

MATHEMATICA

Manipulación y dinámica de expresiones

Ricardo Villafaña Figueroa

Contenido

Deslizador.....	3
Etiquetando la variable de cambio.....	3
Iniciando la variable de entrada.....	3
Cambiando la posición del deslizador	4
Utilizando un campo de entrada:.....	4
Menú de funciones	5
Etiquetando el menú de opciones	6
Menú PopUp	7
Casillas de verificación (Checkbox)	8
Botones (Radio button).....	9
Manipulación combinando campos de entrada con diferentes controles	10
Campos de entrada con Popup Menu.....	10
Campos de entrada con SetterBar	10
Campos de entrada con RadioButton	11
Manipulación en dos dimensiones.....	12
Locator	13
Manipulando Coordenadas Polares	16
Manipulando-graficando la derivada de una función.....	17
Manipulando con objetos gráficos (puntos, líneas):.....	19
Actualización dinámica de expresiones	23

Deslizador

```
Manipulate[Expand[(x + y)^n], {n, 1, 15, 1}]
```

n

$$x^2 + 2xy + y^2$$

Etiquetando la variable de cambio

```
Manipulate[Expand[(x + y)^n],
{n, 1, 15, 1, Appearance -> "Labeled"}]
```

n

$$x^{10} + 10x^9y + 45x^8y^2 + 120x^7y^3 + 210x^6y^4 + 252x^5y^5 + 210x^4y^6 + 120x^3y^7 + 45x^2y^8 + 10xy^9 + y^{10}$$

Iniciando la variable de entrada

```
Manipulate[Expand[(x + y)^n],
{{n, 5}, 1, 15, 1, Appearance -> "Labeled"}
]
```

n

$$x^5 + 5x^4y + 10x^3y^2 + 10x^2y^3 + 5xy^4 + y^5$$

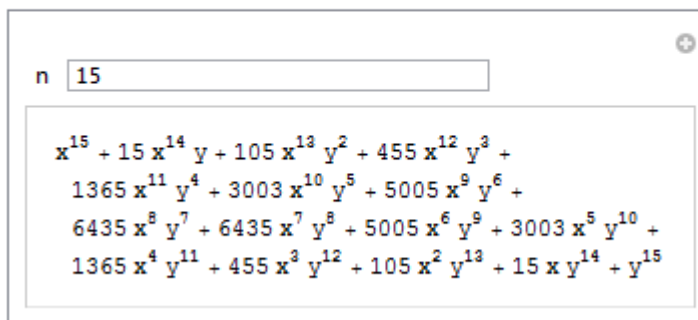
Cambiando la posición del deslizador

```
Manipulate[Expand[(x + y)^n],
{n, 1, 15, 1, Appearance -> "Labeled"},
ControlPlacement -> Bottom]
```

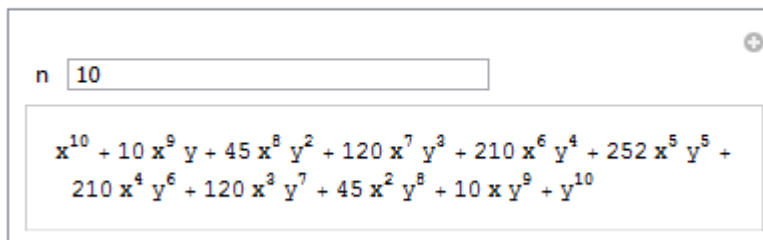


Utilizando un campo de entrada:

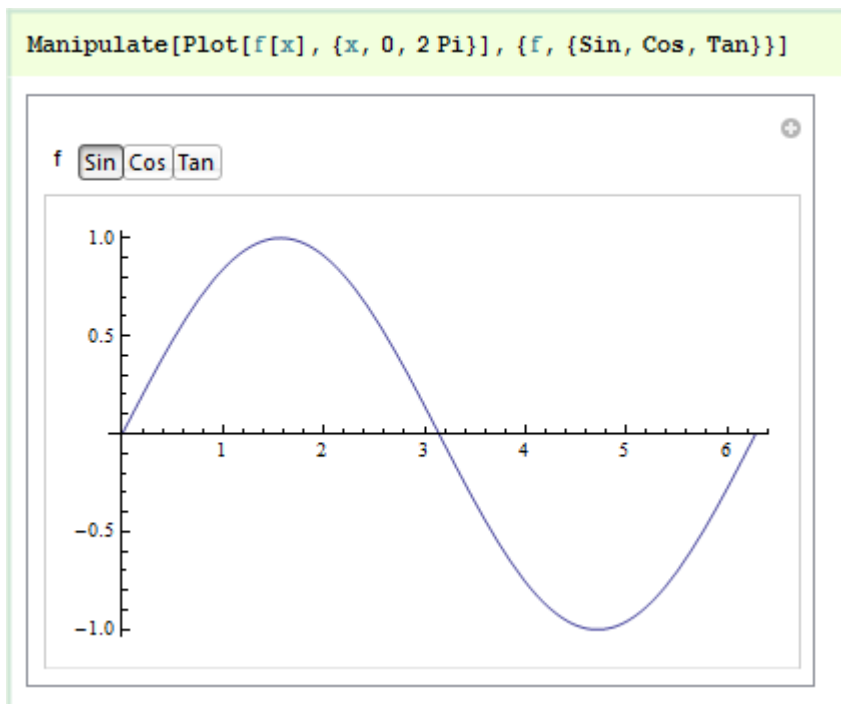
```
Manipulate[Expand[(x + y)^n], {n, 1, 15, 1},
ControlType -> InputField]
```



```
Manipulate[Expand[(x + y)^n], {n, 1, 15, 1, InputField}]
```

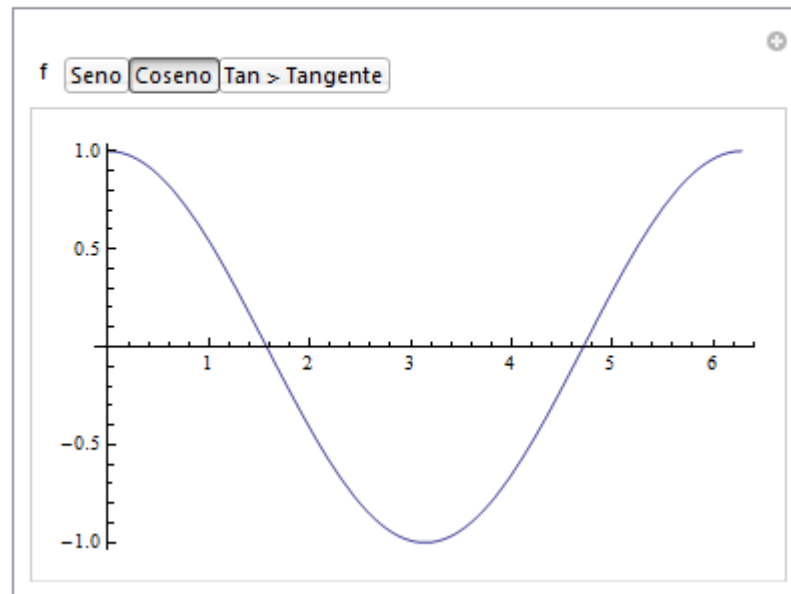


Menú de funciones



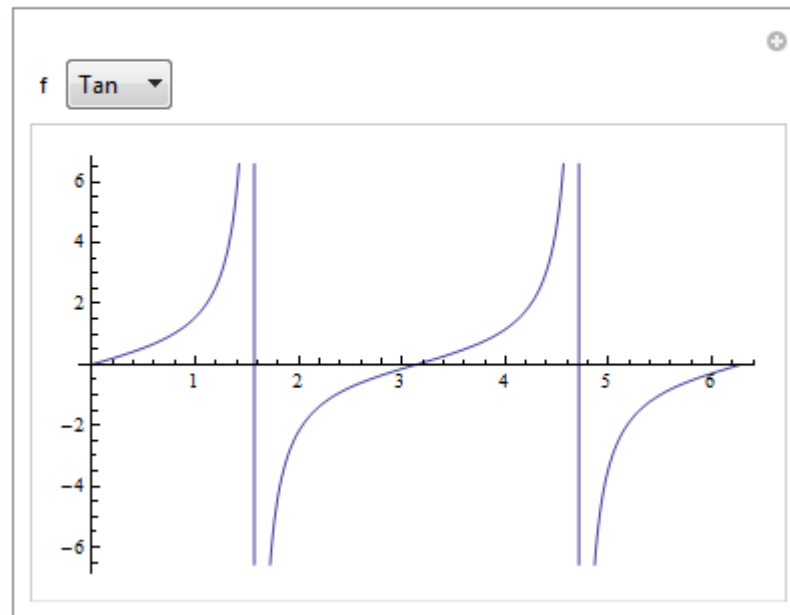
Etiquetando el menú de opciones

```
Manipulate[Plot[f[x], {x, 0, 2 Pi}],  
{f, {Sin → "Seno", Cos → "Coseno", Tan > "Tangente"}}]
```



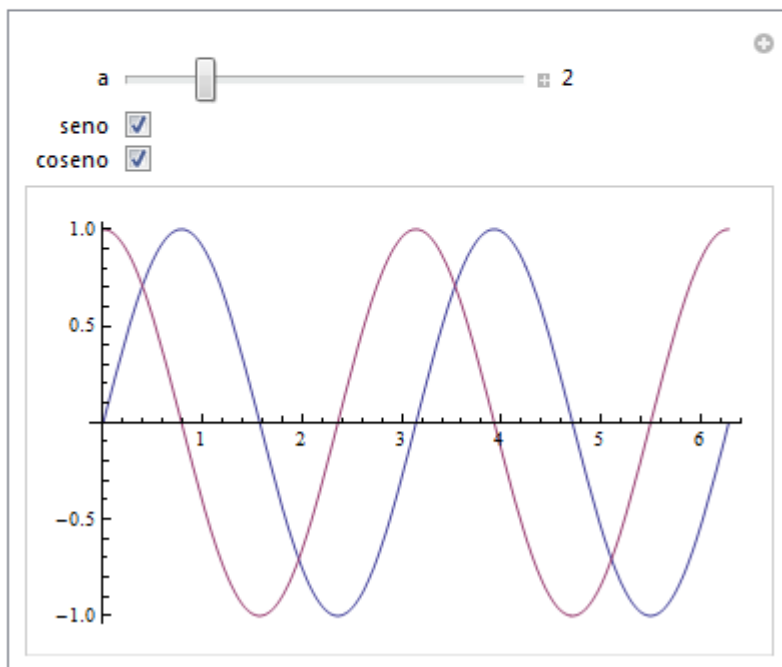
Menú PopUp

```
Manipulate[Plot[f[x], {x, 0, 2 Pi}],  
{f, {Sin, Cos, Tan}, ControlType -> PopupMenu]
```



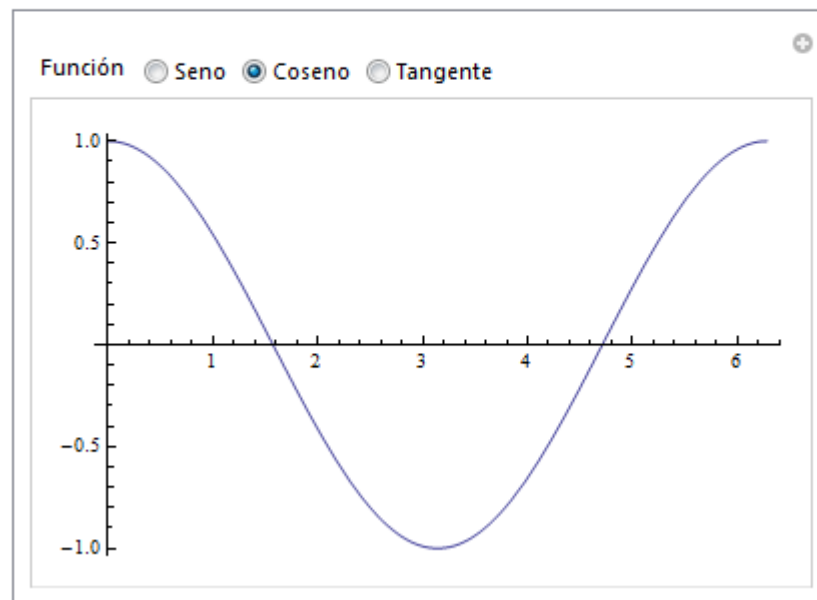
Casillas de verificación (Checkbox)

```
Manipulate[
  Plot[{
    If[seno, Sin[a x]],
    If[coseno, Cos[a x]],
    {x, 0, 2 Pi}},
  {{a, 1}, 1, 2 Pi, 1, Appearance -> "Labeled"},
  {seno, {True, False}},
  {coseno, {True, False}}]
```



Botones (Radio button)

```
Manipulate[Plot[f[x], {x, 0, 2 Pi}],  
  {{f, Sin, "Función"}, {Sin → "Seno", Cos → "Coseno",  
  Tan → "Tangente"}], RadioButtonBar}]
```



Manipulación combinando campos de entrada con diferentes controles

Campos de entrada con Popup Menu

```
Manipulate[funcion[expresion],
  {expresion, 3 (x + 1)},
  {funcion, {Simplify, Expand, Factor}, ControlType -> PopupMenu}]
```

The screenshot shows a Mathematica Manipulate interface. At the top, there is a text input field labeled 'expresion' containing the mathematical expression $3(1+x)$. Below it is a dropdown menu labeled 'funcion' with 'Factor' selected. At the bottom, there is a large text area displaying the result of the operation: $3(1+x)$.

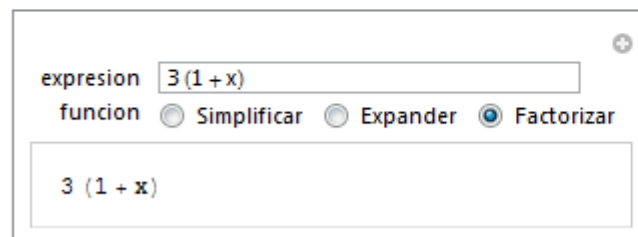
Campos de entrada con SetterBar

```
Manipulate[funcion[expresion],
  {expresion, 3 (x + 1)},
  {funcion, {Simplify -> " Simplificar ", Expand -> " Expander ",
    Factor -> " Factorizar "}}]
```

The screenshot shows a Mathematica Manipulate interface. At the top, there is a text input field labeled 'expresion' containing the mathematical expression $3(1+x)$. Below it is a SetterBar labeled 'funcion' with three buttons: 'Simplificar', 'Expander', and 'Factorizar'. At the bottom, there is a large text area displaying the result of the operation: $3(1+x)$.

Campos de entrada con RadioButton

```
Manipulate[funcion[expresion],  
  {expresion, 3 (x + 1)},  
  {funcion, {Simplify → " Simplificar ", Expand → " Expander ",  
    Factor → " Factorizar "}, ControlType → RadioButton}]
```



expression

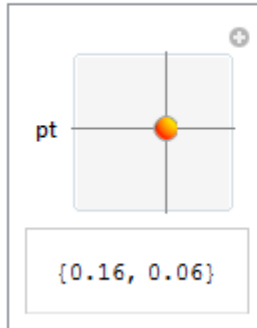
funcion Simplificar Expander Factorizar

3 (1 + x)

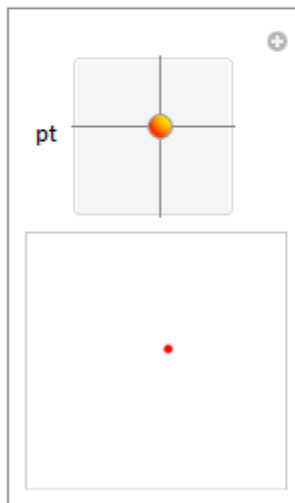
Manipulación en dos dimensiones

$\{var, \{x_{min}, y_{min}\}, \{x_{max}, y_{max}\}\}$

```
Manipulate[pt, {pt, {-1, -1}, {1, 1}}
```



```
Manipulate[
  Graphics[{Red, PointSize[Medium], Point[pt]},
    PlotRange -> 1, ImageSize -> 100],
  {pt, {-1, -1}, {1, 1}}
]
```



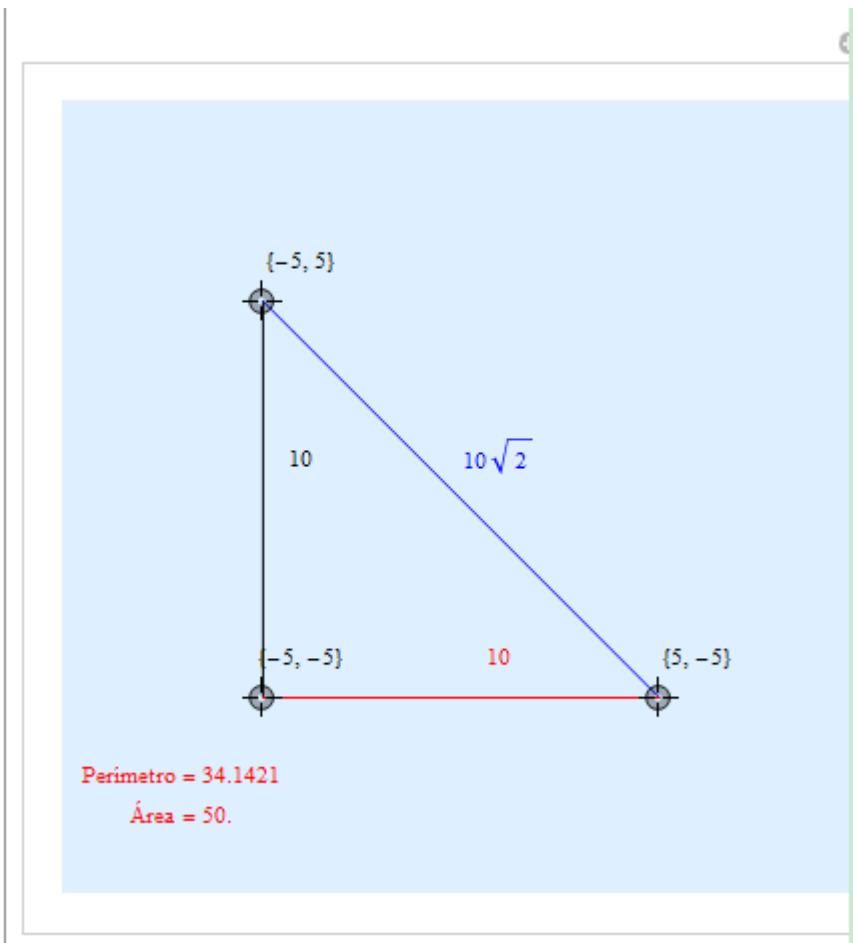
Locator

```

Manipulate[
Module[{d1, d2, d3, rea, perimetro, p, puntoMedio},
  puntoMedio[A_, B_] :=
  {  $\frac{A[[1]] + B[[1]]}{2}$ ,  $\frac{A[[2]] + B[[2]]}{2}$  };
  d1 = EuclideanDistance[p1, p2];
  d2 = EuclideanDistance[p2, p3];
  d3 = EuclideanDistance[p3, p1];
  perimetro = N[d1 + d2 + d3];
  (* fórmula de Herón*)
  p = (d1 + d2 + d3) / 2; (* semiperímetro *)
  area = N[Sqrt[p (p - d1) (p - d2) (p - d3)]];

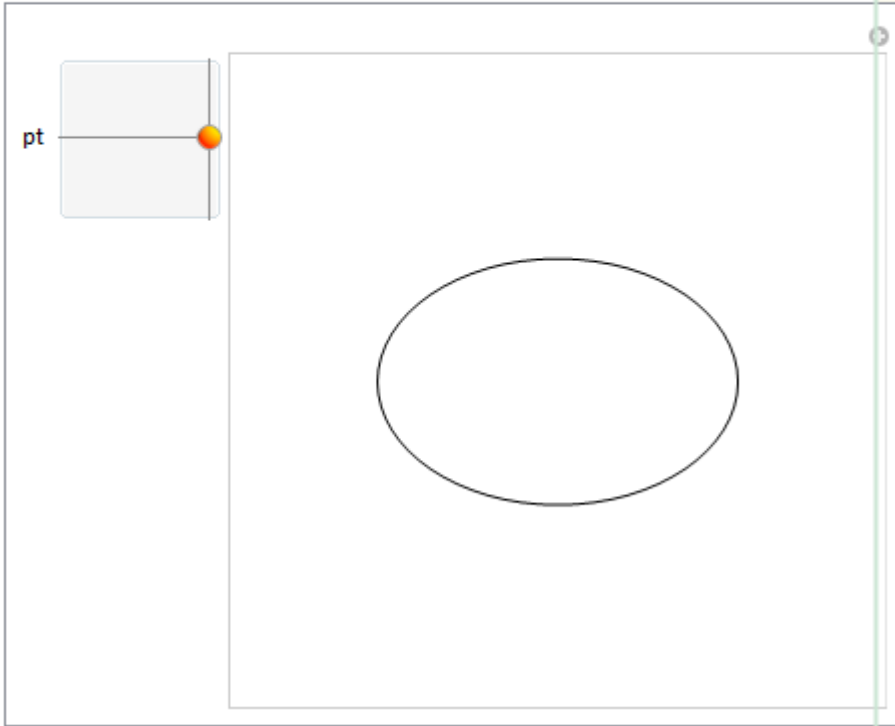
  Graphics[{
    Text[p1, p1 + 1],
    Text[p2, p2 + 1],
    Text[p3, p3 + 1],
    Line[{p1, p2}],
    Text[d1, puntoMedio[p1, p2] + 1],
    Blue,
    Line[{p2, p3}],
    Text[d2, puntoMedio[p2, p3] + 1],
    Red,
    Line[{p3, p1}],
    Text[d3, puntoMedio[p3, p1] + 1],
    Text["Perímetro = " <> ToString[perimetro],
      {-7, -7}],
    Text["Área = " <> ToString[area], {-7, -8}]
  ],
  Background → LightBlue, PlotRange → 10]
],
{{p1, {-5, -5}}, Locator},
{{p2, {-5, 5}}, Locator},
{{p3, {5, -5}}, Locator}
]

```



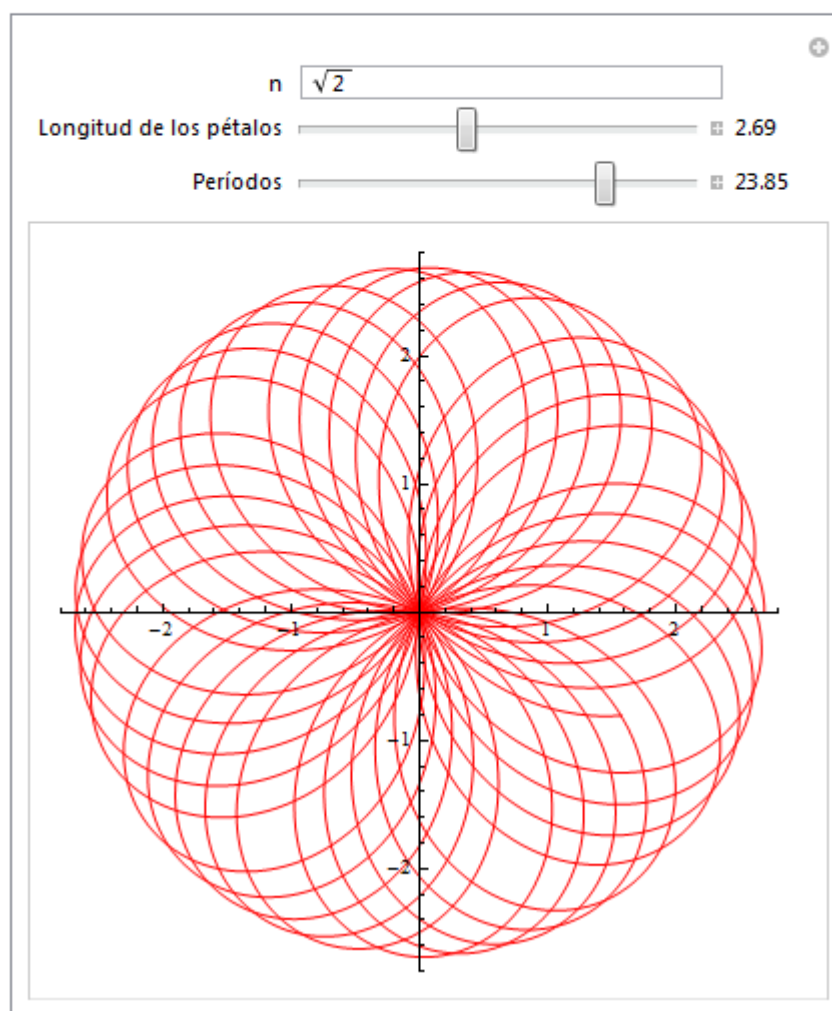
Elipse

```
Manipulate[
  Graphics[Circle[{0, 0}, pt], PlotRange → 5,
    ImageSize → 300],
  {pt, {1, 1}, {3, 3}, ControlPlacement → Left}
]
```



Manipulando Coordenadas Polares

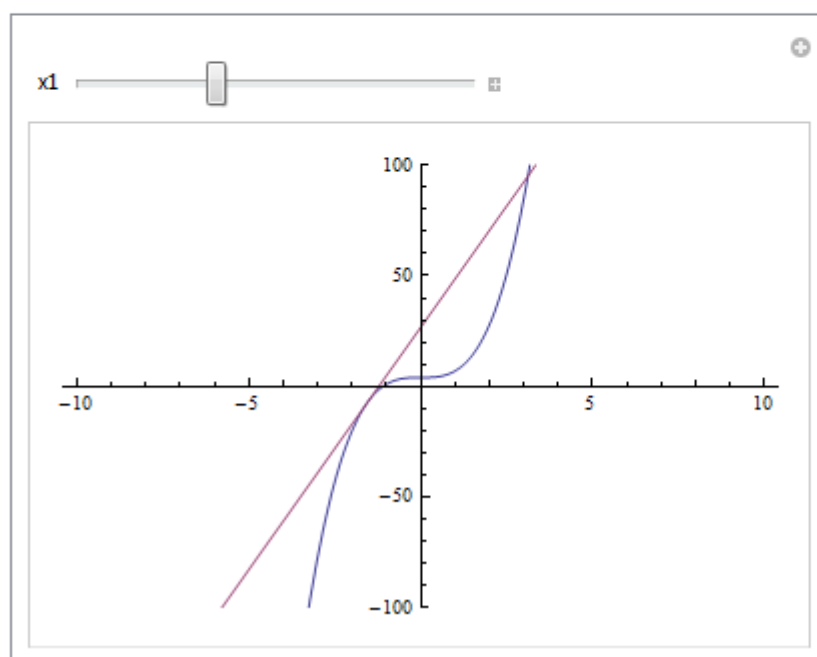
```
Manipulate[
  PolarPlot[a Cos[n  $\theta$ ], { $\theta$ , 0, p  $\pi$ },
    PlotStyle -> Directive[Red]],
  {n, Sqrt[2], 15, ControlType -> InputField},
  {{a, 3, "Longitud de los pétalos"}, 1, 5,
    Appearance -> "Labeled"},
  {{p, 2, "Períodos"}, 1, 30, Appearance -> "Labeled"}
]
```



Manipulando-graficando la derivada de una función

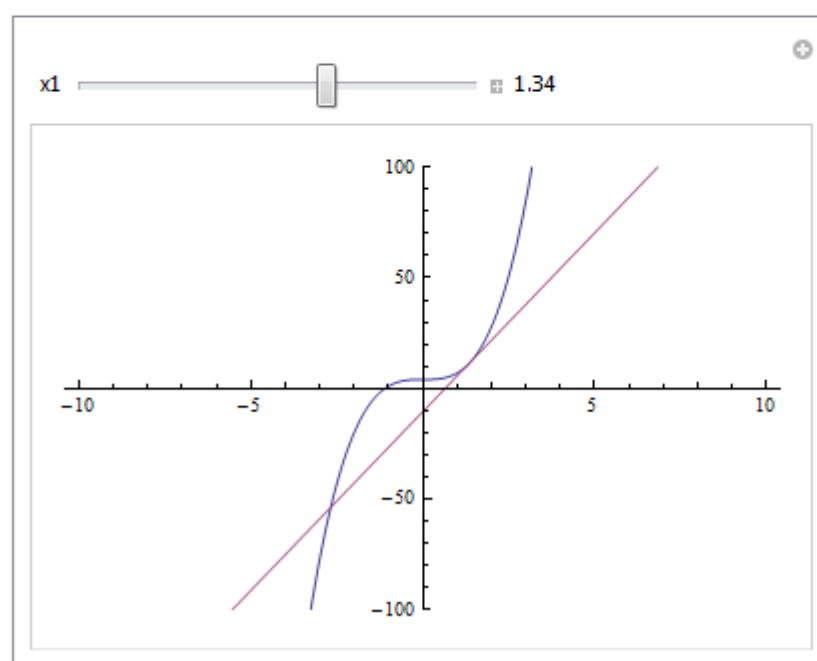
Primera aproximación (uso de paréntesis para agrupar):

```
Manipulate[
(
d = D[f1[x], x];
m = d /. x -> x1;
p1 = {x1, f1[x1]};
l1 = recta[p1, m];
Plot[{f1[x], l1}, {x, -10, 10}, PlotRange -> 100]
),
{x1, -5, 5}]
```



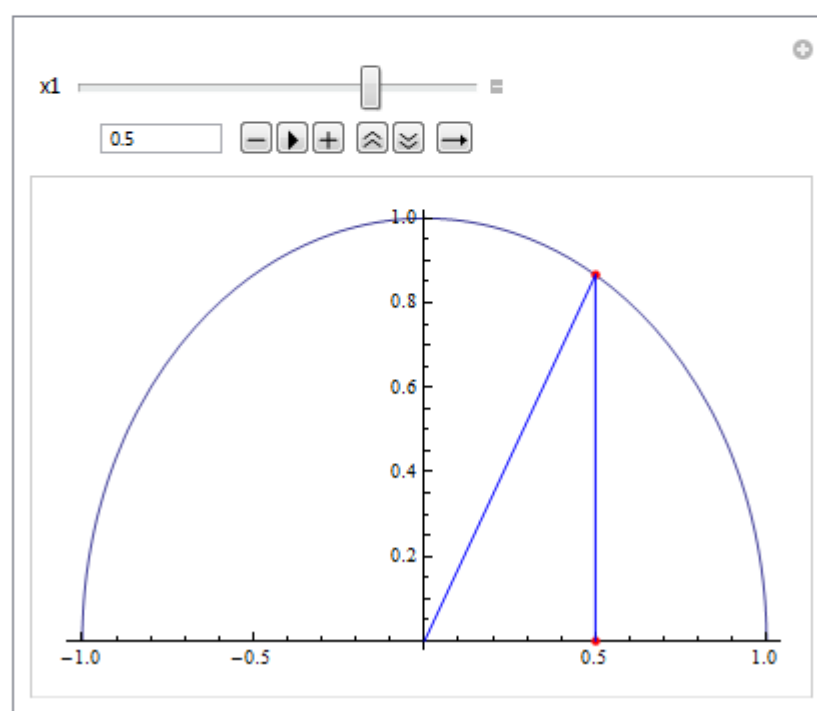
Segunda versión (uso de la función Module-manejo de variables locales):

```
Manipulate[
Module[{f1, recta, d, m, p1, l1},
  f1[x_] := 3 x3 + 4;
  recta[P_, m_] := m (x - P[[1]]) + P[[2]];
  d = D[f1[x], x];
  m = d /. x -> x1;
  p1 = {x1, f1[x1]};
  l1 = recta[p1, m];
  Plot[{f1[x], l1}, {x, -10, 10}, PlotRange -> 100]
],
{x1, -5, 5, Appearance -> "Labeled"}]
```



Manipulando con objetos gráficos (puntos, líneas):

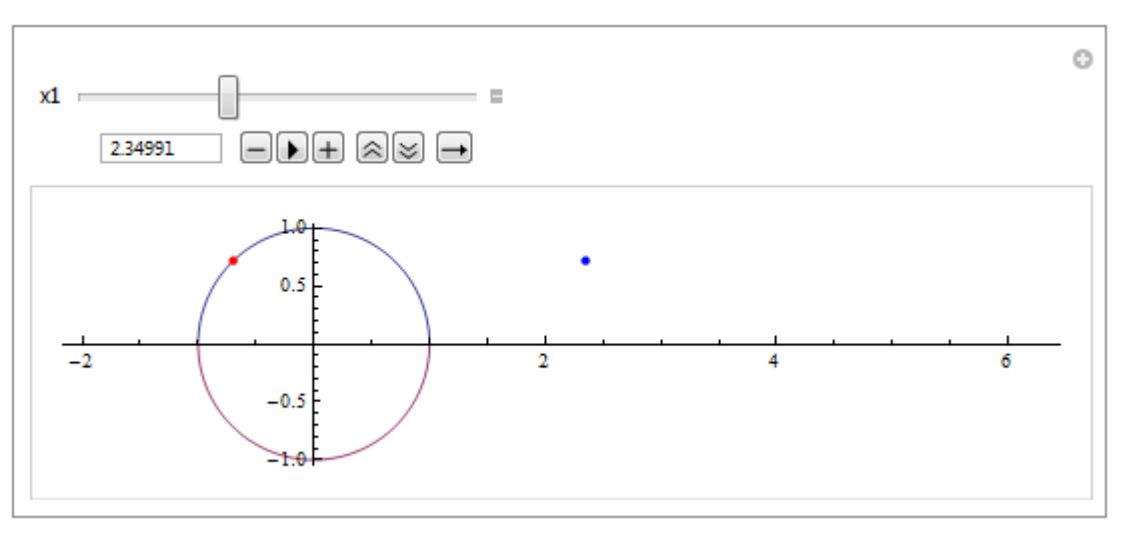
```
Manipulate[
Module[{p1, p2},
  f[x_] :=  $\sqrt{1 - x^2}$ ;
  p1 = {x1, f[x1]};
  p2 = {x1, 0};
  Plot[f[x], {x, -1, 1},
    Epilog -> {Red, PointSize[Medium], Point[p1], Point[p2],
      Blue, Line[{{0, 0}, p1]}, Line[{{p2, p1}]}}]
],
{x1, -1, 1}]
```



```

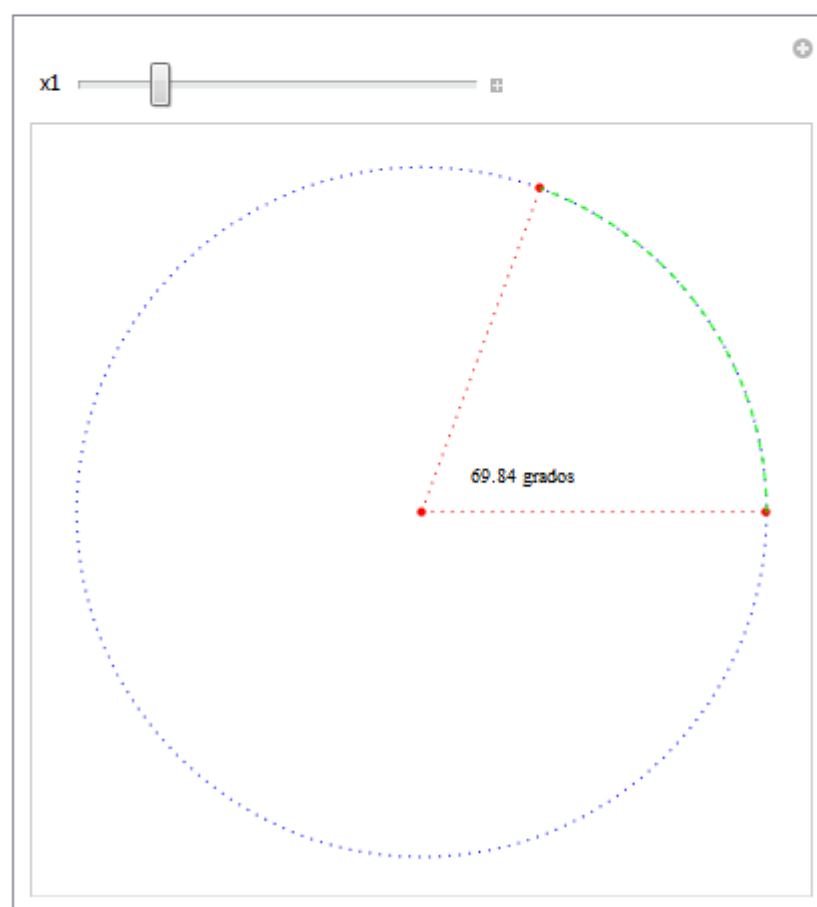
Manipulate[
Module[{f1, f2, p1, p2},
f1[x_] :=  $\sqrt{1 - x^2}$  ;
f2[x_] :=  $-\sqrt{1 - x^2}$  ;
p1 = {Cos[x1], Sin[x1]};
p2 = {x1, Sin[x1]};
Plot[{f1[x], f2[x]}, {x, -2, 2 Pi}, AspectRatio -> Automatic,
Epilog -> {Red, PointSize[Medium], Point[{p1]},
Blue, Point[p2]},
ImageSize -> 500]
],
{x1, 0, 2 Pi}]

```



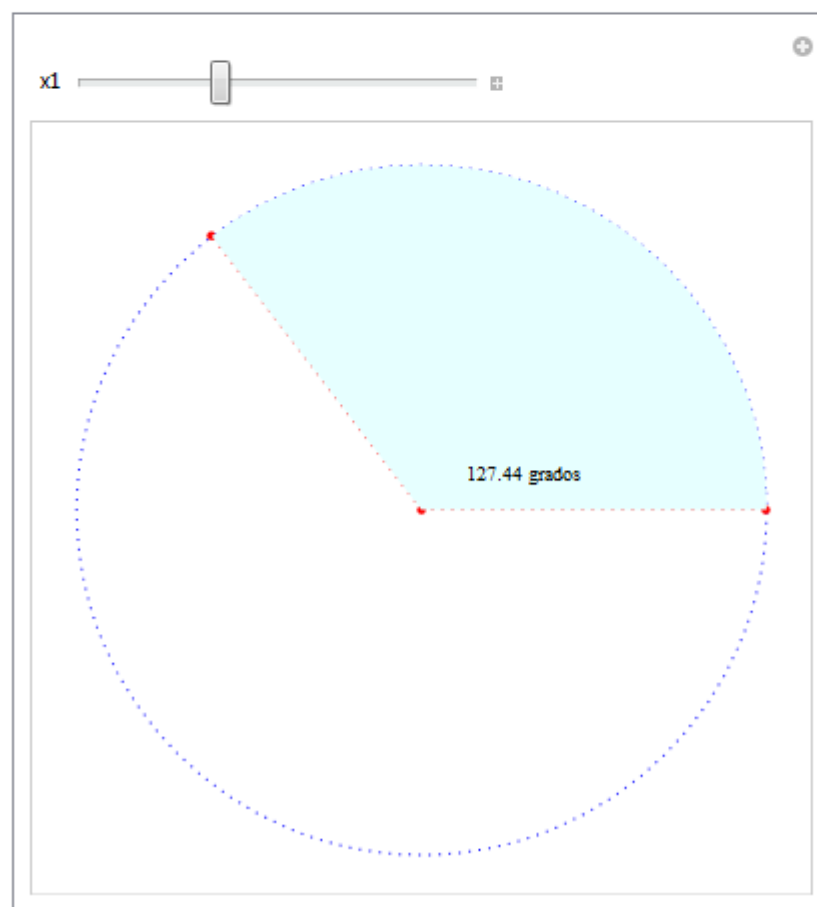
Usando arcos de círculo:

```
Manipulate[
Module[{p1},
  p1 = {Cos[x1], Sin[x1]};
  Graphics[{Blue, Dotted, Circle[{0, 0}, 1],
    Red, PointSize[Medium],
    Point[{0, 0}],
    Point[{1, 0}],
    Point[p1],
    Line[{0, 0}, {1, 0}],
    Line[{0, 0}, p1],
    Green, Dashed, Circle[{0, 0}, 1, {0, x1}],
    Black, Text[ToString[x1 180 / Pi] <> " grados", {0.3, 0.1}]}]
,
{x1, 0, 2 Pi}]
```



Usando sectores de círculo:

```
Manipulate[
Module[{p1},
  p1 = {Cos[x1], Sin[x1]};
  Graphics[{Blue, Dotted, Circle[{0, 0}, 1],
    Red, PointSize[Medium],
    Point[{0, 0}],
    Point[{1, 0}],
    Point[p1],
    Line[{{0, 0}, {1, 0}}],
    Line[{{0, 0}, p1}],
    LightCyan, Disk[{0, 0}, 1, {0, x1}],
    Black, Text[ToString[x1 180 / Pi] <> " grados", {0.3, 0.1}]}]
,
{x1, 0, 2 Pi}]
```



Actualización dinámica de expresiones

Ejemplo

Sin el uso de Dynamic

n = 5

5

Expand [(x + y) ^n]

$$x^5 + 5 x^4 y + 10 x^3 y^2 + 10 x^2 y^3 + 5 x y^4 + y^5$$

Cambio en la variable n, no hay cambio automático/ dinámico en la expresión de expansión:

n = 10

10

Expand [(x + y) ^n]

$$x^5 + 5 x^4 y + 10 x^3 y^2 + 10 x^2 y^3 + 5 x y^4 + y^5$$

Con el uso de Dynamic:

n = 10

10

Dynamic[Expand [(x + y) ^n]]

$$x^{10} + 10 x^9 y + 45 x^8 y^2 + 120 x^7 y^3 + 210 x^6 y^4 + 252 x^5 y^5 + 210 x^4 y^6 + 120 x^3 y^7 + 45 x^2 y^8 + 10 x y^9 + y^{10}$$

Ejemplo

Uso de deslizadores con Dynamic

```
Slider[Dynamic[n], {1, 25, 1}]
```



```
Dynamic[Expand[(x + y)^n]]
```

```
x + y
```

Ejemplo

Actualización dinámica en el manejo de una cuadrática

```
Slider[Dynamic[a], {-10, 20, 1}]
```



```
Slider[Dynamic[b], {-50, 50, 1}]
```



```
Slider[Dynamic[c], {-50, 50, 1}]
```

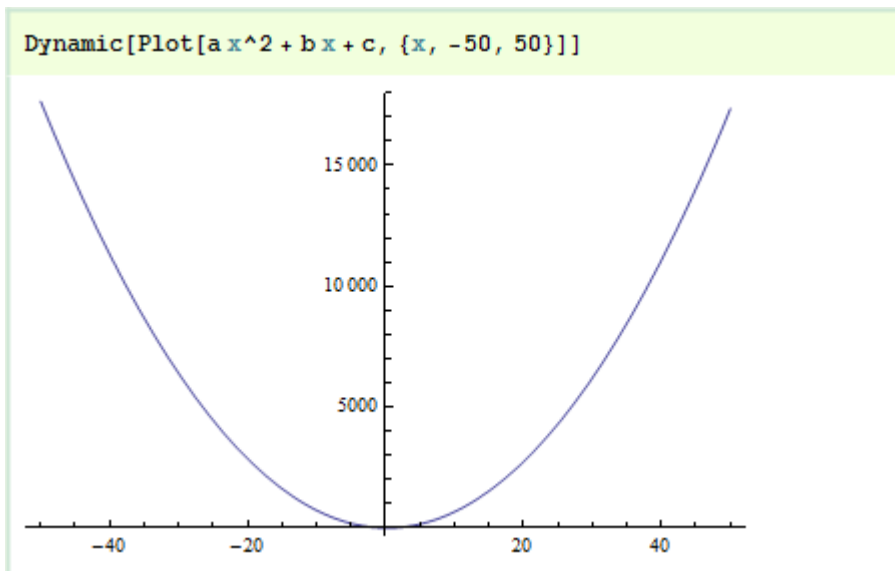


```
{Dynamic[a], Dynamic[b], Dynamic[c]}
```

```
{7, -3, -9}
```

```
Dynamic[a x^2 + b x + c]
```

```
-9 - 3 x + 7 x^2
```

Ejemplo

La cuadrática

Definición de la función cuadrática

```
f = (x + 8) (x - 5) // Expand
```

```
-40 + 3 x + x2
```

Evaluación dinámica de los coeficientes de la cuadrática

```
Dynamic[a = Coefficient[f, x, 2]]
```

```
1
```

```
Dynamic[b = Coefficient[f, x, 1]]
```

```
3
```

```
Dynamic[c = Coefficient[f, x, 0]]
```

```
-40
```

Evaluación dinámica del discriminante

```
Dynamic[discriminante = b2 - 4 a c]
```

```
169
```

Evaluación dinámica de la solución

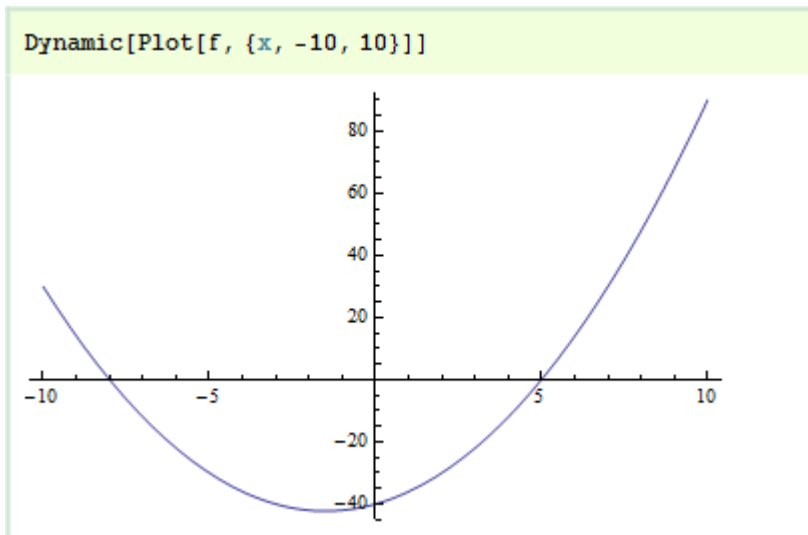
```
Dynamic[solucion = Solve[f == 0, x]]
```

```
{{x → -8}, {x → 5}}
```

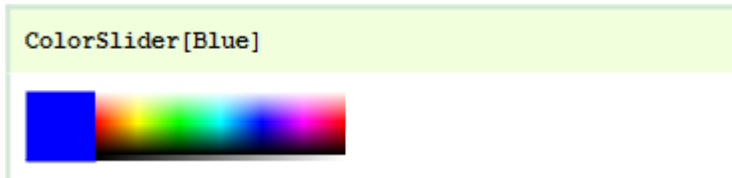
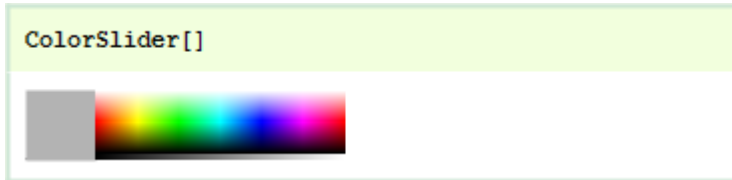
```
Dynamic[Which[discriminante > 0, {"Reales", solucion},  
discriminante == 0, {"Real. Una solución", solucion},  
discriminante < 0, "Imaginarias"]]
```

```
{Reales, {{x → -8}, {x → 5}}}
```

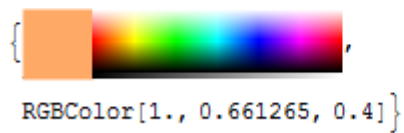
Gráfica dinámica



Dinámica en objetos gráficos



```
{ColorSlider[Dynamic[x]], Dynamic[x]}
```



```
Graphics[  
  {Dynamic[x],  
   Disk[]  
  }, ImageSize -> 200]
```

