

Tutorial Gmsh 4.6.0

Módulo de Gmsh-API



Steven Vanegas Giraldo

Gmsh-API es una interfaz de programación de aplicaciones.

- Lenguaje C.
- Lenguaje C++.
- Lenguaje Python.
- Lenguaje Julia.



► Link

www.anaconda.com/products/individual



Se puede usar **pip**, sistema de gestion de paquetes en Python.

Instalar:

```
pip install gmsh_api
```

Se puede empezar importar el módulo en Python:

```
import gmsh_api.gmsh as gmsh
```



Siempre se debe inicializar y finalizar el módulo. Para inicializar:

`gmsh.initialize()`

Para finalizar:

`gmsh.finalize()`

Para abrir el archivo de malla:

`gmsh.open(nombre_archivo)`

Este comando nos permite acceder a la información de la malla generada en el archivo.

```
eti_n, coor_n, para_coor =  
gmsh.model.mesh.getNodes(dim, eti)
```

Parámetros de entrada:

- **dim**: dimensión de la entidad (entero positivo)
- **eti**: etiqueta numérica de la entidad (entero ≥ 1)

Parámetros de salida:

- **eti_n**: etiquetas numéricas de los nodos (empiezan desde 1)
- **coor_n**: coordenadas de los nodos
- **para_coor**: coordenadas paramétricas del modelo interno de Gmsh

Si **eti** es negativo se retornará las variables de salida de todas las entidades de dimensión **dim**. Si ambas son negativas se retornará las variables de salida de todos los nodos presentes en la malla.

Si no se ingresan parámetros de entrada, se retornará las variables de salida de todos los nodos en la malla.

- **eti_n**: es un array con un tamaño igual al número de nodos de la malla

eti_n = [eti1, eti2, ..., etin]

- **coor_n**: es un array con un tamaño igual a 3 veces el número de nodos de la malla

coor_n = [x1, y1, z1, x2, y2, z2, ..., xn, yn, zn]

```
eti_n, coor_n, para_coor =  
gmsh.model.mesh.getNodesByElementType(tipo_e, eti)
```

Parámetros de entrada:

- **tipo_e**: número que indica el tipo de elemento
- **eti**: etiqueta numérica de la entidad (entero ≥ 1)

Parámetros de salida:

- **eti_n**: etiquetas numéricas de los nodos
- **coor_n**: coordenadas de los nodos
- **para_coor**: coordenadas paramétricas del modelo interno de Gmsh

Si no ingresa **eti** se retorna las variables de salida de todos los elementos especificados por **tipo_e**.

- **eti_n**: es un array con un tamaño igual al número de elementos finitos (n_{ef}) por el número de nodos (n) que tiene el elemento finito especificado por **tipo_e**

$$\mathbf{eti_n} = [\mathbf{eti1}, \mathbf{eti2}, \dots, \mathbf{etin}]$$

- **coor_n**: es un array con un tamaño igual a $n * n_{ef} * 3$ es el número de nodos del elemento finito tratado.

$$\mathbf{coor_n} = [\mathbf{x11}, \mathbf{y11}, \mathbf{z11}, \mathbf{x12}, \mathbf{y12}, \mathbf{z12}, \dots, \mathbf{xnefn}, \mathbf{ynefn}, \mathbf{znefn}]$$

```
eti_elementos, eti_nodos =  
gmsh.model.mesh.getElementsByType(tipo_elemento, eti)
```

Parámetros de entrada:

- **tipo_elemento**: tipo de elemento clasificado por Gmsh
- **eti**: etiqueta numérica de la entidad donde se encuentra este tipo de elemento

Parámetros de salida:

- **eti_elementos**: etiquetas numéricas del elementos buscar
- **eti_nodos**: etiquetas numéricas de los nodos que conforman los elementos

eti_elementos no empieza en 1, tiene en cuenta todos los elementos creados

```
eti_n, coor_n =  
gmsh.model.mesh.getNodesForPhysicalGroup(dim_gf,  
eti_gf)
```

Parámetros de entrada:

- **dim_gf**: dimensión del grupo físico seleccionado
- **eti_gf**: etiqueta numérica del grupo físico seleccionado (entero ≥ 1)

Parámetros de salida:

- **eti_n**: etiquetas numéricas de los nodos de los elementos del grupo físico seleccionado
- **coor_n**: coordenadas de los nodos de los elementos del grupo físico seleccionado

```
eti_elementos =  
gmsh.model.mesh.getElementsByCoordinates(x, y, z, dim,  
exacto)
```

Parámetros de entrada:

- **x, y, z**: las coordenadas x, y, z donde está localizado el elemento
- **dim**: dimensión del elemento a buscar
- **exacto**: variable booleana que me indica si se usa o no una tolerancia en la búsqueda

Parámetros de salida:

- **eti_elementos**: etiquetas numéricas del elementos buscar