

**COMPLEJIDAD ALGORÍTMICA (CC184)****Ciclo 2023-02**

Procedimiento: Instalar y Usar Graphviz. Integrar con VS Code

Elaborado por: Robert Zubieta

Fuente: Propia

Instalar y Usar Graphviz. Integrar con VS Code

DESCARGA DE GRAPHVIZ

<https://graphviz.org/download/>

graphviz.org/download/

Aplicaciones Entrenamiento Estudio Personal Cole Util Casa Exito Diseño Cyan v18 UPC

Graphviz Download Documentation Gallery Forum GitLab

Search this site...

Graphviz

About

Download

Source Code

Documentation

DOT Language

Command Line

Layout Engines

Output Formats

Attributes

Attribute Types

Graph Attributes

Node Attributes

Node Shapes

Cluster Attributes

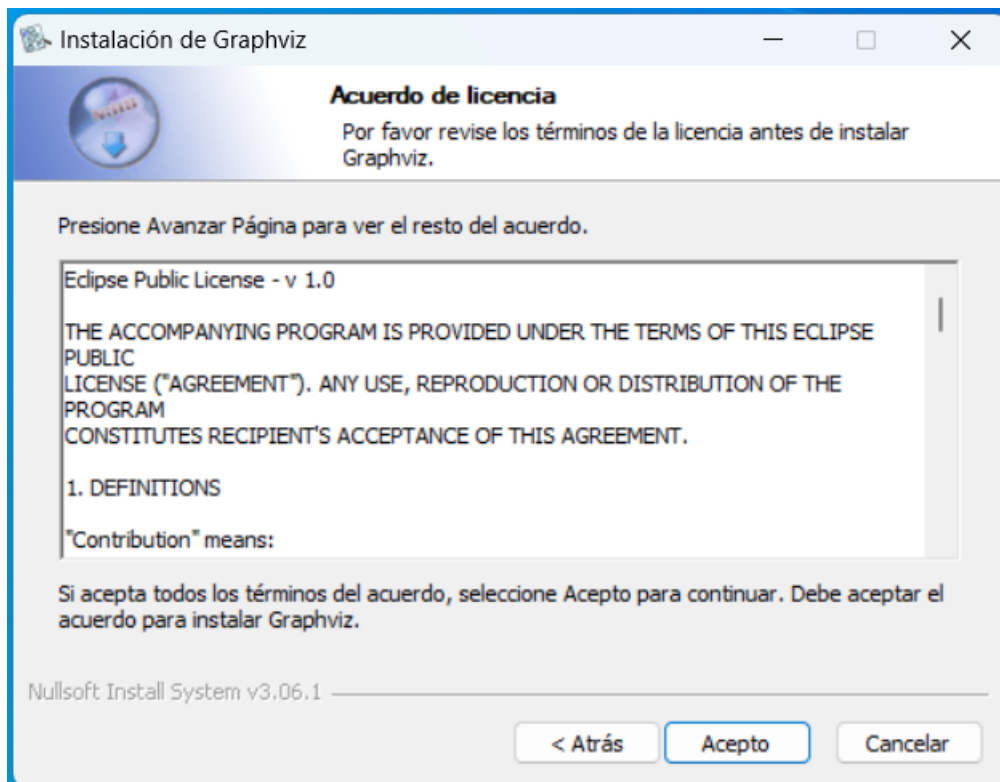
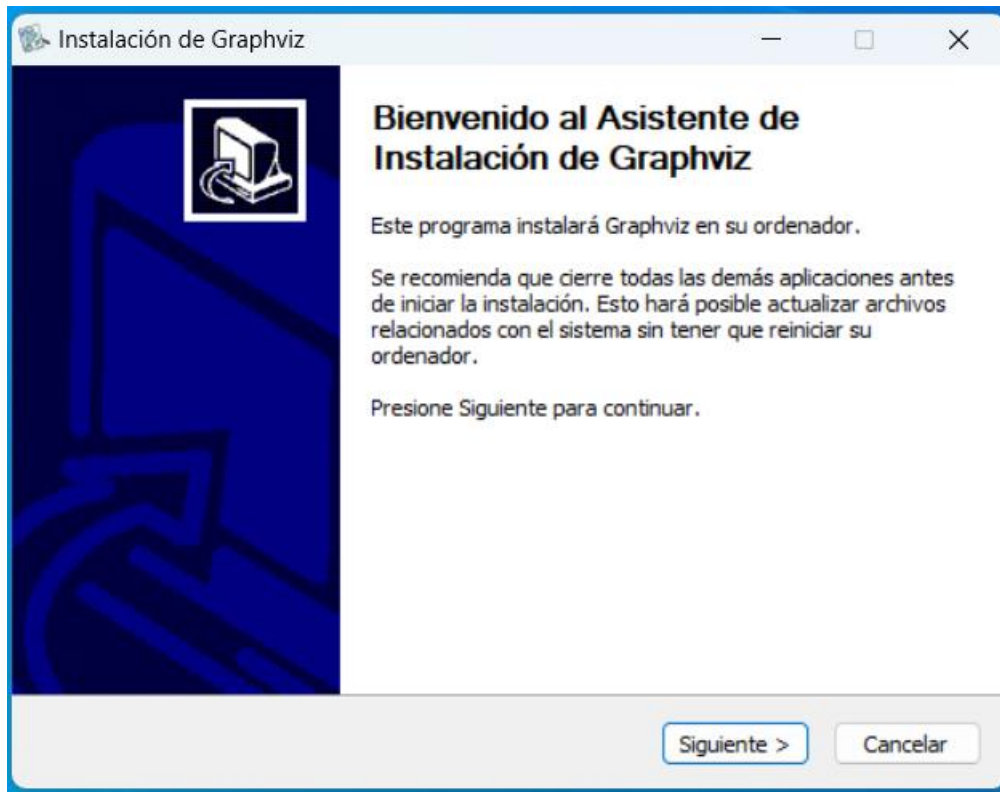
Edge Attributes

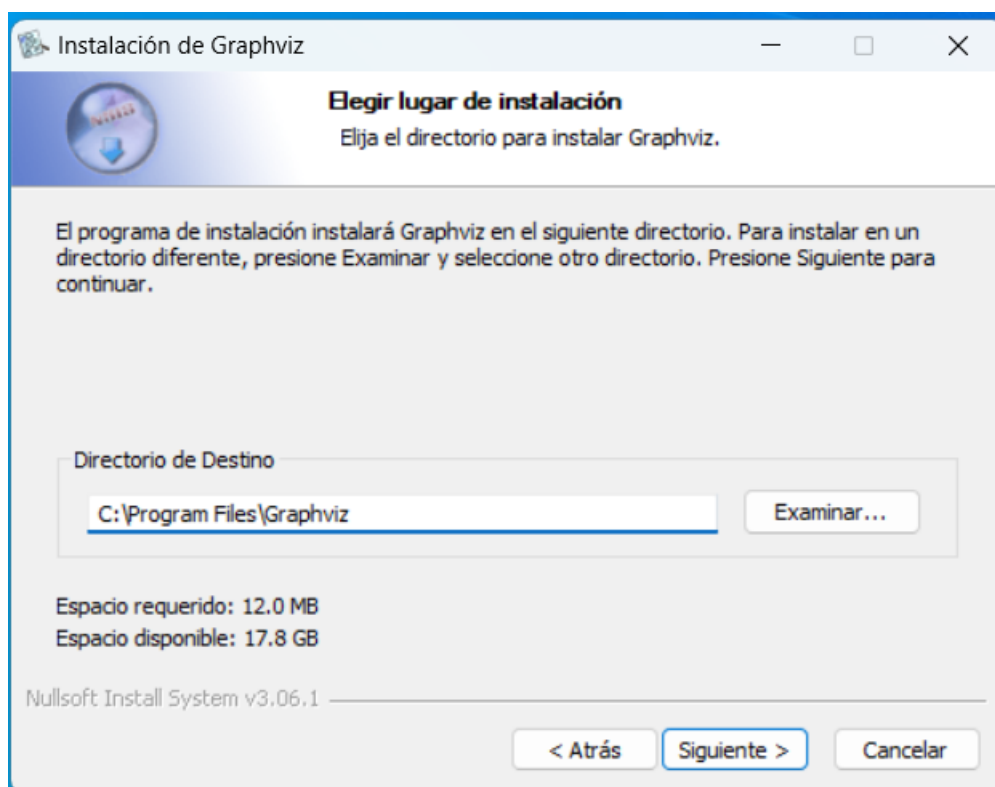
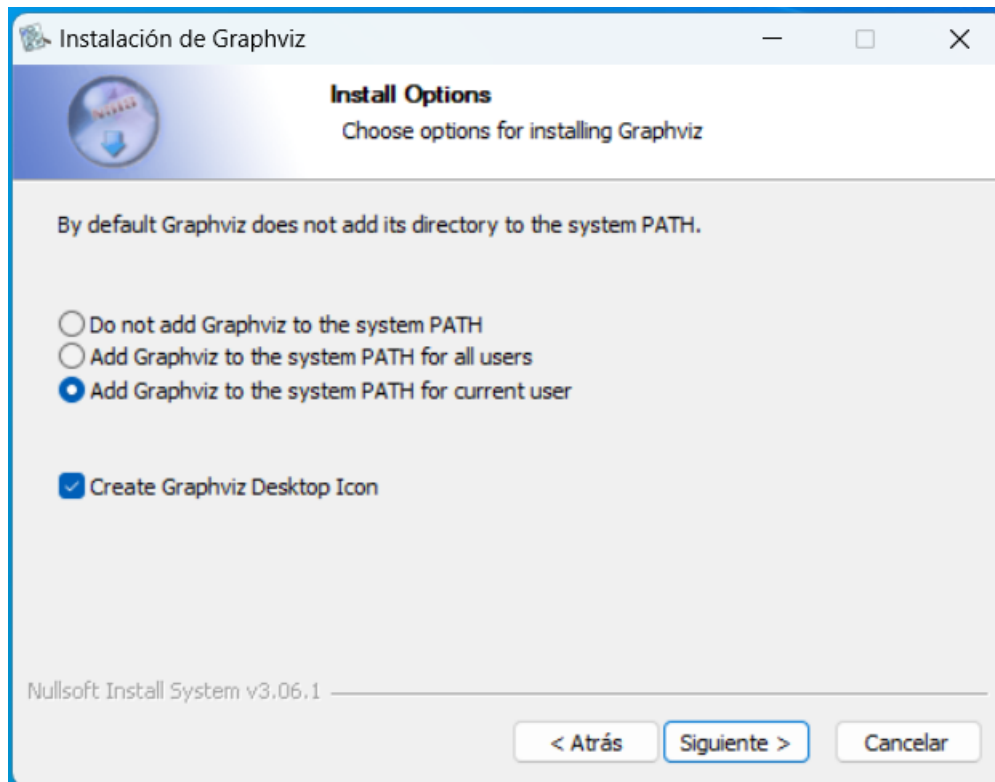
Windows

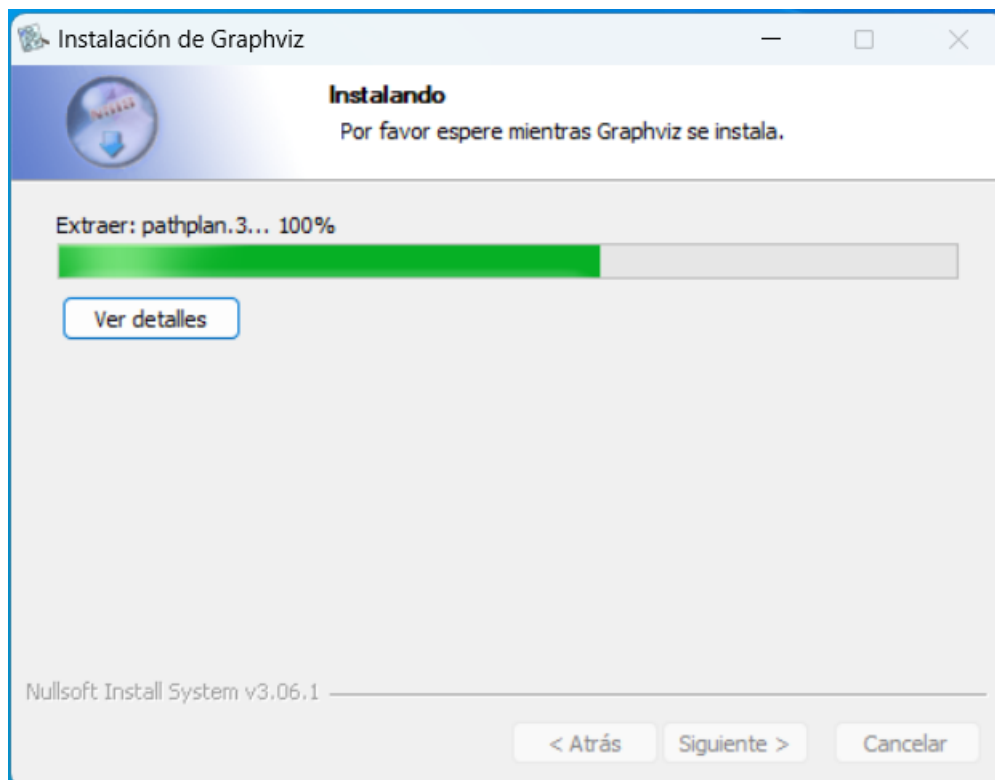
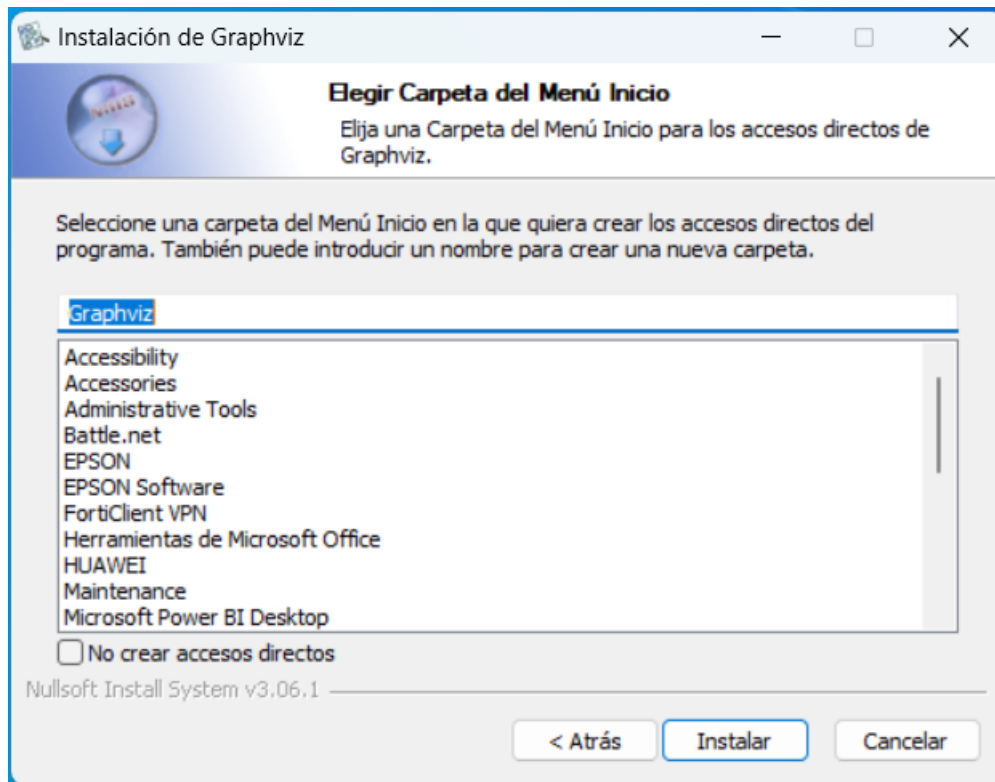
- Stable Windows install packages, built with Microsoft Visual Studio 16 2019:
 - graphviz-7.1.0
 - graphviz-7.1.0 (32-bit) ZIP archive [sha256] (contains all tools and libraries)
 - graphviz-7.1.0 (64-bit) EXE installer [sha256]
 - graphviz-7.1.0 (32-bit) EXE installer [sha256]
 - graphviz-7.0.6
 - graphviz-7.0.6 (32-bit) ZIP archive [sha256] (contains all tools and libraries)
 - graphviz-7.0.6 (32-bit) EXE installer [sha256]
 - graphviz-7.0.6 (64-bit) EXE installer [sha256]
 - graphviz-7.0.5
 - graphviz-7.0.5 (32-bit) ZIP archive [sha256] (contains all tools and libraries)
 - graphviz-7.0.5 (32-bit) EXE installer [sha256]
 - graphviz-7.0.5 (64-bit) EXE installer [sha256]
 - graphviz-7.0.4
 - graphviz-7.0.4 (32-bit) ZIP archive [sha256] (contains all tools and libraries)
 - graphviz-7.0.4 (32-bit) EXE installer [sha256]
 - graphviz-7.0.4 (64-bit) EXE installer [sha256]
 - graphviz-7.0.3

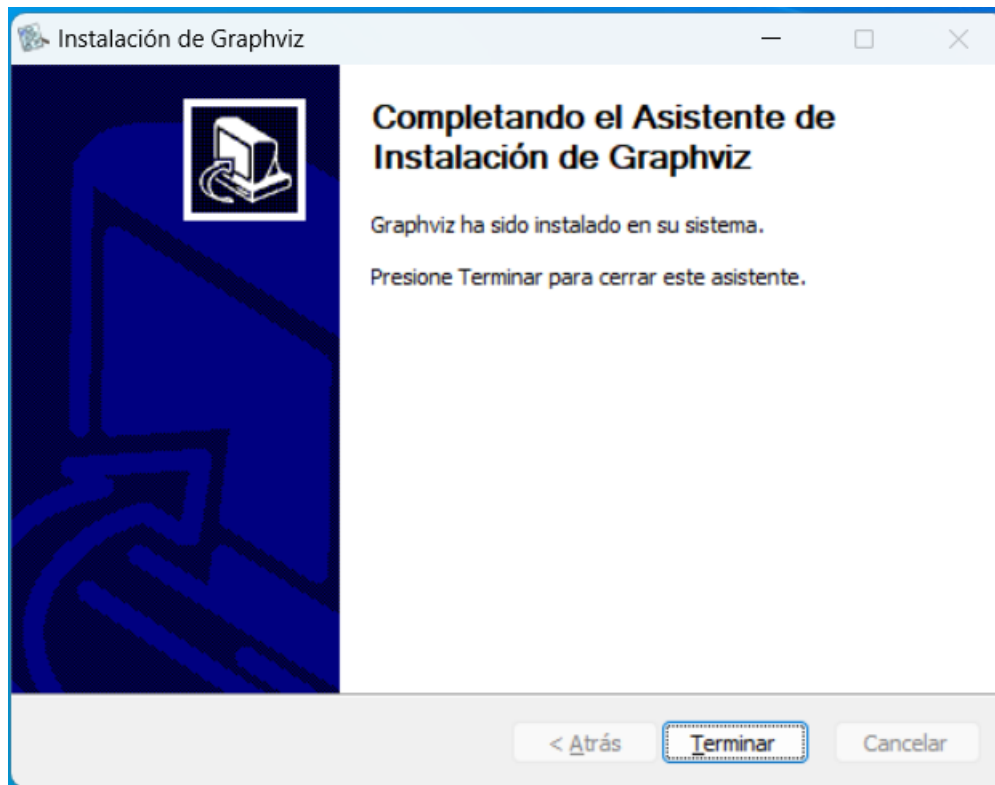
INSTALACIÓN DE GRAPHVIZ

Install



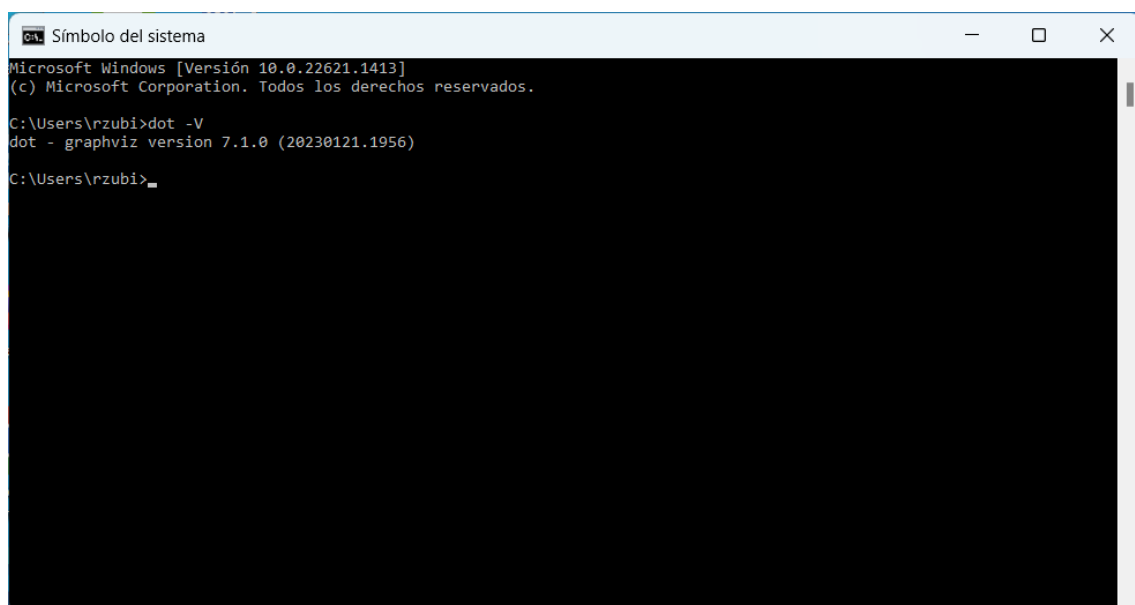






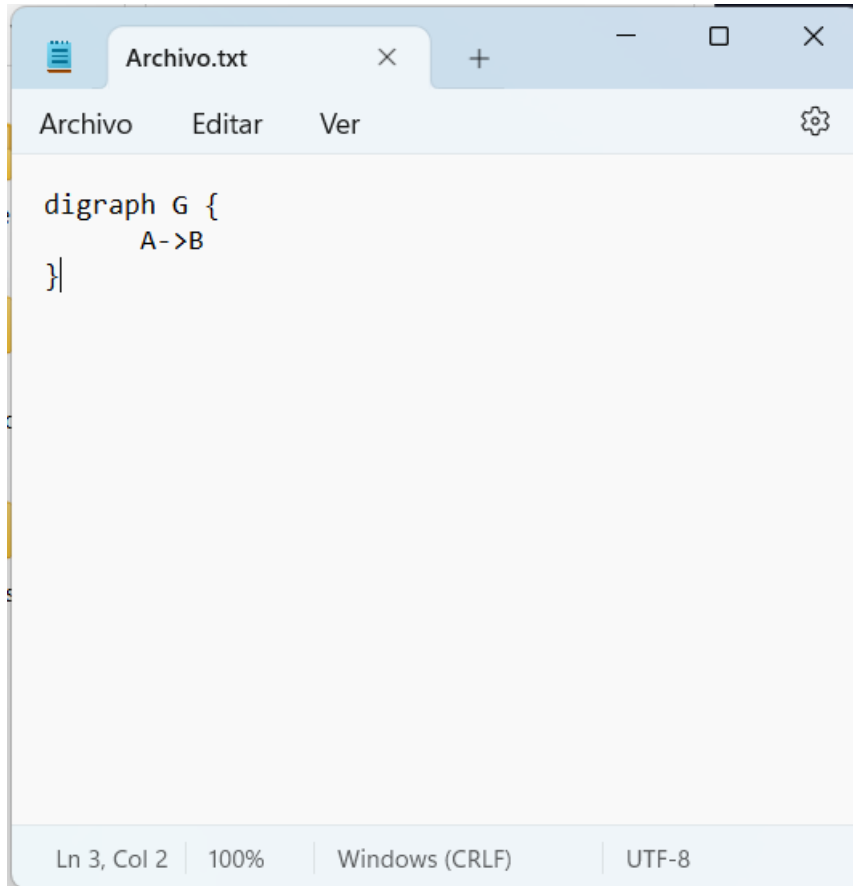
Verificación:

En CMD ejecutar: **dot -V**



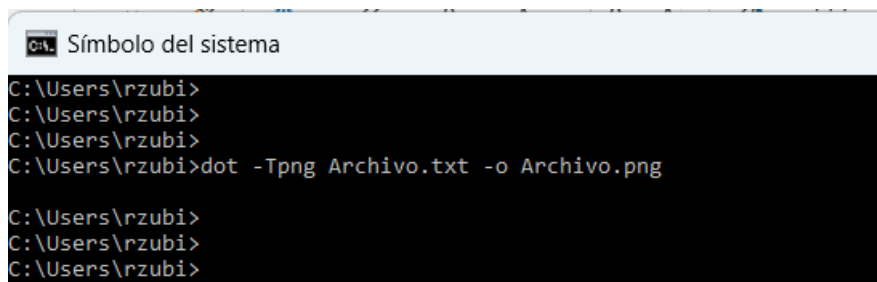
EJECUCIÓN DE GRAPHVIZ

1. Crear archivo con las directrices del Grafico: **Archivo.txt**

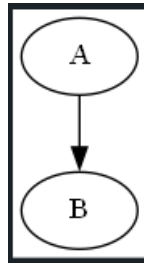


2. EN CMD o Terminal, ejecutar el contenido del archivo:
dot -Tpng Archivo.txt -o Archivo.png

El grafico se obtendrá en un Archivo.png



3. Abrimos el archivo obtenido: **Archivo.png**



Si se desea realizar la ejecución en un directorio particular:

- nos cambiamos al directorio correspondiente, o
- colocamos la ruta absoluta en el comando:

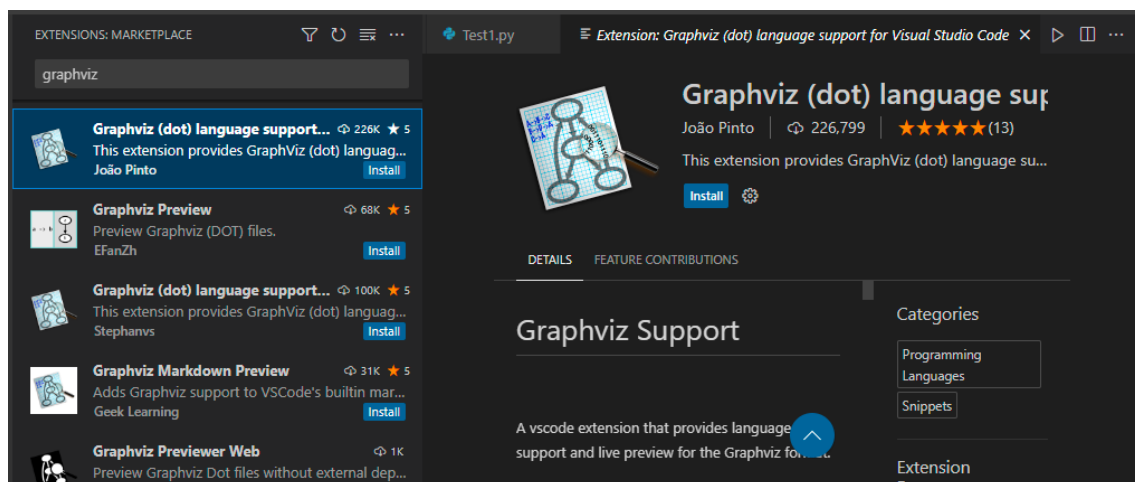
```
dot -Tpng D:\Temporal\Python\Test\Archivo.txt -o  
D:\Temporal\Python\Test\Archivo.png
```

GRAPHVIZ y VS CODE

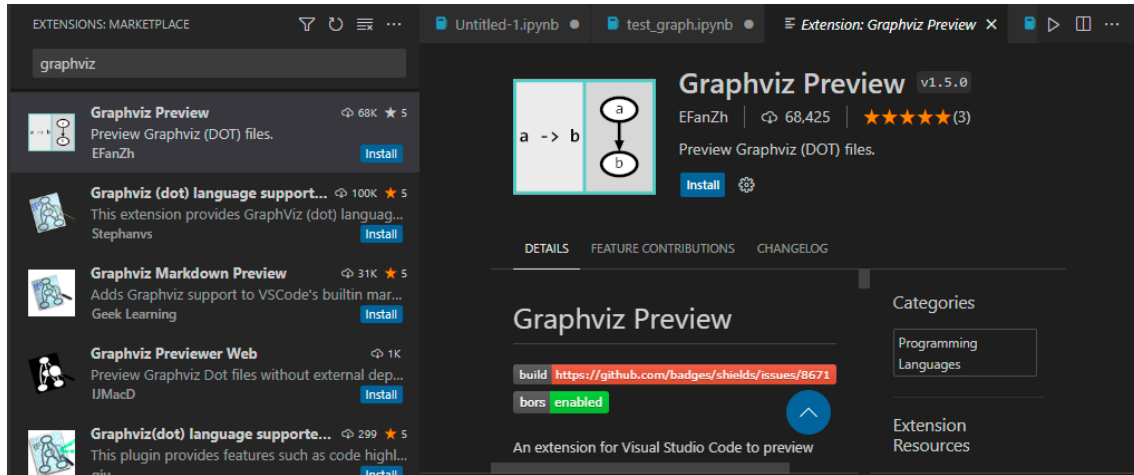
Usar graphviz en un Código Python con VS Code

Instalar Extensiones:

Buscar: “graphviz”



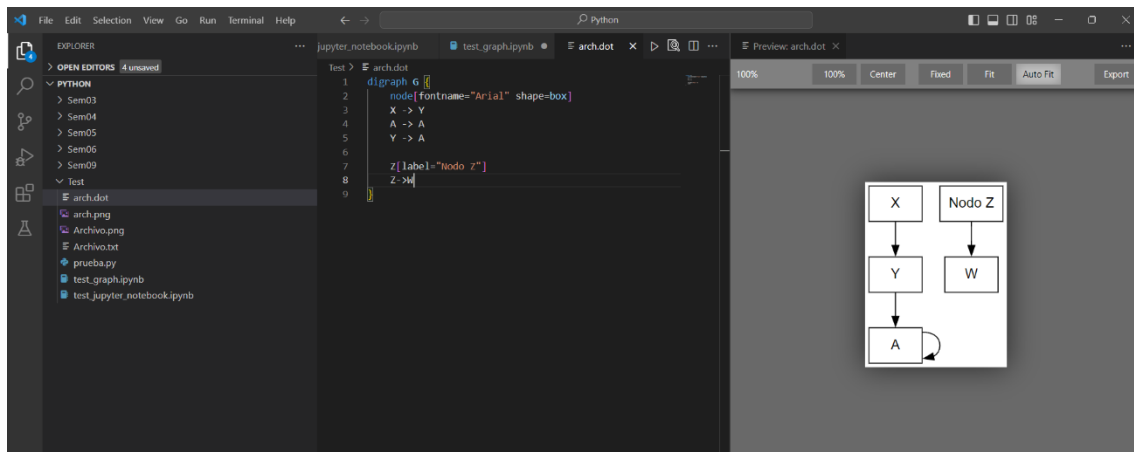
Para preview del gráfico: **Graphviz Preview**



Uso de archivo .dot y Preview en Visual Studio Code

Crear Archivo .dot

Ctrl + Shift + p [para ver el preview del Grafico]



Usar graphviz en código Python de VS Code

IMPORTANTE:

Instalar graphviz por terminal (Adicional a la instalación de la extension)

Si no se realiza este último paso, es probable que la librería graphviz (**import graphviz**) no se reconozca en el código).

En Terminal de VS Code, ejecutar:

pip install graphviz

```
PS D:\Temporal\CC184> pip install graphviz
Collecting graphviz
  Downloading graphviz-0.20.1-py3-none-any.whl (47 kB)
    47.0/47.0 kB 2.3 MB/s eta 0:00:00
Installing collected packages: graphviz
Successfully installed graphviz-0.20.1

[notice] A new release of pip available: 22.3.1 -> 23.0.1
[notice] To update, run: python.exe -m pip install --upgrade pip
PS D:\Temporal\CC184>
PS D:\Temporal\CC184> 
```

Ejemplo 1:

```
import graphviz as gv

dot = gv.Digraph()

dot.node("A")
dot.node("B")

dot.edge("A", "B")

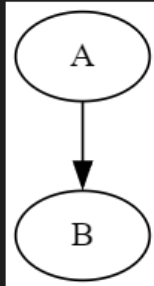
# Imprime como Estructura
print(dot)

# Imprime como Grafico
dot
```

Output:

```
... digraph {  
    A -> B  
}
```

</>



Ejemplo 2 :

```
import graphviz as gv  
  
def drawUF(a):  
    dot = gv.Digraph(comment='Nada')  
    n = len(a)  
    for i in range(n):  
        dot.node(str(i), str(i))  
    for i in range(n):  
        if i != a[i]:  
            dot.edge(str(i), str(a[i]))  
    dot.graph_attr['rankdir'] = 'BT'  
    return dot  
  
dot = drawUF([0, 0, 9, 9, 9, 6, 6, 7, 8, 9])  
  
# Imprime como Estructura  
print(dot)  
  
# Imprime como Grafico  
dot
```

Output:

```
... // Nada
digraph {
  graph [rankdir=BT]
  0 [label=0]
  1 [label=1]
  2 [label=2]
  3 [label=3]
  4 [label=4]
  5 [label=5]
  6 [label=6]
  7 [label=7]
  8 [label=8]
  9 [label=9]
  1 -> 0
  2 -> 9
  3 -> 9
  4 -> 9
  5 -> 6
}
```

</>

