



**LABORATORIO DE DOCUMENTACIÓN GEOMÉTRICA DEL PATRIMONIO**  
Grupo de Investigación en Patrimonio Construido -GPAC- (UPV-EHU)



UPV EHU

Aulario de las Nieves, edificio de Institutos Universitarias  
C/ Nieves Cano 33, 01006 Vitoria-Gasteiz (España-Spain).

Tfno: +34 945 013222 / 013264

e-mail: [jm.valle@ehu.es](mailto:jm.valle@ehu.es) web: <http://www.ldgp.es>

# ARCHIVO DEL LABORATORIO DE DOCUMENTACIÓN GEOMÉTRICA DEL PATRIMONIO

LABORATORY FOR THE GEOMETRIC DOCUMENTATION OF  
HERITAGE'S ARCHIVE


Sección de aplicaciones informáticas / **Software section**

# 7

<b>Información general / General information</b>		
TÍTULO:	Programa para generar ficheros de grafos en formato DOT (v. 1.0)	:TITLE
AUTORES:	Álvaro RODRÍGUEZ MIRANDA José Manuel VALLE MELÓN	:AUTORS
FECHA:	2012 / <b>2012</b>	:DATE
NUMERO:	LDGP_sof_007	:NUMBER
IDIOMA:	español / <b>Spanish</b>	:LANGUAGE

Resumen	
TITULO:	Programa para generar ficheros de grafos en formato DOT (v. 1.0)
RESUMEN:	A partir de un fichero de texto organizado en columnas se convierte a una estructura de grafo en formato DOT.
DESCRIPTORES NATURALES:	topología, grafos
DESCRIPTORES CONTROLADOS:	(Procedentes del Tesauro UNESCO [ <a href="http://databases.unesco.org/thessp/">http://databases.unesco.org/thessp/</a> ]) Topología
LENGUAJE:	Programado en Tcl (v.8). El fichero es un documento ASCII que se convierte en ejecutable si se dispone del intérprete Tcl que puede descargarse de forma gratuita por Internet (preferentemente en el paquete denominado "Active Tcl" [ <a href="http://www.activestate.com/activetcl/">http://www.activestate.com/activetcl/</a> ]).

Abstract	
TITLE:	Software to generate graphs in DOT format (v. 1.0)
ABSTRACT:	From a plain text file (column separated) and generates as output the graph in DOT format.
NATURAL KEYWORDS:	topology, graphs
CONTROLLED KEYWORDS:	(From the UNESCO's thesaurus [ <a href="http://databases.unesco.org/thesaurus/">http://databases.unesco.org/thesaurus/</a> ]) Topology
CODE:	Programmed in Tcl (v.8). It is an ASCII file that becomes self-executable when the Tcl interpreter is available on the system. This interpreter can be freely downloaded from Internet (better with the "Active Tcl" package [ <a href="http://www.activestate.com/activetcl/">http://www.activestate.com/activetcl/</a> ]).

Derechos / Rights		
DERECHOS:	Se permite su utilización en las condiciones que establece una licencia Creative Commons de tipo "Reconocimiento-Comparteigual 3.0". / This software has a Creative Commons licence "Attribution-Share Alike 3.0".	:RIGHTS
		

Renuncia de responsabilidad / Disclaimer		
DESCARGO:	Esta aplicación informática es una prueba de laboratorio, no ha sido completamente probado ni optimizado por lo que los resultados que proporciona pueden no ser correctos y el tiempo de ejecución de algunos procesos excesivamente largo. El uso de esta aplicación se hará bajo la completa responsabilidad del usuario. / This software is an internal test, it has not been thoroughly either checked or optimized, therefore the results may not be correct and the elapsing time too long. Its use will be done under the exclusive responsibility of the user.	:DISCLAIMER

Estructura / <b>Framework</b>		
PERMANENTE:	<small>ID</small> <a href="http://hdl.handle.net/10810/6169">http://hdl.handle.net/10810/6169</a>	:PERMANENT ID
ESTRUCTURA:	<ul style="list-style-type: none"> <li>• <b>ldgp_sof_007_grafos.pdf</b>: (este documento) descripción del programa / <i>(this document) description of the software.</i></li> <li>• <b>ldgp_sof_007_grafos.tcl</b>: código del programa principal (menu) / <i>code of the main application (menu).</i></li> <li>• <b>ldgp_sof_007_ejemplo.zip</b>: datos de ejemplo / <i>test data.</i></li> </ul>	:FRAMEWORK

Cita completa recomendada / <b>Recommended full citation</b>		
CITA:	Laboratorio de Documentación Geométrica del Patrimonio (Universidad del País Vasco-Euskal Herriko Unibertsitatea) –LDGP-. <i>Programa para generar ficheros de grafos en formato DOT (v. 1.0).</i> 2012.	:CITATION

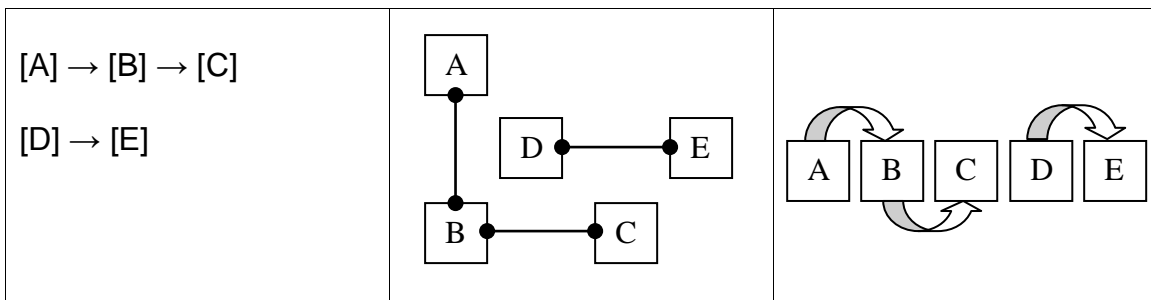
**PROGRAMA PARA GENERAR GRAFOS EN FORMATO DOT**

(v. 1.0)

**1.- Introducción**

Un grafo es una estructura topológica en la cual se dispone espacialmente un conjunto de elementos no por sus coordenadas sino por sus conexiones. Así por ejemplo, si se define un conjunto de cinco elementos: A, B, C, D y E y se establece que existen las conexiones entre A-B, B-C y D-E. El grafo nos indicará que desde el nodo "A" se puede llegar al nodo "B" y también al nodo "C" (pasando previamente por "B") pero no a "D".

Existen infinidad de representaciones gráficas válidas del grafo anterior, como las que se muestran a continuación:



Existen varios formatos informáticos que permiten codificar un grafo. Entre ellos, merece la pena destacar el formato DOT por su simplicidad (se trata de un formato ASCII con una estructura muy lógica), su versatilidad (está soportado por varias aplicaciones informáticas) y el hecho de que se encuentre documentado de forma accesible.

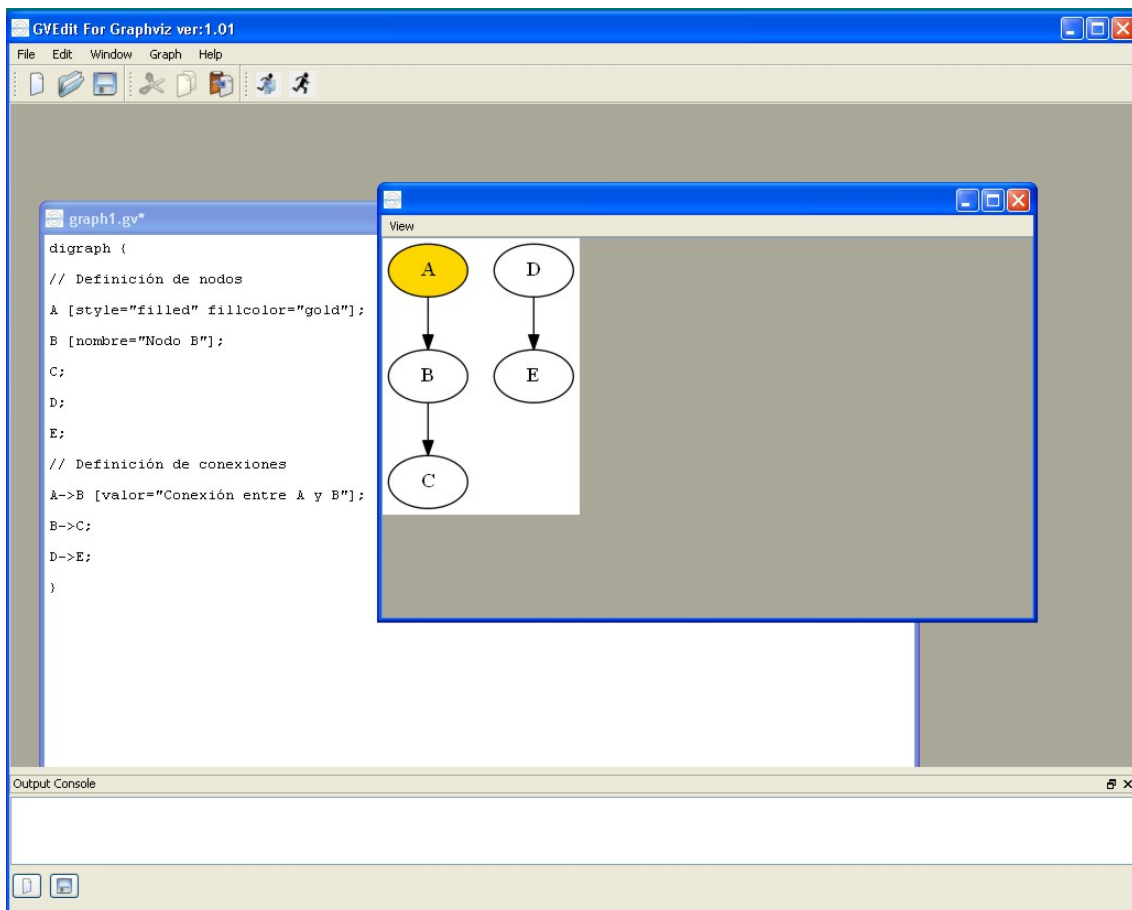
Este formato ha sido desarrollado dentro del proyecto *open source* denominado *GraphViz* [<http://www.graphviz.org/content/dot-language>] aunque, como se ha indicado, es aceptado por otros software de visualización de grafos (*Xglore*, *Tulip*, etc). Como puede verse en el siguiente extracto, los ficheros DOT contienen dos partes contenidas en un elemento "digraph {}":

```
digraph {
  // Definición de nodos
  A;
  B;
  C;
  D;
  E;
  // Definición de conexiones
  A->B;
  B->C;
  D->E;
}
```

Tanto los nodos como las conexiones pueden llevar atributos que están normalizados y marcan cómo se deben representar gráficamente (color, simbología, tipo de línea, fuente de los textos, etc...) o simplemente valores que sean relevantes y que puede definir de forma libre el usuario. Por ejemplo:

```
digraph {
  // Definición de nodos
  A [style="filled" fillcolor="gold"];
  B [nombre="Nodo B"];
  C;
  D;
  E;
  // Definición de conexiones
  A->B [valor="Conexión entre A y B"];
  B->C;
  D->E;
}
```

Una vez generados estos ficheros pueden visualizarse con alguno de los programas que los aceptan, por ejemplo el GraphViz antes comentado tal como se presenta en la siguiente imagen:



## **2.- Objeto del programa**

El objetivo del programa es generar un fichero DOT válido (por lo tanto, que pueda ser utilizado en cualquier software que soporte este formato) a partir de un fichero de texto organizado por columnas que provenga, por ejemplo, de una hoja de cálculo.

	A	B	C	D
1	<b>ID_Nodo</b>	<b>Nombre</b>	<b>Conecta con...</b>	<b>Color</b>
2	1	A		rojo
3	2	B	1	azul
4	3	C	2	verde
5	4	D		rojo
6	5	E	4	azul

Para ello es necesario que las tablas dispongan de una columna con un identificador único para cada nodo y otra que establezca las relaciones. En el programa desarrollado se contemplan dos posibilidades: que un nodo no dependa de ningún otro (en cuyo caso la relación se mantiene vacía) o que proceda de un único nodo.

En el caso de que se necesite que un nodo dependa de más de un antecesor de forma directa será necesario editar el fichero generado según se comenta más adelante.

El programa no lee directamente formatos de hoja de cálculo sino ficheros ASCII por lo que deben exportarse.

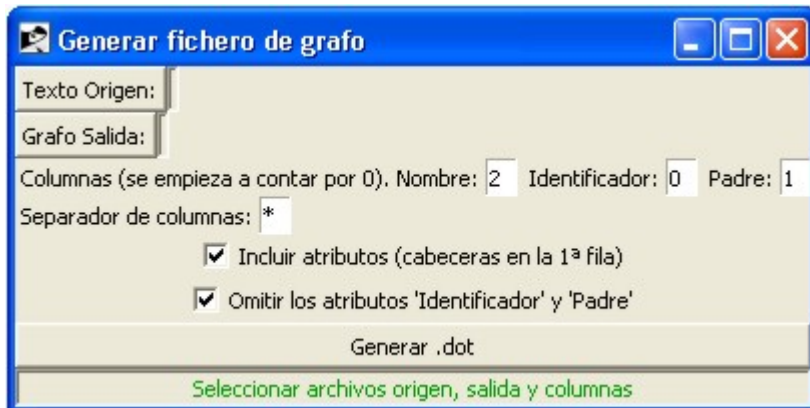
## **3.- Características del programa**

El programa se ha desarrollado en lenguaje Tcl. El código es un fichero de texto ASCII que puede ser leído y editado con cualquier aplicación de tipo "block de notas". Los ficheros Tcl son autoejecutables si se dispone del correspondiente intérprete que puede ser descargado de forma gratuita por Internet.

Se recomienda descargar los paquetes de instalación de Tcl ya preparados para la plataforma de trabajo que se desee (*Windows, Mac, Linux, ...*). Estos paquetes se denominan "*ActiveTcl*" que pueden encontrarse en la siguiente dirección: [<http://www.activestate.com/activetcl>].

#### 4.- Descripción de la interfaz

Al ejecutar el programa, aparecerá la siguiente ventana:



Cuya función se detalla a continuación:

- El botón “Texto Origen” permite seleccionar el fichero de texto con la información organizada por columnas (ver más adelante).
- El botón “Grafo Salida” permite escribir el nombre del fichero de salida que presentará la información del fichero origen según la estructura del formato DOT.
- De entre las columnas que componen el fichero origen deben indicarse la que representa el “Nombre” (texto que aparecerá en la visualización), el “Identificador” (cómo se designa de forma única a cada nodo) y el “Padre” (a qué otro nodo debe conectarse –su identificador-).
- El fichero de texto separará las columnas mediante un carácter especial, por defecto se considera el asterisco “\*” pero puede definirse.
- “Incluir atributos (cabeceras en la 1ª fila)” indica al programa que la primera fila del fichero de texto contiene los títulos de las columnas (no un nodo) y que use estos valores para identificar los atributos en cada nodo.
- “Omitir los atributos ‘Identificador’ y ‘Padre’” está relacionada con la anterior e indica que de la lista de atributos a incluir en cada nodo elimine los que corresponden a estas dos columnas.
- El botón “Generar .dot” crea el fichero con la estructura de grafo descrita.

## 5.- Notas y ejemplo

Se incluyen seguidamente algunas consideraciones sobre el código actual mediante un ejemplo desarrollado.

Se pretende generar un grafo a partir del siguiente fichero de texto que muestra el organigrama de una empresa:

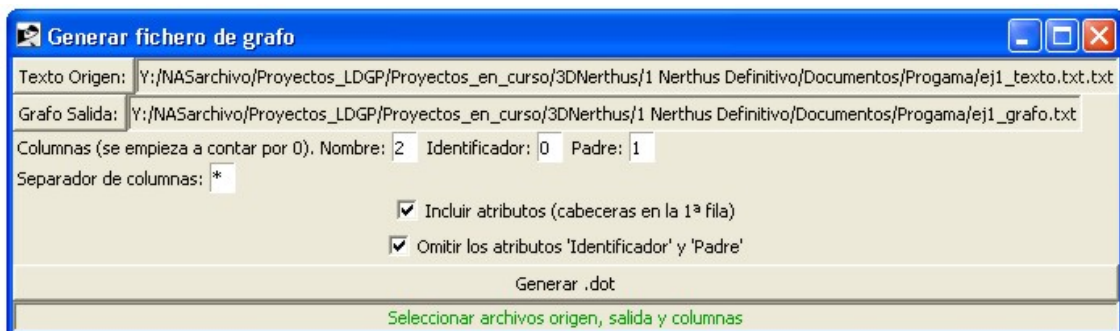
```
ID*"Depende de"*Cargo
1**Presidente
2*1*Vicepresidente
3*2*Jefe Seccion 1
4*2*Jefe Seccion 2
5*3*Empleado 1
6*4*Empleado 2
7*4*Empleado 3
8*4*Empleado 4
```

Como puede observarse, las columnas están delimitadas por asteriscos y contiene una cabecera que muestra el significado de cada una.

En la cabecera, las columnas con espacios deben ir entre paréntesis, pero en el resto de filas debe ir sin ellos.

El nodo inicial (“Presidente”) que no depende de ningún otro tendrá que llevar los dos separadores juntos “\*\*”.

Se ejecuta el programa:





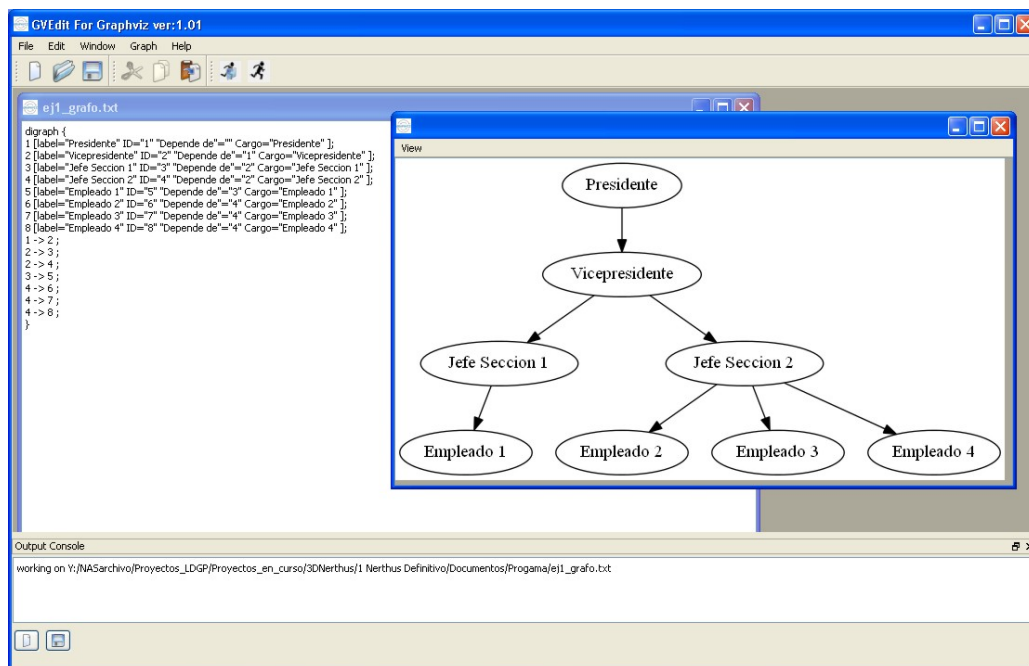
Obteniendo como resultado el siguiente fichero de texto:

```

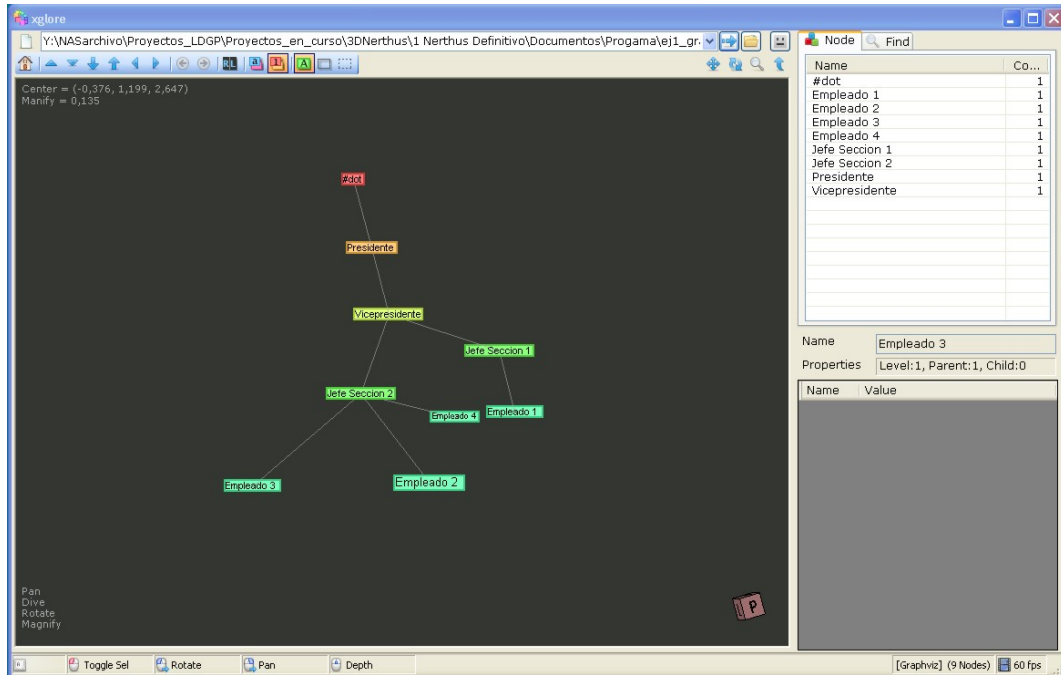
digraph {
1  [label="Presidente"      ID="1"      "Depende de"=""
Cargo="Presidente" ];
2  [label="Vicepresidente" ID="2"      "Depende de"="1"
Cargo="Vicepresidente" ];
3  [label="Jefe Seccion 1" ID="3"      "Depende de"="2"
Cargo="Jefe Seccion 1" ];
4  [label="Jefe Seccion 2" ID="4"      "Depende de"="2"
Cargo="Jefe Seccion 2" ];
5  [label="Empleado 1"    ID="5"      "Depende de"="3"
Cargo="Empleado 1" ];
6  [label="Empleado 2"    ID="6"      "Depende de"="4"
Cargo="Empleado 2" ];
7  [label="Empleado 3"    ID="7"      "Depende de"="4"
Cargo="Empleado 3" ];
8  [label="Empleado 4"    ID="8"      "Depende de"="4"
Cargo="Empleado 4" ];
1  -> 2 ;
2  -> 3 ;
2  -> 4 ;
3  -> 5 ;
4  -> 6 ;
4  -> 7 ;
4  -> 8 ;
}

```

El grafo representado puede visualizarse en cualquiera de los programas que aceptan el formato, por ejemplo el *GraphViz*:



A continuación se puede ver el mismo grafo en *Xglore* [<http://sourceforge.net/projects/xglore>] que lo presenta tridimensionalmente y permite una exploración interactiva (giros, escalado, búsqueda de nodos por el texto de los atributos,...).



Como se ha comentado, el grafo generado es funcional pero puede ser editado manualmente si se quieren incorporar algunas características, por ejemplo que un elemento dependa de más de un antecesor directo.

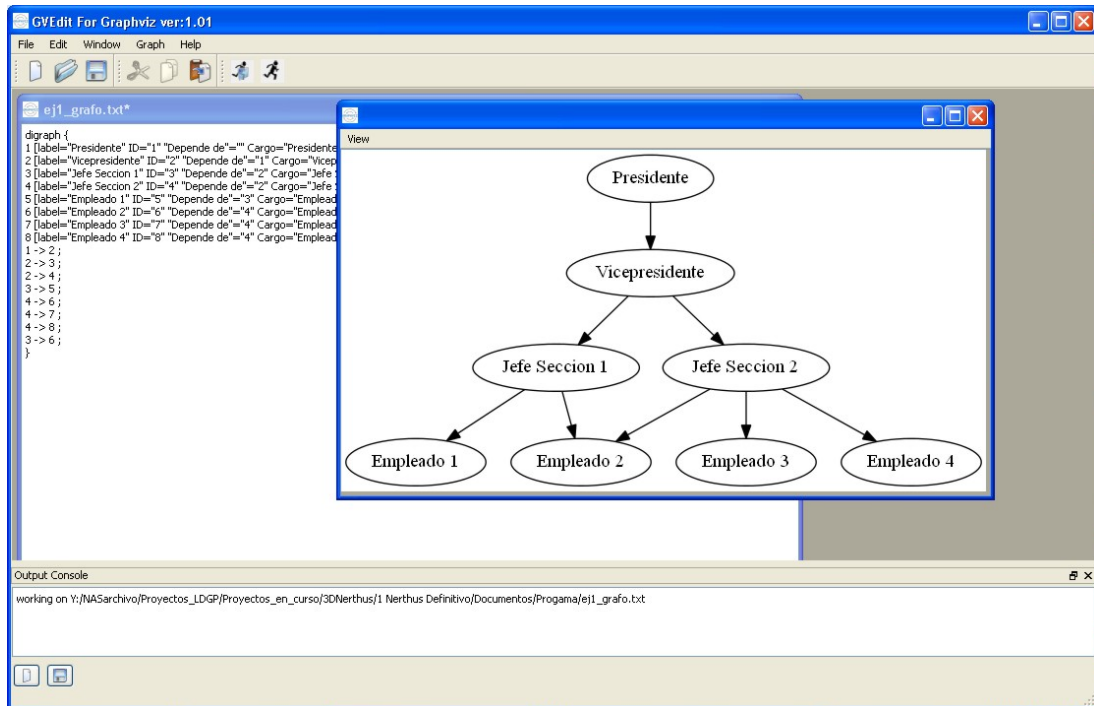
Por ejemplo, en el caso de que se quiera hacer depender el nodo "Empleado 2" de los dos jefes de sección, sería necesario editar el fichero de texto del grafo resultado en el block de notas o en el propio editor del GraphViz incluyendo la relación "3 ->6" como se muestra a continuación.

```
digraph {
1  [label="Presidente"      ID="1"    "Depende de"=" "
Cargo="Presidente" ];
2  [label="Vicepresidente" ID="2"    "Depende de"="1 "
Cargo="Vicepresidente" ];
3  [label="Jefe Seccion 1" ID="3"    "Depende de"="2 "
Cargo="Jefe Seccion 1" ];
4  [label="Jefe Seccion 2" ID="4"    "Depende de"="2 "
Cargo="Jefe Seccion 2" ];
5  [label="Empleado 1"    ID="5"    "Depende de"="3 "
Cargo="Empleado 1" ];
6  [label="Empleado 2"    ID="6"    "Depende de"="4 "
Cargo="Empleado 2" ];
7  [label="Empleado 3"    ID="7"    "Depende de"="4 "
Cargo="Empleado 3" ];
8  [label="Empleado 4"    ID="8"    "Depende de"="4 "
Cargo="Empleado 4" ];
```

```

1 -> 2 ;
2 -> 3 ;
2 -> 4 ;
3 -> 5 ;
4 -> 6 ;
4 -> 7 ;
4 -> 8 ;
3 -> 6 ;
}
    
```

El resultado tras la modificación incluiría esta nueva conexión como se muestra en la imagen:





**LABORATORIO DE DOCUMENTACIÓN GEOMÉTRICA DEL PATRIMONIO**

Grupo de Investigación en Patrimonio Construido -GPAC- (UPV-EHU)

Aulario de las Nieves, edificio de Institutos Universitarias  
C/ Nieves Cano 33, 01006 Vitoria-Gasteiz (España-Spain).  
Tfno: +34 945 013222 / 013264  
e-mail: [jm.valle@ehu.es](mailto:jm.valle@ehu.es) web: <http://www.ldgp.es>



UPV EHU