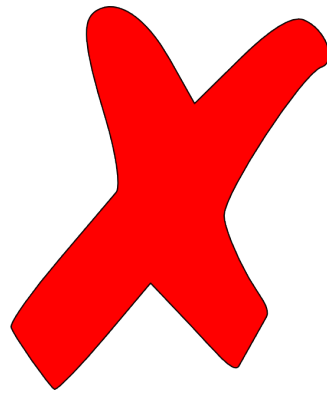




Universidad de
los Andes

Interfaces Gráficas

Redefinición de las variables



```
16  
17  
18  
19  
20  
21  
22
```

```
public void run() {  
    int counter = 0;  
    for (int i = 0; i < 10; i++) {  
        int counter = counter + 1;  
    }  
}
```

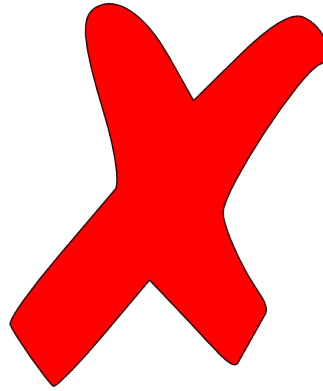
Solo necesitas especificar el tipo de una variable UNA vez.



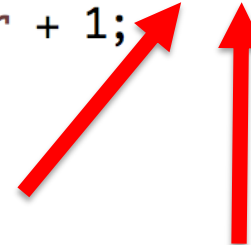
```
16  
17  
18  
19  
20  
21  
22  
23
```

```
public void run() {  
    int counter = 0;  
    for (int i = 0; i < 10; i++) {  
        counter = counter + 1;  
    }  
}
```

A dónde pertenece ; el punto y coma



```
public void run() {  
    int counter = 0;  
    for (int i = 0; i < 10; i++); {  
        counter = counter + 1;  
    }  
}
```

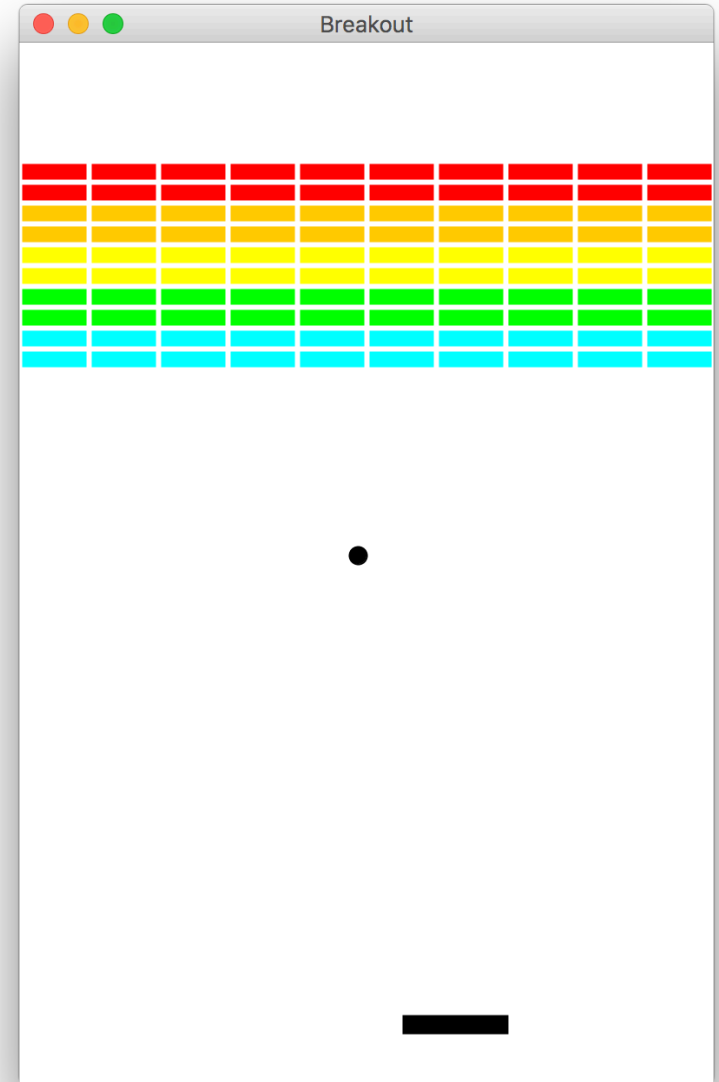
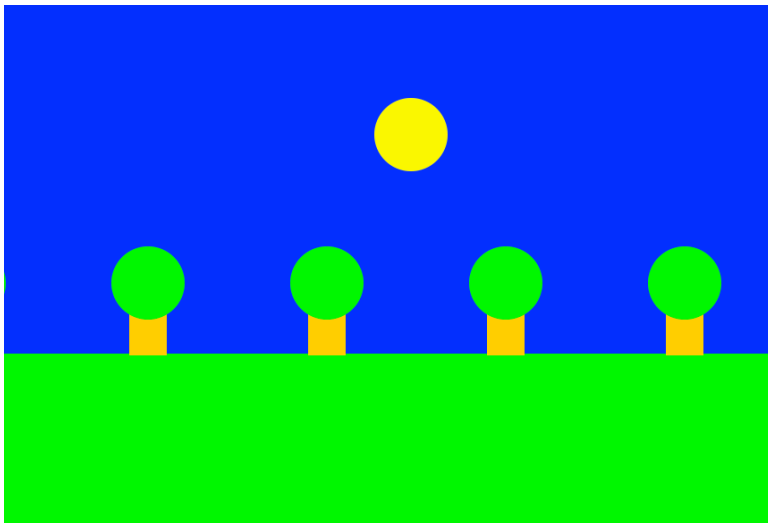
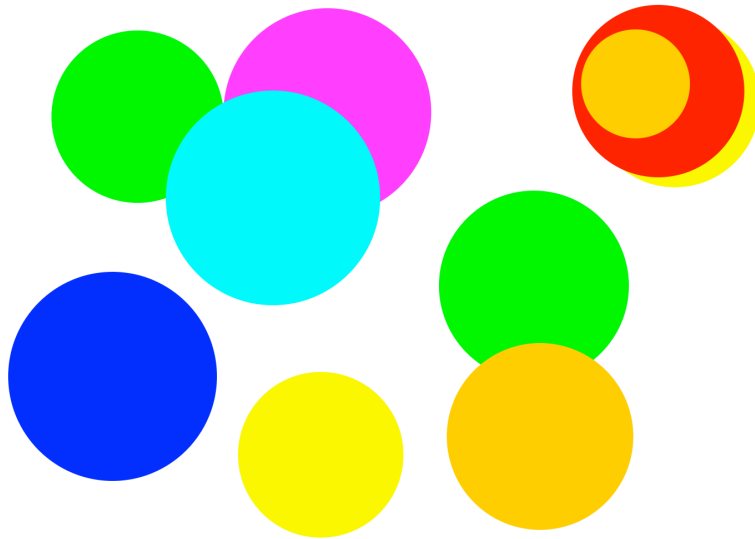


for, while, if, else, y { **NO** son amigos del ;
(punto y coma).



```
public void run() {  
    int counter = 0;  
    for (int i = 0; i < 10; i++) {  
        counter = counter + 1;  
    }  
}
```

Más allá de los programas de consola



Coordenadas

0,0

x 40,20

x 120,40

x 40,120

darAncho () ;

darAlto () ;

Métodos de las interfaces gráficas

<code>agregar (objeto)</code>	Añade el objeto al canvas encima de los otros objetos
<code>agregar (objeto, x, y)</code>	Mueve el objeto a (x, y) y después lo añade al canvas
<code>quitar (objeto)</code>	Remueve el objeto del canvas
<code>quitarTodo ()</code>	Remueve todos los objetos del canvas
<code>darObjetoA (x, y)</code>	Devuelve el objeto que está más encima en (x, y), o <code>null</code> si no hay ninguno.
<code>darAncho ()</code>	Devuelve el ancho en pixeles de la totalidad del canvas
<code>darAlto ()</code>	Devuelve la altura en pixeles de la totalidad del canvas
<code>cambiarFondo (c)</code>	Cambia el color del fondo del canvas a <code>c</code>
<code>pausa (milisegundos)</code>	Pausa el programa por el tiempo especificado en milisegundos
<code>esperarClic ()</code>	Suspende el programa hasta que el usuario hace clic con el mouse

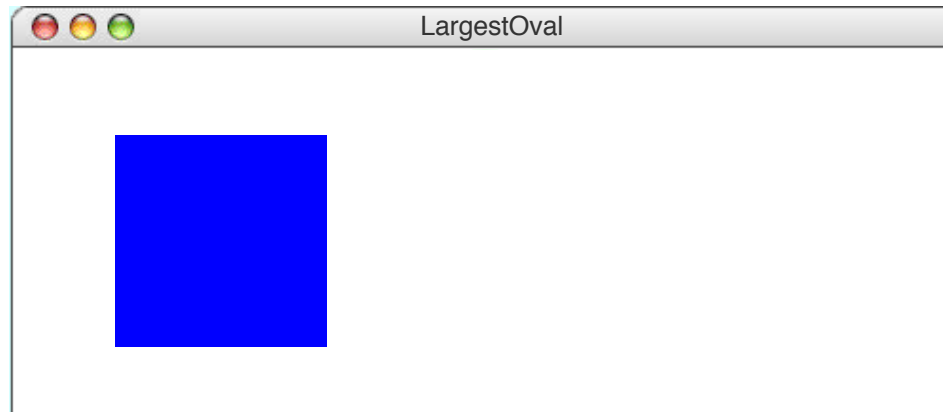
<https://uniandes.csbridge.org/es/handouts/graphics.html>

SRect

SRect es un tipo de variable que almacena un rectángulo.

Por ejemplo, el siguiente método `run` muestra un rectángulo:

```
public void run() {  
    SRect rect = new SRect(200, 200);  
    rect.cambiarRelleno(true);  
    rect.cambiarColor(Color.BLUE);  
    agregar(rect, 50, 50);  
}
```



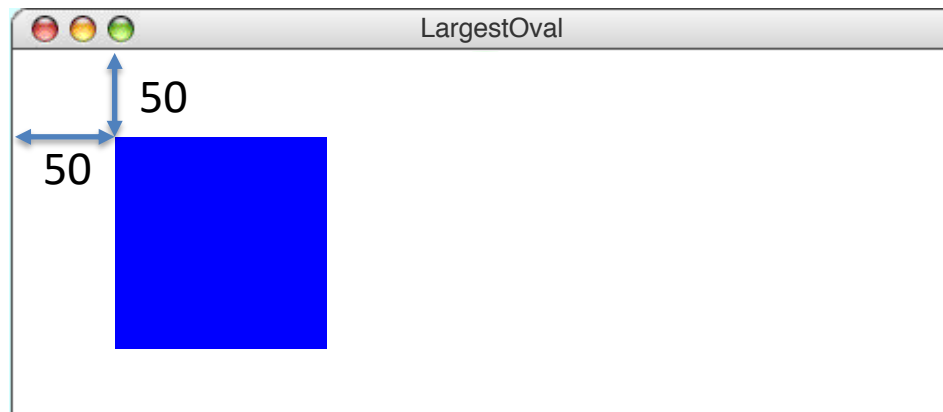
SRect

SRect es un tipo de variable que almacena un rectángulo.

Por ejemplo, el siguiente método `run` muestra un rectángulo:

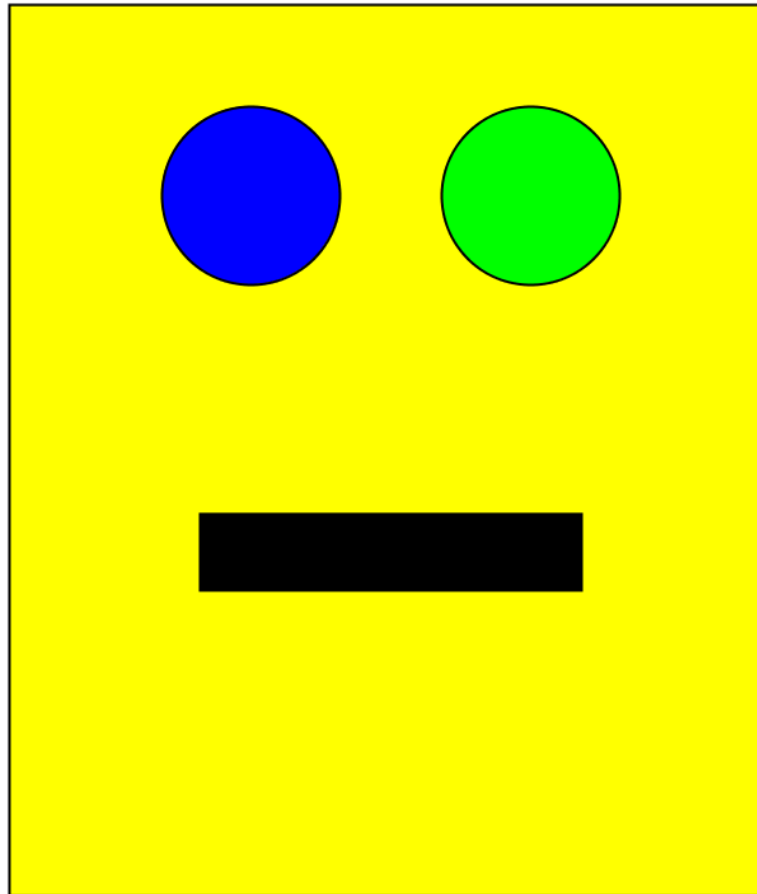
```
public void run() {  
    SRect rect = new SRect(200, 200);  
    rect.cambiarRelleno(true);  
    rect.cambiarColor(Color.BLUE);  
    agregar(rect, 50, 50);  
}
```

El punto de origen
de un SRect es su
esquina superior
izquierda



El abuelo de Karel

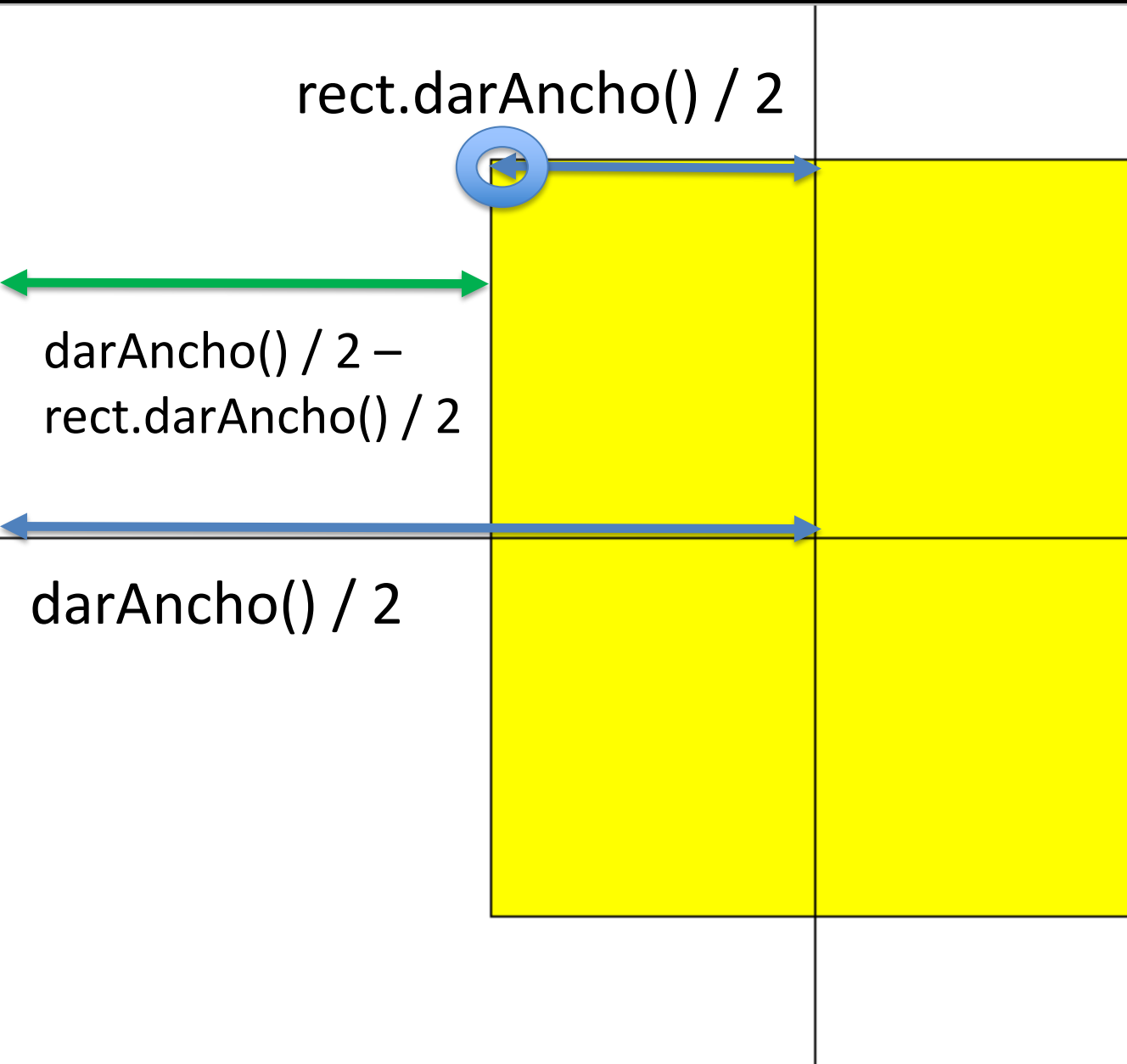
Robot Face



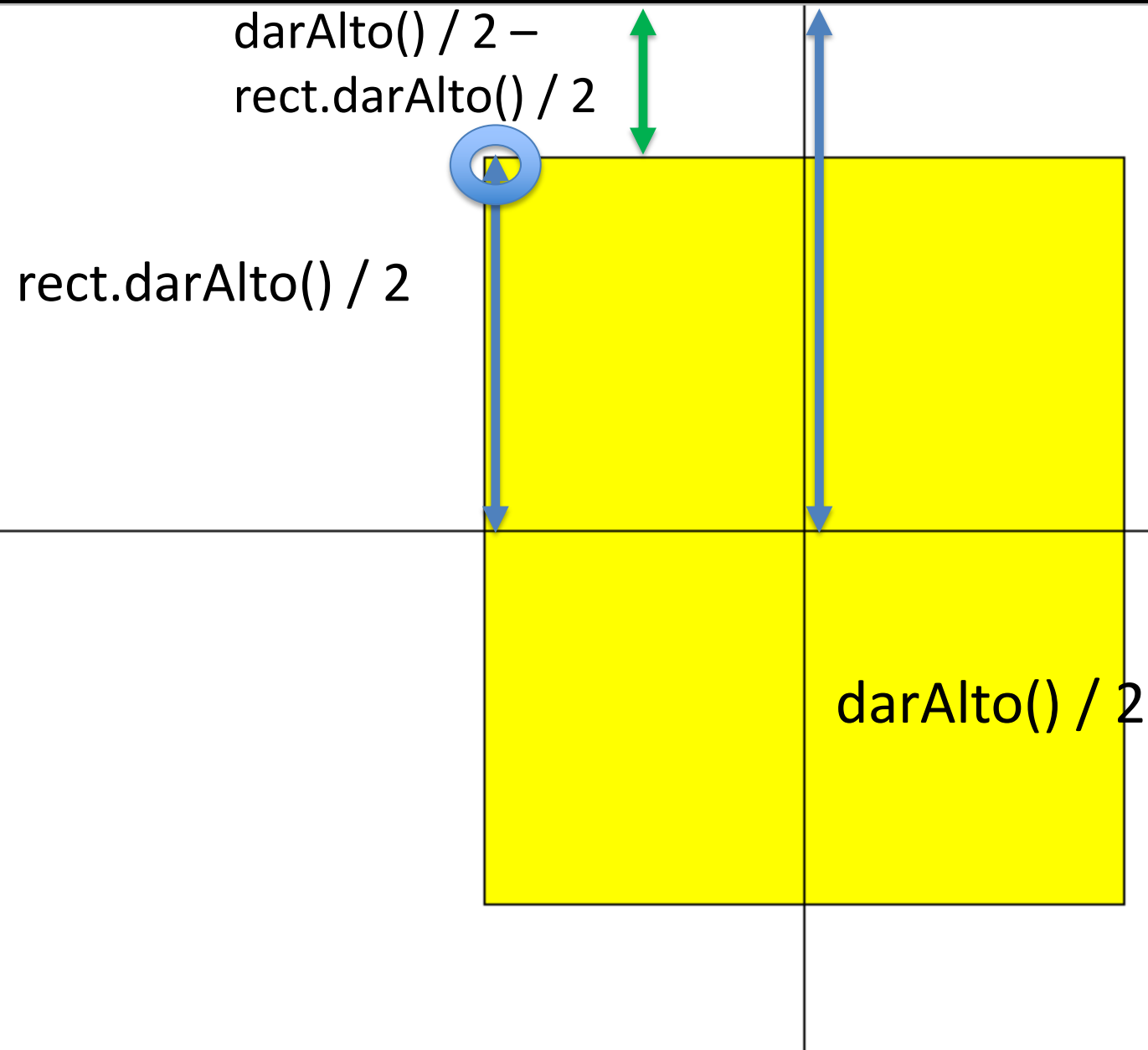
El abuelo de Karel



El abuelo de Karel



El abuelo de Karel

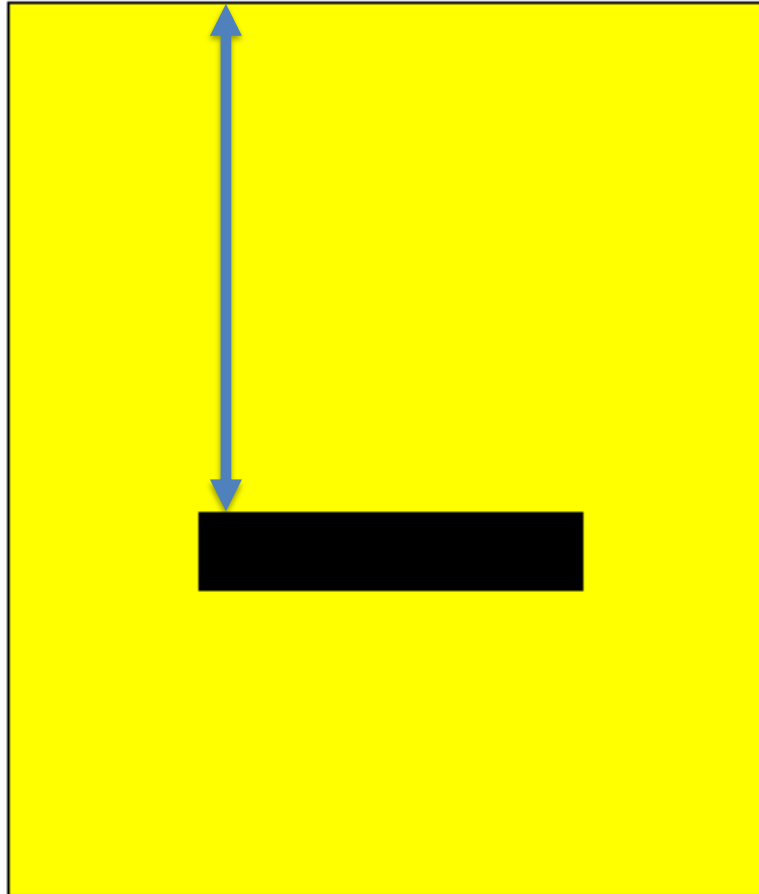


Constantes

```
/*The distance from the top of the head to the top of the mouth:*/  
private static final int MOUTH_Y_OFFSET = 200;
```

El abuelo de Karel

```
/*The distance from the top of the head to the top of the mouth:*/  
private static final int MOUTH_Y_OFFSET = 200;
```



SEtiqueta

Una caja de texto que se puede agregar al canvas

```
public class HolaPrograma extends EsGraphics {  
    public void run() {  
        SEtiqueta etiqueta = new SEtiqueta("hola, mundo");  
        etiqueta.cambiarFuente("SansSerif-36");  
        etiqueta.cambiarColor(Color.RED);  
        agregar(etiqueta, 100, 75);  
    }  
}
```



SEtiqueta

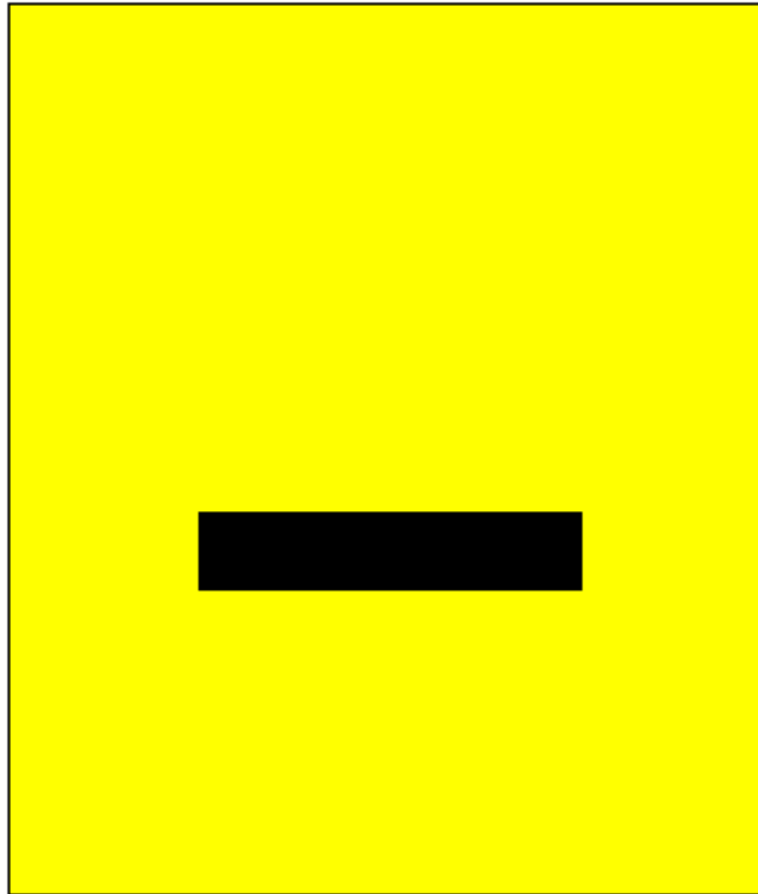
Una caja de texto que se puede agregar al canvas

```
public class HolaPrograma extends EsGraphics {  
    public void run() {  
        SEtiqueta etiqueta = new SEtiqueta("hola, mundo");  
        etiqueta.cambiarFuente("SansSerif-36");  
        etiqueta.cambiarColor(Color.RED);  
        agregar(etiqueta, 100, 75);  
    }  
}
```



El abuelo de Karel

↕ Robot Face



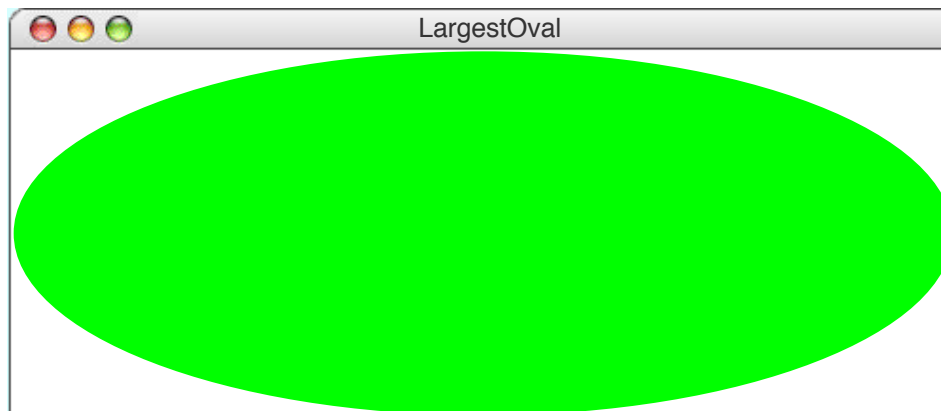
```
/*The distance from the top of the screen to the base of the label:*/  
private static final int LABEL_Y = 50;
```

SOvalo

Una forma *elíptica* que se define por los límites del rectángulo que la rodea.

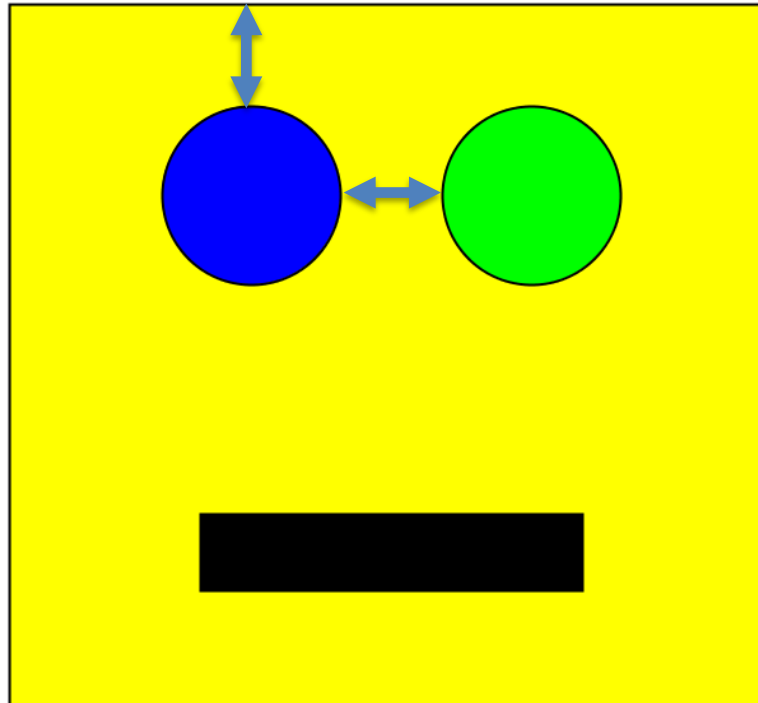
Por ejemplo, el siguiente método `run` crea el óvalo más grande que cabe dentro del canvas:

```
public void run() {  
    SOvalo oval = new SOvalo(darAncho(), darAlto());  
    oval.cambiarRelleno(true);  
    oval.cambiarColor(Color.GREEN);  
    agregar(oval, 0, 0);  
}
```



El abuelo de Karel

Robot Face

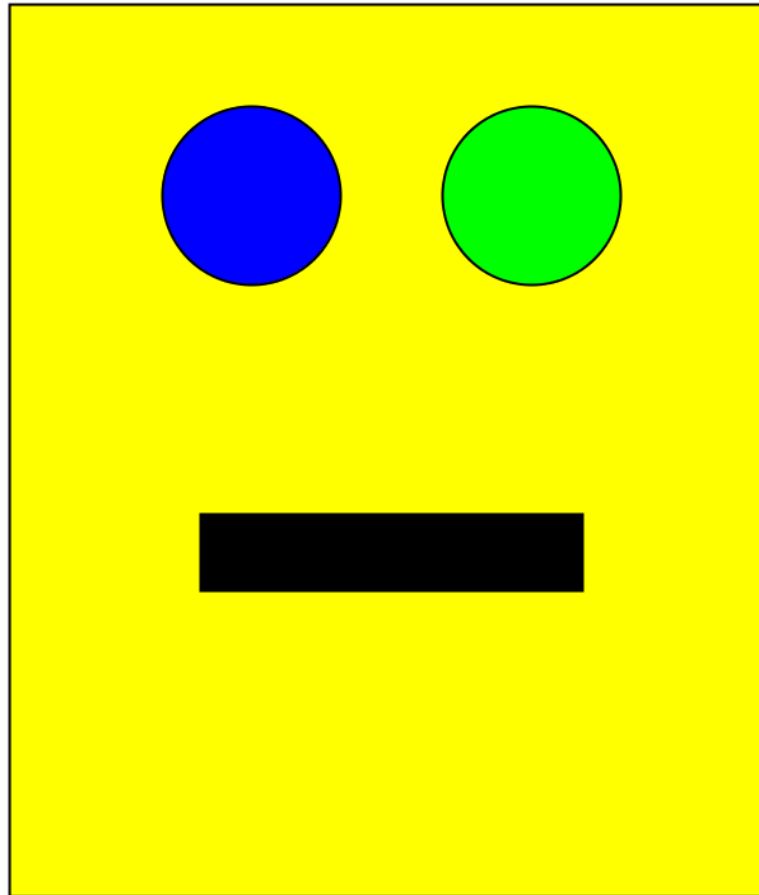


```
/*The distance from the top of the head to the top of the eyes:*/  
private static final int EYE_Y_OFFSET = 40;
```

```
/*The distance in between the two eyes:*/  
private static final int EYE_X_SEPARATION = 40;
```

El abuelo de Karel

Robot Face

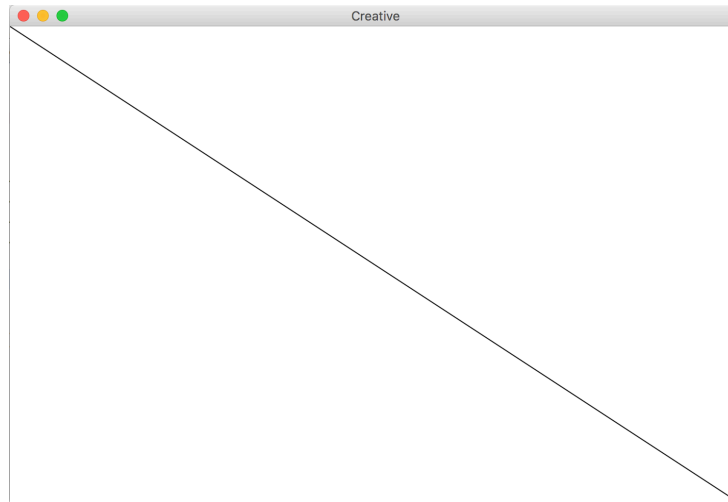


SLinea

El tipo `SLinea` representa una línea que se define por un punto inicial y un punto final.

Por ejemplo, el siguiente método `run` crea una línea diagonal a través del canvas:

```
public void run() {  
    SLinea linea = new SLinea(0,0, darAncho(), darAlto());  
    agregar(linea);  
}
```



El abuelo de Karel

