

Posicionar puntos en coordenadas de la ventana en Geogebra

A. Martín Dinnbier

www.geogebra.org/u/adrianmartin

1 Introducción

Cuando queremos situar un punto en Geogebra lo hacemos empleando el sistema de posicionamiento de Geogebra. Este sistema se basa en el eje de coordenadas del programa. Es decir, los puntos se posicionan respecto a los ejes x e y. Pero hay ocasiones en que queremos colocar los puntos en función de la ventana visible. Por ejemplo, queremos posicionar un punto en una región que estemos seguros de que se encuentra en la zona visible.

En resumen, para crear un sistema de posicionamiento en píxeles en Geogebra tenemos que relacionar dos sistemas de posicionamiento. Uno de ellos es el sistema propio del programa, el que establece el eje de coordenadas que empleamos en nuestras construcciones matemáticas. El otro sistema es el sistema de la pantalla (es decir en píxeles de pantalla)

La idea original para esta construcción está tomada de la entrada de Michel Horvath: ¹. Si quieres descargar un recurso donde se haya implementado puedes hacerlo aquí: <https://ggbm.at/ZsMjTW3Z>

2 Sistema de coordenadas de Geogebra.

Para saber el ancho y alto de la pantalla en un momento dado necesitamos las esquinas de la venta activa (en coordenadas de Geogebra). Para ello vamos a usar la función *Corner* y los números de 1 a 4 que nos dan las coordenadas.

Para simplificar el proceso vamos a crear variables ejecutando comandos en Geogebra por línea de comandos. Estas variables las podemos luego acceder desde Javascript para emplearlas en nuestro programa. Por supuesto, estos comandos se podrían también ejecutar desde Javascript pero hace el proceso menos claro a la hora de entenderlo por primera vez.

Ejecuta los comandos siguientes en Geogebra usando la barra de entrada tal y como se muestra en *Figure1*.

```
1 corner1=Corner (1)
2 corner2=Corner (2)
3 corner3=Corner (3)
4 corner4=Corner (4)
```

Listing 1: Esquinas de la pantalla

¹https://archive.geogebra.org/en/upload/files/english/Michael_Horvath/geogebra_placing_objects_exactly_on_the_screen.htm



```
Input: corner1=corner(1)
```

Figure 1: Barra de entrada

Acabamos de definir las variables `corner1`, `corner2`, `corner3` y `corner4` que nos dan las cuatro esquinas de ventana de Geogebra (en coordenadas de Geogebra). Observa como se disponen (Figura 2) porque es relevante para los pasos posteriores.



Figure 2: Esquinas de la pantalla

Ahora que tenemos las esquinas podemos calcular el ancho total de la ventana pero no en píxeles sino en coordenadas de Geogebra. Esto lo hacemos usando el comando `distance` que nos da la distancia entre dos puntos. Vuelve a usar la barra de entrada de Geogebra para introducir los comandos siguientes:

```
1 distX=Distance(corner1, corner2)
2 distY=Distance(corner1, corner4)
```

Y por último vamos a almacenar en estas variables los valores de las coordenadas `x` e `y` de las esquinas. Las usaremos para posicionar puntos correctamente.

```
1 valuxcorner1= x(corner1)
2 valueycorner4=y(corner4)
```

3 Sistema de coordenadas de la ventana

El comando `corner(5)` crea un punto con las coordenadas de la venta. Vamos a ejecutar un comando en Geogebra para almacenar los valores de la venta.

```
1 window=Corner(5)
```

Ya podemos calcular el alto y ancho de la ventana con estos comandos (en píxeles)

```
1 winHeight=y(window)
2 winWidth=x(window)
```

4 Factores de conversión

Dado que queremos hacer una conversión entre ambos sistemas de posicionamiento, lo que pretendemos es transformar las coordenadas de un punto de la ventana (y por tanto en píxeles) en un punto en coordenadas de Geogebra (por tanto en coordenadas de los ejes de coordenadas)

de Geogebra). Supongamos que quiero colocar un punto en el centro de la pantalla. Como conozco el tamaño de la ventana en píxeles, sé que el punto se encuentra en:

$$\left(\frac{winWidth}{2}, \frac{winHeight}{2}\right) = (x_1, y_1)$$

Pero ese sistema en píxeles no me vale para posicionar el punto. Necesito convertirlo a coordenadas de Geogebra. Fijémonos en la coordenada x_1 del punto y establezcamos la siguiente proporción:

$$\frac{x_1 \text{ píxeles}}{winWidth \text{ píxeles}} = \frac{x \text{ unidades ejes}}{distX \text{ unidades ejes}}$$

Despejando en la igualdad tenemos que el valor que estamos buscando es:

$$x = \frac{x_1 \cdot distX}{winWidth}$$

Luego es fácil ver que el factor de conversión para pasar de coordenadas en píxeles a coordenadas en los ejes es:

$$factorX = \frac{distX}{winWidth}$$

Análogamente:

$$factorY = \frac{distY}{winHeight}$$

Dado que ambos factores son iguales, sólo vamos calcular uno de ellos al que llamaremos $factorXY$: Ejecuta ahora el comando:

```
1 factorXY=distX / winWidth
```

5 Usando Javascript

Ahora puedes usar Javascript en un elemento del escenario (por ejemplo un botón) para crear nuevos elementos y posicionarlos en relación a la ventana.

```
1 //definimos el applet
2 var ap=ggbApplet;
3
4 //obtengo las unidades en píxeles
5 var ancho=ap.getValue("winHeight");
6 var alto=ap.getValue("winWidth");
7
8 //obtengo los factores de conversión calculados en Geogebra (pixels/unidad real)
9 var factorX=ap.getValue("factorXY");
10
11 //Las esquinas contienen las coordenadas de los extremos usadas para situar el punto
12 var valuxcorner1=ap.getValue("valuexcorner1");
```

```

13 var valuexcorner4=ap.getValue("valuexcorner4");
14
15 //dados dos valores x,y en píxeles, los convertimos a coordenadas de los ejes de
    Geogebra haciendo la siguiente conversión.
16 x=valorxdeEsquina1+x*factorXY;
17 y=valorydeEsquina4-y*factorXY;

```

Ahora que lo hemos entendido, para facilitarnos el proceso, crearemos una función en Javascript que hace el proceso más sencillo. La función recibe las coordenadas de los puntos x e y en píxeles y las convierte a coordenadas de los ejes.

```

1
2 function pointToWindow(x, y){
3   var point= new Array();
4   //definimos el applet
5   var ap=ggbApplet;
6
7   //obtengo las unidades en píxeles
8   var ancho=ap.getValue("winHeight");
9   var alto=ap.getValue("winWidth");
10
11  //obtengo el factor de conversión calculado en Geogebra (pixels/unidad real)
12  var factorXY=ap.getValue("factorXY");
13
14  //Las esquinas contienen las coordenadas de los extremos
15  var valuexcorner1=ap.getValue("valuexcorner1");
16  var valueycorner4=ap.getValue("valueycorner4");
17
18  //Dados dos valores x,y en píxeles, los convertimos a coordenadas de los ejes de
    Geogebra haciendo la siguiente conversión.
19  point[0]=valuexcorner1+x*factorXY;
20  point[1]=valueycorner4-y*factorXY;
21
22  return(point)
23 }

```

Si quieres crear una sencilla aplicación para generar puntos aleatorios que siempre se sitúen en la zona visible lo puedes hacer con siguiente código. Y el ejemplo para descargar lo puedes encontrar aquí: <https://ggbm.at/ZsMjTW3Z>

```

1 var ap=ggbApplet;
2
3 //obtengo las unidades en píxeles de la ventana
4 var winWidth=ap.getValue("winWidth");
5 var winHeight=ap.getValue("winHeight");
6
7 //cordenadas del punto que quiero situar en la pantalla (en píxeles)
8 var x=getRandomArbitrary(0, winWidth);
9 var y=getRandomArbitrary(0, winHeight);
10
11 //lo transformo a coordenadas de los ejes de Geogebra
12 var point= new Array();
13 point=pointToWindow(x,y);
14 //alert(point[0]+" "+ point[1]);
15
16 //situamos el punto en Geogebra
17 var nuevoPunto=ap.evalCommandGetLabels("(" +point[0]+"," +point[1]+")");
18
19 //ocultamos la etiqueta del punto
20 ap.evalCommand("ShowLabel("+nuevoPunto+", "+false+")");
21

```

```
22
23
24 function getRandomArbitrary(min, max) {
25     return Math.random() * (max - min) + min;
26 }
27
28
29 function pointToWindow(x, y){
30     var point= new Array();
31     //definimos el applet
32     var ap=ggbApplet;
33
34     //obtengo las unidades en píxeles
35     var ancho=ap.getValue("winHeight");
36     var alto=ap.getValue("winWidth");
37
38     //obtengo el factor de conversión calculado en Geogebra (pixels/unidad real)
39     var factorXY=ap.getValue("factorXY");
40
41     //Las esquinas contienen las coordenadas de los extremos
42     var valuexcorner1=ap.getValue("valuexcorner1");
43     var valueycorner4=ap.getValue("valueycorner4");
44
45     //Dados dos valores x,y en píxeles, los convertimos a coordenadas de los ejes de
46     //Geogebra haciendo la siguiente conversión.
47     point[0]=valuexcorner1+x*factorXY;
48     point[1]=valueycorner4-y*factorXY;
49     return(point)
50 }
```