

CIMAT

GiD, brevísimo tutorial

MSc. Miguel Vargas-Félix

miguelvargas@cimat.mx

<http://www.cimat.mx/~miguelvargas>

Contenido

Descripción

Pre-proceso

Creación de un problem-type

Formato de archivos de resultados para GiD

Formato de archivo de malla

Post-proceso

¿Preguntas?

Descripción

Sirve como **pre-procesador**:

- Diseñar geometrías/dominios de problemas de elemento finito en 1D, 2D y 3D.
- Asignar materiales.
- Definir condiciones iniciales y de frontera.
- Generar mallas para el dominio con diferentes tipos de elementos: líneas, triángulos, cuadriláteros, tetraedros, hexaedros y prismas.
- Crear interfaces de usuario para declarar materiales, condiciones, parámetros, etc.
- Genera archivos de datos con la información anterior.
- Puede llamar programas externos (“solvers”).

Sirve como **post-procesador**:

- Puede leer datos de resultados generados por los “solvers”.
- Visualizar resultados: desplazamientos, campos escalares, campos vectoriales, flujos.
- Generar animaciones.
- Efectos de visualización: cortes, texturas en superficies, etc.

Información sobre el GiD

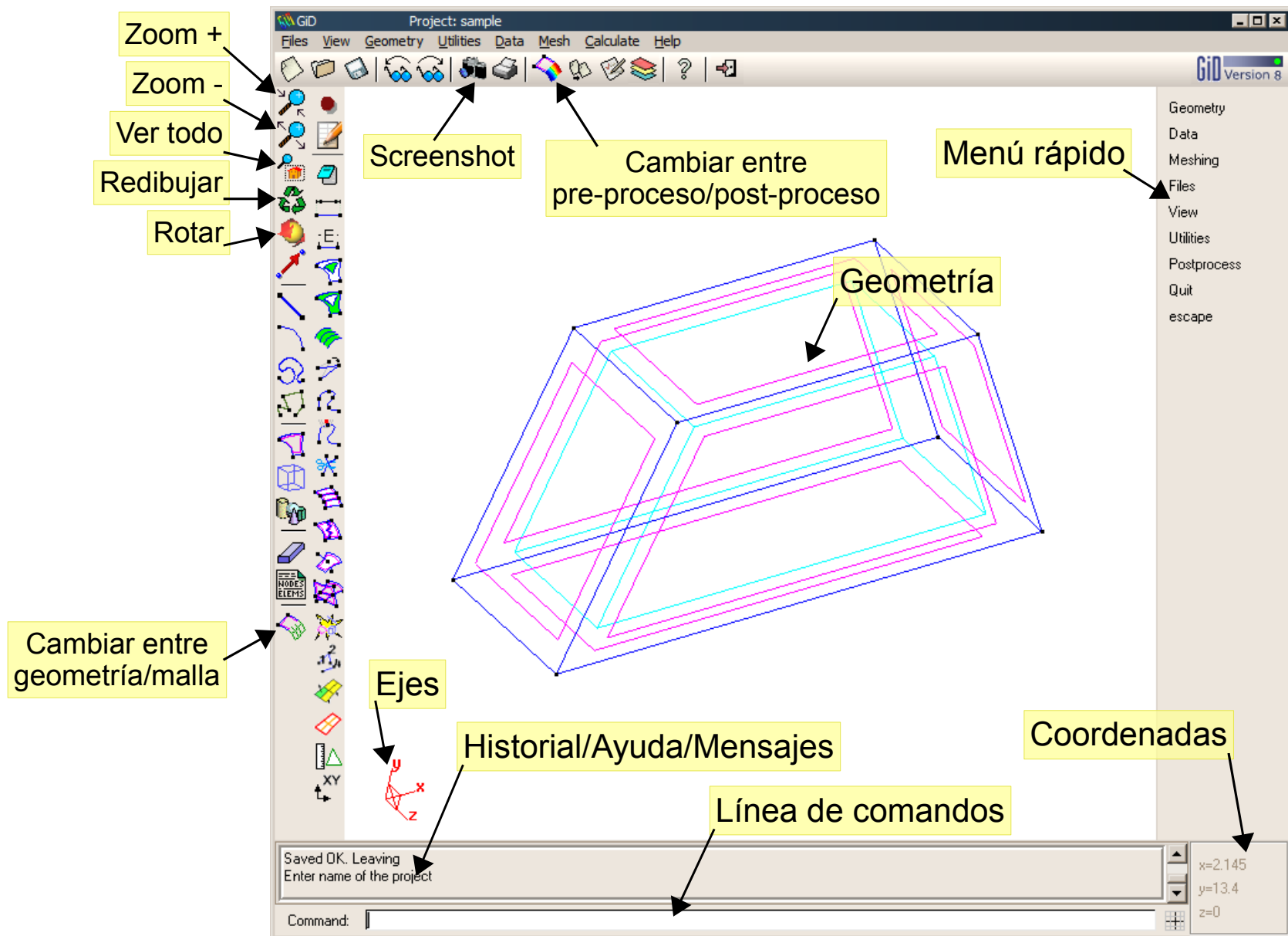
El GiD:

- Es un programa comercial.
- Funciona en computadoras i386, x86-64 con sistemas operativos
 - Windows
 - GNU/Linux
 - Mac OS
- Se puede usar sin licencia, pero tiene un límites en cuanto al tamaño de la geometría/malla.
- Se ofrecen licencias por un mes gratis, renovable. Esto permite utilizar el GiD sin limitaciones.

Página oficial: <http://gid.cimne.upc.es>

- Documentación.
- Descargar el GiD.
- Obtener licencias.
- Foros.

Pre-proceso



Geometría

Se diseña de forma jerárquica:

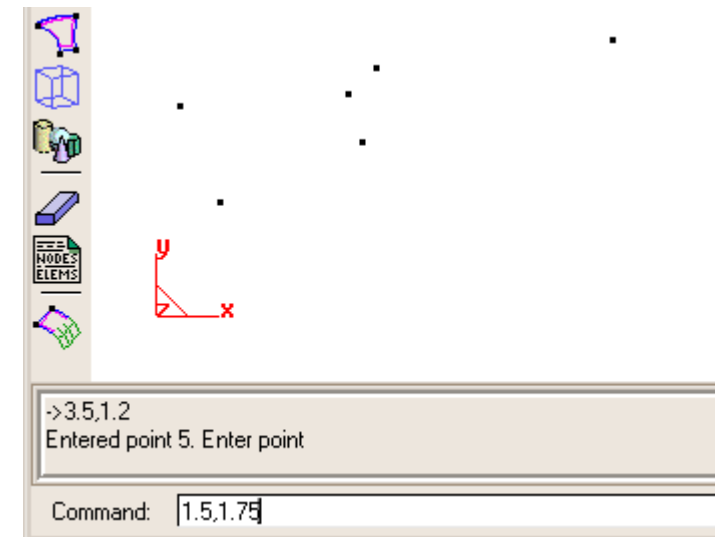
- Puntos
 - Líneas (conectando puntos)
 - Superficies (agrupando líneas)
 - Volúmenes (agrupando superficies)

Para crear un punto por medio del mouse:

- Desde el menú rápido: **Geometry/Create/Point**
- Hacer “click” en el área de diseño.
- Introducir otro punto o [esc] para terminar.

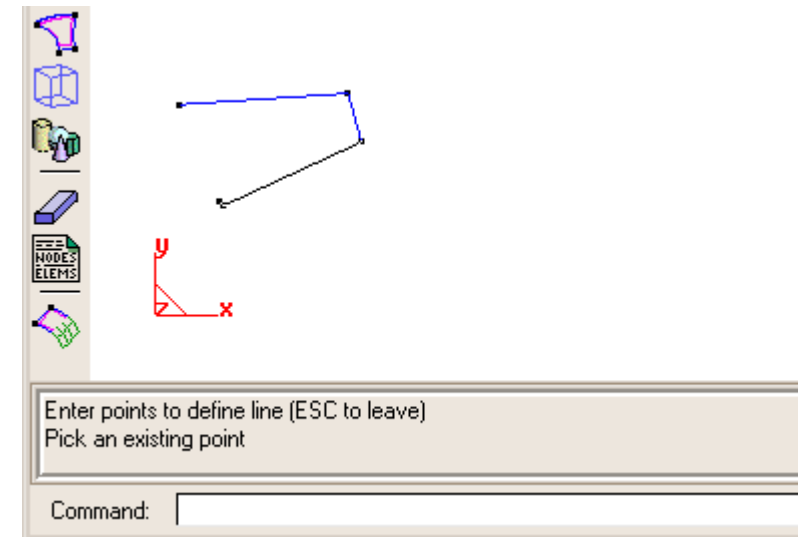
Para crear un punto con coordenadas:

- Desde el menú rápido: **Geometry/Create/Point**
- Introducir en la línea de comandos las coordenada “x,y” o “x,y,z” y [enter].
- Introducir otra coordenada o [esc] para terminar.



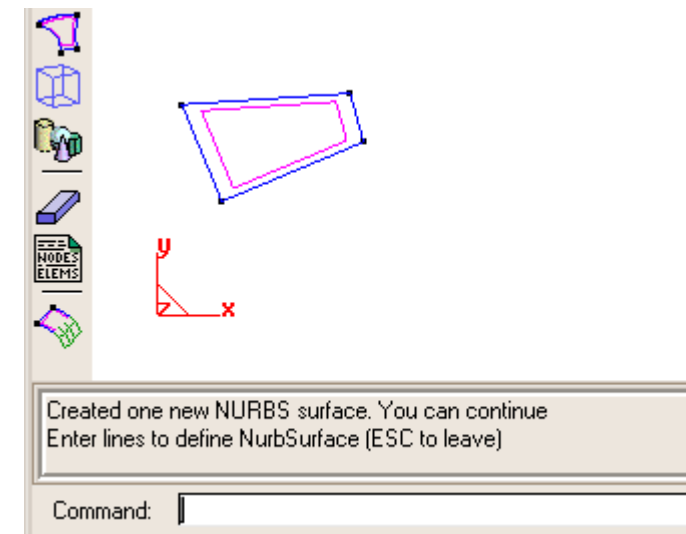
Para crear una línea:

- Desde el menú rápido: **Geometry/Create/Line/Join**
- Seleccionar un punto.
- Seleccionar más puntos o [esc] para terminar.



Para crear una superficie:

- Desde el menú rápido: **Geometry/Create/NurbsSurface**
- Seleccionar las líneas que encierran una superficie.
- Definir otra superficie o [esc] para terminar.

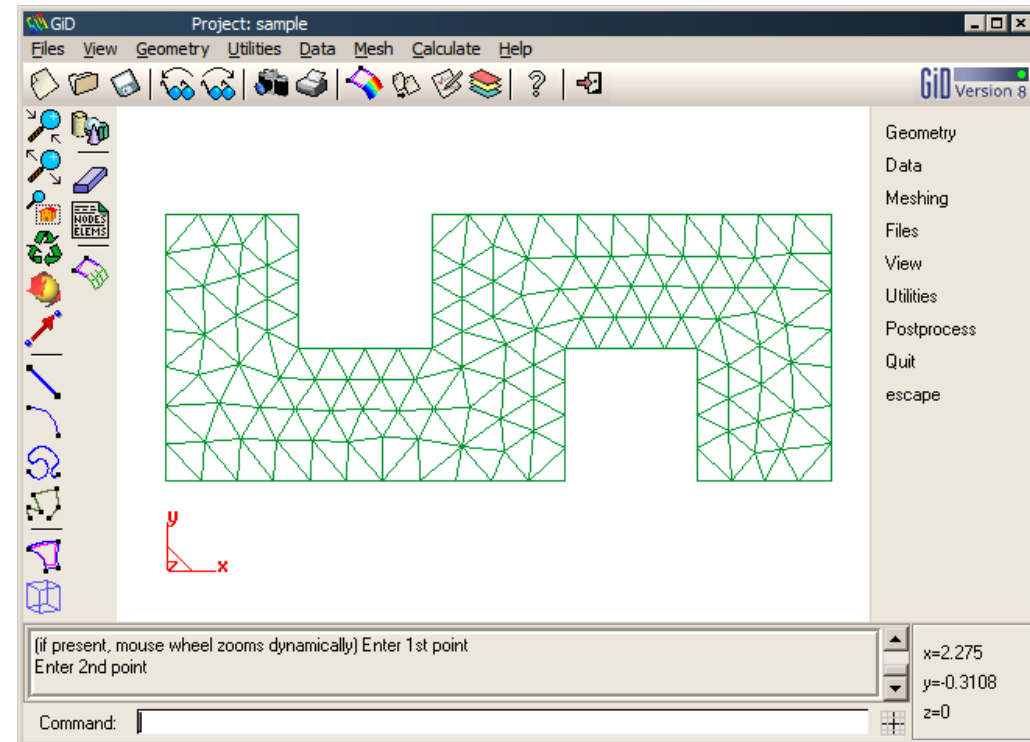
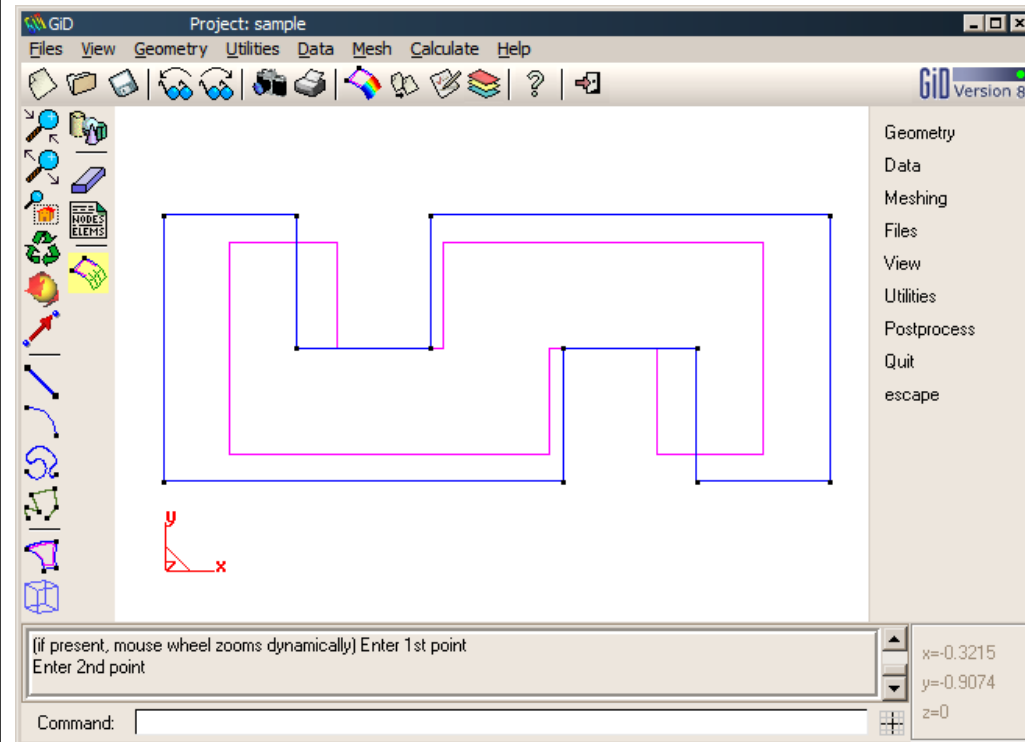


Para crear un volumen:

- Desde el menú rápido: **Geometry/Create/Volume**
- Seleccionar superficies encierran un volumen.
- Definir otro volumen o [esc] para terminar.

Mallado

Desde el menú rápido: **Geometry/Generate**



Creación de un *problem-type*

Los *problem-types* permiten crear interfaces de usuario para problemas específicos de elemento finito.

Permiten capturar datos para después exportarlos en archivos de texto.

Entre otras cosas, se pueden definir:

- Condiciones iniciales
- Condiciones de frontera
- Parámetros para los “solvers”
- Materiales
- Unidades de medida
- Intervalos para soluciones en varios pasos

Se definen mediante archivos de texto.

Vamos a crear un *problem-type* para la ecuación de calor.

1. Crear directorio para el *problem-type*

Vamos a llamar al *problem-type* “Calor”.

El primer paso es crear un directorio “Calor.gid” dentro del directorio “problemtypes” en el directorio instalación de GiD:

```
"C:\Archivos de programa\GiD\problemtypes\Calor.gid"
```

Es importante mencionar que la extensión del directorio debe ser “.gid”.

2. Archivo de datos del problema

Crear un archivo de texto llamado:

`"C:\Archivos de programa\GiD\problemtypes\Calor.gid\Calor.prb"`

Con el siguiente contenido:

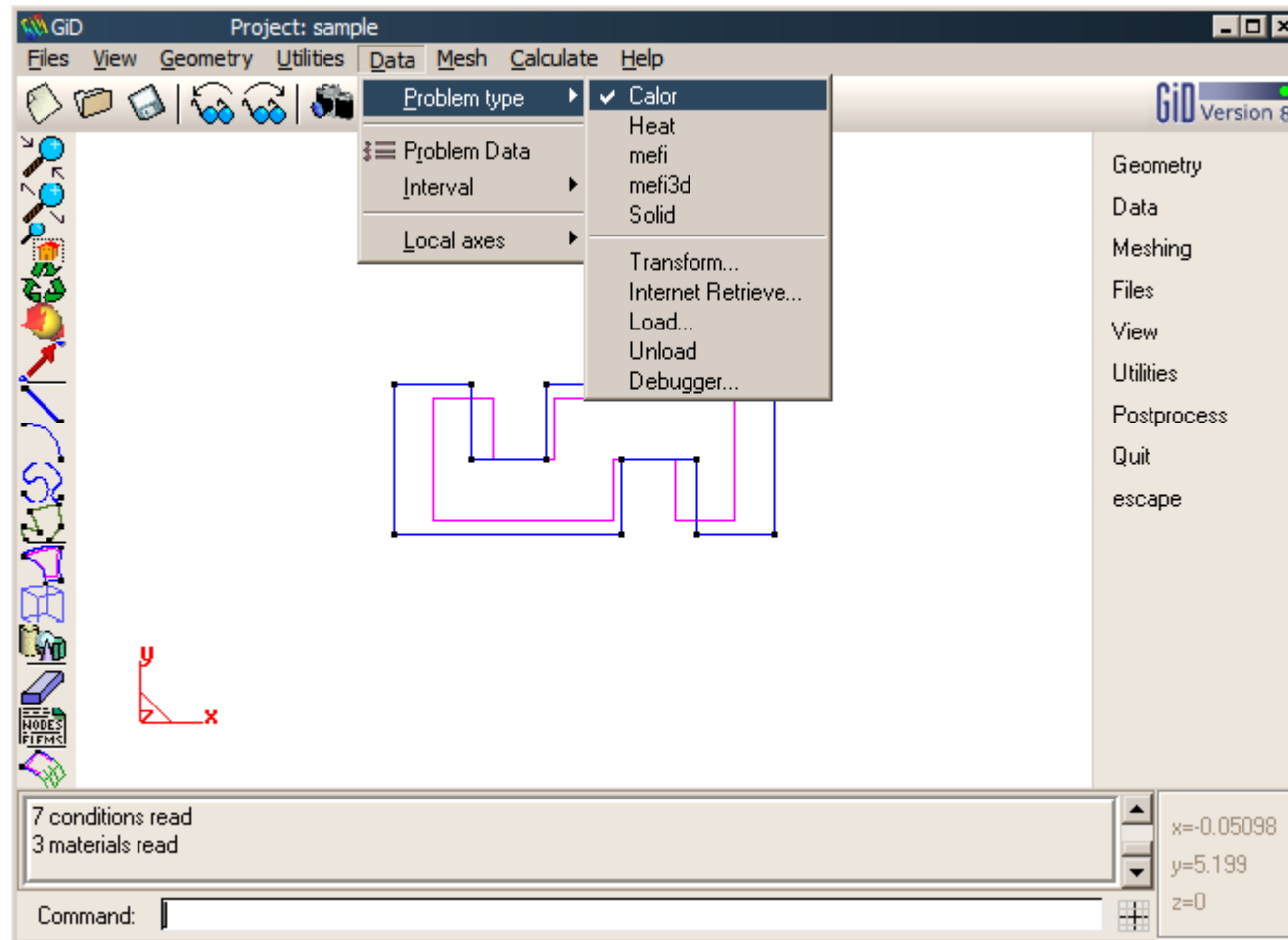
```
problem data
question: Problem_name
value:
question: Solver_type#CB#(Conjugate_gradient,Cholesky,Gauss_Jordan)
value: Conjugate_gradient
question: Tolerance
value: 1e-5
question: Max_steps
value: 10000
end problem data
```

Permite definir un problema el cual tiene varias preguntas:

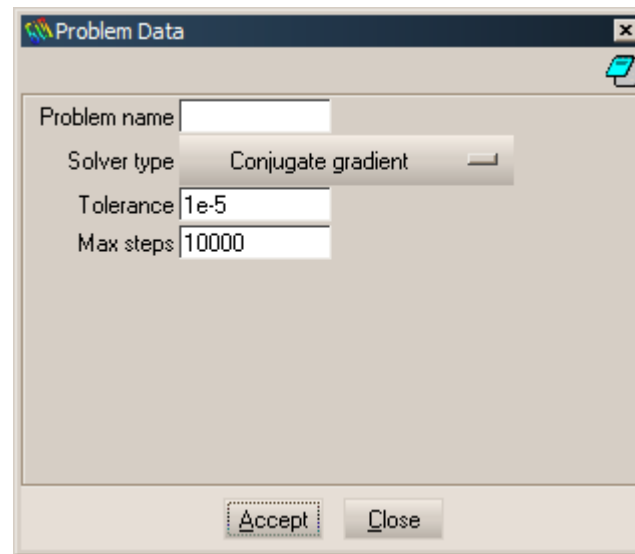
Problem_name, *Solver_type*, *Tolerance*, *Max_steps*.

Es necesario cerrar y volver a ejecutar el GiD para que cargue el problema.

Podemos ahora crear una geometría o leer una ya almacenada. Para asignarle a esta geometría nuestro tipo de problema, es con el menú: **Data/Problem type/Calor**



Ahora, seleccionando el menú: **Data/Problem Data**, aparecerá un cuadro de diálogo con los campos que definimos anteriormente. Nos permite entrar datos generales sobre el problema.



Veamos de nuevo el archivo de datos del problema

```

Encabezado → problem data
Pregunta { question: Problem_name
Pregunta { value:
Pregunta { question: Solver_type#CB#(Conjugate_gradient,Cholesky,Gauss_Jordan)
Pregunta { value: Conjugate_gradient
Pregunta { question: Tolerance
Pregunta { value: 1e-5
Pregunta { question: Max_steps
Cola → end problem data

```

Diagram illustrating the structure of the problem data file with annotations:

- Nombre de la pregunta** points to `question: Problem_name`.
- Campo con varias opciones** points to `Solver_type#CB#(Conjugate_gradient,Cholesky,Gauss_Jordan)`.
- Opción por defecto** points to `Conjugate_gradient` in the `value:` line.
- Valor por defecto** points to `10000` in the `value:` line.

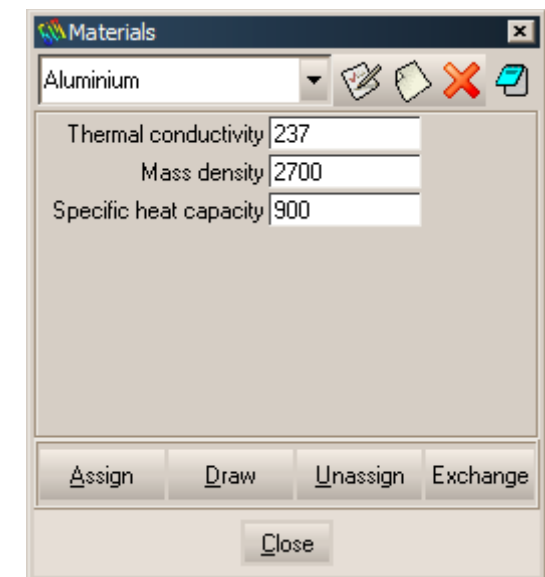
3. Archivo de materiales

Crear un archivo de texto llamado:

"C:\Archivos de programa\GiD\problemtypes\Calor.gid\Calor.mat"

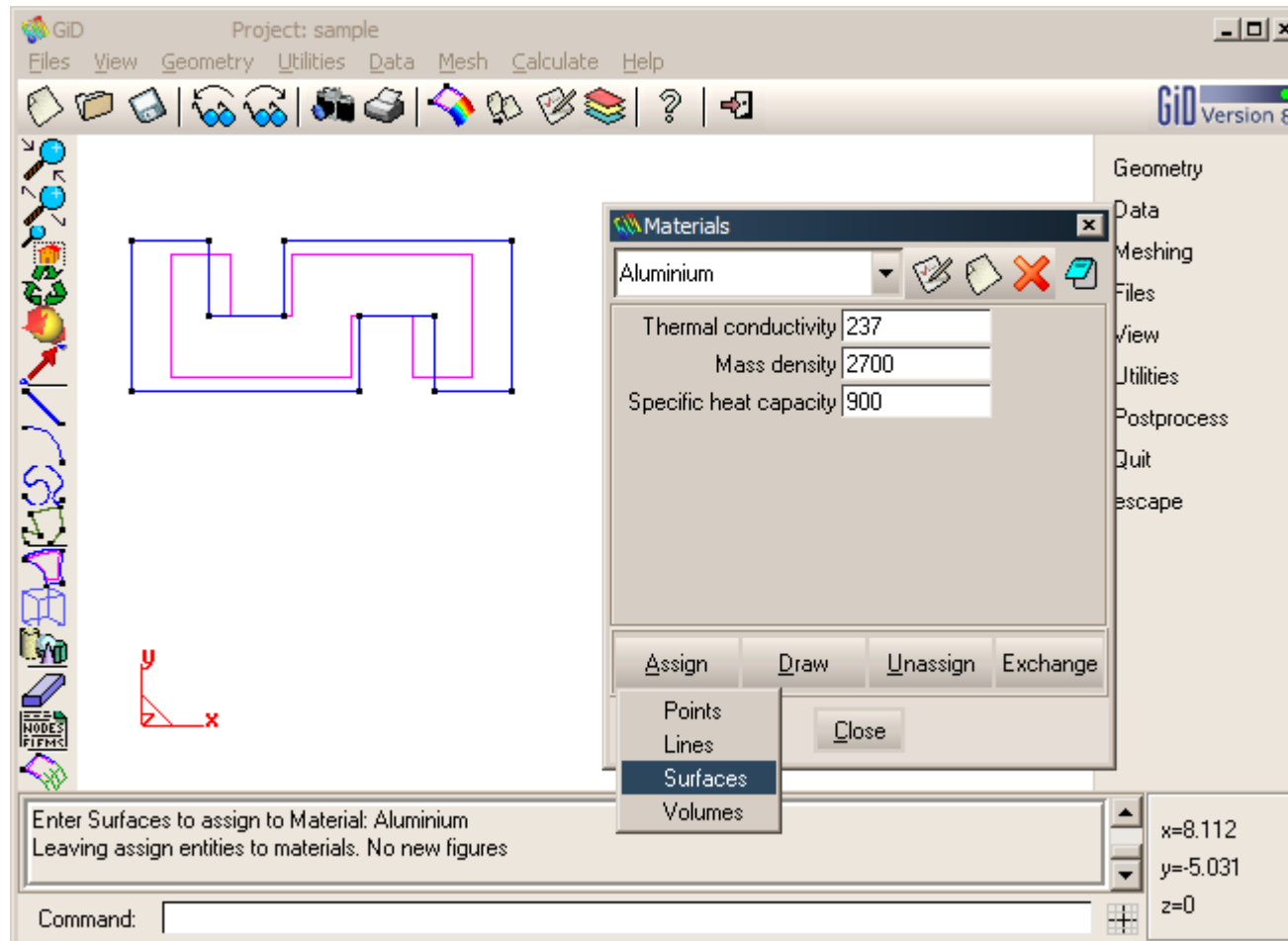
Con el siguiente contenido.

| | | | |
|------------|---|----------------------------------|-----------------------|
| | | Número de material | Nombre del material |
| Encabezado | → | number: 1 | material: Aluminium |
| Pregunta | { | question: Thermal_conductivity | |
| | | value: 237 | |
| Pregunta | { | question: Mass_density | Nombre de la pregunta |
| | | value: 2700 | |
| Pregunta | { | question: Specific_heat_capacity | |
| | | value: 900 | Valor por defecto |
| Cola | → | end material | |
| Encabezado | → | number: 2 | material: Iron |
| Pregunta | { | question: Thermal_conductivity | |
| | | value: 80.2 | |
| Pregunta | { | question: Mass_density | |
| | | value: 7874 | |
| Pregunta | { | question: Specific_heat_capacity | |
| | | value: 440 | |
| Cola | → | end material | |



Para acceder los materiales, en el menú **Data/Materials**.

Podemos ahora asignar un material a una superficie.



Para ver materiales asignados, es con el botón **Draw**.

4. Archivo de condiciones

Crear un archivo de texto llamado:

```
"C:\Archivos de programa\GiD\problemtypes\Calor.gid\Calor.cnd"
```

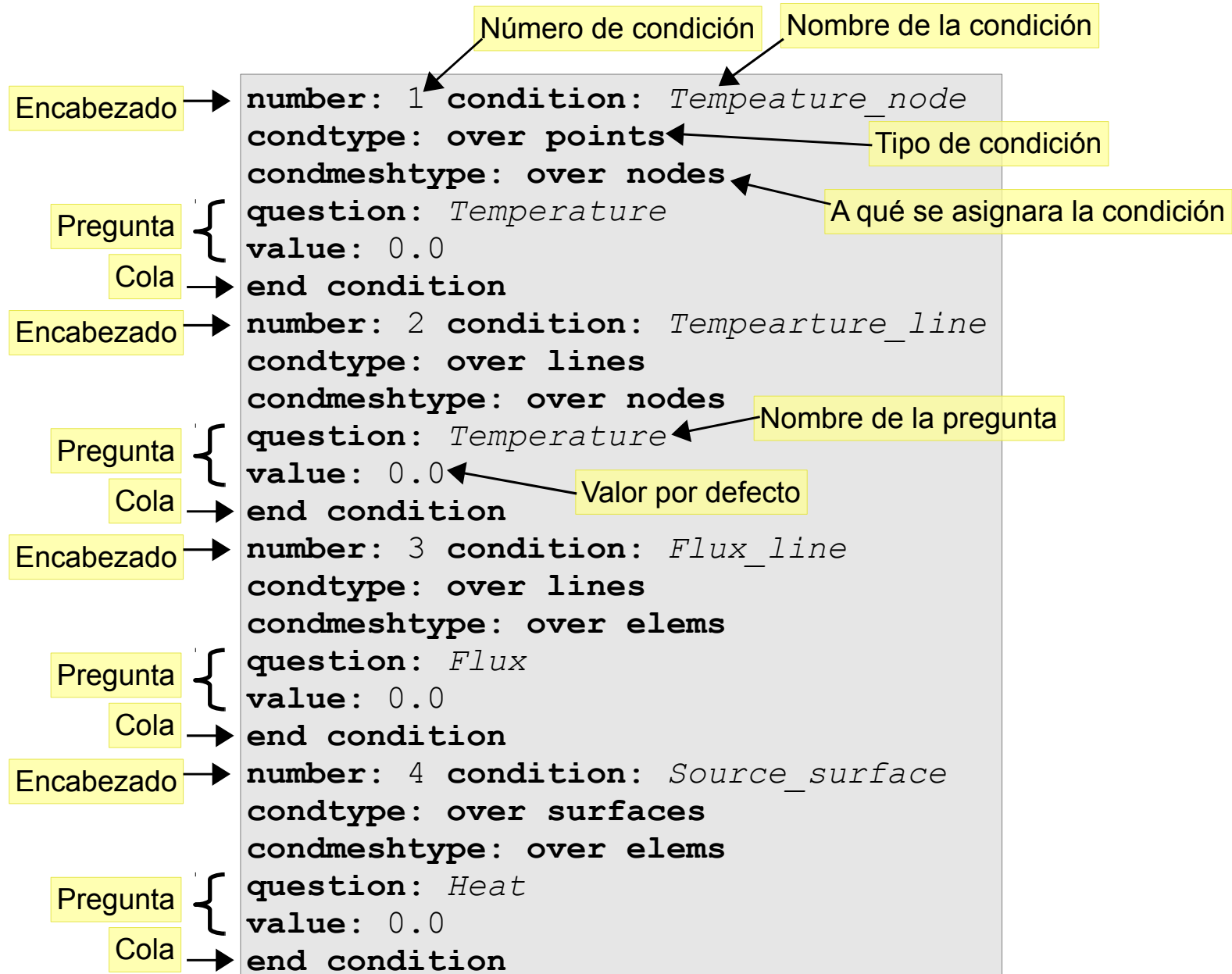
Sirve para definir condiciones de frontera.

Las condiciones se pueden asignar a:

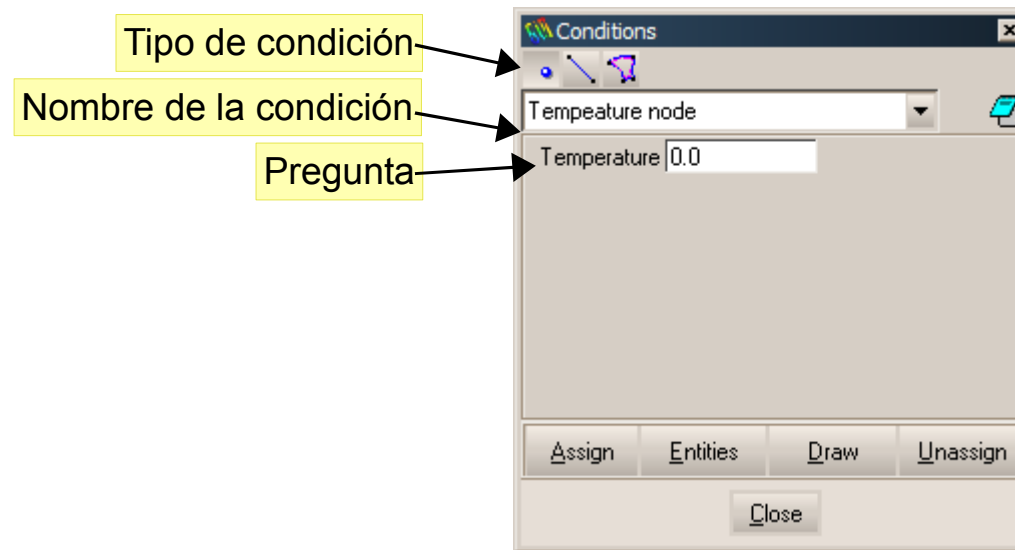
- Puntos
- Líneas
- Superficies
- Volúmenes

Estas condiciones pueden traducirse condiciones aplicadas a elementos o a nodos.

El siguiente archivo define condiciones para nuestro problema



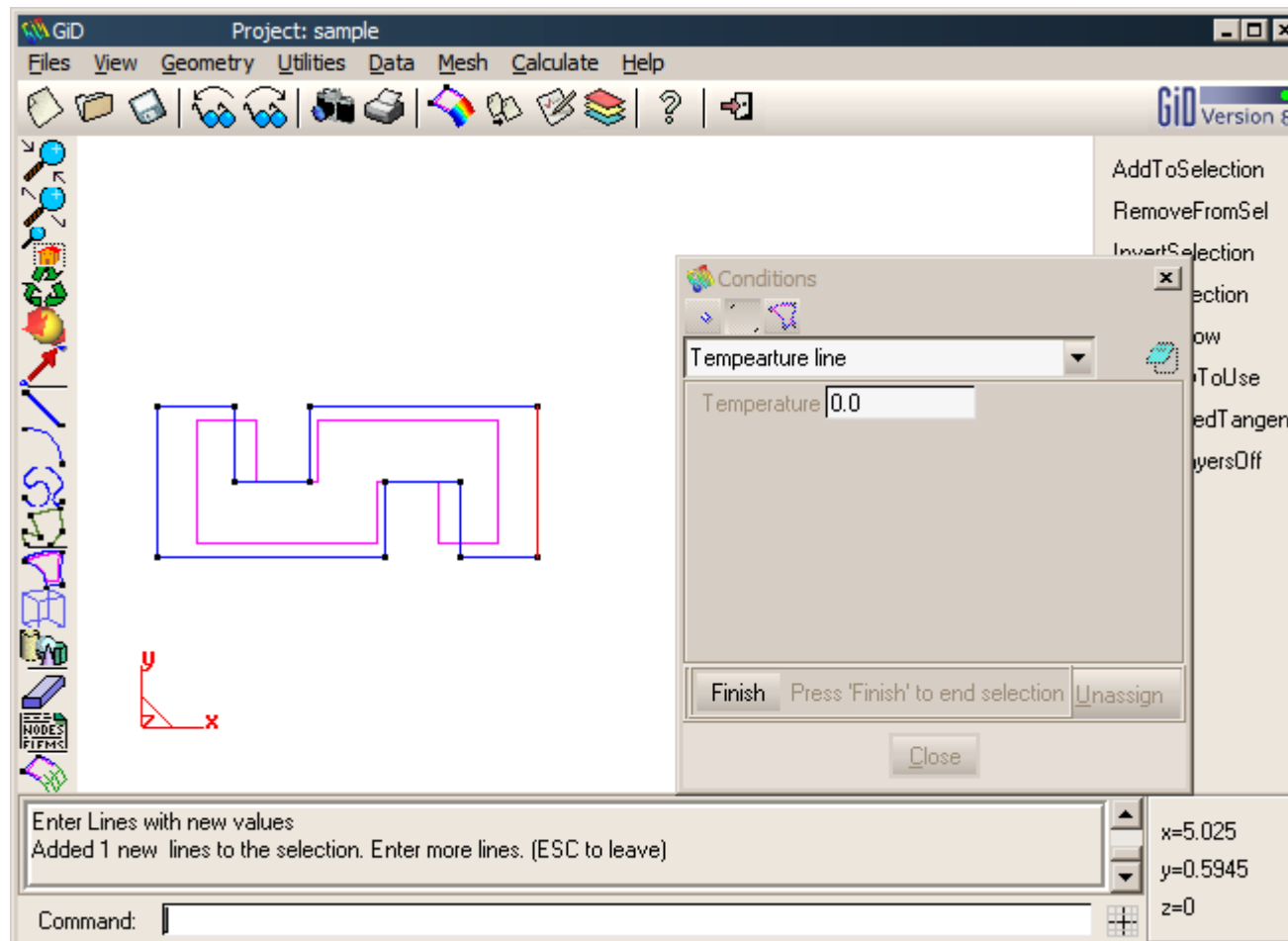
Ésto se traduce a un cuadro de diálogo que permite asignar condiciones directamente en la geometría.



Éste se puede acceder desde el menú **Data/Conditions**.

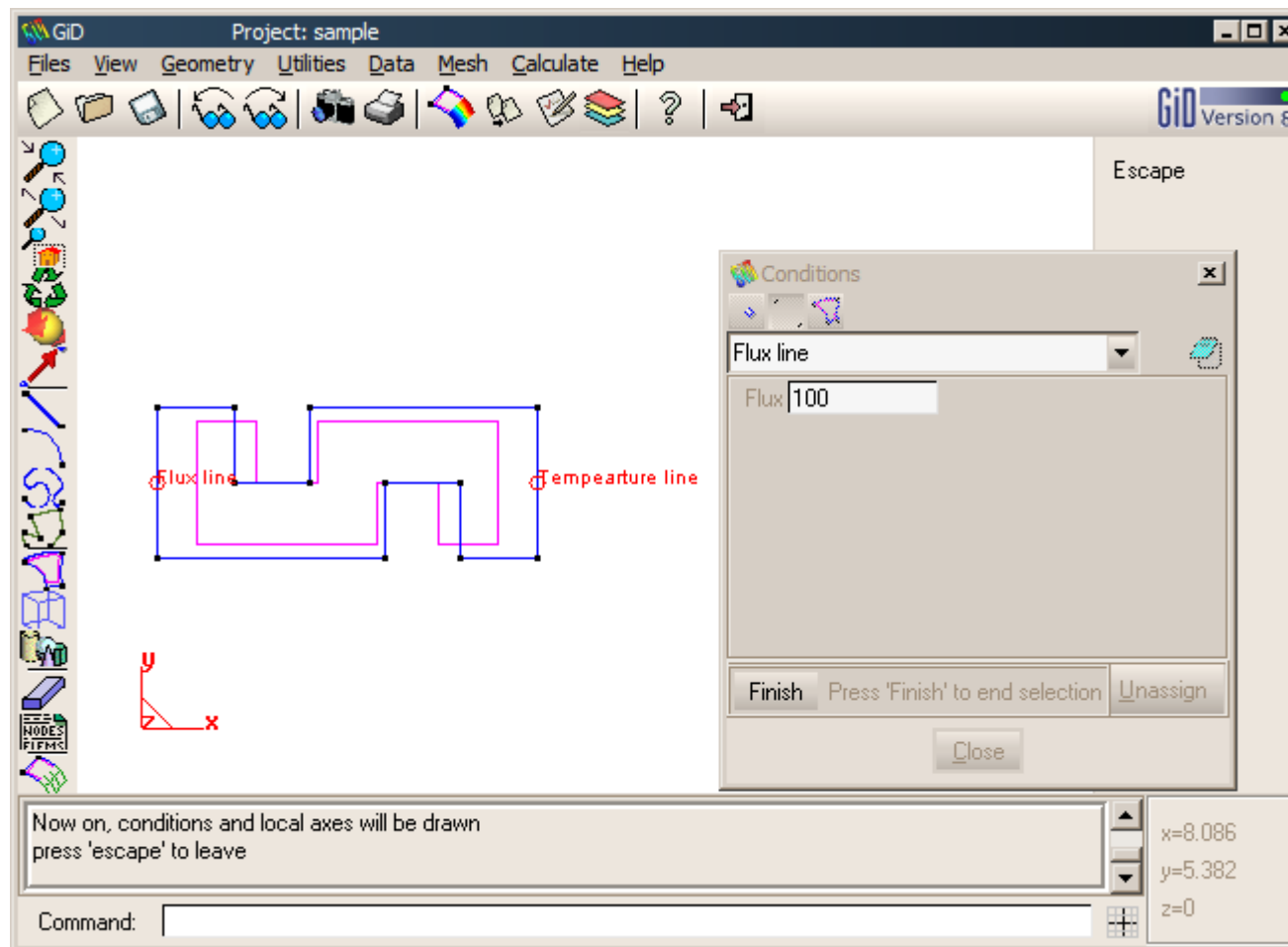
Para asignar condiciones en el dominio:

- Se elige el tipo de condición.
- Se llenan los valores
- Hacer click en el botón **Assign**.
- Y se eligen las entidades (nodos, líneas, superficies, etc). Y se presiona **Finish**.



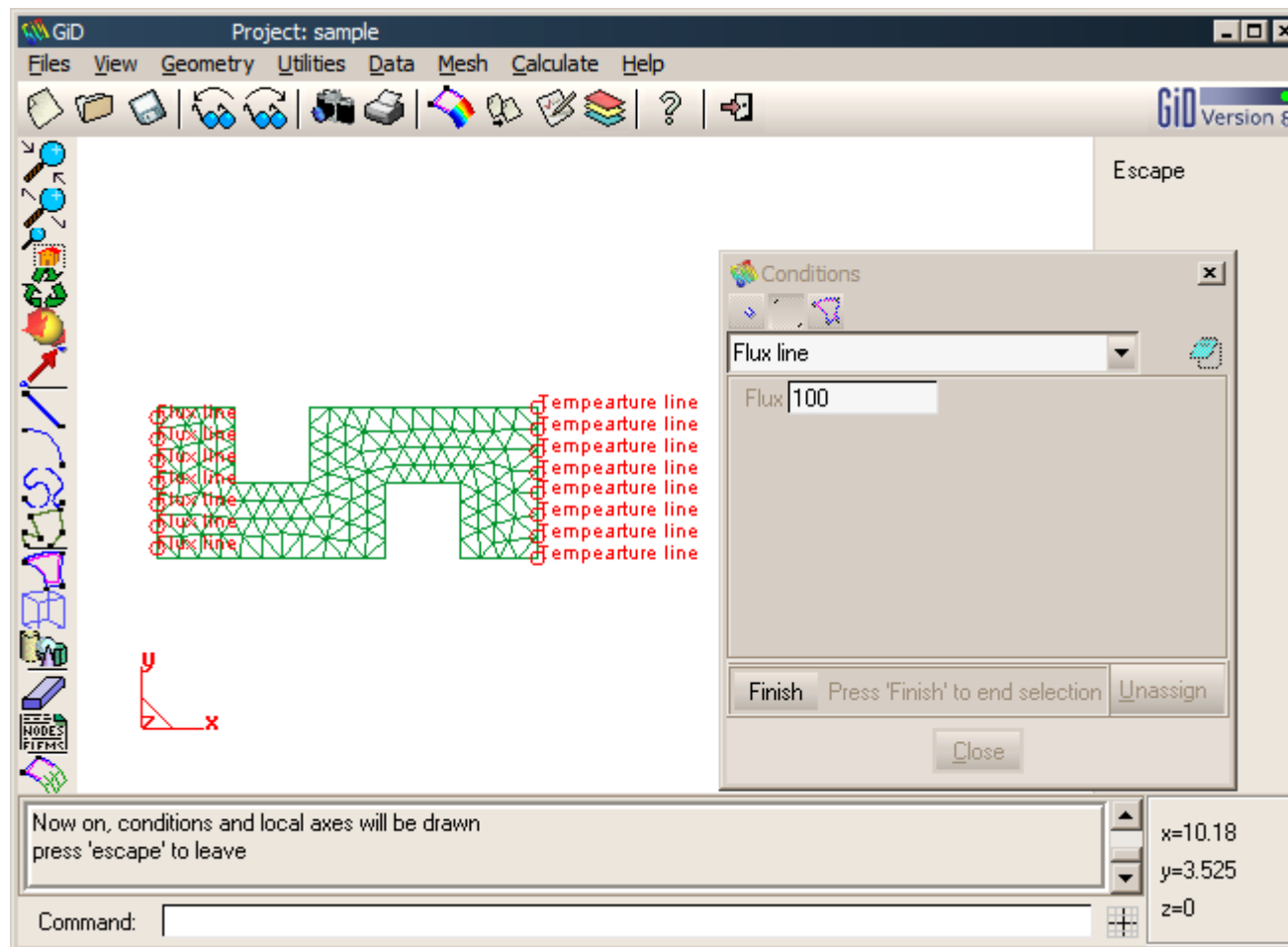
Para visualizar las condiciones en el dominio:

- En el diálogo de **Conditions**, desde el menú **Data/Conditions**.
- Hacer click en el botón **Draw/All conditions/Include local axes**.



Podemos ahora regenerar la malla y ver las condiciones en la malla:

- En el diálogo de **Conditions**, desde el menú **Data/Conditions**.
- Hacer click en el botón **Draw/All conditions/Include local axes**.



5. Archivo de pre-proceso

Crear un archivo de texto llamado:

```
"C:\Archivos de programa\GiD\problemtypes\Calor.gid\Calor.bas"
```

Es el formato con los que se generarán los datos de entrada del “solver”.

El formato es libre, completamente definido por el usuario.

Se pueden generar:

- Datos del programa
- Conectividades de los nodos
- Coordenadas de los nodos
- Materiales
- Condiciones

El GiD toma el formato del archivo de preproceso y lo rellena con los datos de la geometría y los parámetros que hemos agregado.

El siguiente es un formato ejemplo para el archivo de pre-proceso

```
#####
# Archivo de datos para Calor
#####

# Datos del problema #####
# Problem_name
*gendata(Problem_name)
# Solver_type
*gendata(Solver_type)
# Tolerance
*gendata(Tolerance,real)
# Max_steps
*gendata(Max_steps,int)

# Malla #####
# Dimension
*ndime
# Nodos por elemento
*nnode
# Numero de elementos
*nelem
# Listado de elementos
# Elemento Material node1 node2 ...
*loop elems
*elemsnum *elemsmat *elemsconec
*end elems

# Nodos #####
# Numero de nodos
*npoin
# Listado de nodos
# Nodo x1 x2 ...
*loop nodes
*nodesnum *nodescoord
*end nodes

# Materiales #####
# Numero de materiales
*nmats

# Listado de materiales
# Material Thermal_conductivity Mass_density Specific_heat
*loop materials
*matnum *matprop(Thermal_conductivity,real)
*matprop(Mass_density,real)
*matprop(Specific_heat_capacity,real)
*end

# Temperaturas #####
*Set Cond Tempeature_node *nodes
*Add Cond Tempearture_line *nodes
# Numero de condiciones
*condnumentities
# Listado de condiciones
# Nodo Temperature
*loop nodes *OnlyInCond
*NodesNum *cond(Temperature,real)
*end

# Flux #####
*Set Cond Flux_line *elems
# Numero de condiciones
*condnumentities
# Listado de condiciones
# Element Flux FaceNode1 FaceNode2 ...
*loop elems *OnlyInCond
*ElementsNum *cond(Flux,real) *globalnodes
*end

# Terminos fuente #####
*Set Cond Source_surface *elems
# Numero de condiciones
*condnumentities
# Listado de condiciones
# Element Heat
*loop elems *OnlyInCond
*ElementsNum *cond(Heat,real)
*end
```

Para llenar el archivo de pre-proceso es con el menú Calculate/Calculate.

Vamos a verl por partes como se llena este archivo.

En la primer parte del archivo ponemos los datos del problema. A la izquierda está el código del archivo formato, a la derecha el resultado:

```
#####
# Archivo de datos para Calor
#####

# Datos del problema #####
# Problem_name
*gendata(Problem_name)
# Solver_type
*gendata(Solver_type)
# Tolerance
*gendata(Tolerance, real)
# Max_steps
*gendata(Max_steps, int)
```

```
#####
# Archivo de datos para Calor
#####

# Datos del problema #####
# Problem_name
Ejemplo simple
# Solver_type
Cholesky
# Tolerance
1e-05
# Max_steps
10000
```

Las funciones de GiD comienzan con un “*”.

La función ***gendata**(*<nombre_de_pregunta>*) es rellenada con el valor de la pregunta de los datos del problema.

Se puede además indicar si el valor es de tipo entero (**int**) o real (**real**).

Ahora veamos como se generan los datos de conectividades de la malla:

```
# Malla #####
# Dimension
*ndime
# Nodos por elemento
*nnode
# Numero de elementos
*nelem
# Listado de elementos
# Elemento Material node1 node2 ...
*loop elems
*elemsnum *elemsmat *elemsconec
*end elems
```

```
# Malla #####
# Dimension
2
# Nodos por elemento
3
# Numero de elementos
226
# Listado de elementos
# Elemento Material node1 node2 ...
1 1 27 33 24
2 1 76 70 84
3 1 81 82 71
4 1 112 111 120
      :
226 1 94 89 95
```

Las variables de GiD comienzan con un “*”.

La variable ***ndime** indica la dimensión de la geometría 2D o 3D (si hay puntos con $z \neq 0$).

***nnode** indica el número de nodos por elemento, 3 para triángulo, 4 para cuadrilátero, etc.

***nelem** representa el total de elementos en la malla.

***loop elems** hace un ciclo para todos los elementos.

***elemsnum** indica el número de elemento dentro del ciclo.

***elemsmat** es el material asignado e dicho elemento.

***elemsconec** enlista los nodos que se conforman el elemento.

***end elems** cierra el ciclo.

Las coordenadas de los nodos:

```
# Nodos #####
# Numero de nodos
*npoin
# Listado de nodos
# Nodo x1 x2 ...
*loop nodes
*nodesnum *nodescoord
*end nodes
```

```
# Nodos #####
# Numero de nodos
148
# Listado de nodos
# Nodo x1 x2 ...
1      0      0
2      0.27273      0
3      0      0.28571
4      0.29879      0.29455
5      0.54545      0
6      0      0.57143
7      0.567      0.32472
8      0.40757      0.58545
9      0.81818      0
10     0.25391      0.80232
11     0      0.85714
      :
148     5      2
```

***npoin** indica el número total de nodos en la malla.

***loop nodes** hace un ciclo para todos los nodos.

***nodesnum** indica el número de nodo dentro del ciclo.

***nodescoord** enlista las coordenadas del nodo.

***end nodes** cierra el ciclo.

Los materiales:

```
# Materiales #####
# Numero de materiales
*nmats
# Listado de materiales
# Material Thermal_conductivity Mass_density Specific_heat_capacity
*loop materials
*matnum *matprop(Thermal_conductivity,real) *matprop(Mass_density,real) *matprop(Specific_heat_capacity,real)
*end
```

```
# Materiales #####
# Numero de materiales
1
# Listado de materiales
# Material Thermal_conductivity Mass_density Specific_heat_capacity
1          237          2700          900
```

***nmats** indica el número total de materiales utilizados en la malla.

***loop materials** hace un ciclo para todos materiales utilizados en la malla.

***matnum** indica el número material dentro del ciclo.

***matprop(<nombre_de_pregunta>)** es rellenada con el valor de la pregunta del material. Se puede además indicar si el valor es de tipo entero (**int**) o real (**real**).

***end** cierra el ciclo.

Las condiciones de temperatura en los nodos:

```
# Temperaturas #####
*Set Cond Tempeature_node *nodes
*Add Cond Tempearture_line *nodes
# Numero de condiciones
*condnumentities
# Listado de condiciones
# Nodo Temperature
*loop nodes *OnlyInCond
*NodesNum *cond(Temperature,real)
*end
```

```
# Temperaturas #####
# Numero de condiciones
8
# Listado de condiciones
# Nodo Temperature
139          1.2
141          1.2
142          1.2
143          1.2
144          1.2
146          1.2
147          1.2
148          1.2
```

***Set Cond Tempeature_node *nodes** selecciona todos los nodos con condiciones *Tempeature_node*.

***Add Cond Tempeature_line *nodes** agrega a la selección los nodos con condiciones *Tempeature_node*.

***condnumentities** indica el número total de condiciones en la selección.

***loop nodes *OnlyInCond** hace un ciclo para los nodos en la selección de condiciones.

***nodesnum** indica el número de nodo dentro del ciclo.

***cond(Temperature,real)** despliega el valor de temperatura en el nodo.

***end** cierra el ciclo.

Las condiciones de flujo en los elementos:

```
# Flux #####
*Set Cond Flux_line *elems
# Numero de condiciones
*condnumentities
# Listado de condiciones
# Element Flux FaceNode1 FaceNode2 ...
*loop elems *OnlyInCond
*ElemsNum *cond(Flux,real) *globalnodes
*end
```

```
# Flux #####
# Numero de condiciones
7
# Listado de condiciones
# Element Flux FaceNode1 FaceNode2 ...
28          100 48 38
188          100 3 1
210          100 6 3
211          100 38 28
213          100 17 11
215          100 28 17
220          100 11 6
```

- *Set Cond Flux_line *elems** selecciona todos los elementos con condiciones *Flux_line*.
- *condnumentities** indica el número total de condiciones en la selección.
- *loop elems *OnlyInCond** hace un ciclo para los elementos en la selección de condiciones.
 - *ElemsNum** indica el número de elemento dentro del ciclo.
 - *cond(Flux,real)** despliega el valor de flujo en la cara del elemento.
 - *globalnodes** enlista los nodos de la cara del elemento en la que se aplicó la condición.
- *end** cierra el ciclo.

Los términos fuente en los elementos:

```
# Terminos fuente #####
*Set Cond Source_surface *elems
# Numero de condiciones
*condnumentities
# Listado de condiciones
# Element Heat
*loop elems *OnlyInCond
*ElemsNum *cond(Heat,real)
*end
```

```
# Terminos fuente #####
# Numero de condiciones
0
# Listado de condiciones
# Element Heat
```

***Set Cond Source_surface *elems** selecciona todos los elementos con condiciones *Source_surface*.

***condnumentities** indica el número total de condiciones en la selección.

***loop elems *OnlyInCond** hace un ciclo para los elementos en la selección de condiciones.

***ElemsNum** indica el número de elemento dentro del ciclo.

***cond(Heat,real)** despliega el valor del calor en el elemento.

***end** cierra el ciclo.

6. Scripts para la ejecución del “solver”

El paso final es escribir los scripts que llamarán al “solver”.

Para Windows. Crear un archivo de texto llamado:

"C:\Archivos de programa\GiD\problemtypes\Calor.gid\Calor.win.bat"

Con el siguiente contenido:

```
@echo off

del /Q %1.log
del /Q %1.err
del /Q %1.post.res

rem OutputFile: %1.log
rem ErrorFile: %1.err

%3\Calor.exe %1.dat %.post.res > %1.log
```

Aquí el nombre del ejecutable del solver es “Calor.exe”

%1 Contiene el nombre del problema.

%3 Contiene el path al ejecutable del solver.

El ejecutable del solver debe estar ubicado en

"C:\Archivos de programa\GiD\problemtypes\Calor.gid\Calor.exe"

Para GNU/Linux el contenido del archivo sería “Calor.unix.bat”, con contenido:

```
#!/bin/sh

rm -f $1.log
rm -f $1.err
rm -f $1.post.res

# OutputFile: $1.log
# ErrorFile: $1.err

$3/Calor.bin $1.dat $1.post.res > $1.log
```

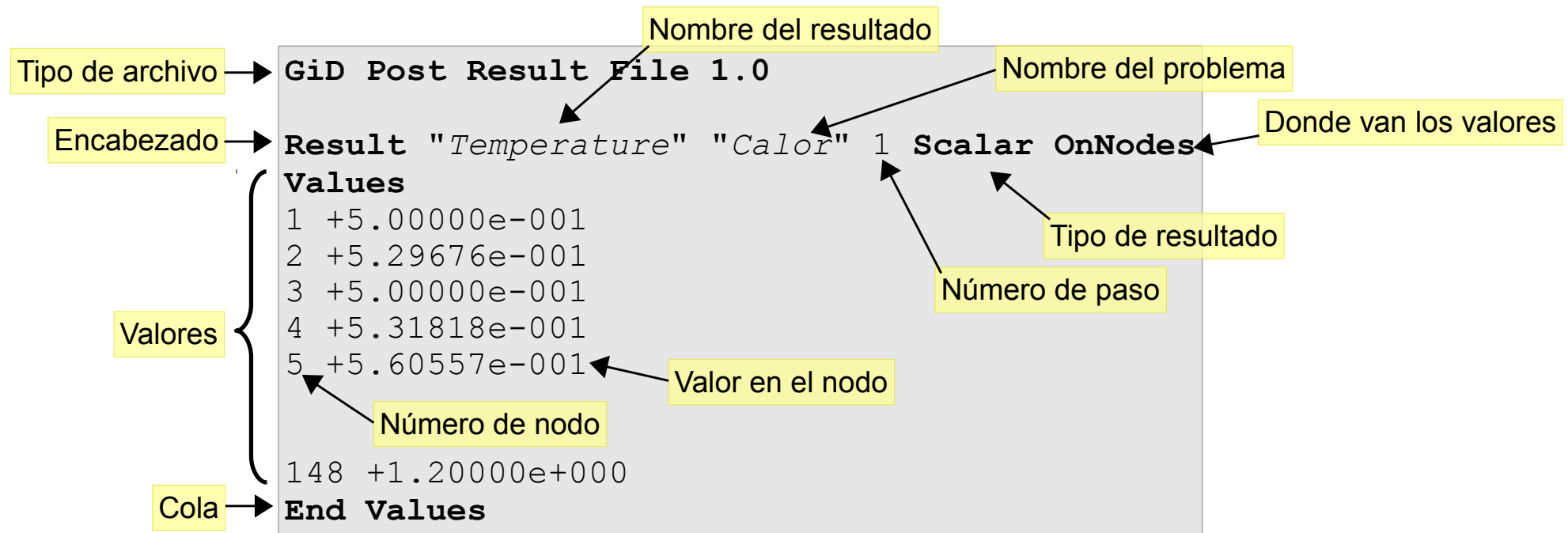
Aquí el nombre del ejecutable del solver es “Calor.bin”

\$1 Contiene el nombre del problema.

\$3 Contiene el path al ejecutable del solver.

Formato de archivos de resultados para GiD

El siguiente es un ejemplo de archivo de resultados que GiD puede leer:



Estos archivos deben tener extensión “<problema>.post.res”

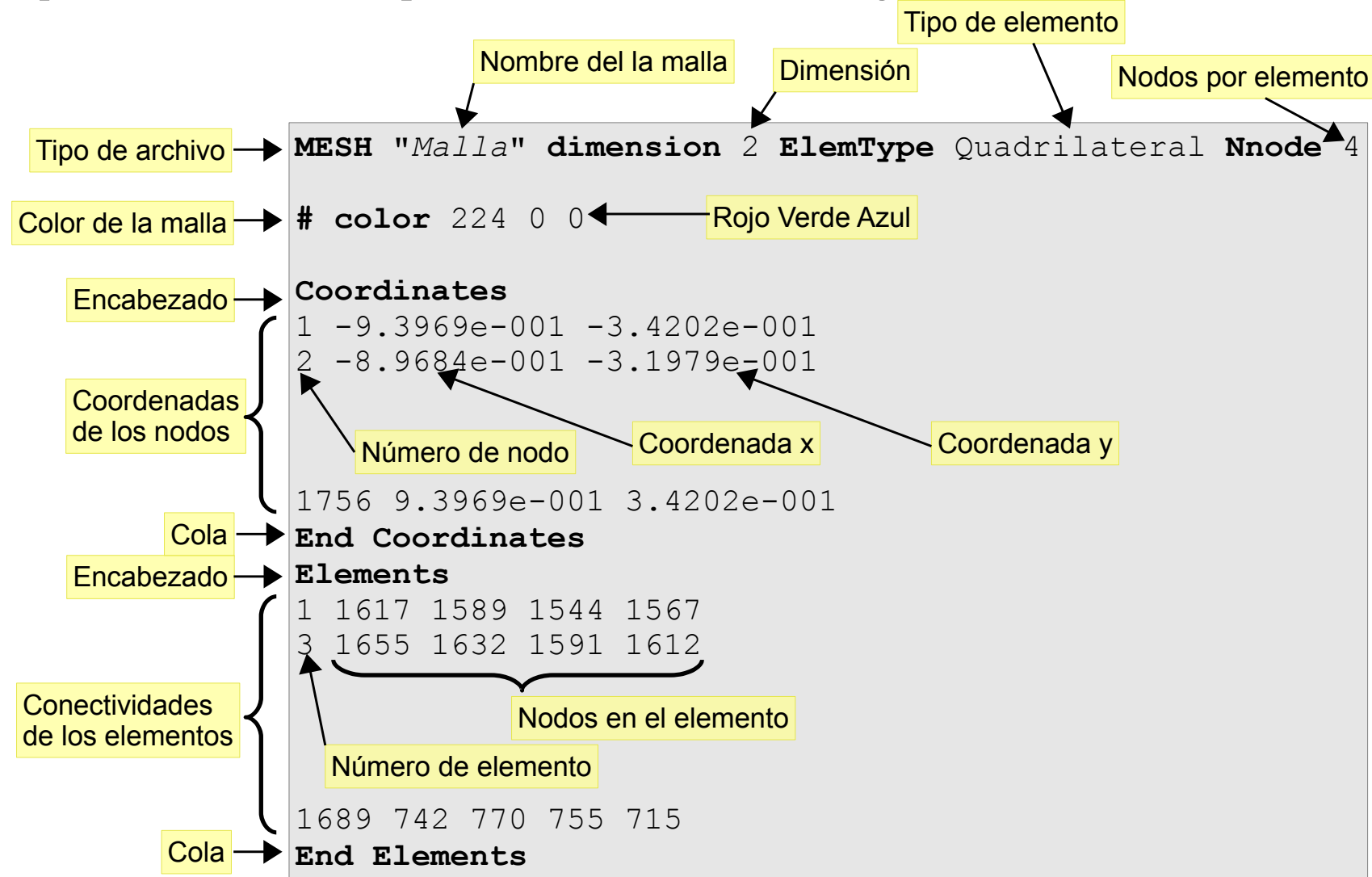
Los tipos de resultados pueden ser **Scalar**, **Vector**, **Matrix**, etc.

Se pueden tener resultados **OnNodes**, **OnGaussPoints**.

Un archivo puede tener varios bloques de resultados.

Formato de archivo de malla

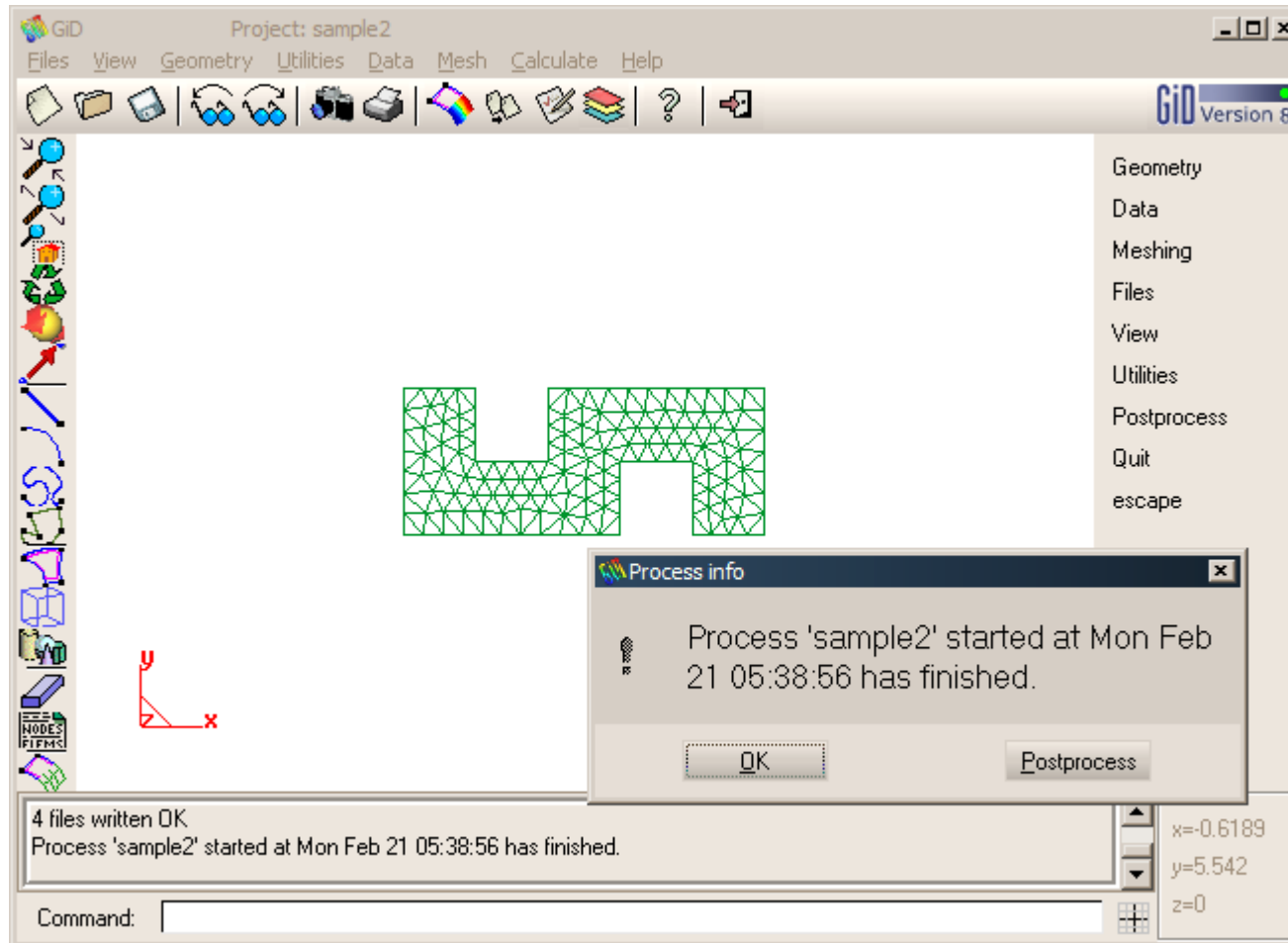
El archivo que contiene la descripción de la malla tiene el siguiente formato:



Estos archivos deben tener extensión “<problema>.post.msh”

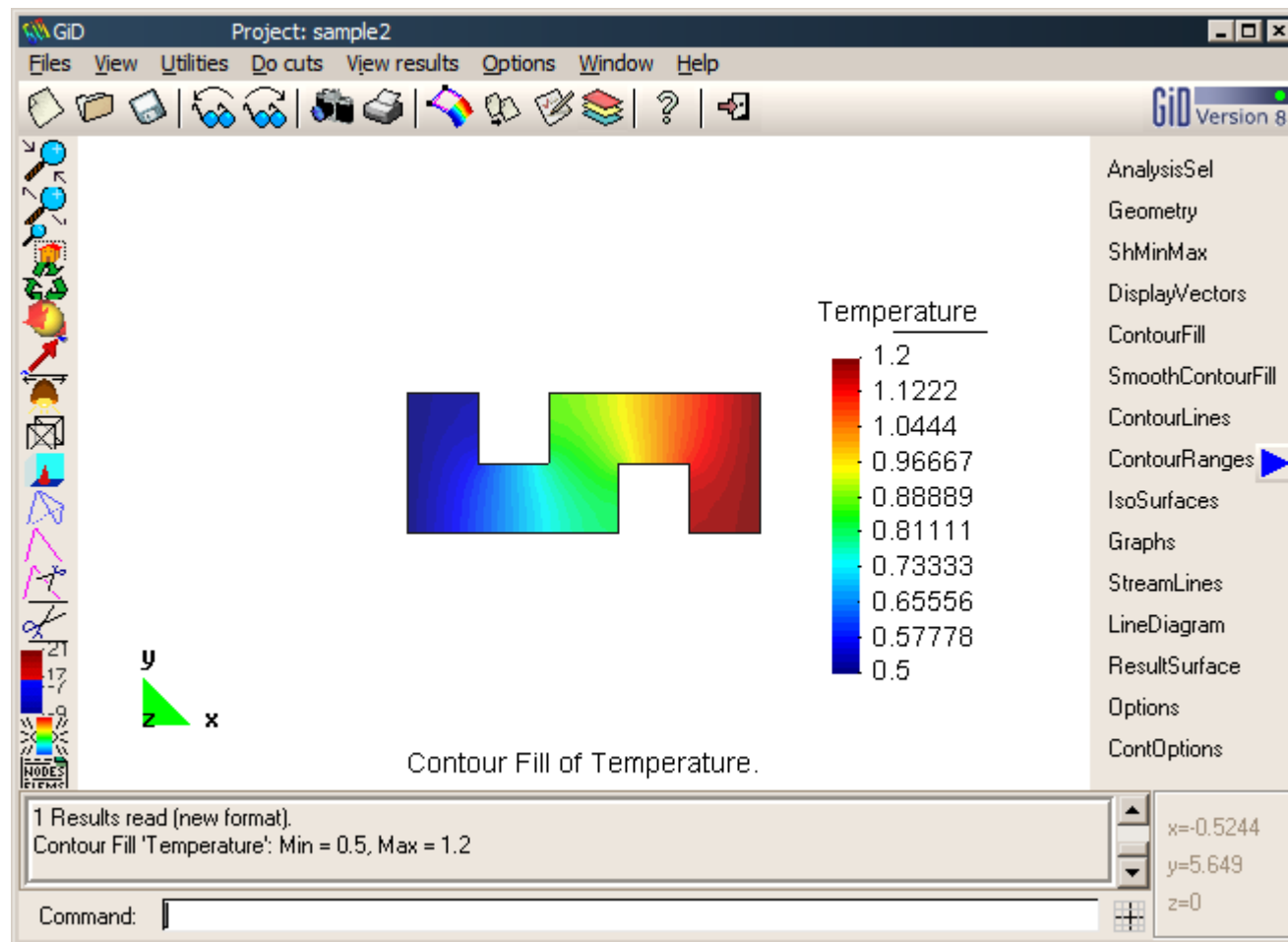
Post-proceso

Al presionar el menú **Calculate/Calculate**, se genera el archivo de datos, y se llama al “solver”.



Cuando el “solver” termina, el GiD lo indica con una cuadro de diálogo. Seleccionamos **Postprocess** para ver los resultados.

En el menú **View/Contour fill/Temperature** podemos ver el campo de temperaturas a color.



¿Preguntas?

<http://gid.cimne.upc.es/support/manuals>

<http://listas.cimne.upc.edu/cgi-bin/mailman/listinfo/gidlist>

miguelvargas@cimat.mx

<http://www.cimat.mx/~miguelvargas>