
  - Realizar la acción de trazo con *stroke()*





- Finalizar el trazo con `closePath()`
- Conocidos los límites en el eje horizontal (los datos de entrada **txtLI** y **txtLS**) se deben obtener los límites en el eje vertical como los valores máximo y mínimo de la función. Para ello se debe calcular la función a través de todos los valores de  $x$  posibles y así poder verificar cual es el valor máximo de la función así como también el valor mínimo
- La gráfica de la función se hará trazando pequeñas líneas con una distancia horizontal de sólo 1 pixel lo cual dará el efecto de curva de la función. Para ello se calculan dos valores de la variable  $x$  contiguos (con una diferencia de  $dX$ ) y sus respectivos valores  $f(x)$  y se traza una línea. Seguido, se traza la línea siguiente y así sucesivamente de izquierda a derecha hasta llegar al último píxel que corresponde al valor  $x_f$ . En este caso, se obtendrán dichas coordenadas cartesianas, luego serán convertidas a las coordenadas en pixeles del objeto de despliegue y luego se traza la línea respectiva

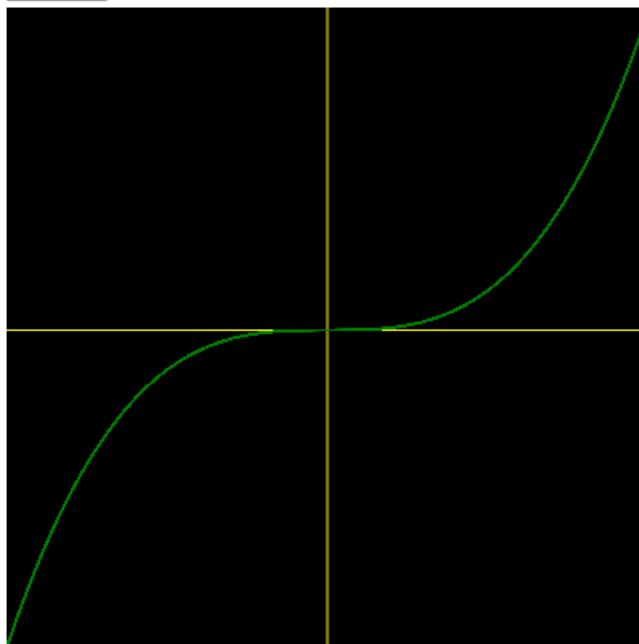
La ejecución luciría así al seleccionar la función cúbica:

#### Gráfica de Funciones

Función

X inicial

X Final





2. Desarrollar un aplicativo que permita representar **Fracciones Impropias** como un vector de círculos segmentados y como una **Fracción Mixta**

Una fracción impropia es una fracción en la cual el numerador (número de arriba) es mayor que o igual al denominador (número de abajo).

Una fracción mixta es la combinación de un número entero y una fracción.

Para convertir una fracción impropia en mixta, se siguen estos pasos:

- Dividir el numerador entre el denominador.
- Escribir el cociente como un número entero.
- Después escribir el resto encima del denominador.

La interfaz podría tener el siguiente aspecto al momento de ejecutarse:

Fracciones Impropias a Fracciones Mixtas:

Denominador:  Fraccion Mixta:

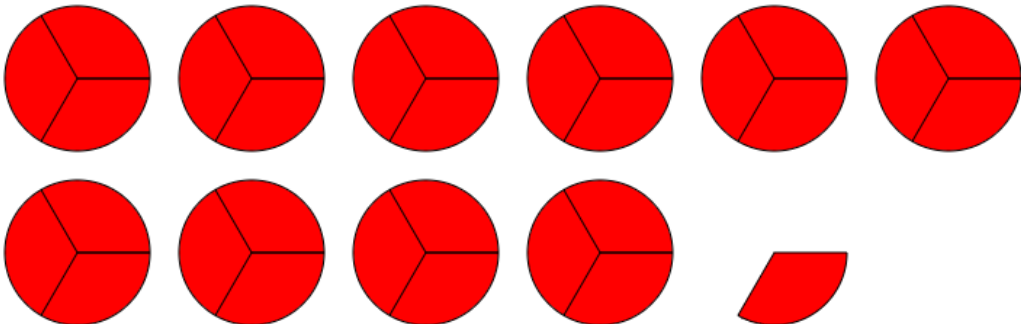
Numerador:

Al hacer clic en el botón “Mostrar” deben aparecer el vector de círculos segmentados y la fracción mixta:

Fracciones Impropias a Fracciones Mixtas:

Denominador:  Fraccion Mixta: 10  $\frac{1}{3}$

Numerador:



Tenga en cuenta que los círculos se segmentan de acuerdo al valor del denominador.



Fracciones Impropias a Fracciones Mixtas:

Denominador:

Numerador:

Fraccion Mixta:  $8 \frac{3}{6}$

