

**GERS**

**GUIA GENERAL DE PROGRAMACIÓN EN DIGSILENT -  
DPL**

Sebastian Campos Bustos  
M.Sc. Carlo Viggiano

31/10/2018

---

Introducción y conceptos generales.

**GERS**

# DPL – DIgSILENT Programming Library

---

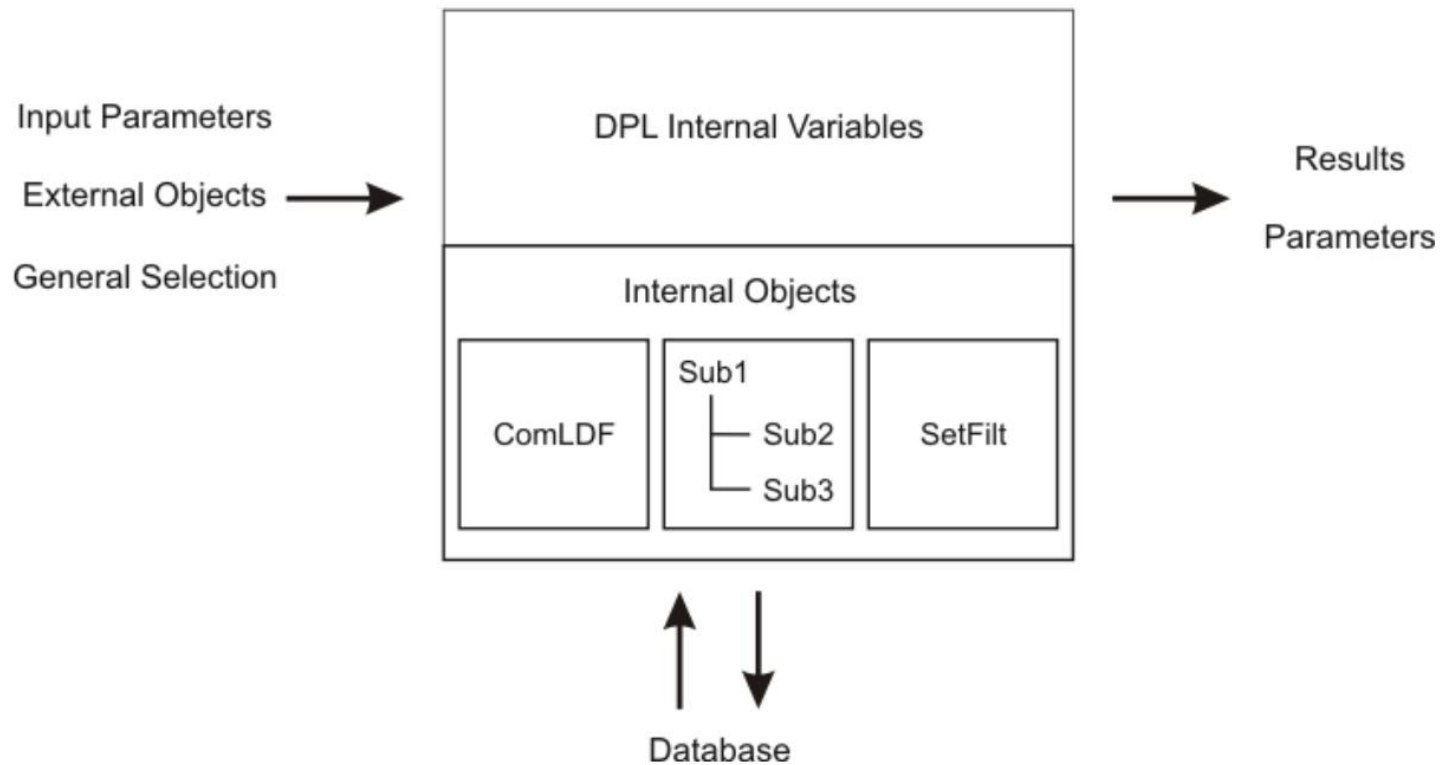
- Librería de programación de DIgSILENT que ofrece al usuario la posibilidad de automatizar tareas dentro de PowerFactory. Permite el acceso a funciones y elementos de PowerFactory mediante scripts.
- Los scripts son desarrollados en el lenguaje de programación DPL Script Language, el cual es similar a C++.
- La programación del script se realiza al utilizar el objeto DPL Command Object.

# Scripts

---

- Un script es un archivo o documento que es escrito en un lenguaje de programación específico, el cual contiene instrucciones que para poder ser ejecutadas se necesita un interprete.
- Las instrucciones pueden ser ejecutadas por lotes o una por una.

# DPL Command Object



Estructura principal del DPL Command Object.  
(Tomado del Manual de usuario de DIGSILENT  
PowerFactory)

# ¿En qué consiste DPL Script Language?

---

- Lenguaje de programación propio de DlgSILENT similar a C++.
- Es un lenguaje de programación intuitivo, fácil de leer y fácil de aprender.
- La sintaxis de este lenguaje se puede dividir en:
  - Definición de variables
  - Asignaciones y expresiones
  - Instrucciones y llamada de métodos.
- Utiliza un editor propio y el script esta asociado a el objeto DPL Command Object. Por lo tanto, es necesario tener DlgSILENT para poder realizar un script.

# Tipo de datos

---

- DPL Script Language sólo permite los siguientes tipos de datos:
  - double : Número real de hasta 15 dígitos.
  - int : Número entero.
  - string : Cadena de texto.
  - object : Objeto que hace referencia a un objeto de DIGSILENT.
  - Set : Contenedor de objetos.

# Funciones matemáticas

function	description	example
<b>sin(x)</b>	sine	sin(1.2)=0.93203
<b>cos(x)</b>	cosine	cos(1.2)=0.36236
<b>tan(x)</b>	tangent	tan(1.2)=2.57215
<b>asin(x)</b>	arcsine	asin(0.93203)=1.2
<b>acos(x)</b>	arccosine	acos(0.36236)=1.2
<b>atan(x)</b>	arctangent	atan(2.57215)=1.2
<b>sinh(x)</b>	hyperbolic sine	sinh(1.5708)=2.3013
<b>cosh(x)</b>	hyperbolic cosine	cosh(1.5708)=2.5092
<b>tanh(x)</b>	hyperbolic tangent	tanh(0.7616)=1.0000
<b>exp(x)</b>	exponential value	exp(1.0)=2.718281
<b>ln(x)</b>	natural logarithm	ln(2.718281)=1.0
<b>log(x)</b>	log10	log(100)=2
<b>sqrt(x)</b>	square root	sqrt(9.5)=3.0822
<b>sqr(x)</b>	power of 2	sqr(3.0822)=9.5
<b>pow(x,y)</b>	power of y	pow(2.5, 3.4)=22.5422
<b>abs(x)</b>	absolute value	abs(-2.34)=2.34
<b>min(x,y)</b>	smaller value	min(6.4, 1.5)=1.5
<b>max(x,y)</b>	larger value	max(6.4, 1.5)=6.4
<b>modulo(x,y)</b>	remainder of x/y	modulo(15.6,3.4)=2
<b>trunc(x)</b>	integral part	trunc(-4.58823)=-4.0000
<b>frac(x)</b>	fractional part	frac(-4.58823)=-0.58823
<b>round(x)</b>	closest integer	round(1.65)=2.000
<b>ceil(x)</b>	smallest larger integer	ceil(1.15)=2.000
<b>floor(x)</b>	largest smaller integer	floor(1.78)=1.000

Funciones basicas de DPL.

(Tomado del manual de usuario de DigSILENT PowerFactory).

pi()	pi
twopi()	2 pi
e()	e

Constantes de DPL  
(Tomado del manual de usuario de DigSILENT PowerFactory).



---

# Guía para crear un DPL script en DIGSILENT

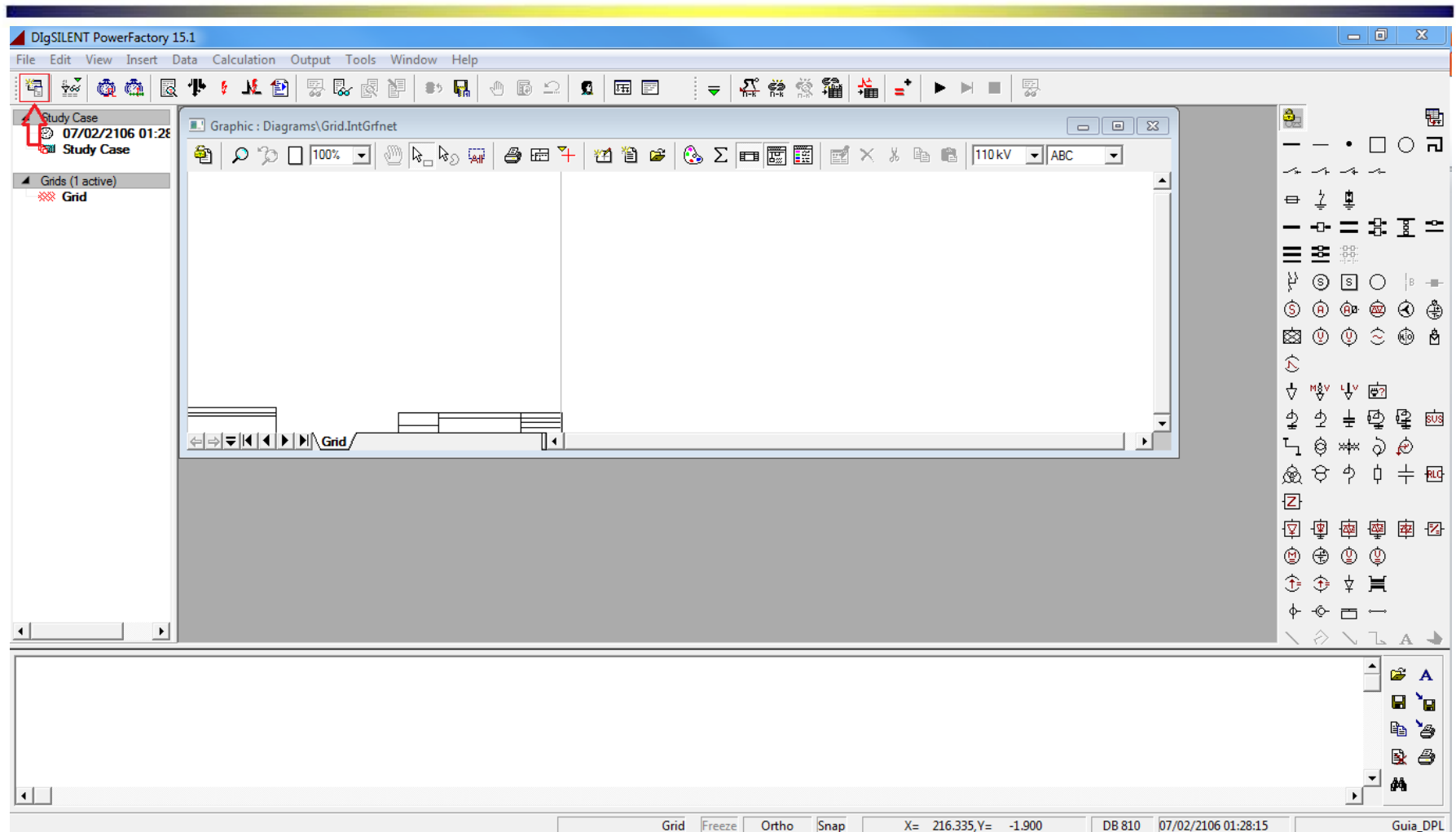
# Pasos

---

- Es necesario tener el programa PowerFactory instalado.
- Una vez el programa instalado y un proyecto activo, se realizan los siguientes pasos:

Nota: No es necesaria la instalación de software externo de programación.

# Presionar clic en el ícono para abrir el Data Manager del proyecto.



# Seleccionar el nombre del proyecto.

The screenshot shows the DigSILENT PowerFactory 15.1 interface. The main window displays a diagram titled "Graphic : Diagrams\Grid.IntGrfnet". A "Data Manager" window is open, showing a tree view on the left and a table on the right. The tree view shows a folder structure under "Database" with "sebastian.campos" expanded, and "Guia\_DPL" selected. The table lists the following data:

Name	Type	Object modified	Object modified by
Guia_DPL		17/10/2018 07:46:11	sebastian.campos
LV Distribution Network		03/10/2018 09:17:06	sebastian.campos
Project		12/10/2018 11:56:26	sebastian.campos
Tutorial Exercise 2		09/10/2018 12:25:41	sebastian.campos
Tutorial Exercise 7 finished		09/10/2018 12:24:38	sebastian.campos
Tutorial_HelloWorld		17/10/2018 07:46:11	sebastian.campos
e		05/10/2018 15:45:24	sebastian.campos
Recycle Bin		03/10/2018 09:15:30	Administrator
Settings		03/10/2018 09:15:30	Administrator

The status bar at the bottom shows "Grid Freeze Ortho Snap X= 114.027,Y= 123.529 DB 810 07/02/2106 01:28:15 Guia\_DPL".

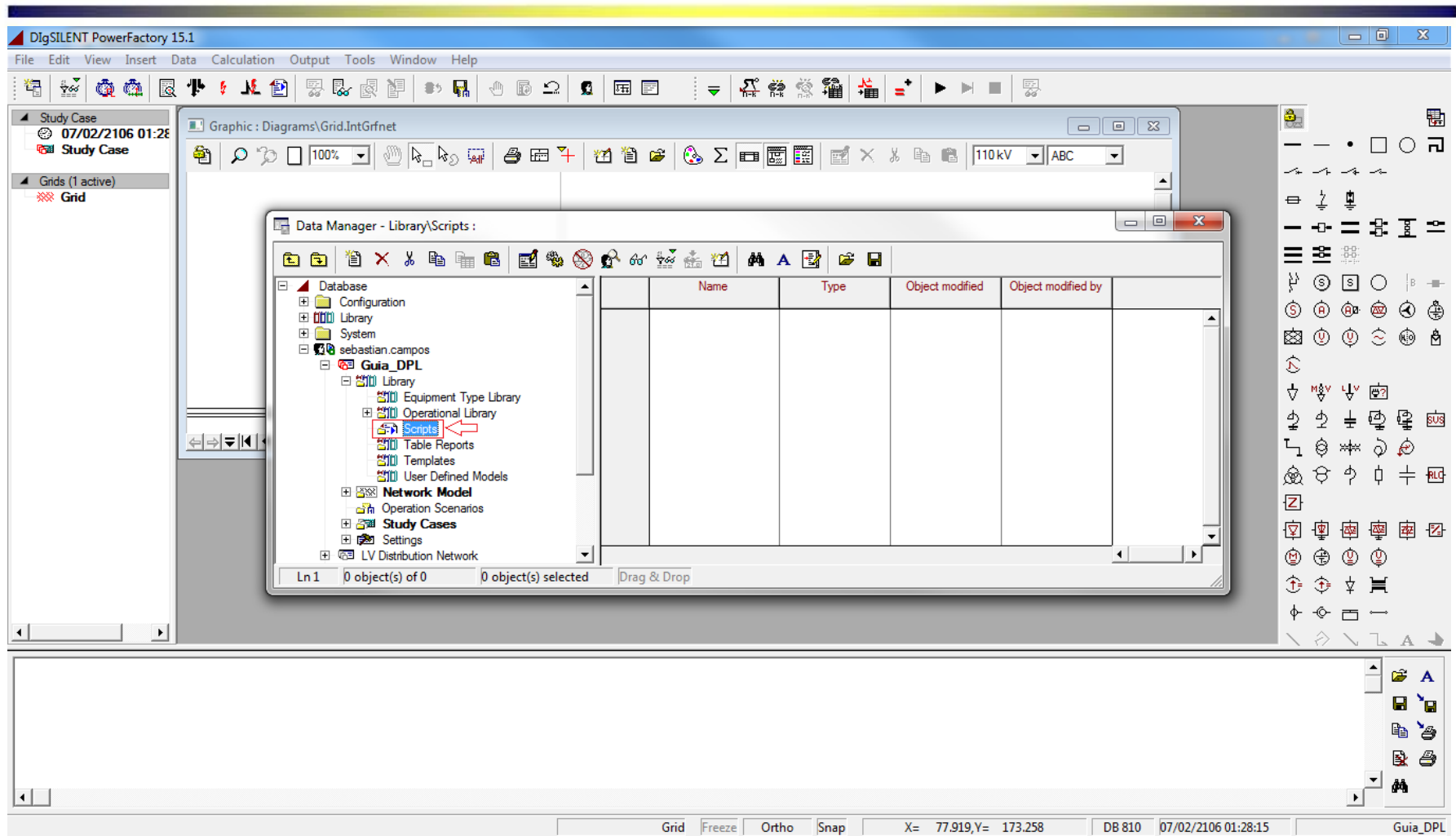
# Seleccionar el menú desplegable llamado Library.

The screenshot shows the DigSILENT PowerFactory 15.1 interface. The 'Data Manager' window is open, displaying a tree view on the left and a table on the right. The tree view shows a hierarchy starting with 'Database', followed by 'Configuration', 'Library', 'System', and 'sebastian.campos'. Under 'sebastian.campos', 'Guia\_DPL' is expanded, and 'Library' is selected, indicated by a red arrow. The table on the right lists the following objects:

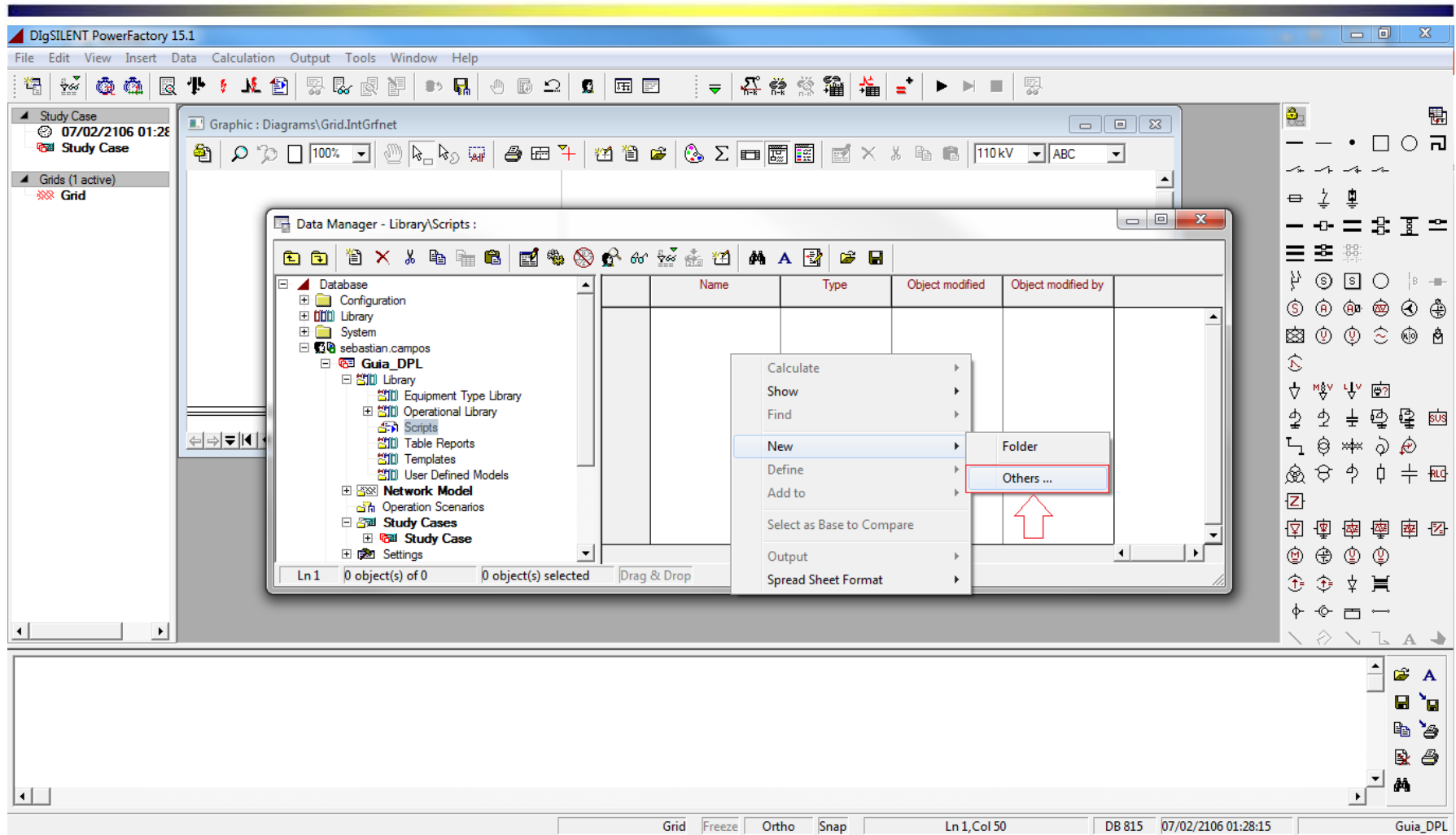
Name	Type	Object modified	Object modified by
Guia_DPL		17/10/2018 07:46:11	sebastian.campos
LV Distribution Network		03/10/2018 09:17:08	sebastian.campos
Project		12/10/2018 11:56:29	sebastian.campos
Tutorial Exercise 2		09/10/2018 12:25:41	sebastian.campos
Tutorial Exercise 7 finished		09/10/2018 12:24:38	sebastian.campos
Tutorial_HelloWorld		17/10/2018 07:46:10	sebastian.campos
e		05/10/2018 15:45:24	sebastian.campos
Recycle Bin		03/10/2018 09:15:30	Administrator
Settings		03/10/2018 09:15:30	Administrator

The status bar at the bottom of the Data Manager window shows 'Ln 1', '9 object(s) of 9', '1 object(s) selected', and 'Drag & Drop'. The main interface also shows a menu bar (File, Edit, View, Insert, Data, Calculation, Output, Tools, Window, Help) and a toolbar with various icons. The bottom status bar of the application shows 'Grid', 'Freeze', 'Ortho', 'Snap', 'X= 55.430, Y= -12.986', 'DB 810', '07/02/2106 01:28:15', and 'Guia\_DPL'.

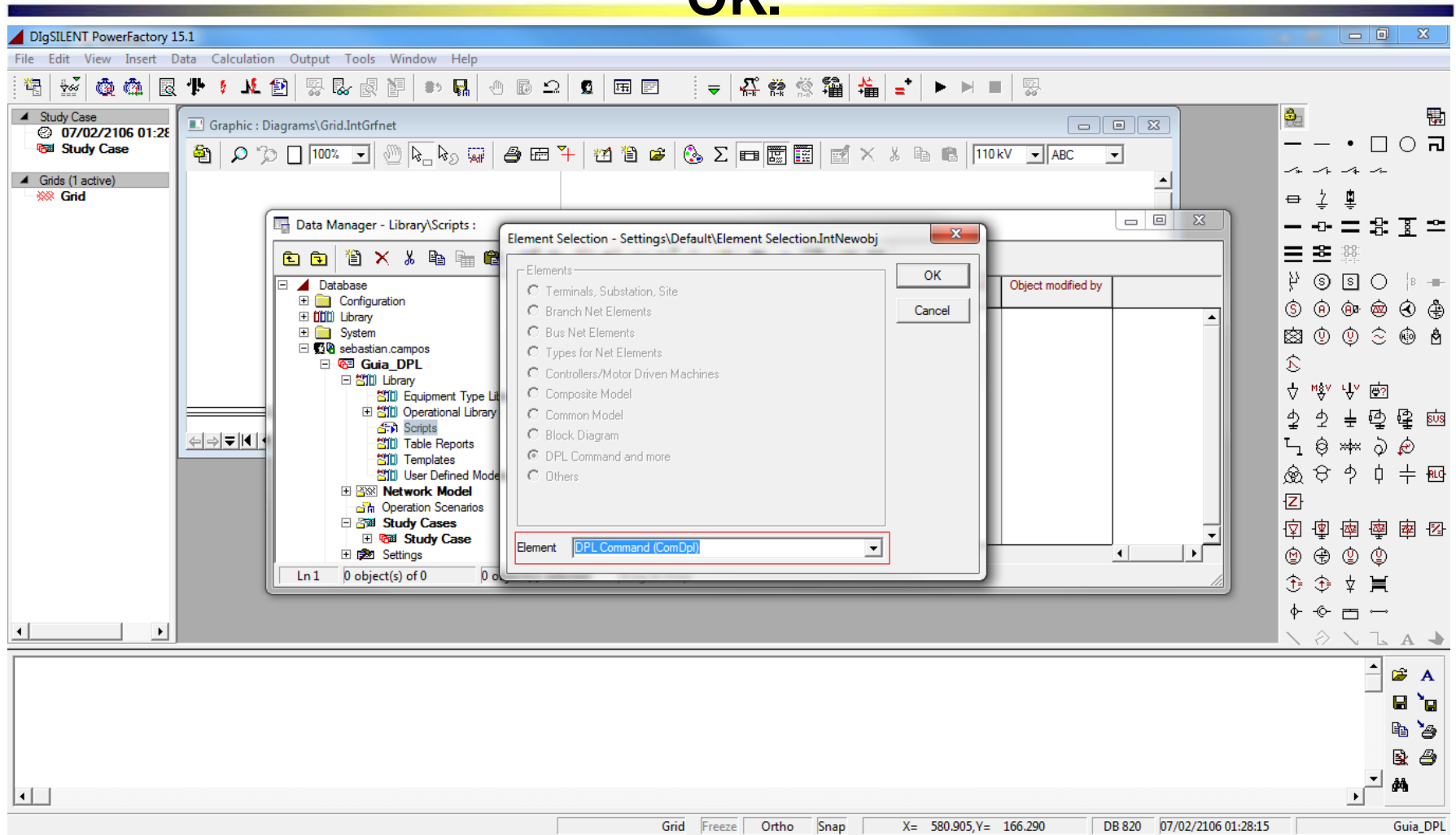
# Seleccionar el menú desplegable llamado Scripts.



# Presionar clic derecho dentro del área que pertenece a scripts y seleccionar new-others.

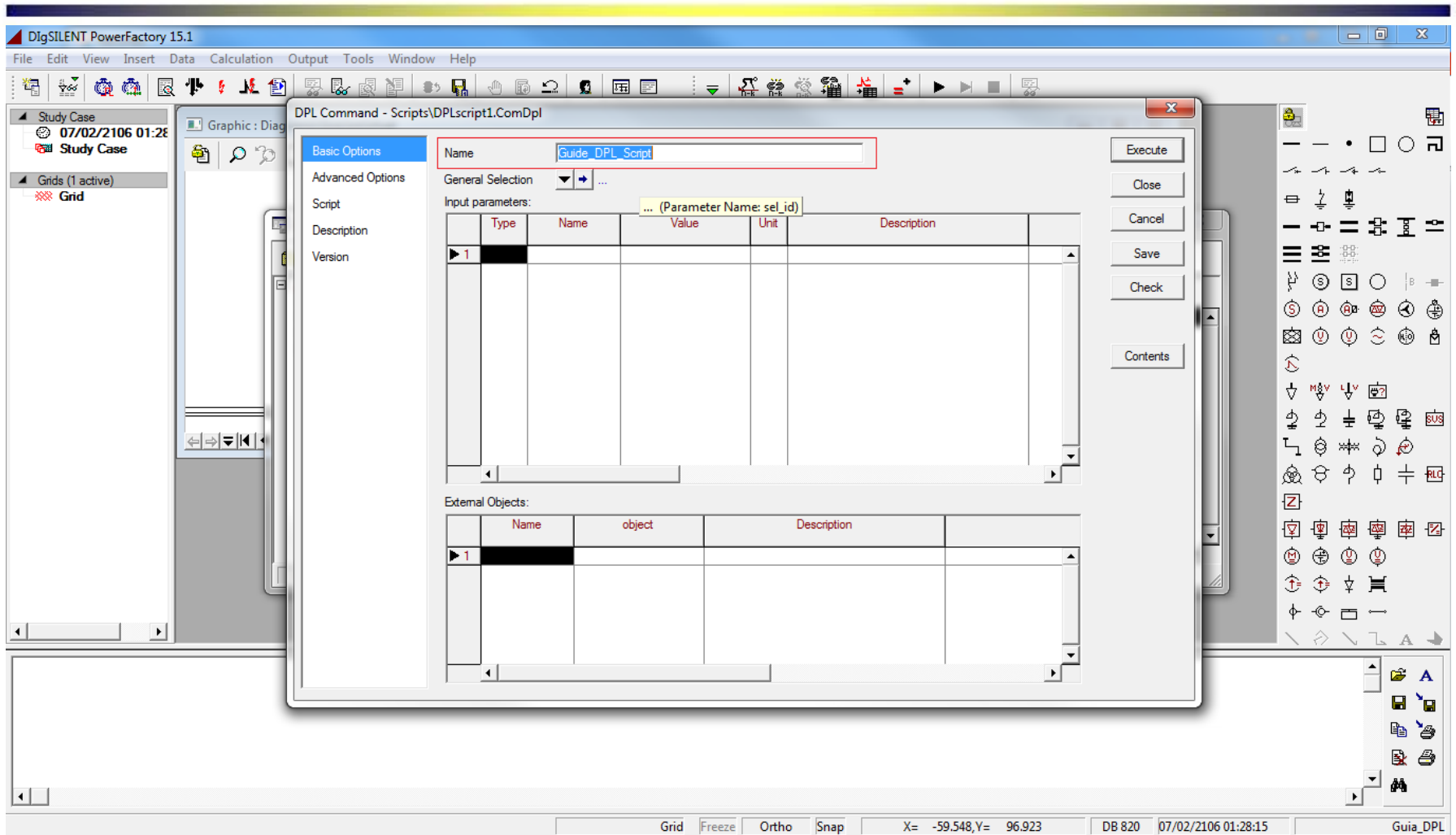


# En la opción Element de la ventana emergente seleccionar DPL Command (ComDpl) y dar clic en OK.



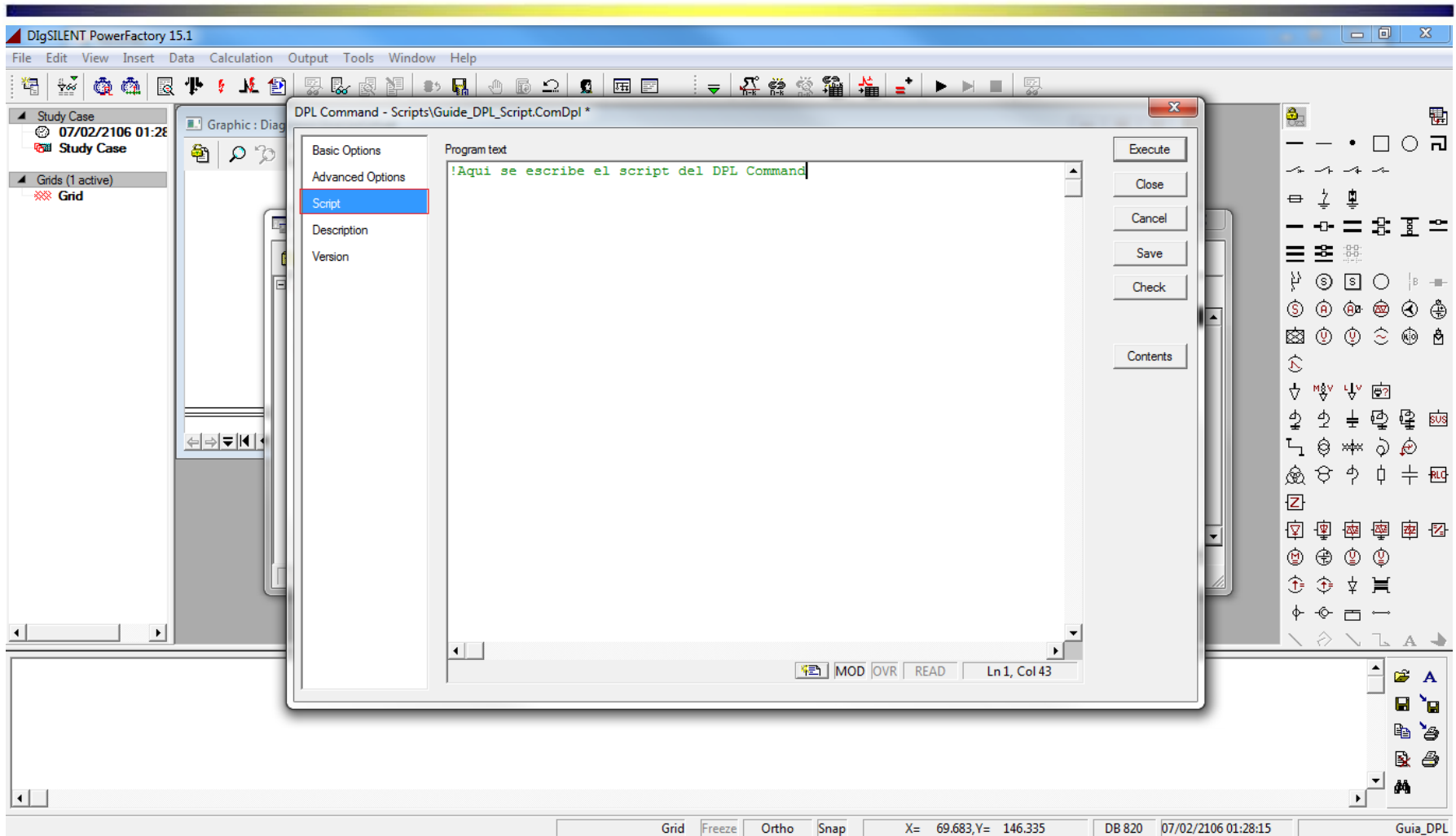


# Seleccionar el campo llamado Name. Se digita el nombre que se asignará al DPL Command.



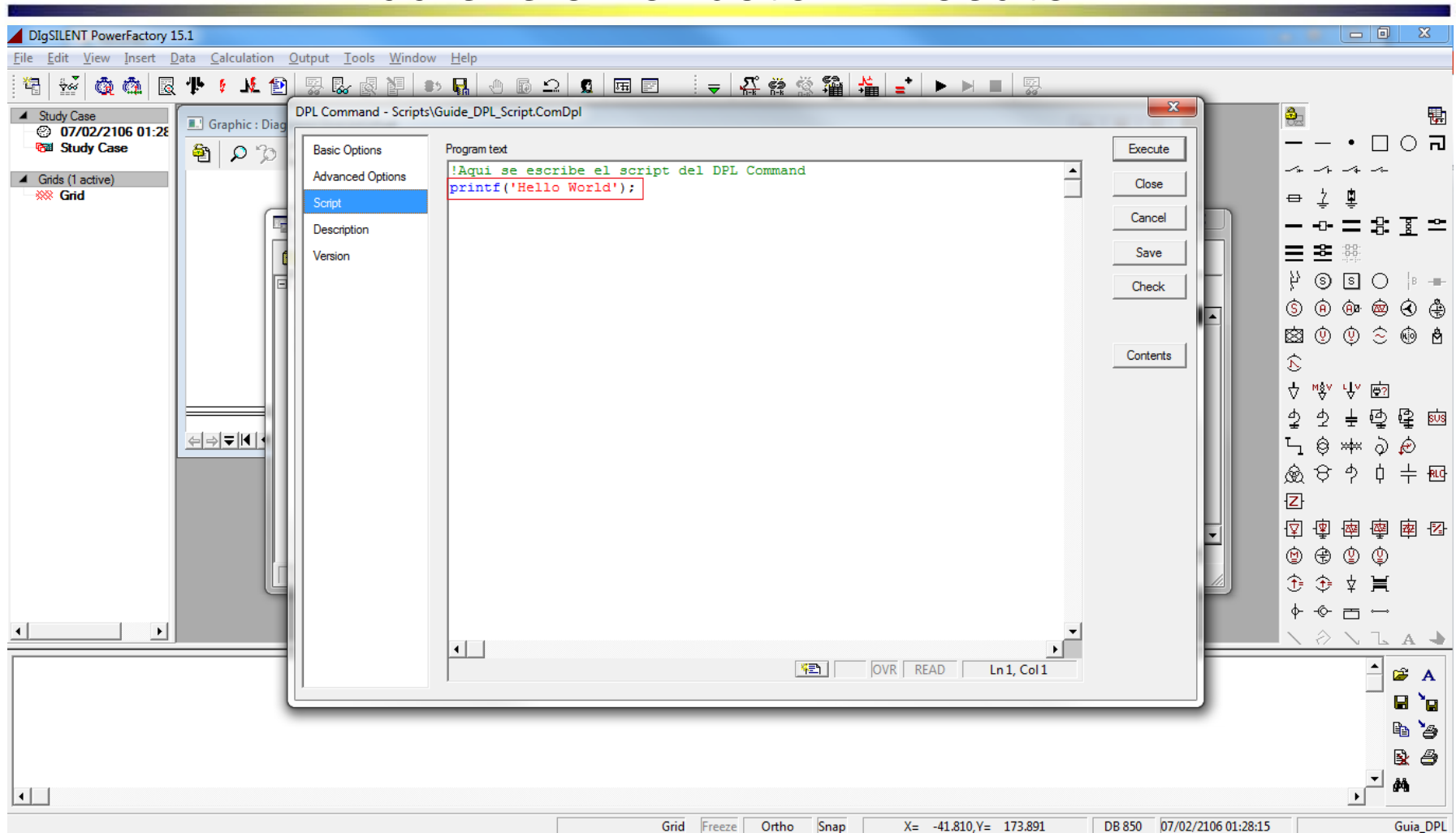
En caso de ser necesario en esta ventana se pueden definir los parámetros de entrada y objetos externos que se usarán.

# Seleccionar la opción Script.



En caso de ser necesario, en la opción Advanced Options se pueden añadir los parámetros de salida o asociar el DPL Command a un script remoto u otro script.

En el program text se escribe el código que se muestra en la imagen encerrado en color rojo y se da clic en el botón Execute.



# Se debe obtener el mensaje Hello World en la ventana de salida de DigSILENT.

The screenshot displays the DigSILENT PowerFactory 15.1 interface. The main window shows a diagram titled 'Graphic : Diagrams/Grid.IntGrfnet'. A 'Data Manager - Library/Scripts' window is open, showing a tree view of the library structure. The 'Scripts' folder is expanded, and the 'Guide\_DPL\_Script' is selected. The console window at the bottom shows the following output:

```
DigSI/info - DPL Program 'Guide_DPL_Script' started  
Hello World  
DigSI/info - DPL program 'Guide_DPL_Script' successfully executed
```

A red arrow points to the 'Hello World' message in the console output.

Name	Type	Object modified	Object modified by
Guide_DPL_Script		17/10/2018 08:18:41	sebastian.campos

---

Ejemplo 1: Modificar un parámetro de una línea, en este caso se modificará la longitud.

Nota: El código se puede encontrar en la carpeta de la presentación con el nombre: `Modify_length_of_line.txt`

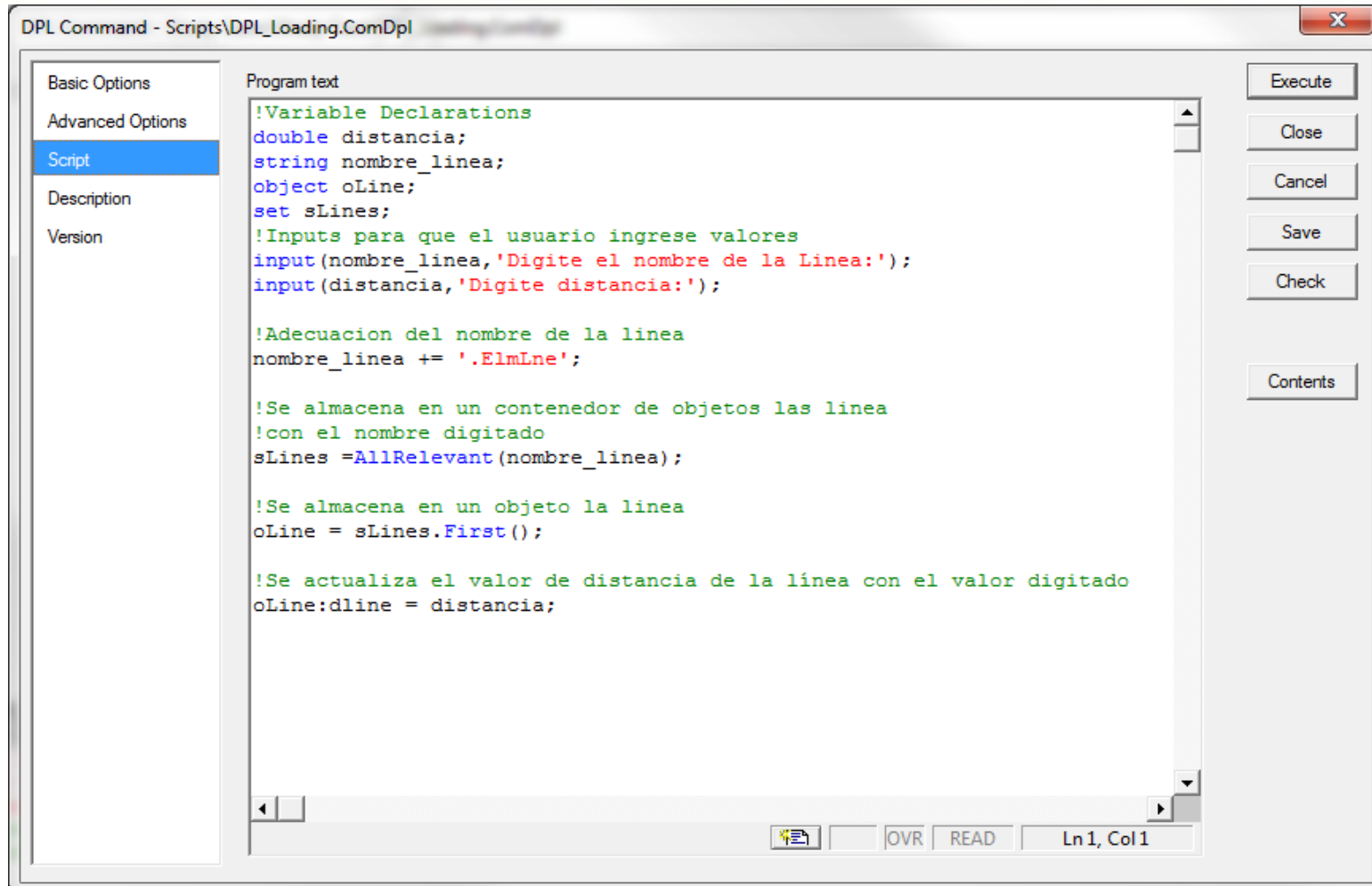
# Para una línea existente en el proyecto se verifica el valor de la longitud de la línea.

The screenshot displays the 'Line - Grid\Line1-2.ElmLine\*' dialog box in DlgSILENT PowerFactory 15.1. The 'Length of Line' field is set to 20 km. The 'Resulting Values' table provides the following data:

Resulting Values	
Rated Current (act.)	0.367 kA
Pos. Seq. Impedance, Z1	4.022875 Ohm
Pos. Seq. Impedance, Angle	39.00818 deg
Pos. Seq. Resistance, R1	3.126 Ohm
Pos. Seq. Reactance, X1	2.532124 Ohm
Zero Seq. Resistance, R0	12.506 Ohm
Zero Seq. Reactance, X0	10.1285 Ohm
Earth-Fault Current, Ice	50.99678 A
Earth Factor, Magnitude	1.000129
Earth Factor, Angle	-0.00597667 deg

Se puede observar que la línea llamada Linea\_Prueba tiene una longitud de 20 km.

# En el program text escribe el siguiente código:



The screenshot shows a software window titled "DPL Command - Scripts\DPL>Loading.ComDpl". On the left, there is a sidebar with menu items: "Basic Options", "Advanced Options", "Script" (highlighted in blue), "Description", and "Version". The main area is a text editor titled "Program text" containing the following code:

```
!Variable Declarations
double distancia;
string nombre_linea;
object oLine;
set sLines;
!Inputs para que el usuario ingrese valores
input(nombre_linea, 'Digite el nombre de la Linea:');
input(distancia, 'Digite distancia:');

!Adecuacion del nombre de la linea
nombre_linea += '.ElmLne';

!Se almacena en un contenedor de objetos las linea
!con el nombre digitado
sLines =AllRelevant(nombre_linea);

!Se almacena en un objeto la linea
oLine = sLines.First();

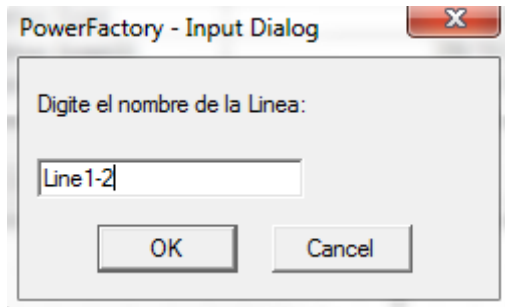
!Se actualiza el valor de distancia de la línea con el valor digitado
oLine:dline = distancia;
```

On the right side of the window, there are several buttons: "Execute", "Close", "Cancel", "Save", "Check", and "Contents". At the bottom right, there is a status bar showing "Ln 1, Col 1" and some icons.

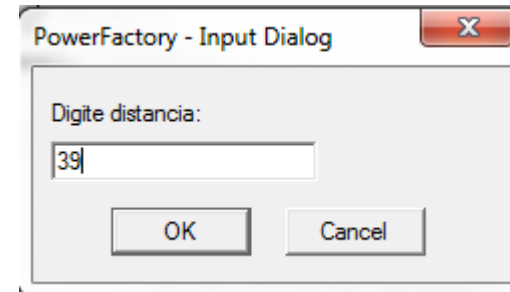
Para conocer el nombre de los parámetros de cualquier elemento en DigSILENT, se abre el elemento y se posiciona el mouse sobre el parámetro. Se despliega un cuadro amarillo donde se observa el nombre.

# Se da click en ejecutar y se obtiene:

---



Primero aparece la ventana emergente donde se digita el nombre de la línea. En este caso es Linea\_Prueba.

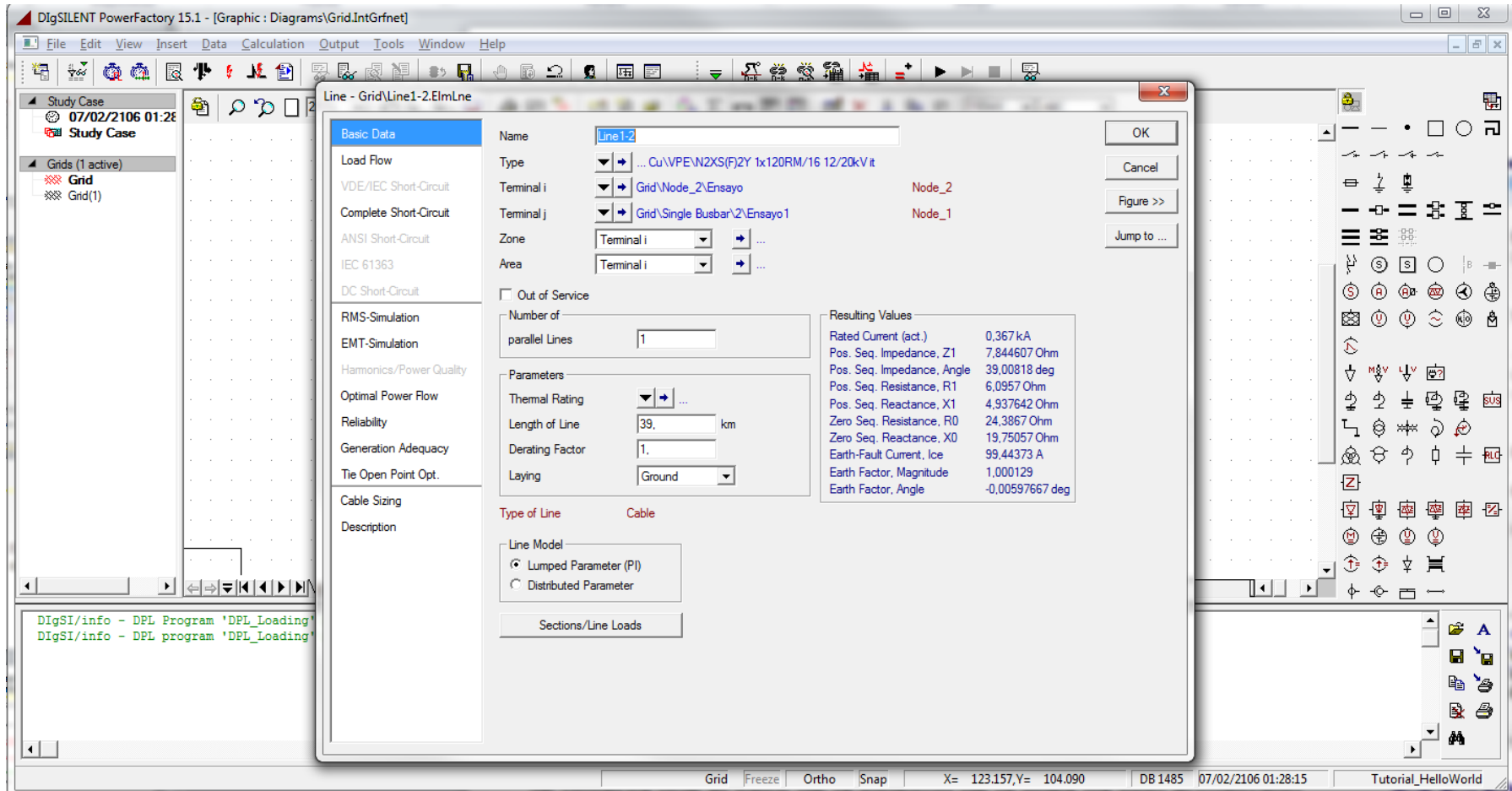


Después de digitar el nombre, sale esta nueva ventana donde se digita la distancia que tendrá la línea en km, en este caso será de 39 km.

En la ventana de salida se obtienen dos mensajes informativos indicando que el Programa DPL comenzó y se ejecuto con éxito.



# Se revisa que la longitud de la línea llamada Linea\_Prueba haya sido actualizada.



Se observa que el valor de la longitud de la línea ha sido actualizado de manera correcta.

---

## Ejemplo 2: Exportar algunos parámetros de la línea a Excel.

Nota: El código se puede encontrar en la carpeta de la presentación con el nombre: `Export_to_Excel.txt`

# En el program text se escribe el siguiente código:

## Primera Parte

---

```
!string class, attributes, nameattributes, s, desc, type, sval, sep, numberFormat;
set objs;
object obj, oval;
double dval;
int error, pos, i, t, row, col, maxRow, maxCol;
!export definition
class = 'ElmLne';
attributes = 'loc_name, typ_id, bus1, bus2, dline, c:loading, m:P:bus2, m:Q:bus2';
nameattributes = 'Nombre_Linea, Tipo_Linea, Bus1, Bus2, Distancia, Carga, P (MW), Q (MVar)';
error = xlStart();
if(error){
    Error('Unable to start MS Excel application');
}

!get decimal separator and buil number format used here
sep = xlGetDecimalSeparator();
numberFormat = sprintf('0%s000', sep);

!create new workbook
xlNewWorkbook();
xlSetWorksheetName(1, class);

!iterate over attributes and write header row
row = 1;
col = 1;
s = strtok(nameattributes, ',', pos, col);
```

# Segunda Parte

```
while(pos > -1){
    xlSetValue(col,row,s);
    col+=1;
    s = strtok(nameattributes,',',pos,col);
}
maxCol = col-1;
!change format of header row
xlSetTextStyle(1,1,maxCol,1,1); !bold
xlSetFillColor(1,1,maxCol,1,0,255,0); !red
!xlSetFillColor(1,1,maxCol,1,255,255,150); !yellow
xlSetBorder(1,1,maxCol,1,1,1,2,0,0,0); !border at bottom

!export data
row = 2;
objs = AllRelevant(class);
for (obj=objs.First(); obj; obj=objs.Next()){
    col = 1;
    s = strtok(attributes,',',pos,col);
    while(pos > -1){
        obj.GetVarType(s,type);
        t = strcmp(type,'string');
        if (t=0){
            obj.GetVal(sval,s);
            xlSetValue(col,row,sval);
        }
        t=strcmp(type,'object');
        if (t=0){
            obj.GetVal(oval,s);
            sval = oval:loc name:
        }
    }
}
```

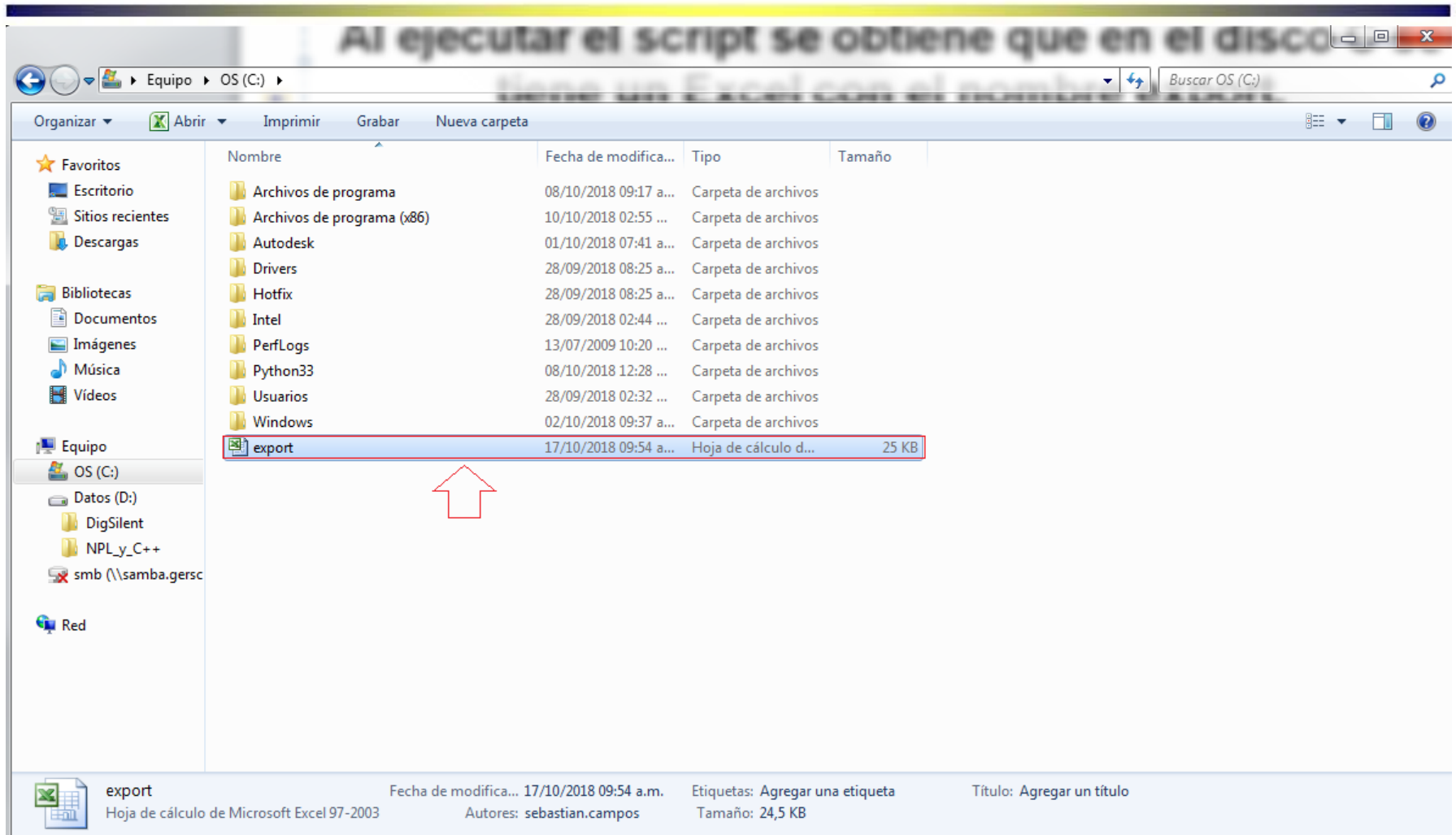
# Tercera parte

---

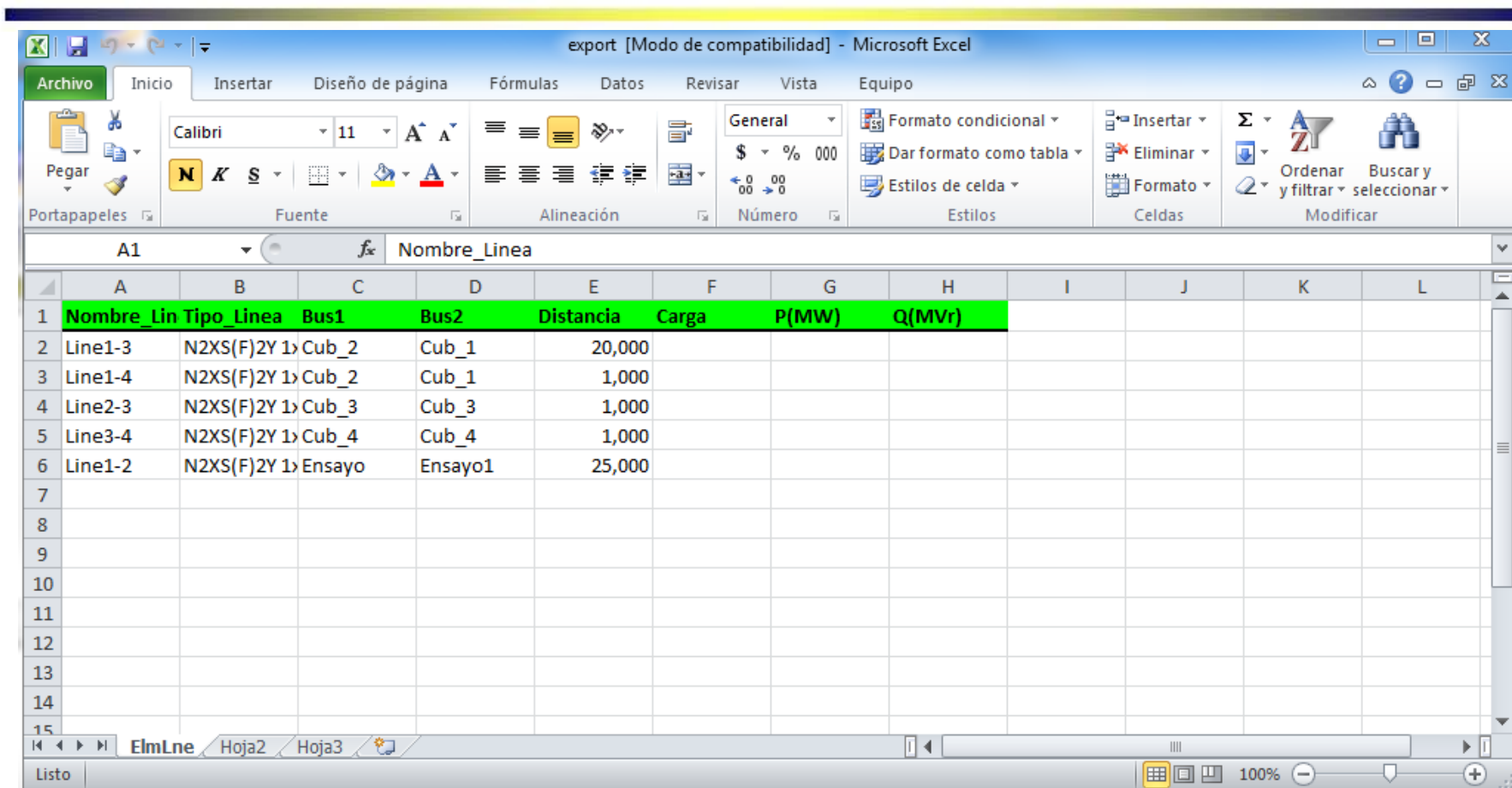
```
        xlSetValue(col, row, sval);
    }
    t=strcmp(type, 'double');
    if(t=0){
        obj.GetVal(dval, s);
        xlSetValue(col, row, dval);
        xlSetNumberFormat(col, row, numberFormat);
    }
    col+=1;
    s = strtok(attributes, ',', pos, col);
}
row+=1;
}

!save and exit
error = xlSaveWorkbookAs('c:\export.xls');
if (error){
    Error('Workbook could not be saved');
}
xlTerminate();
```

# Al ejecutar el script se obtiene que en el disco C se tiene un Excel con el nombre export



# Al abrir el archivo de Excel se obtiene lo siguiente:



The screenshot shows a Microsoft Excel spreadsheet with the following data:

1	Nombre_Lin	Tipo_Linea	Bus1	Bus2	Distancia	Carga	P(MW)	Q(MVr)
2	Line1-3	N2XS(F)2Y 1	Cub_2	Cub_1	20,000			
3	Line1-4	N2XS(F)2Y 1	Cub_2	Cub_1	1,000			
4	Line2-3	N2XS(F)2Y 1	Cub_3	Cub_3	1,000			
5	Line3-4	N2XS(F)2Y 1	Cub_4	Cub_4	1,000			
6	Line1-2	N2XS(F)2Y 1	Ensayo	Ensayo1	25,000			
7								
8								
9								
10								
11								
12								
13								
14								
15								

Los valores de carga, P y Q se encuentran vacíos debido a que no se ha ejecutado un flujo de carga. En caso de ejecutar el script después de realizar un flujo de carga, estos campos ya deberían tener algún valor.

---

## Ejemplo 3: Leer un archivo txt.

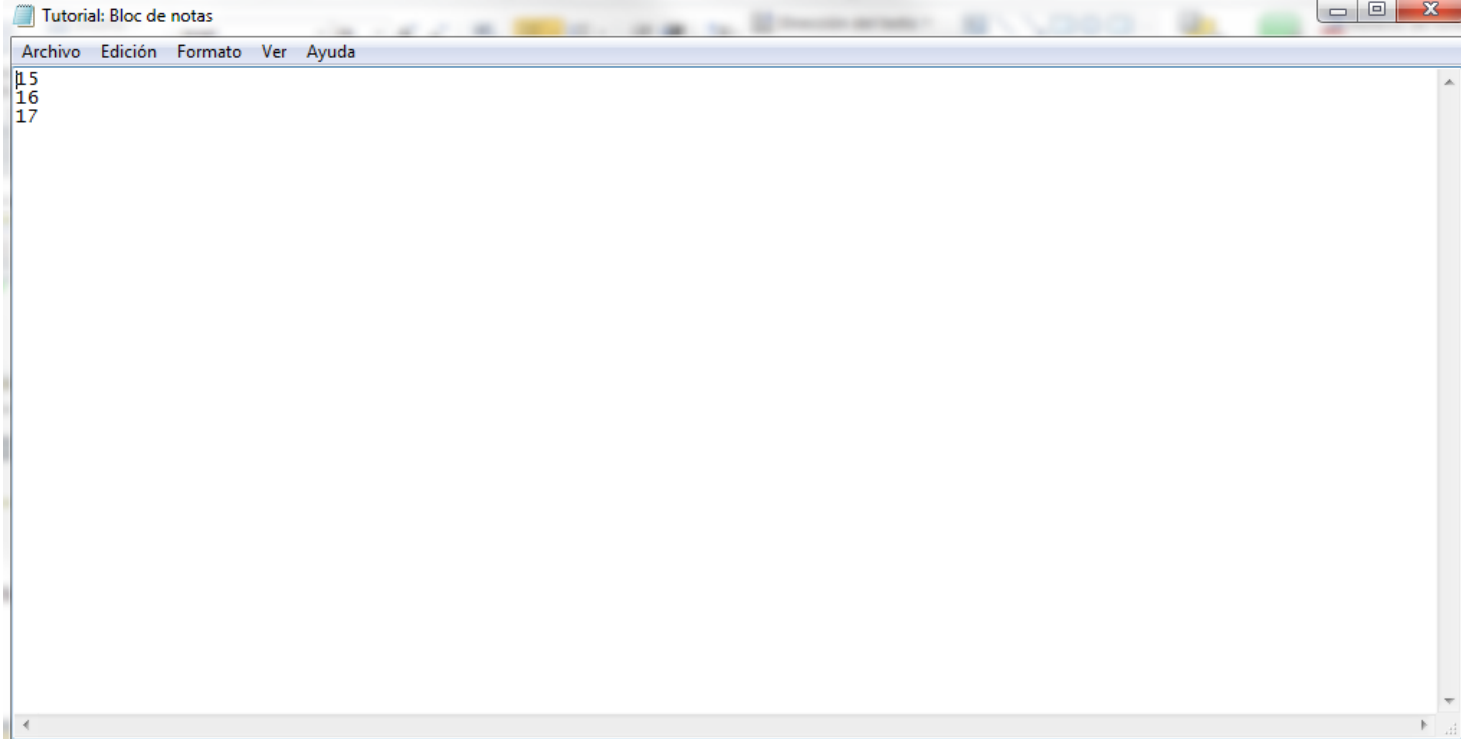
Nota: El código se puede encontrar en la carpeta de la presentación con el nombre: `Read_txt_file.txt`



# Crear un archivo txt y llamarlo Tutorial

---

- Dentro del archivo creado escribir un valor por fila, por ejemplo:



A screenshot of a Windows Notepad application window. The title bar reads "Tutorial: Bloc de notas". The menu bar includes "Archivo", "Edición", "Formato", "Ver", and "Ayuda". The text area contains three lines of text: "15", "16", and "17", each on a new line. The window has standard Windows window controls (minimize, maximize, close) in the top right corner.

# En la casilla program text escribir el siguiente código:

---

```
!Variable Declarations
double distancia;
int iRet;
!Open the txt file in D:\DigSilent\Tutoriales\Tutorial.txt
fopen('D:\DigSilent\Tutoriales\Tutorial.txt','r',0);
while(iRet>-1){
!Scan the txt file
iRet = fscanf(0,'%d',distancia);
printf('Distancia: %d', distancia);
}
!Close the txt file
fclose(0);
```

Código para leer un archivo txt

# Una vez ejecutado el script se obtiene el siguiente resultado en la ventana de salida:

---

```
DIgSI/info - DPL Program 'DPL>Loading' started
Distancia: 15
Distancia: 16
Distancia: 17
DIgSI/info - DPL program 'DPL>Loading' successfully executed
```

Resultados obtenidos al ejecutar el script

---

Ejemplo 4: Ejecutar contingencias n-1,  
donde los elementos a desconectar son  
todas las líneas una a una.

Nota: El código se puede encontrar en la carpeta de la  
presentación con el nombre: Lines\_Contingencies.txt

# En la casilla program text escribir el siguiente código:

```
Program text
!Variable Declarations
double distancia;
string nombre_linea;
object oLine, ldf, comparador, trafo, nodo, line, elemento, falla;
set sLines, objs, trafos, nodos, lines, elementos, fallas;
int numero_lineas, i;
string class, attributes, attributest, attributesn, nameattributes, nameattrib
object obj, oval;
double dval;
int error, pos, t, row, col, rowt, colt, rown, coln, maxRow, maxCol;
!export definition
class = 'ElmLne';
attributes = 'loc_name, typ_id, bus1, bus2, dline, c:loading, m:P:bus2, m:Q:bus2';
attributest = 'loc_name, m:P:bush, m:I:bush, m:I:bus1';
attributesn = 'loc_name, m:u, m:U';
nameattributes = 'Nombre_Linea, Tipo_Linea, Bus1, Bus2, Distancia, Carga, P (MW)';
nameattributesn = 'Nombre_Nodo, Voltaje_pu, Voltaje_kV';
nameattributest = 'Nombre_Trafo, Potencia_Activa_Alta, Corriente_Alta, Corri
error = xlStart();
if(error){
    Error('Unable to start MS Excel application');
}

!get decimal separator and buil number format used here
sep = xlGetDecimalSeparator();
numberFormat = sprintf('0%s000', sep);

!create new workbook
```

MOD OVR READ Ln 210, Col 15

# En la casilla program text escribir el siguiente código (continuación):

```
Program text

!create new workbook
xlNewWorkbook();
xlSetWorksheetName(1, 'Lineas');
xlSetWorksheetName(2, 'Trafos');
xlSetWorksheetName(3, 'Nodos');

!iterate over attributes and write header row
sLines =AllRelevant(class);
lines =AllRelevant('*.ElmLne');
trafos = AllRelevant('*.ElmTr2');
nodos = AllRelevant('*.ElmTerm');
ldf = GetCaseObject('ComLdf');

Lineas.Clear();
Distancias.Clear();

row = 1;
rowt=1;
rown = 1;
i=0;

for(oLine = lines.First(); oLine; oLine = lines.Next()){
i+=1;
oLine:outserv = 1;
ldf.Execute();
xlActivateWorksheet(1);
xlSetFillColor(1, row, 4, row, 255, 0, 0); !red
xlSetValue(1, row, 'Contingencia');
```

Ln 210, Col 15

# En la casilla program text escribir el siguiente código (continuación):

```
xlSetValue(2, row, i);
xlSetValue(3, row, 'Desconexión del elemento llamado');
xlSetValue(4, row, oLine:loc_name);
row +=1;
col=1;
s = strtok(nameattributes, ',', pos, col);
while(pos > -1){
    xlSetValue(col, row, s);
    col+=1;
    s = strtok(nameattributes, ',', pos, col);
}
maxCol = col-1;
xlSetTextStyle(1, row, maxCol, row, 1); !bold
xlSetFillColor(1, row, maxCol, row, 255, 255, 150); !yellow
xlSetBorder(1, row, maxCol, row, 1, 1, 2, 0, 0, 0); !border at bottom
row+=1;
for(comparador=sLines.First();comparador; comparador=sLines.Next()){
    if(comparador = oLine){}
    else{
        !export data
        obj = comparador;
        col = 1;
        s = strtok(attributes, ',', pos, col);
        while(pos > -1){
            obj.GetVarType(s, type);
            t = strcmp(type, 'string');
            if (t=0){
                obj.GetVal(sval, s);
            }
        }
    }
}
```

# En la casilla program text escribir el siguiente código (continuación):

```
xlSetValue(col, row, sval);
}
t=strcmp(type, 'object');
if (t=0){
    obj.GetVal(oval, s);
    sval = oval:loc_name;
    xlSetValue(col, row, sval);
}
t=strcmp(type, 'double');
if(t=0){
    obj.GetVal(dval, s);
    xlSetValue(col, row, dval);
    xlSetNumberFormat(col, row, numberFormat);
}
col+=1;
s = strtok(attributes, ',', pos, col);
}
row+=1;
}
}
row+=1;

xlActivateWorksheet(2);
xlSetFillColor(1, rowt, 4, rowt, 255, 0, 0); !red
xlSetValue(1, rowt, 'Contingencia');
xlSetValue(2, rowt, i);
xlSetValue(3, rowt, 'Desconexión del elemento llamado');
xlSetValue(4, rowt, oLine:loc_name);
```

Ln 210, Col 15



# En la casilla program text escribir el siguiente código (continuación):

```
rowt +=1;
colt =1;
s = strtok(nameattributest, ',', pos, colt);
while(pos > -1){
    xlSetValue(colt, rowt, s);
    colt+=1;
    s = strtok(nameattributest, ',', pos, colt);
}
maxCol = colt-1;
xlSetTextStyle(1, rowt, maxCol, rowt, 1); !bold
xlSetFillColor(1, rowt, maxCol, rowt, 255, 255, 150); !yellow
xlSetBorder(1, rowt, maxCol, rowt, 1, 1, 2, 0, 0, 0); !border at bottom
rowt+=1;
for(trafo=trafos.First();trafo; trafo=trafos.Next()){
    if(trafo = oLine){}
    else{
        !export data
        obj = trafo;
        colt = 1;
        s = strtok(attributest, ',', pos, colt);
        while(pos > -1){
            obj.GetVarType(s, type);
            t = strcmp(type, 'string');
            if (t=0){
                obj.GetVal(sval, s);
                xlSetValue(colt, rowt, sval);
            }
            t=strcmp(type, 'object');
        }
    }
}
```

Ln 210, Col 15

# En la casilla program text escribir el siguiente código (continuación):

```
if (t=0){
    obj.GetVal (oval,s);
    sval = oval:loc_name;
    xlSetValue (colt,rowt,sval);
}
t=strcmp (type, 'double');
if(t=0){
    obj.GetVal (dval,s);
    xlSetValue (colt,rowt,dval);
    xlSetNumberFormat (colt,rowt,numberFormat);
}
colt+=1;
s = strtok (attributest, ',', pos, colt);
}
rowt+=1;
}
}
rowt+=1;

xlActivateWorksheet (3);
xlSetFillColor (1, rown, 4, rown, 255, 0, 0); !red
xlSetValue (1, rown, 'Contingencia');
xlSetValue (2, rown, i);
xlSetValue (3, rown, 'Desconexión del elemento llamado');
xlSetValue (4, rown, oLine:loc_name);
rown +=1;
coln=1;
s = strtok (nameattributesn, ',', pos, coln);
```

# En la casilla program text escribir el siguiente código (continuación):

```
while (pos > -1) {
    xlSetValue (coln, rown, s);
    coln+=1;
    s = strtok (nameattributesn, ',', pos, coln);
}
maxCol = coln-1;
xlSetTextStyle (1, rown, maxCol, rown, 1); !bold
xlSetFillColor (1, rown, maxCol, rown, 255, 255, 150); !yellow
xlSetBorder (1, rown, maxCol, rown, 1, 1, 2, 0, 0, 0); !border at bottom
rown+=1;
for (nodo=nodos.First (); nodo; nodo=nodos.Next ()) {
    if (nodo = oLine) {}
    else {
        !export data
        obj = nodo;
        coln = 1;
        s = strtok (attributesn, ',', pos, coln);
        while (pos > -1) {
            obj.GetVarType (s, type);
            t = strcmp (type, 'string');
            if (t=0) {
                obj.GetVal (sval, s);
                xlSetValue (coln, rown, sval);
            }
            t=strcmp (type, 'object');
            if (t=0) {
                obj.GetVal (oval, s);
                sval = oval:loc_name;
            }
        }
    }
}
```

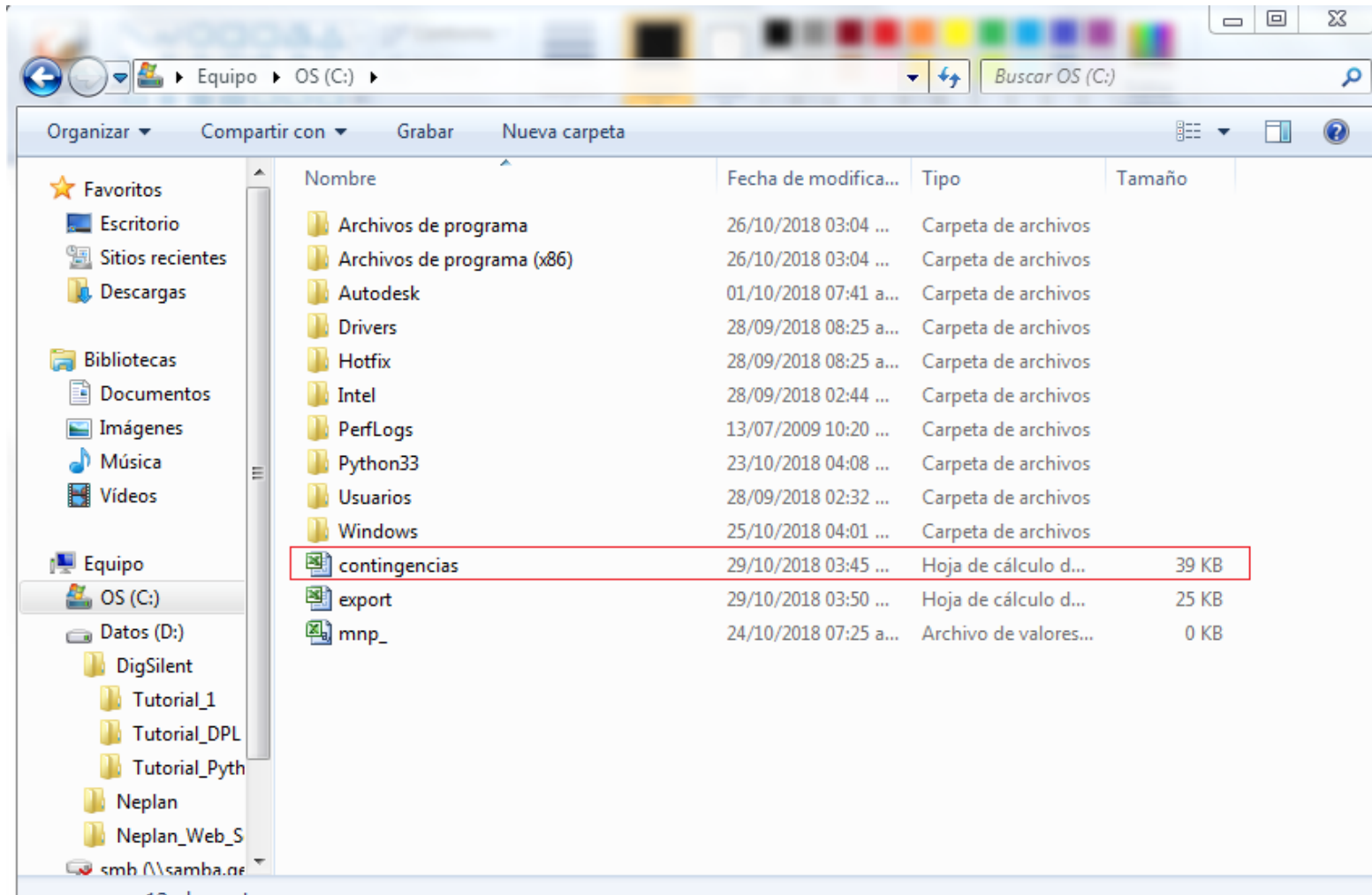
Ln 210, Col 15

# En la casilla program text escribir el siguiente código (continuación):

```
        xlSetValue (coln, rown, sval);
    }
    t=strcmp (type, 'double');
    if (t=0) {
        obj.GetVal (dval, s);
        xlSetValue (coln, rown, dval);
        xlSetNumberFormat (coln, rown, numberFormat);
    }
    coln+=1;
    s = strtok (attributesn, ',', pos, coln);
}
rown+=1;
}
}
rown+=1;
oLine:outserv = 0;
}
!save and exit
error = xlSaveWorkbookAs ('c:\contingencias.xls');
if (error) {
    Error ('Workbook could not be saved');
}
xlTerminate ();
```

Ln 195, Col 74

# Archivo de Excel con resultados:



Al ejecutar el script se obtiene el archivo en el disco local C con el nombre: Contingencias.

# Al abrir el archivo se obtiene:

contingencias [Modo de compatibilidad] - Microsoft Excel

Archivo Inicio Insertar Diseño de página Fórmulas Datos Revisar Vista Equipo

Calibri 11 Fuente Alineación Número Estilos

Formato condicional Dar formato como tabla Estilos de celda Insertar Eliminar Formato Celdas

Autosuma Rellenar Borrar Ordenar y filtrar Buscar y seleccionar Modificar

A1 Contingencia

1	Contingenci. 1 Desconexiór Line1-3							
2	Nombre_Lin	Tipo_Linea	Bus1	Bus2	Distancia	Carga	P(MW)	Q(MVr)
3	Line1-4	N2XS(F)2Y 1	Cub_2	Cub_1	1,000	45,301	3,802	4,297
4	Line2-3	N2XS(F)2Y 1	Cub_3	Cub_3	1,000	6,941	-0,698	-0,531
5	Line3-4	N2XS(F)2Y 1	Cub_4	Cub_4	1,000	40,257	-3,066	-4,064
6	Line1-2	N2XS(F)2Y 1	Ensayo	Ensayo1	25,000	6,169	0,304	-0,013
7								
8	Contingenci. 2 Desconexiór Line1-4							
9	Nombre_Lin	Tipo_Linea	Bus1	Bus2	Distancia	Carga	P(MW)	Q(MVr)
10	Line1-3	N2XS(F)2Y 1	Cub_2	Cub_1	20,000	29,841	2,414	2,345
11	Line2-3	N2XS(F)2Y 1	Cub_3	Cub_3	1,000	11,350	0,820	1,098
12	Line3-4	N2XS(F)2Y 1	Cub_4	Cub_4	1,000	6,239	0,714	0,247
13	Line1-2	N2XS(F)2Y 1	Ensayo	Ensayo1	25,000	24,213	1,897	1,697
14								
15	Contingenci. 3 Desconexiór Line2-3							
16	Nombre_Lin	Tipo_Linea	Bus1	Bus2	Distancia	Carga	P(MW)	Q(MVr)
17	Line1-3	N2XS(F)2Y 1	Cub_2	Cub_1	20,000	5,120	0,249	0,012
18	Line1-4	N2XS(F)2Y 1	Cub_2	Cub_1	1,000	33,726	2,843	3,180
19	Line3-4	N2XS(F)2Y 1	Cub_4	Cub_4	1,000	28,716	-2,118	-2,956
20	Line1-2	N2XS(F)2Y 1	Ensayo	Ensayo1	25,000	12,811	1,018	0,539
21								
22	Contingenci. 4 Desconexiór Line3-4							
23	Nombre_Lin	Tipo_Linea	Bus1	Bus2	Distancia	Carga	P(MW)	Q(MVr)
24	Line1-3	N2XS(F)2Y 1	Cub_2	Cub_1	20,000	26,695	1,984	2,171
25	Line1-4	N2XS(F)2Y 1	Cub_2	Cub_1	1,000	6,005	0,713	0,243

Lineas Trafos Nodos

Listo 100%

ES 04:27 p.m. 29/10/2018

# Datos de transformadores:

Contingencia 1 Desconexión del elemento llamado Line1-3			
Nombre_Tra	Potencia_Activa_Alta	Corriente_Alta	Corriente_Baja
Reductor_Tra	2,367	0,124	6,197
Reductor_Tra	0,713	0,022	1,099
Contingencia 2 Desconexión del elemento llamado Line1-4			
Nombre_Tra	Potencia_Activa_Alta	Corriente_Alta	Corriente_Baja
Reductor_Tra	2,419	0,133	6,622
Reductor_Tra	0,714	0,023	1,139
Contingencia 3 Desconexión del elemento llamado Line2-3			
Nombre_Tra	Potencia_Activa_Alta	Corriente_Alta	Corriente_Baja
Reductor_Tra	2,365	0,124	6,179
Reductor_Tra	0,713	0,022	1,098
Contingencia 4 Desconexión del elemento llamado Line3-4			
Nombre_Tra	Potencia_Activa_Alta	Corriente_Alta	Corriente_Baja
Reductor_Tra	2,411	0,131	6,554
Reductor_Tra	0,713	0,022	1,096
Contingencia 5 Desconexión del elemento llamado Line1-2			
Nombre_Tra	Potencia_Activa_Alta	Corriente_Alta	Corriente_Baja
Reductor_Tra	2,367	0,124	6,196
Reductor_Tra	0,713	0,022	1,099

# Datos de Nodos:

Contingenci	1 Desconexión del elemento llamado Line1-3	
Nombre_No	Voltaje_pu	Voltaje_kV
T1.4	1,000	11,547
T1.3	1,000	11,547
4	1,000	11,547
2	1,000	11,547
3	1,000	11,547
T2.4	1,000	11,547
T2.3	1,000	11,547
T2.1	1,000	11,547
T2.2	1,000	11,547
1(1)	1,000	11,547
T2.1(1)	1,000	11,547
T2.3(1)	1,000	11,547
5	1,000	11,547
T2.3(2)	1,000	11,547
T2.5	1,000	11,547
Node_1	1,000	11,547
Node_2	0,994	11,480
Node_3	0,995	11,485
Node_4	0,997	11,514
Node_6	0,807	0,224
Node_5	0,621	0,172
LineaCreada	0,000	0,000
Nodo	0,000	0,000