

Introducción al comando **Manipulate** en *Mathematica*

José Luis Gómez-Muñoz
<http://homepage.cem.itesm.mx/lgomez>

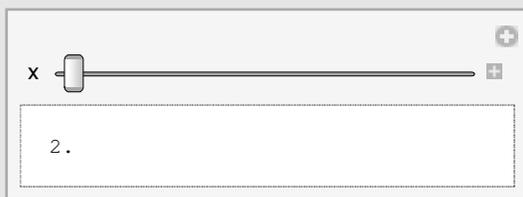
Manipulando números

Aquí se usa **Manipulate** para mostrar valores de x desde 2 hasta 5.

Usa el ratón para arrastrar (botón izquierdo oprimido) la barra deslizador

In[10]:= **Manipulate**[x , { x , 2, 5}]

Out[10]=



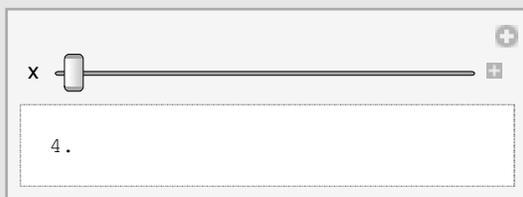
Aquí se usa **Manipulate** para evaluar x^2 en valores de x desde 2 hasta 5.

Usa el ratón para arrastrar (botón izquierdo oprimido) la barra deslizador

Oprime el signo + que está adentro de un cuadro para abrir controles de animación

In[11]:= **Manipulate**[x^2 , { x , 2, 5}]

Out[11]=



■ Ejercicio 1

Utiliza **Manipulate** para evaluar la expresión $\sqrt{1-x^2}$ para valores de x desde -1 hasta 1

Manipulando puntos (parejas de números) y vectores en gráficas

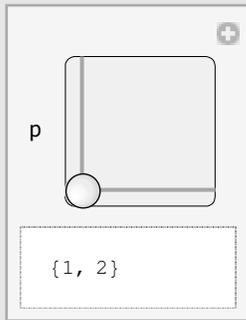
Aquí se usa **Manipulate** para mostrar puntos (parejas de números) variando desde **P(1,2)** hasta **Q(11,9)**.

Usa el ratón para arrastrar (botón izquierdo oprimido) el punto de control en dos dimensiones

In[27]:=

```
Manipulate[p, {p, {1, 2}, {11, 9}}]
```

Out[27]=



Recordemos de la práctica anterior como se pueden dibujar vectores (flechas) en *Mathematica*.

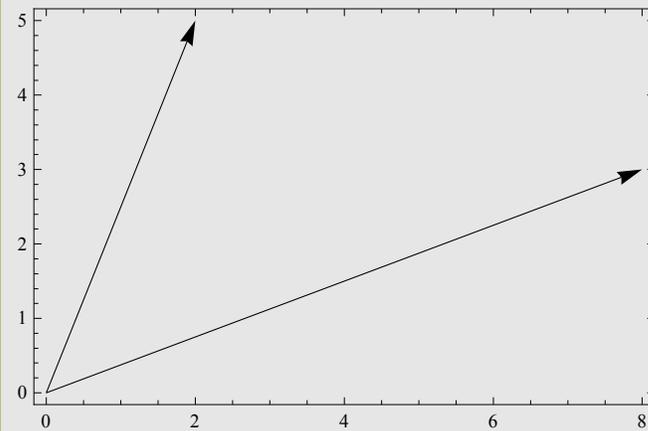
```
Graphics[ {
  Arrow[{{0, 0}, {2, 5}}],
  Arrow[{{0, 0}, {8, 3}}]
}, Frame -> True]
```

Observa que al seguir escribiendo después del "guión pequeño" y el "mayor que" \rightarrow , estos se convierten en una flecha \rightarrow

In[28]:=

```
Graphics [ {
  Arrow[{{0, 0}, {2, 5}}],
  Arrow[{{0, 0}, {8, 3}}]
}, Frame  $\rightarrow$  True]
```

Out[28]=



Ahora combinamos los dos ejemplos anteriores para producir una imagen donde podemos manipular con el ratón a uno de los vectores.

Observa que **la primera flecha se modificó**, ahora es $\text{Arrow}[\{\{0,0\},\mathbf{p}\}]$, donde \mathbf{p} es el punto que será manipulado.

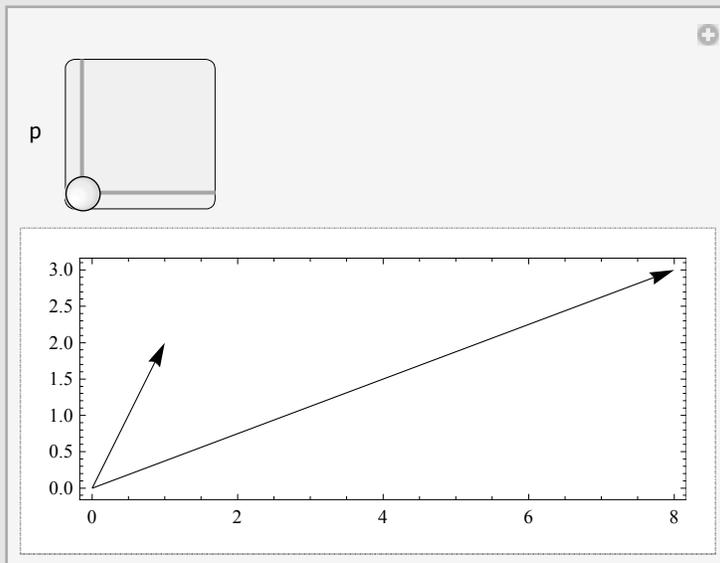
Observa también que este primer ejemplo no es del todo satisfactorio, porque cuando \mathbf{p} es manipulado para ser un punto que está más a la derecha o más arriba que el punto $P(8,3)$, la escala de la gráfica comienza a cambiar para que quepan las dos flechas.

Usa el ratón para arrastrar (botón izquierdo oprimido) el punto de control en dos dimensiones. La primera flecha debe cambiar de tamaño y dirección.

In[29]:=

```
Manipulate[
  Graphics[ {
    Arrow[{{0, 0}, p}],
    Arrow[{{0, 0}, {8, 3}}]
  }, Frame -> True],
  {p, {1, 2}, {11, 9}}
```

Out[29]=

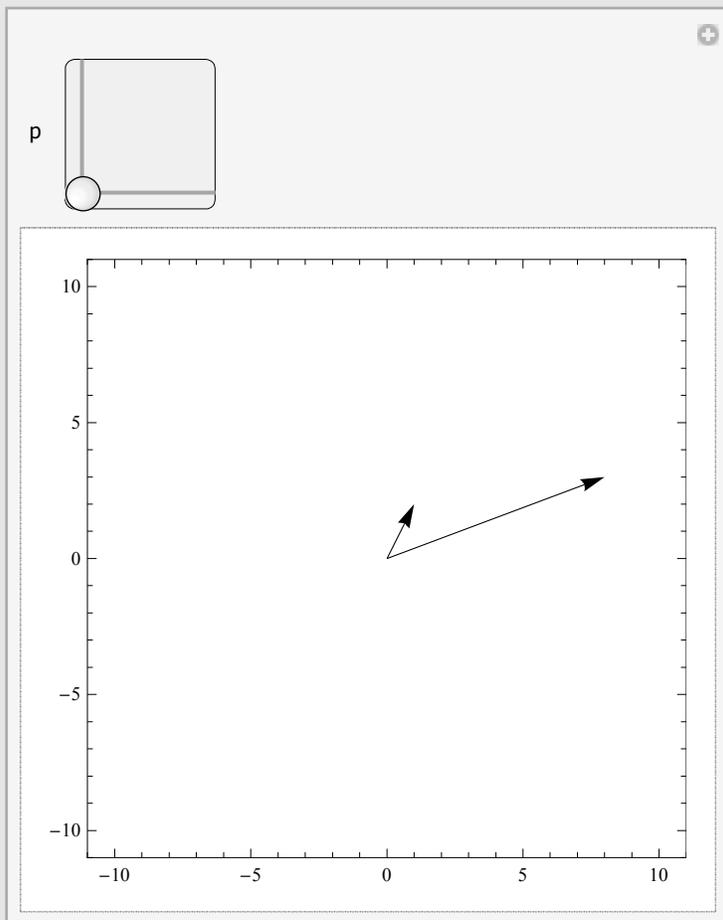


Aquí está una posible mejora al ejemplo de arriba, añadiendo la opción **PlotRange->11** adentro de **Graphics**. De esta manera el dibujo siempre se muestra para $P(x,y)$ desde $P(-11,-11)$ hasta $P(11,11)$:

In[39]:=

```
Manipulate[
  Graphics[ {
    Arrow[{{0, 0}, p}],
    Arrow[{{0, 0}, {8, 3}}]
  }, Frame → True, PlotRange → 11],
  {p, {1, 2}, {11, 9}}
```

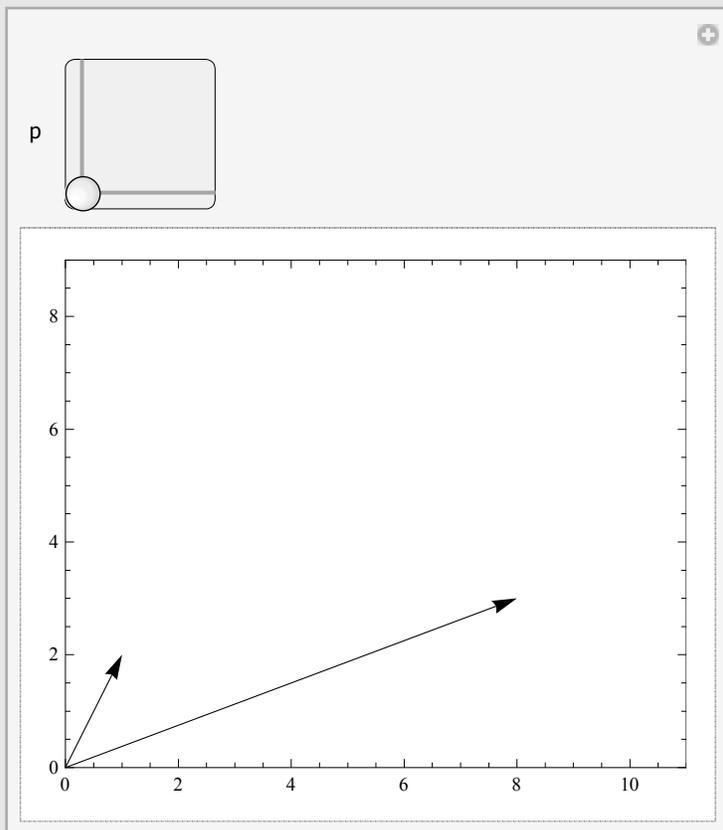
Out[39]=



La opción `PlotRange->{{xmin,xmax},{ymin,ymax}}` permite escoger un mejor rango de valores para este dibujo:

In[38]:=

```
Manipulate[
Graphics[ {
  Arrow[{{0, 0}, p}],
  Arrow[{{0, 0}, {8, 3}}]
}, Frame → True, PlotRange → {{0, 11}, {0, 9}}],
{p, {1, 2}, {11, 9}}
```



Out[38]=

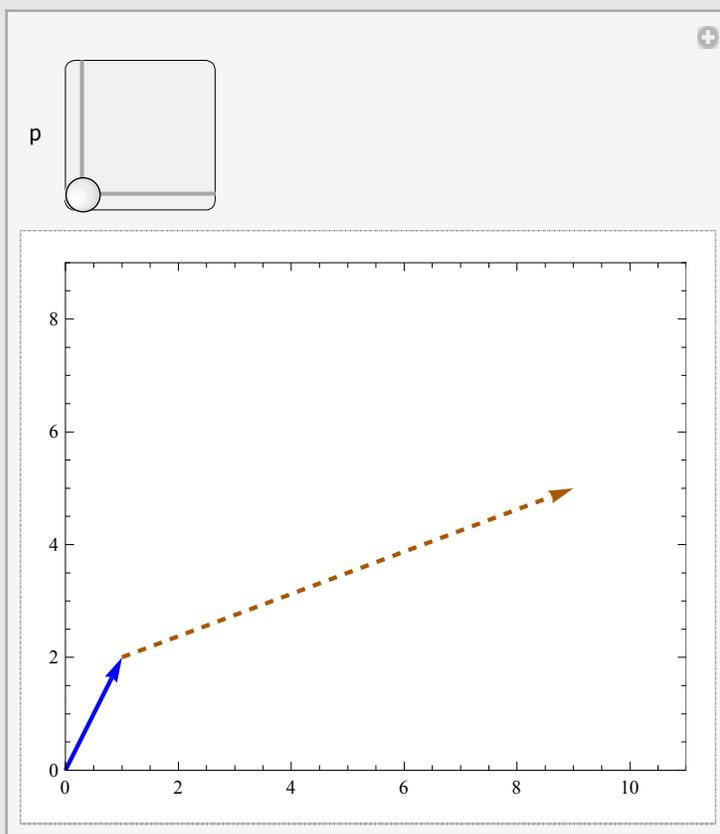
Ahora modificamos la segunda flecha `Arrow[$\{p, \{8, 3\}\}$]` para que sea dibujada desde la punta de la primera flecha, como en la suma de vectores. Además se añaden colores y otras especificaciones (directivas) para el estilo de las flechas

In[52]:=

```

Manipulate[
  Graphics[ {
    Thick,
    Blue,
    Arrow[{{0, 0}, p}],
    Dashed,
    Darker[Orange],
    Arrow[{p, p + {8, 3}}]
  }, Frame → True, PlotRange → {{0, 11}, {0, 9}},
  {p, {1, 2}, {11, 9}}]

```

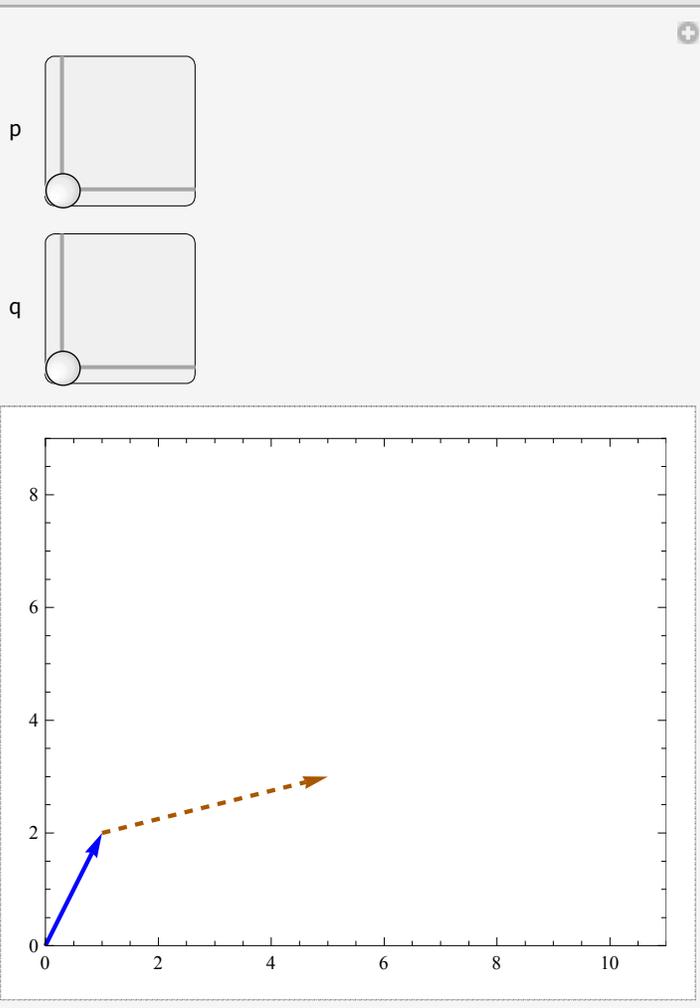


Out[52]=

Para manipular las dos flechas, modificamos otra vez la segunda flecha **Arrow**[{ p , $p + q$] y añadimos $\{q, \{1, 1\}, \{8, 3\}\}$ adentro del **Manipulate**:

In[99]:=

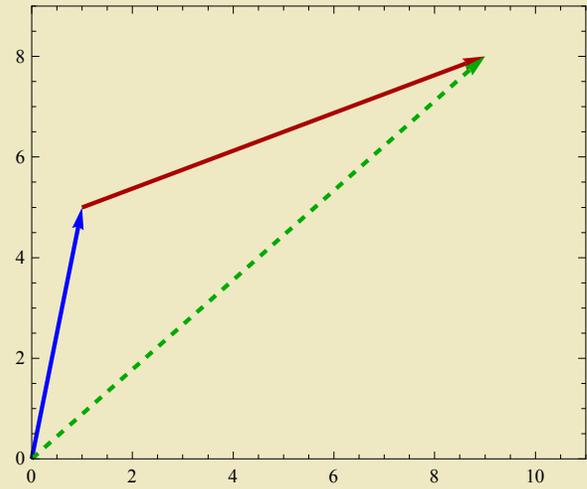
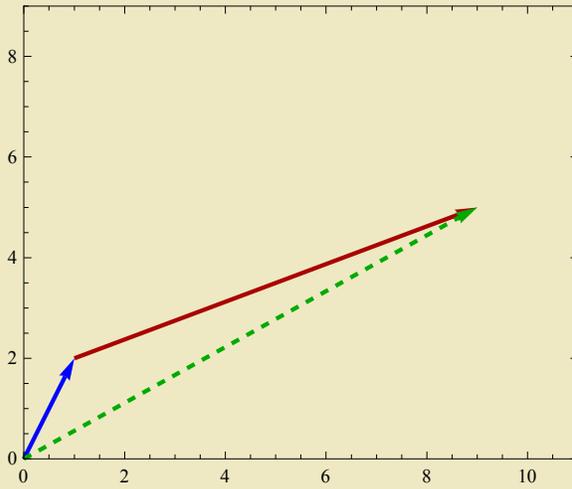
```
Manipulate[
  Graphics[ {
    Thick,
    Blue,
    Arrow[{{0, 0}, p}],
    Dashed,
    Darker[Orange],
    Arrow[{p, p + q}]
  }, Frame → True, PlotRange → {{0, 11}, {0, 9}},
  {p, {1, 2}, {11, 9}},
  {q, {4, 1}, {8, 3}}
]
```



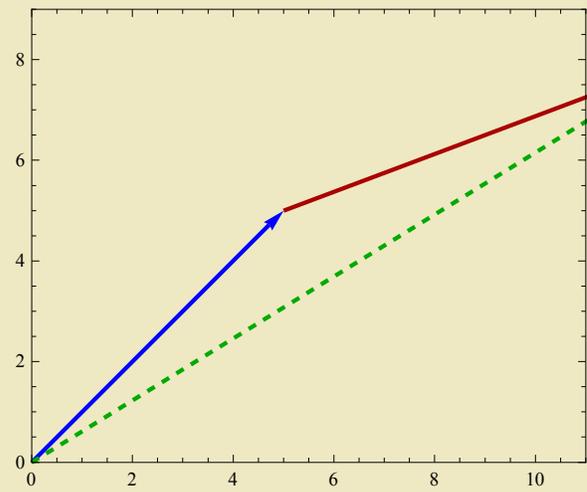
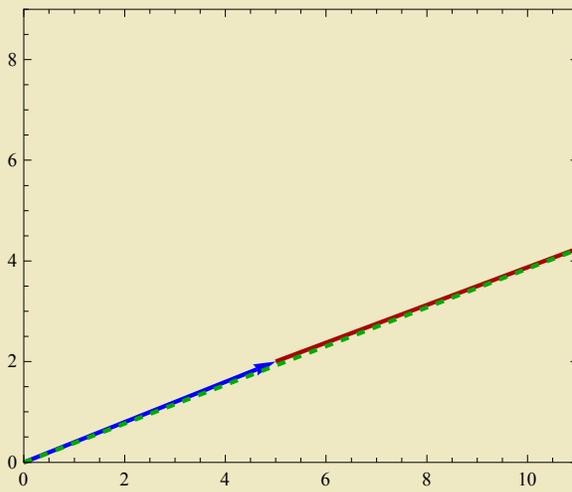
Out[99]=

Ejercicio 2

Usa **Manipulate** para dibujar tres flechas, representando dos vectores y el vector que resulta de sumarlos:



Out[58]=



Locator

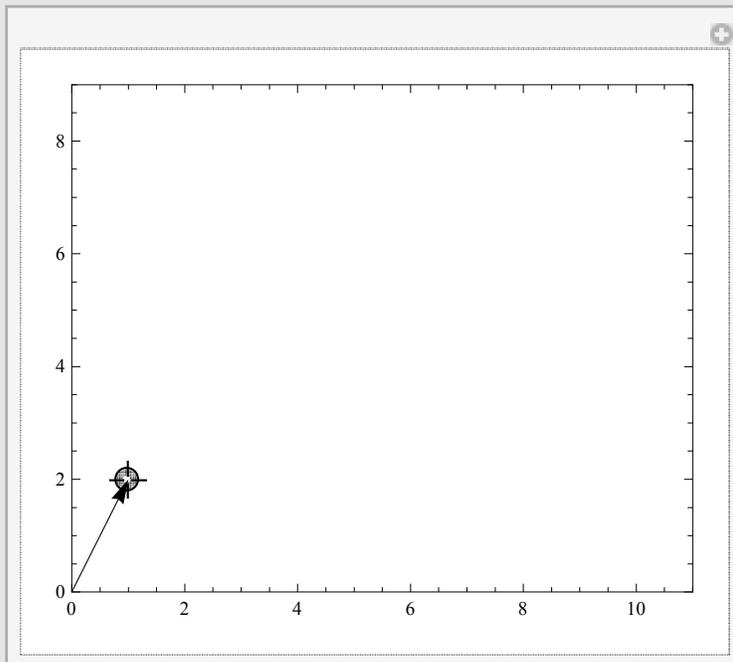
El comando **Locator** permite manipular la flecha oprimiendo con el ratón adentro del dibujo.

Usa el ratón para arrastrar (botón izquierdo oprimido) DIRECTAMENTE la punta de la flecha

In[73]:=

```
Manipulate[
  Graphics[ {
    Arrow[{{0, 0}, p}]
  }, Frame → True, PlotRange → {{0, 11}, {0, 9}},
  {{p, {1, 2}}, Locator}]
```

Out[73]=



A continuación se usan dos locators para manipular ambas flechas, además se añadió una tercera flecha representando la suma vectorial de las dos primeras

Usa el ratón para arrastrar (botón izquierdo oprimido) DIRECTAMENTE la puntas de la flechas

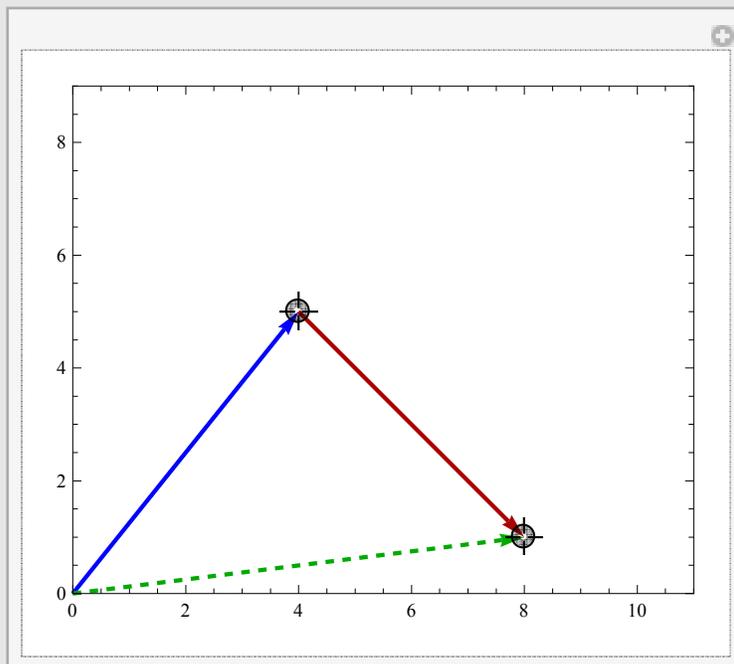
In[86]:=

```

Manipulate[
  Graphics[ {
    Thick,
    Blue,
    Arrow[{{0, 0}, p}],
    Darker[Red],
    Arrow[{p, suma}],
    Darker[Green],
    Dashed,
    Arrow[{{0, 0}, suma}]
  ], Frame → True, PlotRange → {{0, 11}, {0, 9}},
  {{p, {4, 5}}, Locator},
  {{suma, {8, 1}}, Locator}
]

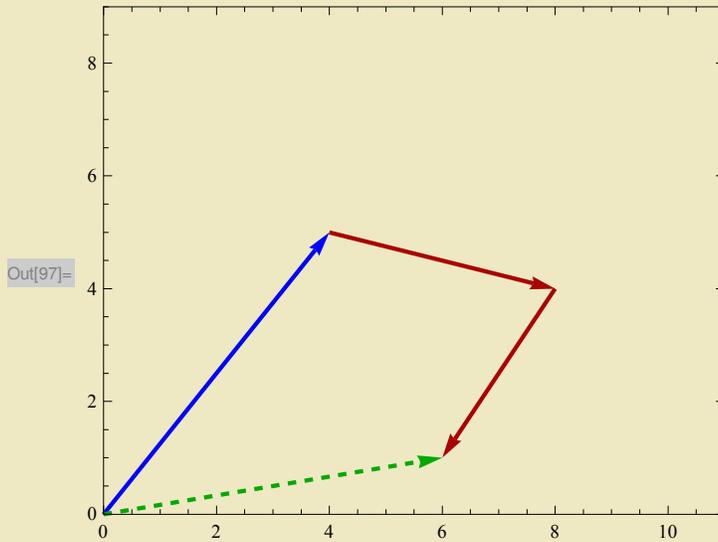
```

Out[86]=



■ Ejercicio 3

Haz una **Manipulate** con **Locators** para tres flechas, que cada flecha comience donde terminó la anterior, y que también se dibje una **cuarta flecha** representando la suma vectorial de las tres:



Flechas con "vida propia"

La norma (magnitud) del vector $(3,4)=3i+4j$ viene dada por:

In[108]:= $\sqrt{3^2 + 4^2}$

Out[108]= 5

Podemos obtener el mismo resultado con el comando Norm

In[107]:= **Norm**[{ 3, 4 }]

Out[107]= 5

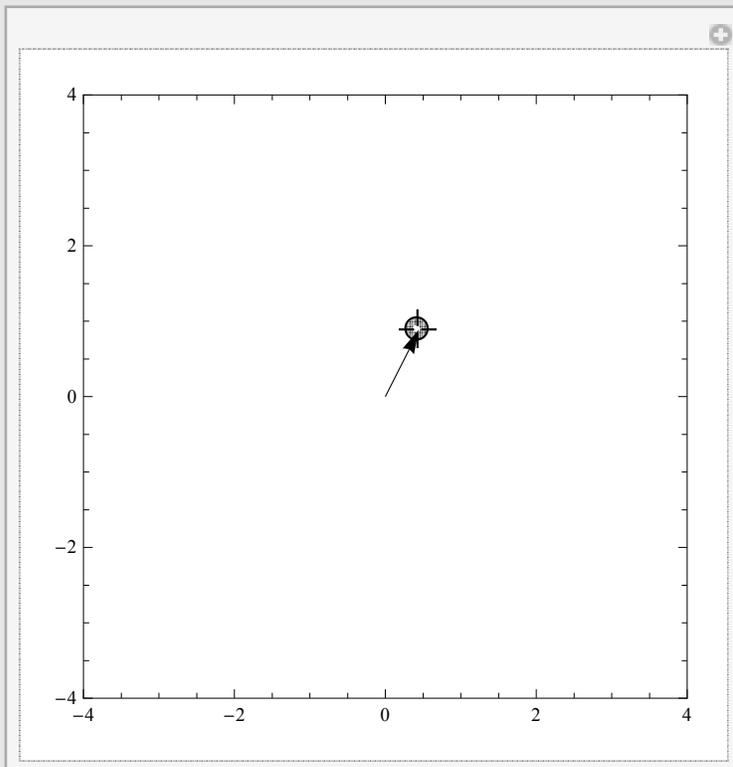
Con los comandos **If** y **Norm** podemos construir una flecha que automáticamente **reduce su tamaño un 99%**, y **lo continua reduciendo** hasta tener una magnitud de 1.

Usa el ratón para aumentar el tamaño de la flecha arrastrando (botón izquierdo oprimido) su punta, y observa como la flecha se vuelve a hacer pequeña cuando la "sueitas"

In[119]:=

```
Manipulate[
  If[Norm[p] > 1, p = 0.99 p];
  Graphics[ {
    Arrow[{{0, 0}, p}]
  }, Frame → True, PlotRange → 4],
  {{p, {1, 2}}, Locator}]
```

Out[119]=



Podemos extraer el primer elemento de un punto con el comand **Part**

In[120]:=

```
Part[{3, 4}, 1]
```

Out[120]=

3

Podemos extraer el segundo elemento de un punto con el comand **Part**

In[115]:=

```
Part[{3, 4}, 2]
```

Out[115]=

4

Podemos extraer el segundo elemento de un punto con el comand **Part**

```
In[122]:= Part[{123, 987}, 2]
```

```
Out[122]= 987
```

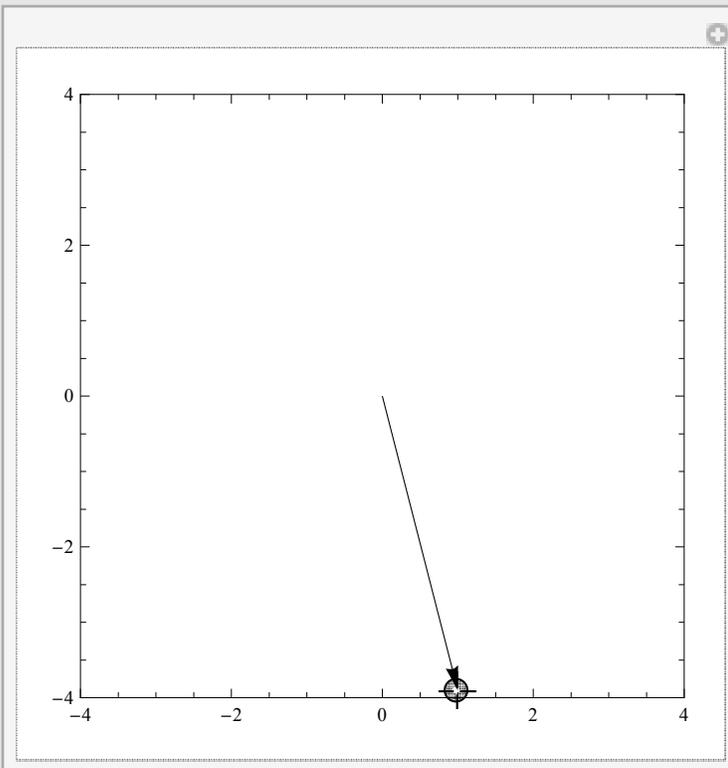
Con los comandos **If** y **Part** podemos construir una flecha que automáticamente **cae a velocidad constante**, y **continúa cayendo** hasta el fondo del dibujo.

Usa el ratón para aumentar el tamaño de la flecha arrastrando (botón izquierdo oprimido) su punta, y observa como la flecha "cae" cuando la "sueeltas"

NOTA: Si cae muy rápido, cambia el valor de 0.05 por un valor más pequeño para que caiga más despacio

```
In[129]:= Manipulate[
  If[Part[p, 2] > -3.9, p = p - {0, 0.05}];
  Graphics[ {
    Arrow[{{0, 0}, p}]
  }, Frame -> True, PlotRange -> 4],
  {{p, {1, 2}}, Locator}]
```

```
Out[129]=
```



■ Ejercicio 4

Realiza un **Manipulate** de una flecha que puedas arrastrar hacia cualquier punto y cuando la sueltes **flote solita hasta la parte superior del dibujo**