

# GRAPHIC v.1.0

Julio Alejandro Santos Corona

14 de febrero de 2021

La clase crea gráficas de puntos de forma simple, las gráficas pueden ser exploradas o segmentadas.

## 1. Instancia

La instancia se crea a través del constructor de la función **GEOMETRYR.GRAPHIC.main** al que se le pasa un objeto como argumento.

```
1 var graphic = new GEOMETRYR.GRAPHIC.main (config);
```

### 1.1. Configuración

La configuración de la gráfica se realiza a través de un objeto pasado durante la creación de instancia con la siguiente estructura:

Propiedad	Tipo	Descripción
width	number	Ancho de la gráfica.
height	number	Alto de la gráfica.
nameX	string	Nombre del eje x.
nameY	string	Nombre del eje y.
marginL	number	Margen izquierdo (configura el espacio dedicado al nombre del eje y).
marginR	number	Margen derecho.
maxUnitsX	number	Número máximo de segmentos calculados para el eje x.
maxUnitsY	number	Número máximo de segmentos calculados para el eje y.
directionX	string	Dirección del eje x ( + o - ).
directionY	string	Dirección del eje y ( + o - ).
notationX	function	Callback que toma el valor de escala y lo transforma según la función.
notationY	function	Callback que toma el valor de escala y lo transforma según la función.

```
1 var graphic = new GEOMETRYR.GRAPHIC.main ({
2   width: 1350,
3   height: 580,
4   nameX: "ops",
5   nameY: "min",
6   marginL: -30,
7   marginR: -12,
8   maxUnitsX: 13,
9   maxUnitsY: 10,
10  directionX: "+",
11  directionY: "-",
12  notationX: function (val) { return val; },
13  notationY: function (val) { return val; }
14 });
```

## 1.2. Métodos

Al crear la instancia se contarán con los siguientes métodos:

- **build** - Construcción de la gráfica, soporta un booleano como argumento para centrarse en un segmento o sólo construir dicho segmento.
- **clear** - Elimina las gráficas mostradas.
- **onBuild** - Recibe una función como argumento que será ejecutado en cada construcción de puntos.
- **create** - Clase que permite crear espacios geométricos.

## 2. Espacios geométricos

El espacio geométrico es un conjunto independiente de puntos que serán graficados.

Dicho espacio es instanciado a partir de la clase **create** que requiere un objeto con la siguiente configuración.

Propiedad	Tipo	Descripción			
name	string	Nombre del espacio de puntos.			
color	string	Color de línea en lenguaje css.			
fill	object	width	number	Anchura del patrón de relleno.	
		height	number	Altura del patrón de relleno.	
		color	string	Color de relleno en formato css.	
		type	string	l = líneas, c = círculos	
		center	object	Activo sólo si type es c. { x: 0, y: 0 }	
		line	string	Activo sólo si type es l.	

```
1 let g1 = new graphic.create ({
2   name: "G1",
3   color: "rgb(67, 124, 156)",
4   fill: {
5     width: 5,
6     height: 5,
7     color: "rgb(67, 124, 156)",
8     type: "l",
9     line: "M 5 0, 0 5"
10  }
11 });
```

El objeto contiene los siguientes métodos:

- **getID** - Arroja el id del espacio geométrico.
- **length** - Cantidad de puntos del espacio.
- **active** - Recibe un booleano para indicar si será graficado o no.
- **getStatus** - Indica si el espacio está activo o no.
- **setPoint** - Crea un nuevo punto en el espacio geométrico.
- **getPoints** - Obtiene los puntos pertenecientes al espacio geométrico.
- **setRange** - Indica un rango de puntos a mostrar.
- **reset** - Elimina todos los puntos

### 3. Uso

Para crear una gráfica se debe tener un espacio geométrico distinto del vacío.

La forma de crear puntos se realiza a través del método **setPoint** con la siguiente estructura:

Parámetro	Tipo	Opcional	Descripción
x	number	no	Valor del eje x
y	number	no	Valor del eje y
i	object	si	Información adicional del punto (Se visualizarán en el cuadro de información).

En caso de requerir un rango específico se puede utilizar un rango a través del método **setRange**.

Parámetro	Tipo	Opcional	Descripción
start	number	no	Número de punto inicial (empezando desde el 0)
end	number	si	Número de punto final, en caso de no declararlo se tomará el punto final

```
1 let gl = new graphic.create (config);
2
3 gl.setPoint (1, 6);
4 gl.setPoint (6.1, 1);
5 gl.setPoint (7.22, 9);
6 gl.setPoint (8.9, 4.2);
7 gl.setPoint (10.1, 3);
8 gl.setPoint (15, 4.28);
9
10 gl.setRange (3, 4); //Se seleccionan los puntos (8.9, 4.2) y (10.1, 3)
11
12 gl.active (true);
```

Para la construcción de la gráfica se utiliza el método **build** de la clase **GRAPHIC**

Parámetro	Tipo	Opcional	Descripción
range	bool	si	En caso de ser true se tomará el rango configurado permitiendo moverse por la gráfica, caso contrario sólo se mostrará dicho rango.

```
1 let graphic = new GEOMETRY.GRAPHIC.main (config);
2 let gl = new graphic.create (configGraphic);
3
4 gl.setPoint (1, 6);
5 gl.setPoint (6.1, 1);
6 gl.setPoint (7.22, 9);
7 gl.setPoint (8.9, 4.2);
8 gl.setPoint (10.1, 3);
9 gl.setPoint (15, 4.28);
10
11 gl.setRange (3, 4); //Se seleccionan los puntos (8.9, 4.2) y (10.1, 3)
12
13 gl.active (true);
14
15 graphic.build (true);
```

### 4. Extra

#### 4.1. Funciones de notación

Durante la instanciación de la clase **GRAPHIC** se incluye la propiedad **notationX** y **notationY**; dicha propiedad recibe como argumento el valor del punto y es útil cuando se quiere cambiar la representación de la escala.

El ejemplo siguiente muestra cómo cambiar la escala a meses:

```
1  function monthScale (value) {  
2      let months = [  
3          "ene",  
4          "feb",  
5          "mar",  
6          "abr",  
7          "may",  
8          "jun",  
9          "jul",  
10         "ago",  
11         "sep",  
12         "oct",  
13         "nov",  
14         "dic"  
15     ];  
16  
17     return months[Math.floor (value % 12)];  
18 }
```

## 4.2. Validaciones extra

La clase `GRAPHIC` también cuenta con el método **onBuild** que solicita como argumento una función que recibe de el objeto `circle` y los valores del punto, dicha función es ejecutada cada vez que se imprime un punto sobre la gráfica.

A continuación se muestra un método que verifica cierto rango de valores en `y`, en caso de ser menores a 11 el punto cambiará su color a rojo.

```
1  let graphic = new GEOMETRY.GRAPHIC.main (config);  
2  
3  graphic.onBuild (function (circle , point) {  
4      if (point.info.y < 11) {  
5          circle.setAttribute ("style", "stroke-width: 2px;");  
6          circle.setAttribute ("stroke", "rgb(254, 0, 0)");  
7      }  
8  });
```