

## REPRESENTACIÓN GRÁFICA DE NÚMEROS COMPLEJOS

`ReIm [z]` descompone el número complejo  $z$  como una lista de dos elementos {Real, Imaginario}

```
In[*]:= ReIm[z]
[partes real e imaginaria]
```

```
Out[*]= {Re[z], Im[z]}
```

Ejemplos:

```
In[*]:= ReIm[-3 + 4 I]
[partes real e... [núr
```

```
Out[*]= {-3, 4}
```

```
In[*]:= ReIm[I]
[parte... [número i
```

```
Out[*]= {0, 1}
```

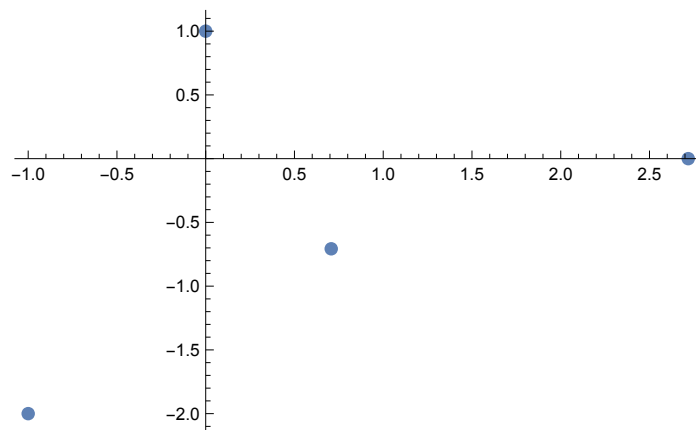
```
In[*]:= ReIm[2]
[partes real e imaginaria]
```

```
Out[*]= {2, 0}
```

`ListPlot` nos permite hacer una representación gráfica de los números complejos:

```
In[*]:= ListPlot[ReIm[{I, E, -1 - 2 I, Sqrt[-I]}], PlotStyle -> PointSize[Large]]
[representa... [partes... [... [número e [raiz c... [núme... [estilo de repre... [tamaño de ... [grande
```

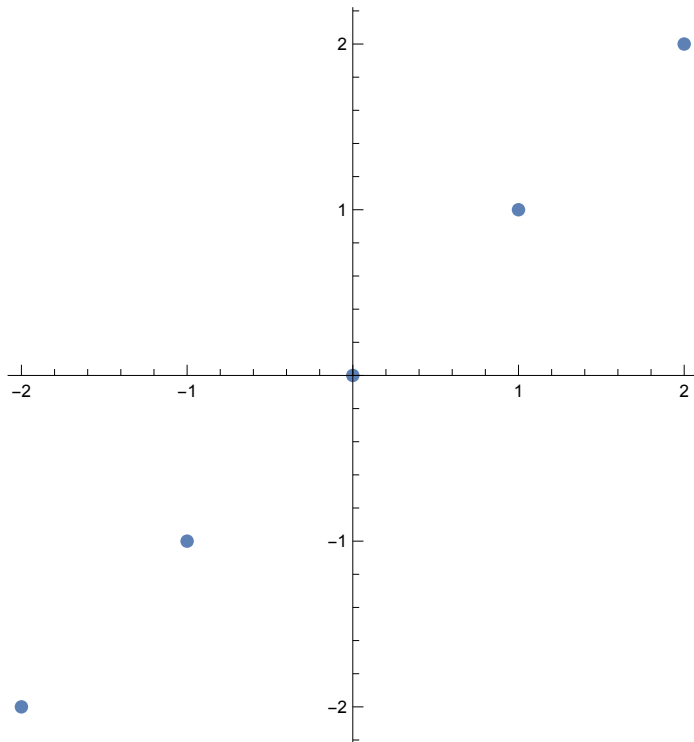
```
Out[*]=
```



Otra forma de representar gráficamente los números complejos:

```
In[*]:= ComplexListPlot[{-2 - 2 I, -1 - I, 0, 1 + I, 2 + 2 I}, PlotStyle -> PointSize[Large]]
[representación compleja de lista [número i [número i [número i [n... [estilo de repre... [tamaño de ... [grande
```

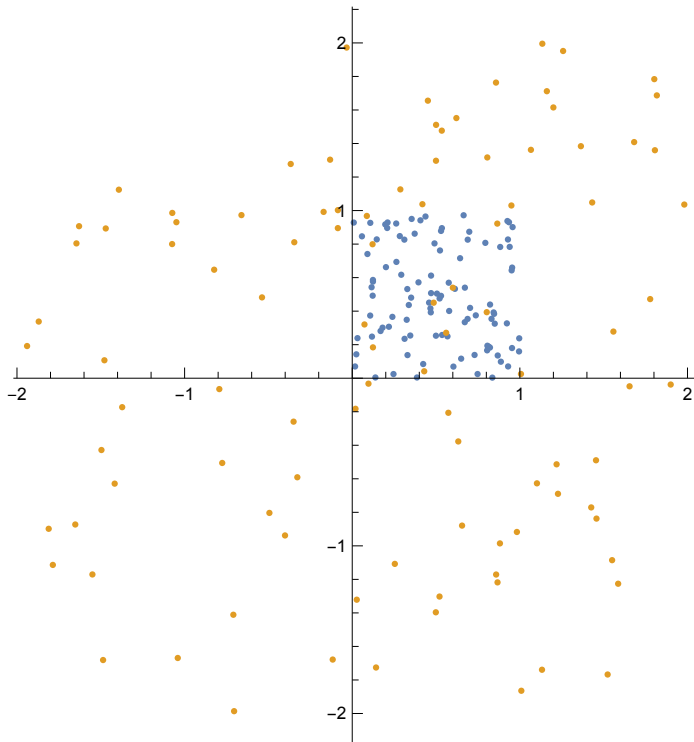
Out[\*]=



En sus múltiples variedades:

```
In[*]:= ComplexListPlot[
[representación compleja de lista
{RandomComplex[{0, 1 + I}, 100], RandomComplex[{-2 - 2 I, 2 + 2 I}, 100]}]
[número complejo aleatorio [número i [número complejo aleatorio [número i [número i
```

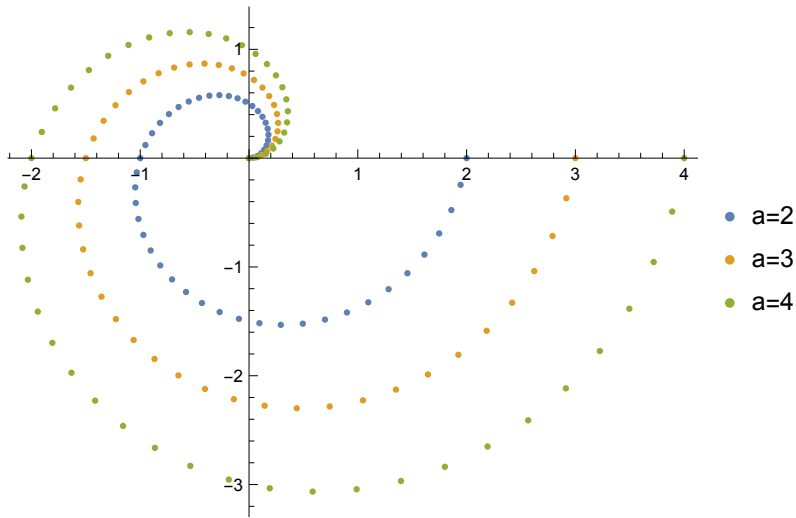
Out[\*]=



```
In[*]:= data = Table[a t Exp[2 π I t], {a, 2, 4}, {t, 0, 1, 0.02}];
```

```
ComplexListPlot[data, PlotLegends → {"a=2", "a=3", "a=4"}]
```

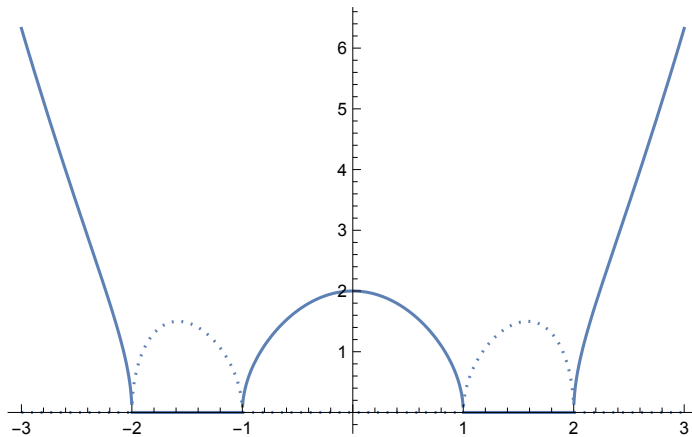
Out[\*]=



Podemos representar gráficamente la parte real e imaginaria de un complejo como un valor de una función de variable real:

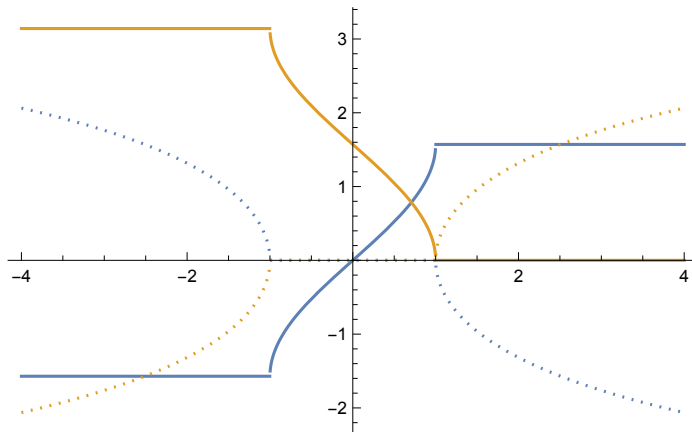
```
In[*]:= ReImPlot[Sqrt[(x^2 - 1) (x^2 - 4)], {x, -3, 3}]
```

Out[\*]=



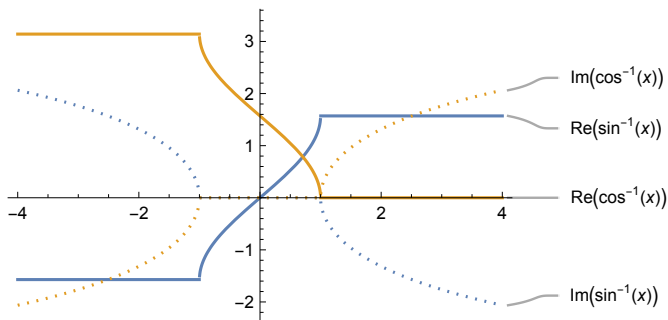
```
In[*]:= ReImPlot[{ArcSin[x], ArcCos[x]}, {x, -4, 4}]
```

Out[\*]=



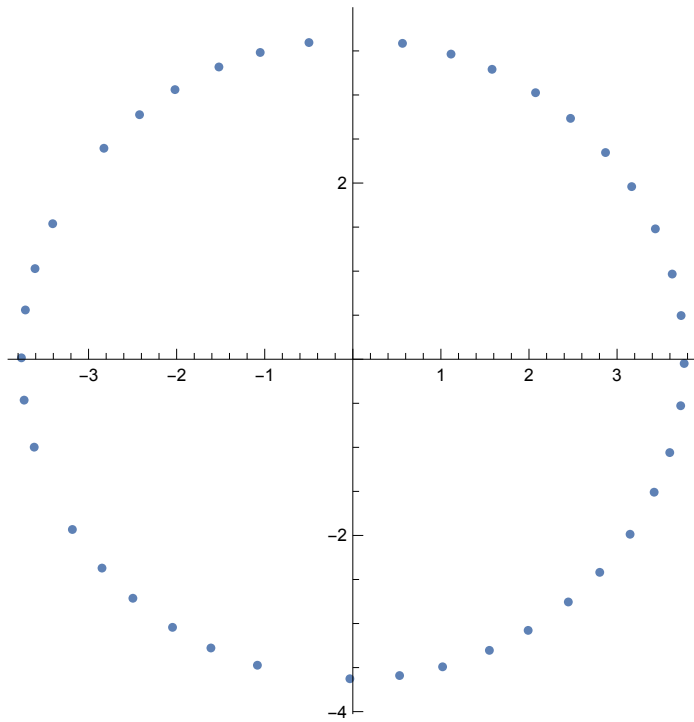
```
In[*]:= ReImPlot[{ArcSin[x], ArcCos[x]}, {x, -4, 4}, PlotLabels -> "Expressions"]
```

Out[\*]=

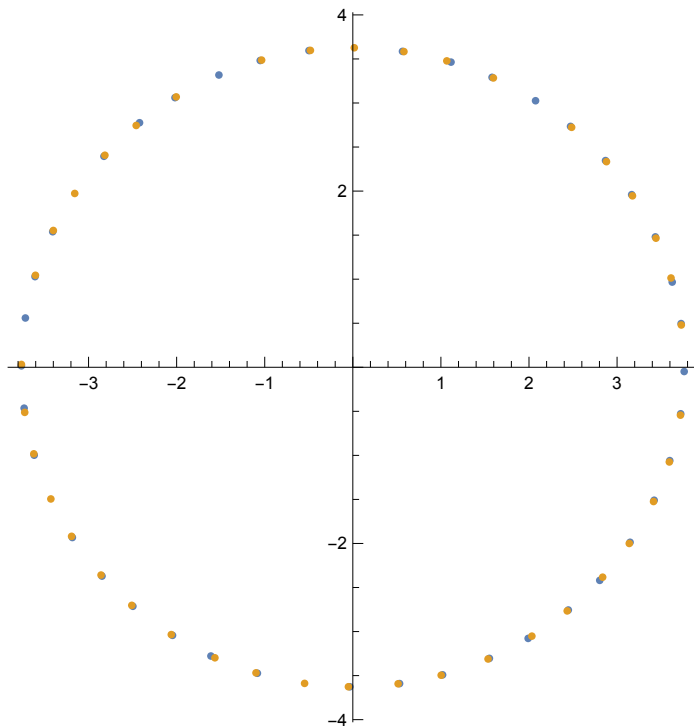


Dada una lista de valores complejos  $\{z_1, z_2, \dots\}$  se puede representar como una lista de pares  $\{\text{Re}[z_1], \text{Im}[z_1]\}$ :

```
In[*]:= ComplexListPlot[Sin[Range[40] + 2 I]]
[representación complej... [seno [rango] [número]
Out[*]=
```



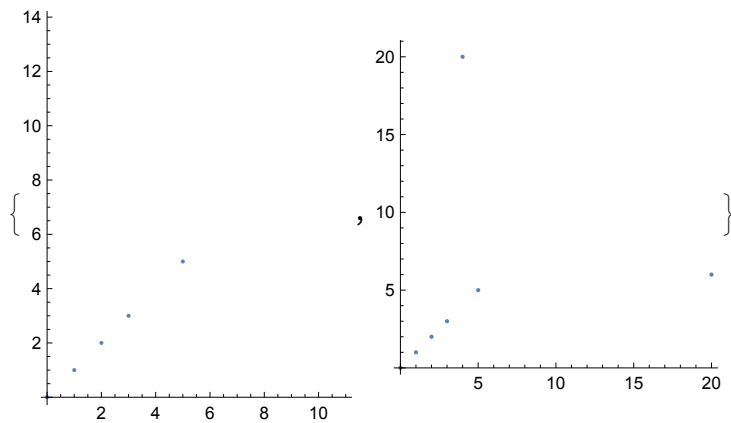
```
In[*]:= ComplexListPlot[{Sin[Range[40] + 2 I], Cos[Range[40] + 2 I]}]
[representación complej... [seno [rango] [número] [coseno [rango] [número]
Out[*]=
```



```

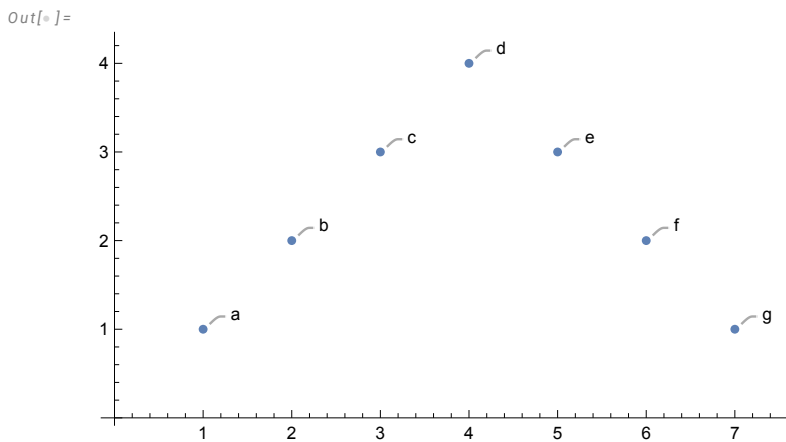
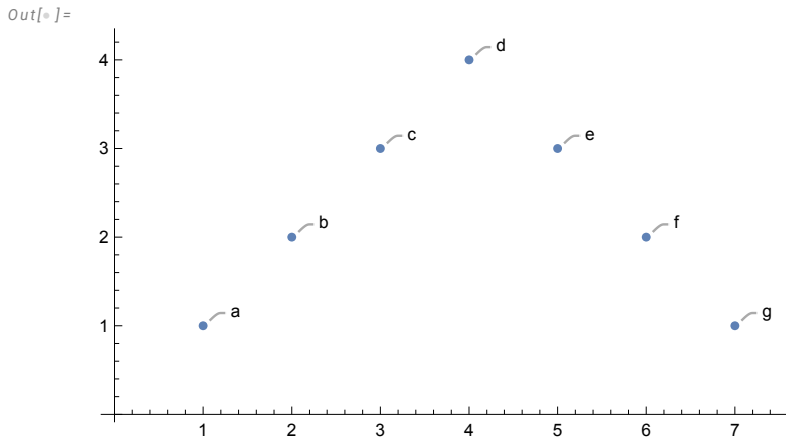
In[*]:= {ComplexListPlot[{0, 1 + I, 2 + 2 I, 3 + 3 I, 4 + 20 I, 5 + 5 I, 20 + 6 I}],
  [representación compleja de lista [número i [número i [número i [número i [número i [número
  ComplexListPlot[
  [representación compleja de lista
  {0, 1 + I, 2 + 2 I, 3 + 3 I, 4 + 20 I, 5 + 5 I, 20 + 6 I}, PlotRange -> All]}
  [número i [número i [número i [número i [número i [n... [rango de repr... [todo
  
```

Out[\*]=



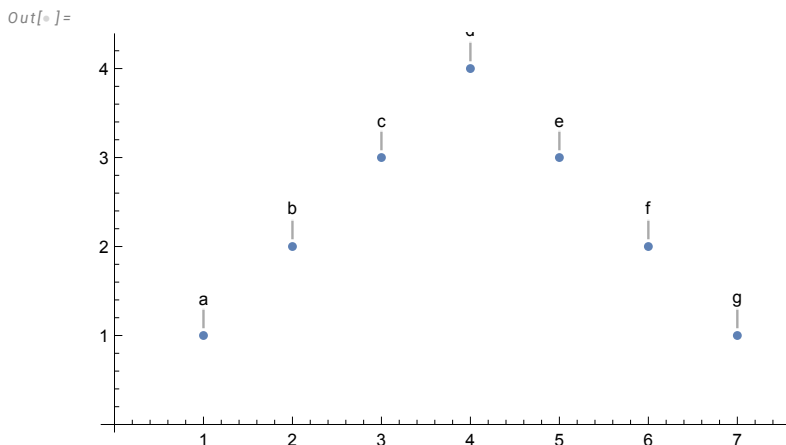
Podemos especificar etiquetas:

```
In[*]:= ComplexListPlot[{1 + I → "a", 2 + 2 I → "b",
  [representación compleja de l·[número i] [número i]
    3 + 3 I → "c", 4 + 4 I → "d", 5 + 3 I → "e", 6 + 2 I → "f", 7 + I → "g"}]
  [número i] [número i] [número i] [número i] [número i]
ComplexListPlot[{1 + i, 2 + 2 i, 3 + 3 i, 4 + 4 i, 5 + 3 i, 6 + 2 i, 7 + i} →
  [representación compleja de lista]
  {"a", "b", "c", "d", "e", "f", "g"}]
```



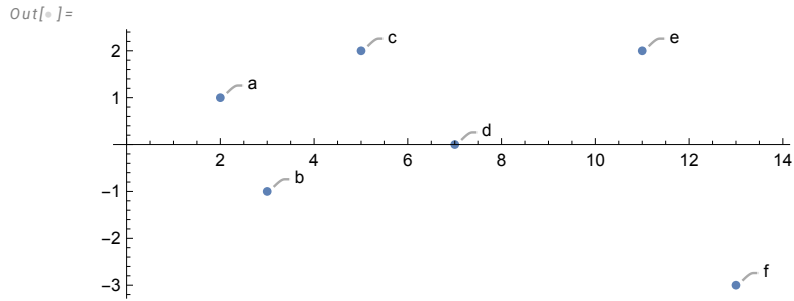
Se puede especificar la localización de las etiquetas:

```
In[*]:= ComplexListPlot[{1 + i, 2 + 2 i, 3 + 3 i, 4 + 4 i, 5 + 3 i, 6 + 2 i, 7 + i} →
  [representación compleja de lista]
  {"a", "b", "c", "d", "e", "f", "g"}, LabelingFunction → Above]
  [función de etiquetado] [encima]
```

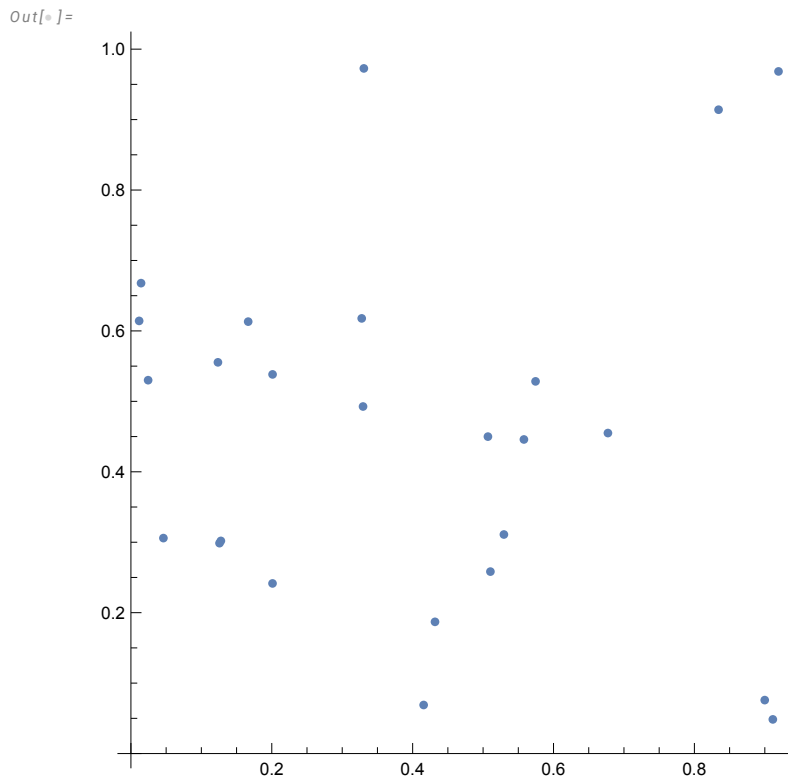


Los valores numéricos se pueden asociar con coordenadas (x, y):

```
In[*]:= ComplexListPlot[<|"a" → 2 + I, "b" → 3 - I,  
[representación compleja de lista [número i [número i  
"c" → 5 + 2 I, "d" → 7, "e" → 11 + 2 I, "f" → 13 - 3 I]>]  
[número i [número i [número i
```



```
In[*]:= ComplexListPlot[SparseArray[RandomComplex[{0, 1 + i}, 25]]]  
[representación compl... [arreglo disperso [número complejo aleatorio
```

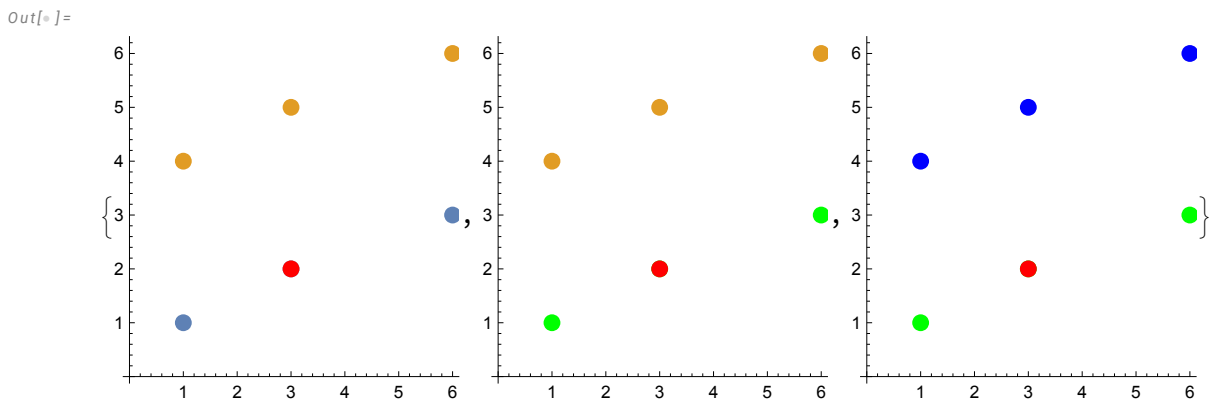




```

In[*]:= {ComplexListPlot[{{1 + I, Style[3 + 2 I, Red], 6 + 3 I}, {1 + 4 I, 3 + 5 I, 6 + 6 I}},
  PlotStyle -> PointSize[0.05]],
  ComplexListPlot[{Style[{1 + I, Style[3 + 2 I, Red], 6 + 3 I}, Green],
    {1 + 4 I, 3 + 5 I, 6 + 6 I}}, PlotStyle -> PointSize[0.05]],
  ComplexListPlot[
    Style[{Style[{1 + I, Style[3 + 2 I, Red], 6 + 3 I}, Green], {1 + 4 I, 3 + 5 I, 6 + 6 I}},
      Blue], PlotStyle -> PointSize[0.05]]]

```

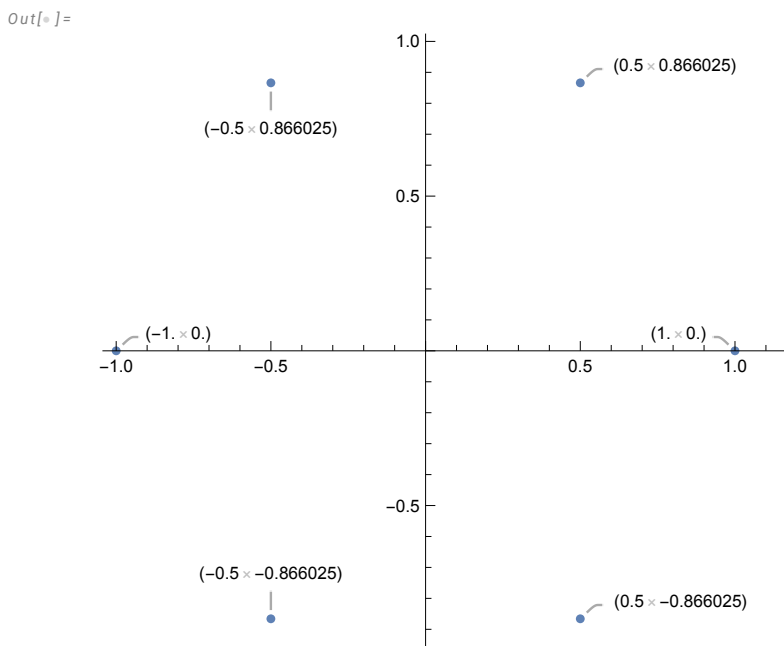


Se pueden calcular resultados de una ecuación y proceder a representar los resultados, etiquetando los valores de los complejos:

```

In[*]:= ComplexListPlot[z /. Solve[z^6 == 1, z],
  LabelingFunction -> (DisplayForm[RowBox[{"(", #1[[1]], #1[[2]], ")"}]] &)]

```

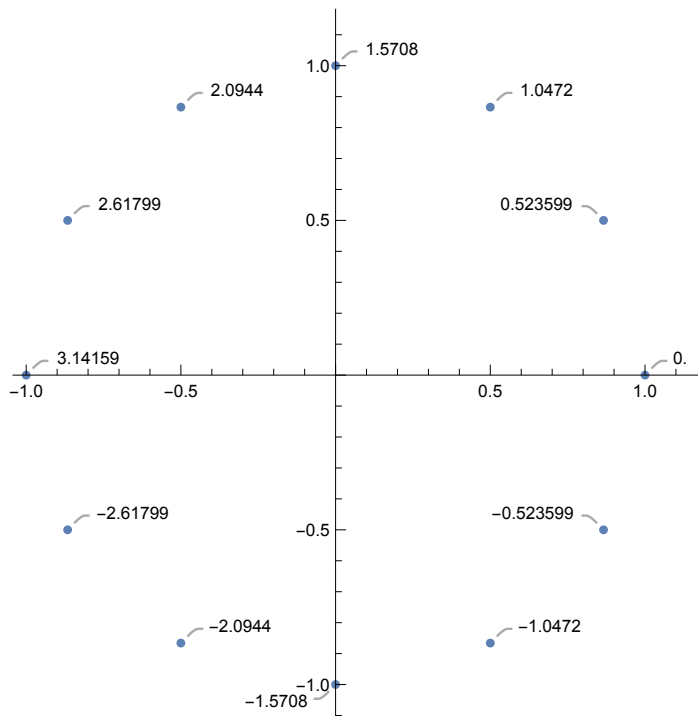


```

In[*]:= ComplexListPlot[z /. Solve[z12 == 1, z], LabelingFunction -> (Arg[#1[[1]] + I #1[[2]] &)]
[representación compleja de l... [resuelve] [función de etiquetado] [argumento c... [número i]

```

Out[\*]=

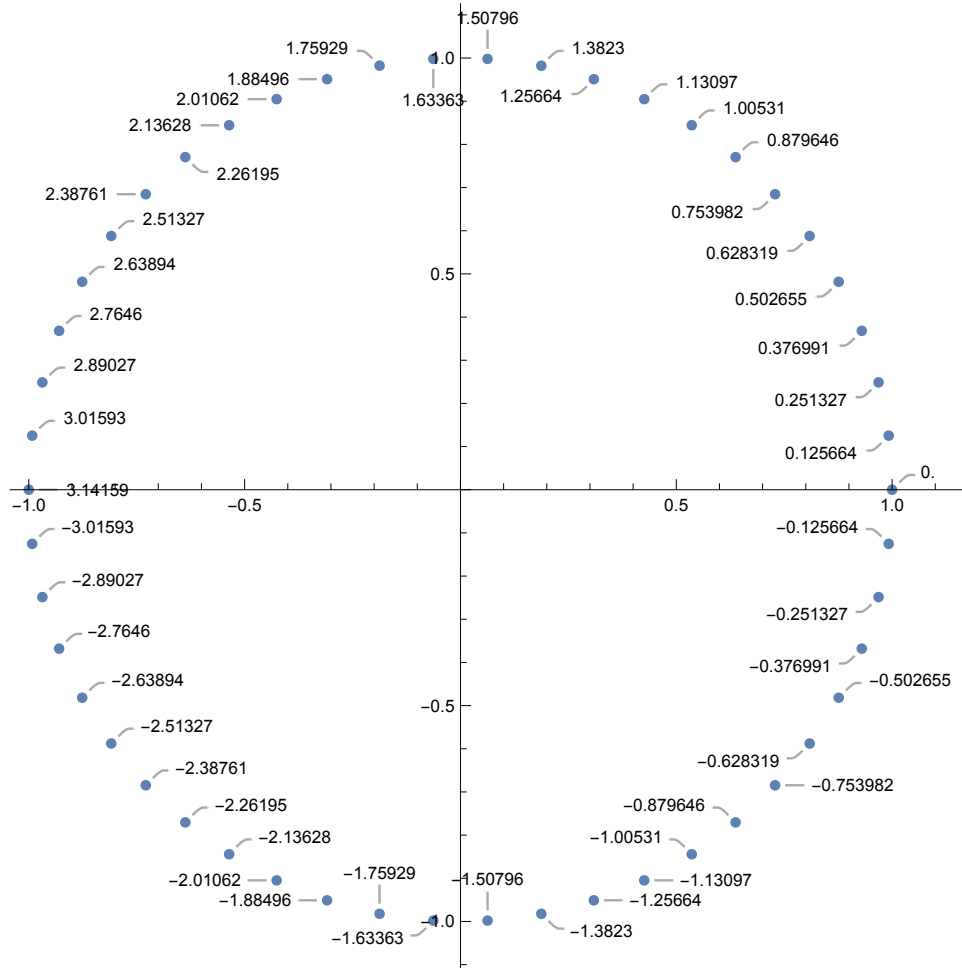


```

In[*]:= Quiet@ComplexListPlot[z /. Solve[z50 == 1, z],
[silenci... [representación compleja de l... [resuelve
LabelingFunction -> (Arg[#1[[1]] + I #[[2]] &), ImageSize -> 500]
[función de etiquetado [argumento c... [número i [tamaño de imagen

```

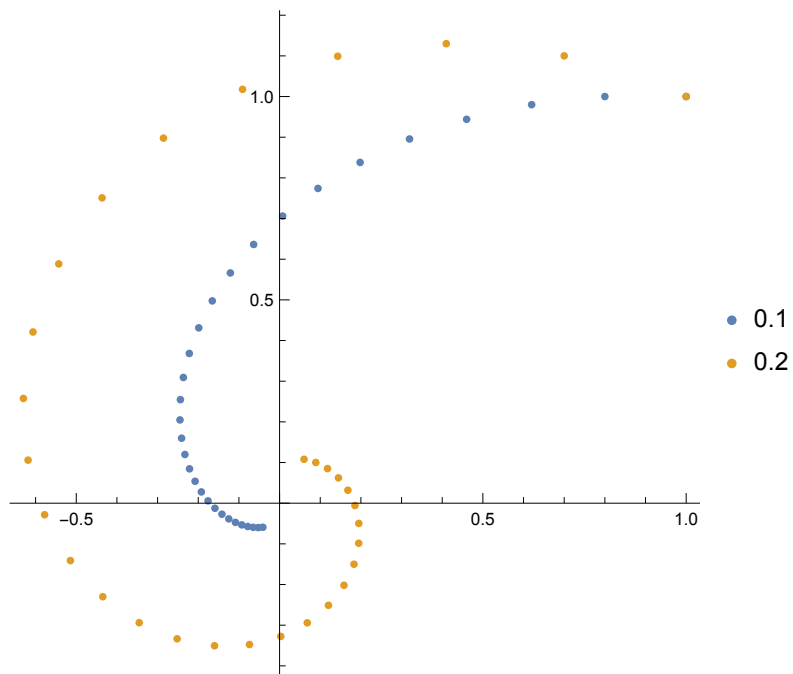
Out[\*]=



Incluimos leyendas:

```
In[*]:= ComplexListPlot[{  
  [representación compleja de lista  
    NestList[(0.9 + 0.1 I) # &, 1 + I, 30], NestList[(0.9 + 0.2 I) # &, 1 + I, 30]  
    [lista de resultados anidados [número i [número i [lista de resultados anidados [número i [número i  
  ], PlotLegends -> {0.1, 0.2}]  
    [leyendas de representación
```

Out[\*]=



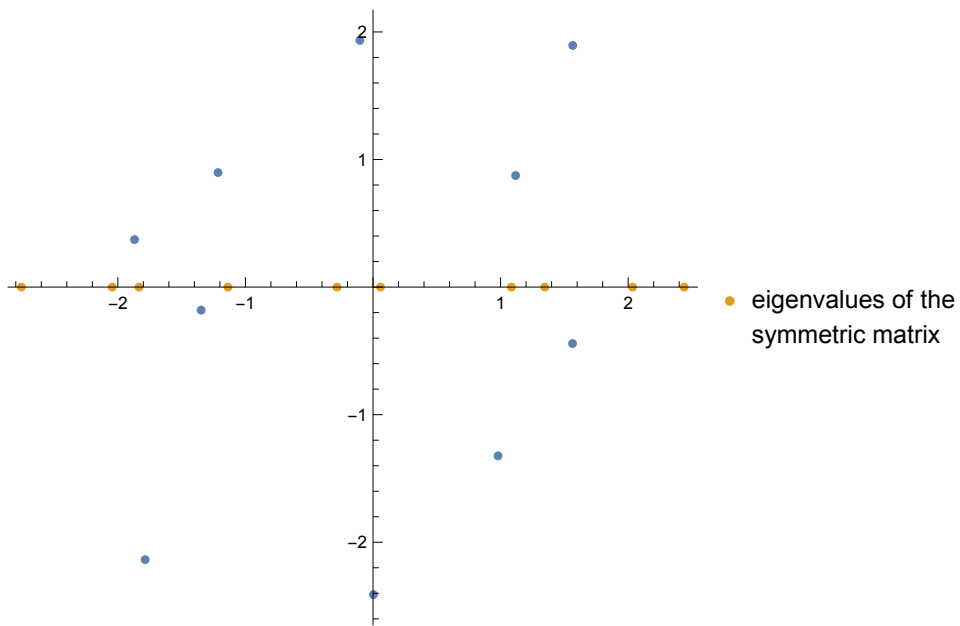
```

In[*]:= matrix = RandomComplex[{-1 - I, 1 + I}, {10, 10}];
          [número complejo aleatorio [número... [número i
evals1 = Eigenvalues[matrix];
          [autovalores
evals2 = Eigenvalues[ $\frac{\text{matrix} + \text{ConjugateTranspose}[\text{matrix}]}{2}$ ]];
          [autovalores

ComplexListPlot[
  [representación compleja de lista
  {evals1, Legended[evals2, "eigenvalues of the\nsymmetric matrix"]}
  [con leyenda

```

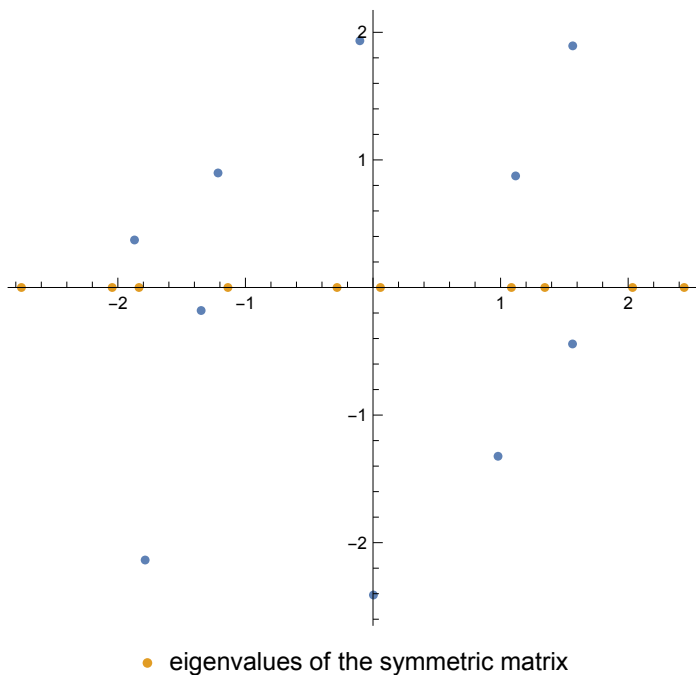
Out[\*]=



Utilizando Placed podemos cambiar la leyenda de posición:

```
In[*]:= ComplexListPlot[{evals1,  
[representación compleja de lista  
Legended[evals2, Placed["eigenvalues of the symmetric matrix", Below]]]}]  
[con leyenda [colocado [debajo]]]
```

Out[\*]=



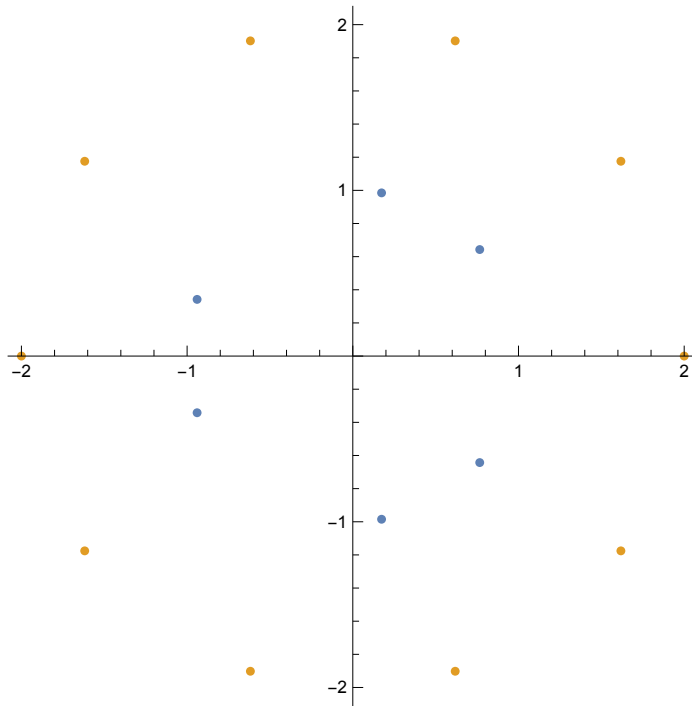
Podemos utilizar diferentes estilos:

```

In[*]:= f =  $\frac{z^6 + z^3 + 1}{z^{10} - 1024}$ ;
zeros = z /. Solve[f == 0, z];
      [resuelve]
poles = z /. Solve[Denominator[f] == 0, z];
      [resuelve [denominador]
ComplexListPlot[{zeros, poles}]
[representación compleja de lista

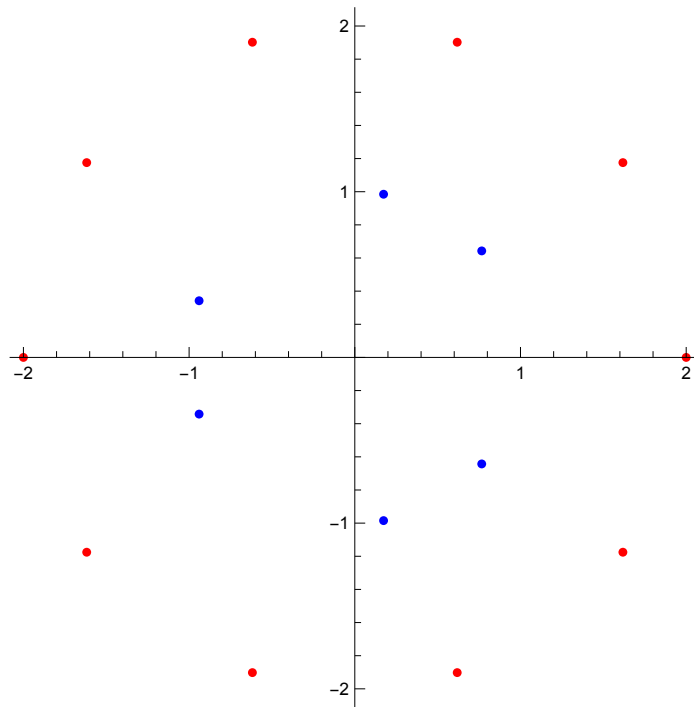
```

Out[\*]=



```
In[*]:= ComplexListPlot[{zeros, poles}, PlotStyle -> {Blue, Red}]  
[representación compleja de lista] [estilo de represe... [azul] [rojo]
```

Out[\*]=



Como aplicaciones:

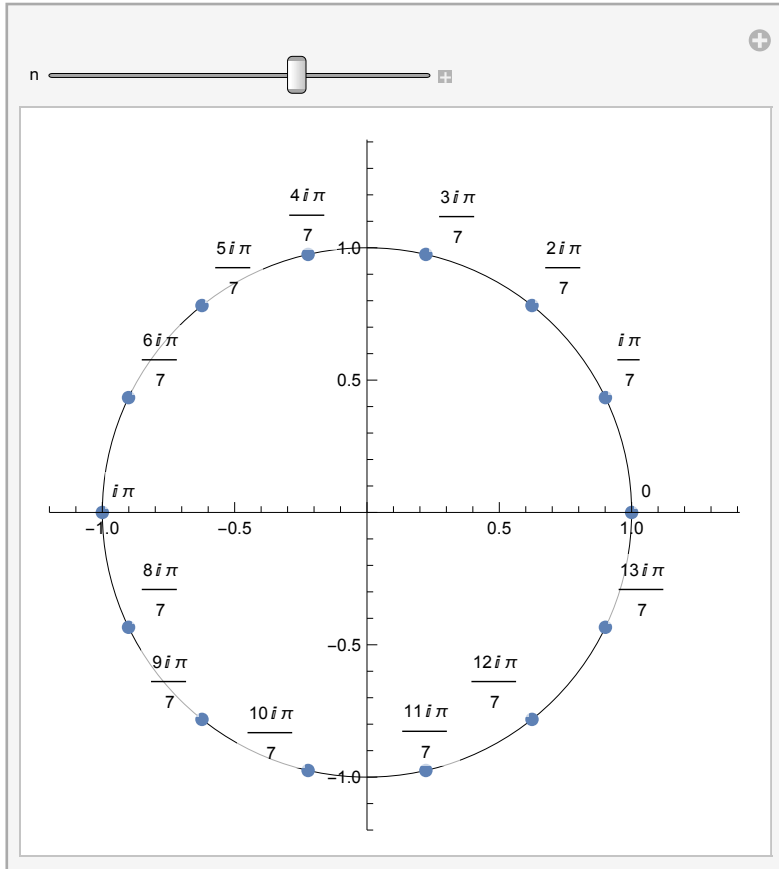


```

In[*]:= Manipulate[ComplexListPlot[Labeled[Exp[#], #] & /@ Table[2 Pi I k / n, {k, 0, n - 1}],
  [manipula [representación compl... [etiquetado [exponencial [tabla [... [número i
    Prolog -> {Circle[]}, ImageSize -> Medium, PlotMarkers -> {Automatic, Medium},
    [prólogo [círculo [tamaño de im... [tamaño... [marcadores de rep... [automático [tamaño medio
    PlotRange -> {{-1.2, 1.2}, {-1.2, 1.2}}, {{n, 6}, 2, 20, 1}]
    [rango de representación
  ]

```

Out[\*]=



```
In[*]:= ComplexListPlot[z /. Solve[z2 + z + 1 == 0, z],  
[representación compleja de l· [resuelve  
LabelingFunction -> (DisplayForm[RowBox[{"(", #1[[1]], #1[[2]], ")"}]] &)]  
[función de etiquetado [muestra [caja de fila
```

Out[\*]=

