



GID

Manual de Instalación

Tabla de Contenido

1.	Introducción	3
2.	Inicio Instalación de Software Kratos Multiphysics	3
3.	Instalación GiD	10

1. Introducción

GiD es una interfaz gráfica de usuario interactiva utilizada para la definición, preparación y visualización de todos los datos relacionados con una simulación numérica. Estos datos incluyen la definición de la geometría, los materiales, las condiciones, la información de la solución y otros parámetros. El programa puede generar una malla adecuada para varios métodos numéricos (elemento finito, volumen finito o diferencia finita, métodos basados en partículas, entre otros) y escribir la información para una simulación numérica.

El presente manual contiene una serie de pasos e información para instalar GiD junto con los componentes básicas y aplicaciones necesarias para su correcto funcionamiento en el servidor de cálculo FIULS, “SynergyGrid”, cuyo sistema operativo corresponde a Linux.

2. Inicio Instalación de Software Kratos Multiphysics

Para la instalación del software GiD se debe realizar la instalación del software Kratos.

KRATOS Multiphysics es un marco virtual de trabajo compuesto por un conjunto estandarizado de conceptos, criterios y elementos para la construcción de softwares de simulación.

Antes de cualquier acción en el ordenador, se debe realizar una actualización del sistema con ayuda del comando:

`sudo apt-get update`

Para la instalación de Kratos es necesario obtener su código fuente con la ayuda del software **git**, a través del siguiente comando en Linux:

`sudo apt-get install git`

Una vez instalado git, es necesario descargar Kratos a través del siguiente comando:

`git clone https://github.com/KratosMultiphysics/Kratos Kratos`

Posteriormente se debe realizar la instalación de las dependencias principales de Kratos como son Python3-dev, C++11 compiler, Cmake y Boost siendo estas necesarias para compilar el núcleo de Kratos y la mayoría de sus aplicaciones.

En la instalación de la dependencia Python se debe considerar como requisito mínimo la versión 3.5 (3.7 o 3.8 recomendada), para Cmake la versión 3.14 en adelante y para Boost la versión 1.67 en adelante. El siguiente comando dará comienzo a la instalación:

sudo apt-get install python3-dev gcc g++ cmake libboost-all-dev

Se recomienda monitorear el estado de instalación de las dependencias y su información con el comando "**apt-cache policy nombreaplicación**".

A continuación, se requiere la instalación del software "pip" con el comando "**sudo apt-get install python3-pip**". Este sistema de administrador de paquetes permite instalar paquetes de Python necesarios para GiD.

Paquetes requeridos:

Pandas	"pip3 install pandas"
Numpy	"pip3 install numpy"
Matplotlib	"pip3 install matplotlib"

Una vez ejecutados los comandos anteriores se debe hacer la instalación de las librerías **Lapack** y **Blas**, cuyo propósito será el de resolver operaciones algebraicas de diferente complejidad utilizado en los procesos de GiD. Para su instalación se debe descargar la versión reciente de **Lapack** a través del siguiente enlace: <http://www.netlib.org/lapack/>. Una vez descargado es necesario mover el archivo a la carpeta de inicio y descomprimir a través del comando:

tar -xvf nombredelarchivo

De esta manera se debe abrir la carpeta descomprimida desde el terminal escribiendo "**cd nombrecarpeta**" e ingresar el siguiente comando:

cp make.inc.example make.inc

Una vez terminado el proceso es necesario escribir "**make blaslib**" dentro del terminal con la carpeta que abrimos anteriormente y de la misma forma se debe escribir "**make lapacklib**". Para finalizar se requiere ingresar en el mismo terminal el siguiente comando:

sudo ln -s \$HOME/lapack-3.9.0/librefblas.a /usr/local/lib/libblas.a

y posteriormente el comando

sudo ln -s \$HOME/lapack-3.9.0/liblapack.a /usr/local/lib/liblapack.a

```
engineer-user@engineer-007: ~/lapack-3.10.1
Archivo  Editar  Ver  Buscar  Terminal  Ayuda
engineer-user@engineer-007:~$ cd lapack-3.10.1
engineer-user@engineer-007:~/lapack-3.10.1$ cp make.inc.example make.inc
engineer-user@engineer-007:~/lapack-3.10.1$ make blaslib
make -C BLAS
make[1]: se entra en el directorio '/home/engineer-user/lapack-3.10.1/BLAS'
make -C SRC
make[2]: se entra en el directorio '/home/engineer-user/lapack-3.10.1/BLAS/SRC'
make[2]: No se hace nada para 'all'.
make[2]: se sale del directorio '/home/engineer-user/lapack-3.10.1/BLAS/SRC'
make[1]: se sale del directorio '/home/engineer-user/lapack-3.10.1/BLAS'
engineer-user@engineer-007:~/lapack-3.10.1$ make lapacklib
make -C SRC
make[1]: se entra en el directorio '/home/engineer-user/lapack-3.10.1/SRC'
make[1]: se sale del directorio '/home/engineer-user/lapack-3.10.1/SRC'
engineer-user@engineer-007:~/lapack-3.10.1$ sudo ln -s $HOME/lapack-3.9.0/libre
fbblas.a /usr/local/lib/libblas.a
[sudo] contraseña para engineer-user:
ln: fallo al crear el enlace simbólico '/usr/local/lib/libblas.a': El archivo y
a existe
engineer-user@engineer-007:~/lapack-3.10.1$
```

Tras ejecutar los comandos anteriores, se debe modificar el archivo de configuración de Kratos denominado "standard_configure.sh", este debería tener la siguiente ruta: **/home/engineer-user/Kratos/scripts/**

Dentro del archivo es necesario agregar en la sección "**# Set basic configuration**" (líneas 31 y 32) las siguientes líneas:

```
export BOOST_ROOT=${BOOST_ROOT:-"usr/include/boost"}
export KRATOS_INSTALL_PYTHON_USING_LINKS=ON
```

```
28 # Set basic configuration
29 export KRATOS_BUILD_TYPE=${KRATOS_BUILD_TYPE:-"Release"}
30 export PYTHON_EXECUTABLE=${PYTHON_EXECUTABLE:-"/usr/bin/python3"}
31 export BOOST_ROOT=${BOOST_ROOT:-"usr/include/boost"}
32 export KRATOS_INSTALL_PYTHON_USING_LINKS=ON
```

Una vez agregadas, hay que verificar la ruta de los comandos anteriormente escritos en "**# Set basic configuration**" con la ayuda del comando "**whereis nombreaplicación**", de no ser correcta se debe modificar reescribiendo la ruta correspondiente.

```
engineer-user@engineer-007: ~
Archivo  Editar  Ver  Buscar  Terminal  Ayuda
engineer-user@engineer-007:~$ whereis boost
boost: /usr/include/boost
```

Posteriormente se requiere instalar las aplicaciones con las que se quiere trabajar, cuya selección dependerá del usuario y sus objetivos.

La instalación se debe realizar dentro del archivo "**standard_configure.sh**" en la sección "**# Set applications to compile**" (línea 34), cuya estructura debe ser "**add_app \${KRATOS_APP_DIR}/Aplicación**".

```
34 # Set applications to compile
35 export KRATOS_APPLICATIONS=
36 add_app ${KRATOS_APP_DIR}/LinearSolversApplication
37 add_app ${KRATOS_APP_DIR}/StructuralMechanicsApplication
38 add_app ${KRATOS_APP_DIR}/DEMAApplication
39 add_app ${KRATOS_APP_DIR}/FluidDynamicsApplication
40 add_app ${KRATOS_APP_DIR}/MeshingApplication
41 add_app ${KRATOS_APP_DIR}/SwimmingDEMAApplication
42 add_app ${KRATOS_APP_DIR}/DemStructuresCouplingApplication
```

GID puede trabajar con una basta cantidad de aplicaciones, las cuales se actualizan y/o modifican con frecuencia.

En el siguiente listado se muestra las aplicaciones que pueden ser instalados en Kratos y utilizados en GID.

```
add_app ${KRATOS_APP_DIR}/CSharpWrapperApplication
add_app ${KRATOS_APP_DIR}/CableNetApplication
add_app ${KRATOS_APP_DIR}/ChimeraApplication
add_app ${KRATOS_APP_DIR}/CoSimulationApplication
add_app ${KRATOS_APP_DIR}/CompressiblePotentialFlowApplication
add_app ${KRATOS_APP_DIR}/ConstitutiveLawsApplication
add_app ${KRATOS_APP_DIR}/ConstitutiveModelsApplication
add_app ${KRATOS_APP_DIR}/ContactMechanicsApplication
add_app ${KRATOS_APP_DIR}/ContactStructuralMechanicsApplication
add_app ${KRATOS_APP_DIR}/ConvectionDiffusionApplication
add_app ${KRATOS_APP_DIR}/DEMAApplication
add_app ${KRATOS_APP_DIR}/DamApplication
add_app ${KRATOS_APP_DIR}/DelaunayMeshingApplication
add_app ${KRATOS_APP_DIR}/DemStructuresCouplingApplication
add_app ${KRATOS_APP_DIR}/DropletDynamicsApplication
add_app ${KRATOS_APP_DIR}/EigenSolversApplication
add_app ${KRATOS_APP_DIR}/ExaqueteSandboxApplication
add_app ${KRATOS_APP_DIR}/FSIApplication
add_app ${KRATOS_APP_DIR}/FemToDemApplication
add_app ${KRATOS_APP_DIR}/FluidDynamicsApplication
add_app ${KRATOS_APP_DIR}/FluidDynamicsBiomedicalApplication
add_app ${KRATOS_APP_DIR}/FluidTransportApplication
add_app ${KRATOS_APP_DIR}/FreeSurfaceApplication
add_app ${KRATOS_APP_DIR}/GeoMechanicsApplication
add_app ${KRATOS_APP_DIR}/HDF5Application
add_app ${KRATOS_APP_DIR}/IgaApplication
add_app ${KRATOS_APP_DIR}/LinearSolversApplication
add_app ${KRATOS_APP_DIR}/MappingApplication
```

```

add_app ${KRATOS_APP_DIR}/MeshMovingApplication
add_app ${KRATOS_APP_DIR}/MeshingApplication
add_app ${KRATOS_APP_DIR}/MetisApplication
add_app ${KRATOS_APP_DIR}/MultilevelMonteCarloApplication
add_app ${KRATOS_APP_DIR}/ParticleMechanicsApplication
add_app ${KRATOS_APP_DIR}/PfemApplication
add_app ${KRATOS_APP_DIR}/PfemFluidDynamicsApplication
add_app ${KRATOS_APP_DIR}/PfemSolidMechanicsApplication
add_app ${KRATOS_APP_DIR}/PoromechanicsApplication
add_app ${KRATOS_APP_DIR}/RANSApplication
add_app ${KRATOS_APP_DIR}/RomApplication
add_app ${KRATOS_APP_DIR}/ShallowWaterApplication
add_app ${KRATOS_APP_DIR}/ShapeOptimizationApplication
add_app ${KRATOS_APP_DIR}/SolidMechanicsApplication
add_app ${KRATOS_APP_DIR}/StatisticsApplication
add_app ${KRATOS_APP_DIR}/StructuralMechanicsApplication
add_app ${KRATOS_APP_DIR}/SwimmingDEMApplication
add_app ${KRATOS_APP_DIR}/ThermalDEMApplication
add_app ${KRATOS_APP_DIR}/TopologyOptimizationApplication
add_app ${KRATOS_APP_DIR}/TrilinosApplication
add_app ${KRATOS_APP_DIR}/ULFApplication
add_app ${KRATOS_APP_DIR}/UmatApplication
add_app ${KRATOS_APP_DIR}/WindEngineeringApplication

```

Se debe tener en consideración que algunas aplicaciones requieren de ciertas dependencias y/o la instalación de otras aplicaciones para su correcto funcionamiento. Es recomendable informarse sobre la aplicación que se quiere instalar y estar atento ante cualquier modificación. Para esto se puede recurrir al siguiente enlace: <https://github.com/KratosMultiphysics> .

Una vez instaladas las aplicaciones es necesario ingresar en la sección "**# Configure**" los siguientes directorios:

```

-DBLAS_LIBRARIES="/usr/local/lib/liblapack.a" \
-DLAPACK_LIBRARIES="/usr/local/lib/libblas.a" \
-DUSE_TRIANGLE_NONFREE_TPL=ON \

```

```

50 # Configure
51 cmake -H"${KRATOS_SOURCE}" -B"${KRATOS_BUILD}/${KRATOS_BUILD_TYPE}" \
52 -DUSE_MPI=OFF \
53 -DUSE_TRIANGLE_NONFREE_TPL=ON \
54 -DUSE_EIGEN_MKL=OFF \
55 -DBLAS_LIBRARIES="/usr/local/lib/liblapack.a" \
56 -DLAPACK_LIBRARIES="/usr/local/lib/libblas.a" \

```

Se debe revisar la ruta del directorio con el comando "**whereis nombreaplicación**" y modificar en caso de ser necesario.

```
engineer-user@engineer-007:~$ whereis liblapack
liblapack: /usr/lib/x86_64-linux-gnu/liblapack.a /usr/lib/x86_64-linux-gnu/liblapack.so /usr/local/lib/liblapack.a
engineer-user@engineer-007:~$ whereis libblas
libblas: /usr/lib/x86_64-linux-gnu/libblas.a /usr/lib/x86_64-linux-gnu/libblas.so /usr/local/lib/libblas.a
engineer-user@engineer-007:~$ █
```

De este modo se requiere ingresar los directorios de las aplicaciones que se han instalado en la sección "# Set applications to compile".

En el siguiente listado se muestra los directorios de cada aplicación para permitir su uso.

```
-DCSHARP_WRAPPER_APPLICATION=ON \
-DCABLE_NET_APPLICATION=ON \
-DCHIMERA_APPLICATION=ON \
-DCO_SIMULATION_APPLICATION=ON \
-DCOMPRESSIBLE_POTENTIAL_FLOW_APPLICATION=ON \
-DCONSTITUTIVE_LAWS_APPLICATION=ON \
-DCONSTITUTIVE_MODELS_APPLICATION=ON \
-DCONTACT_MECHANICS_APPLICATION=ON \
-DCONTACT_STRUCTURAL_MECHANICS_APPLICATION=ON \
-DCONVECTION_DIFFUSION_APPLICATION=ON \
-DDEM_APPLICATION=ON \
-DDAM_APPLICATION=ON \
-DDELAUNAY_MESHING_APPLICATION=ON \
-DDEM_STRUCTURES_COUPLING_APPLICATION=ON \
-DDROPLET_DYNAMICS_APPLICATION=ON \
-DEIGEN_SOLVERS_APPLICATION=ON \
-DEXAQUATE_SANDBOX_APPLICATION=ON \
-DFSI_APPLICATION=ON \
-DFEM_TO_DEM_APPLICATION=ON \
-DFLUID_DYNAMICS_APPLICATION=ON \
-DFLUID_DYNAMICS_BIOMEDICAL_APPLICATION=ON \
-DFLUID_TRANSPORT_APPLICATION=ON \
-DFREE_SURFACE_APPLICATION=ON \
-DGEO_MECHANICS_APPLICATION=ON \
-DHDF5_APPLICATION=ON \
-DIGA_APPLICATION=ON \
-DLINEAR_SOLVERS_APPLICATION=ON \
-DMAPPING_APPLICATION=ON \
-DMESH_MOVING_APPLICATION=ON \
-DMESHING_APPLICATION=ON \
-DMETIS_APPLICATION=ON \
-DMULTILEVEL_MONTECARLO_APPLICATION=ON \
-DPARTICLE_MECHANICS_APPLICATION=ON \
-DPFEM_APPLICATION=ON \
-DPFEM_FLUID_DYNAMICS_APPLICATION=ON \
```



```

-DPFEM_SOLIDMECHANICS_APPLICATION=ON \
-DPOROMECHANICS_APPLICATION=ON \
-DRANS_APPLICATION=ON \
-DROM_APPLICATION=ON \
-DSHALLOW_WATER_APPLICATION=ON \
-DSHAPE_OPTIMIZATION_APPLICATION=ON \
-DSOLID_MECHANICS_APPLICATION=ON \
-DSTATISTICS_APPLICATION=ON \
-DSTRUCTURAL_MECHANICS_APPLICATION=ON \
-DSWIMMING_DEM_APPLICATION=ON \
-DTHERMAL_DEM_APPLICATION=ON \
-DTOPOLOGY_OPTIMIZATION_APPLICATION=ON \
-DTRILINOS_APPLICATION=ON \
-DULF_APPLICATION=ON \
-DUMAT_APPLICATION=ON \
-DWIND_ENGINEERING_APPLICATION=ON \

```

```

50 # Configure
51 cmake -H"${KRATOS_SOURCE}" -B"${KRATOS_BUILD}/${KRATOS_BUILD_TYPE}" \
52 -DUSE_MPI=OFF \
53 -DUSE_TRIANGLE_NONFREE_TPL=ON \
54 -DUSE_EIGEN_MKL=OFF \
55 -DBLAS_LIBRARIES="/usr/local/lib/liblapack.a" \
56 -DLAPACK_LIBRARIES="/usr/local/lib/libblas.a" \
57 -DLINEAR_SOLVERS_APPLICATION=ON \
58 -DSTRUCTURAL_MECHANICS_APPLICATION=ON \
59 #-DFLUID_DYNAMICS_APPLICATION=ON \
60 -DDEM_APPLICATION=ON \
61 -DDEM_STRUCTURES_COUPLING_APPLICATION=ON \
62 -DSWIMMING_DEM_APPLICATION=ON \
63 -DMESHING_APPLICATION=ON \
64 -DTRILINOS_INCLUDE_DIR="/usr/include/trilinos" \
65 -DTRILINOS_LIBRARY_DIR="/usr/lib/x86_64-linux-gnu" \
66 -DTRILINOS_LIBRARY_PREFIX="trilinos" \

```

Finalizada la modificación del archivo "**standard_configure.sh**" se recomienda guardar con un nuevo nombre. Para los siguientes pasos el nombre del archivo "**standard_configure.sh**" será modificado a "**configure.sh**".

Una vez agregados los directorios, es necesario hacer la compilación de las aplicaciones instaladas. Para esto se debe ingresar desde el terminal a la carpeta de scripts en Kratos con el comando "**cd Kratos/scripts**". Una vez listo, se requiere cambiar los permisos del archivo "**configure.sh**" para que así pueda ser ejecutable con el siguiente comando:

```
sudo chmod +x configure.sh
```

A continuación, es necesario iniciar la compilación del archivo "**configure.sh**" en el mismo terminal con la carpeta de "**scripts**" abierta utilizando el comando:

```
./configure.sh
```

```

engineer-user@engineer-007:~/Kratos/scripts$ cd
engineer-user@engineer-007:~$ cd Kratos/scripts
engineer-user@engineer-007:~/Kratos/scripts$ sudo chmod +x configure.sh
engineer-user@engineer-007:~/Kratos/scripts$ ./configure.sh

```

Para finalizar la compilación se debe ubicar al archivo “**~/.bashrc**” el cual se encuentra en la carpeta de inicio (home/engineer-user/) en los archivos ocultos. Posteriormente se requiere agregar al final del archivo las siguientes líneas:

```

export PYTHONPATH=$PYTHONPATH:$HOME/Kratos/bin/Release
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$HOME/Kratos/bin/Release/libs

```

```

117fi
118
119export PYTHONPATH=$PYTHONPATH:$HOME/Kratos/bin/Release
120export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$HOME/Kratos/bin/Release/libs

```

Finalmente, para el testeo de Kratos se debe ingresar a Python a través del terminal e ingresar el siguiente comando:

```
from KratosMultiphysics import *
```

Como resultado debe entregar:

```

engineer-user@engineer-007:~$ python3
Python 3.8.10 (default, Jun 22 2022, 20:18:18)
[GCC 9.4.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> from KratosMultiphysics import *
| /
| / \ | \ | \ | \ | \
| \ | \ | \ | \ | \ | \
| \ \ | \ \ | \ \ | \ \ /
Multi-Physics 9.1."3"-0c392b3f39-Release
Compiled with threading support.
Maximum number of threads: 20.
>>> exit()
engineer-user@engineer-007:~$

```

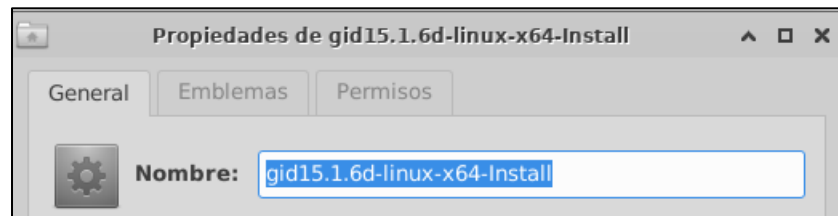
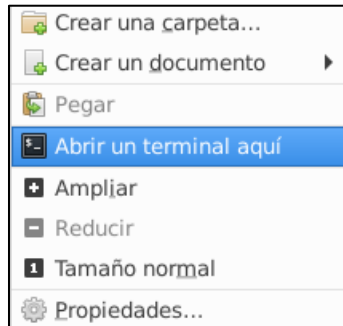
3. Instalación GiD

Para iniciar con la instalación se debe descargar GiD a través del siguiente enlace: <https://www.gidsimulation.com/gid-for-science/downloads/>.

Se recomienda la instalación de "GiD Developer version" porque contiene la versión más actualizada del programa. Hay que considerar que la "Developer version" está en continua actualización y puede presentar errores que serán solucionados a medida que se hagan las correspondientes pruebas de validación por parte de sus desarrolladores.

A continuación, es necesario abrir un terminal en la carpeta que contiene el archivo previamente descargado o escribir en el terminal el comando "**cd nombrecarpeta**", luego se debe ingresar el siguiente comando:

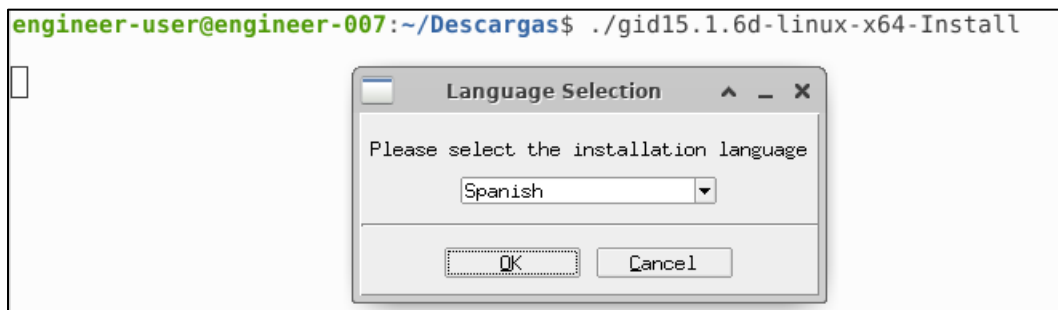
sudo chmod +x nombredelarchivodescargado



```
engineer-user@engineer-007:~/Descargas$ sudo chmod +x gid15.1.6d-linux-x64-Install
[sudo] contraseña para engineer-user:
engineer-user@engineer-007:~/Descargas$
```

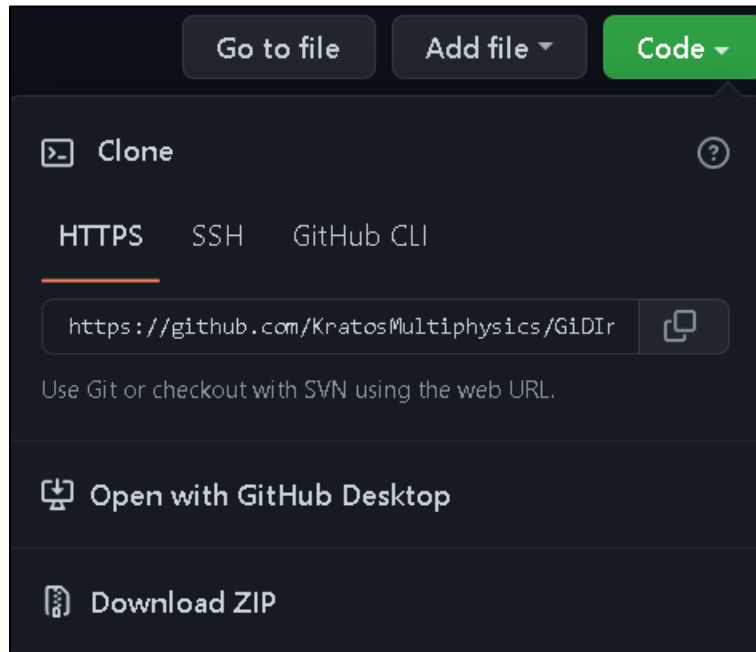
Posteriormente para abrir el instalador de GiD se requiere ingresar el siguiente comando en la misma carpeta:

./nombredelarchivodescargado



Finalizada la instalación es necesario clonar la carpeta de "**GiD Interface**" para la lectura de Kratos, ingresando al siguiente enlace: <https://github.com/KratosMultiphysics/GiDInterface>.

Dentro de la página se debe copiar el enlace que está en la sección "Code".



Una vez copiado se requiere abrir un nuevo terminal e ingresar el enlace copiado junto con el siguiente comando:

"git clone enlacecopiado"

```
engineer-user@engineer-007:~$ git clone https://github.com/KratosMultiphysics/GiDIr
fatal: la ruta de destino 'GiDIr' ya existe y no es un directorio vacío.
```

Finalmente se deben crear accesos directos de Kratos en las carpetas "**problemtypes**" ubicada en "**GiD**" y en la carpeta "**exec**" ubicada en "**GiDIInterface**" con el siguiente procedimiento:

- 1.- Abrir terminal en la carpeta "**problemtypes**".
- 2.- Ingresar el siguiente comando:

ln -s /home/engineer-user/GiDIInterface/kratos.gid kratos.gid

```
engineer-user@engineer-007:~/GIDx64/gid-15.1.6d/problemtypes$ ln -s /home/engineer-user/GiDIInterface/kratos.gid kratos.gid
engineer-user@engineer-007:~/GIDx64/gid-15.1.6d/problemtypes$ █
```

3.- Revisar en la carpeta "**problemtypes**" el acceso directo de "**kratos.gid**".

/home/engineer-user/GiDx64/gid-15.1.6d/problemtypes/			
	Nombre	Tamaño	Tipo
	abaqus.gid	4,0 KiB	carpeta
	ansys55	4,0 KiB	carpeta
	Caltep.gid	4,0 KiB	carpeta
	Examples	4,0 KiB	carpeta
	FEniCS.gid	4,0 KiB	carpeta
	Fluent.gid	4,0 KiB	carpeta
	Frame3DD.gid	4,0 KiB	carpeta
	kratos.gid	4,0 KiB	Enlace a /home/engineer-user/GiDInterface/kratos.gid
	Isdyna.gid	4,0 KiB	carpeta
	nastran.gid	4,0 KiB	carpeta
	OpenFoam.gid	4,0 KiB	carpeta
	SAP2000.gid	4,0 KiB	carpeta

4.-Abrir terminal en la carpeta "**exec**".

5.-Ingresar el siguiente comando:

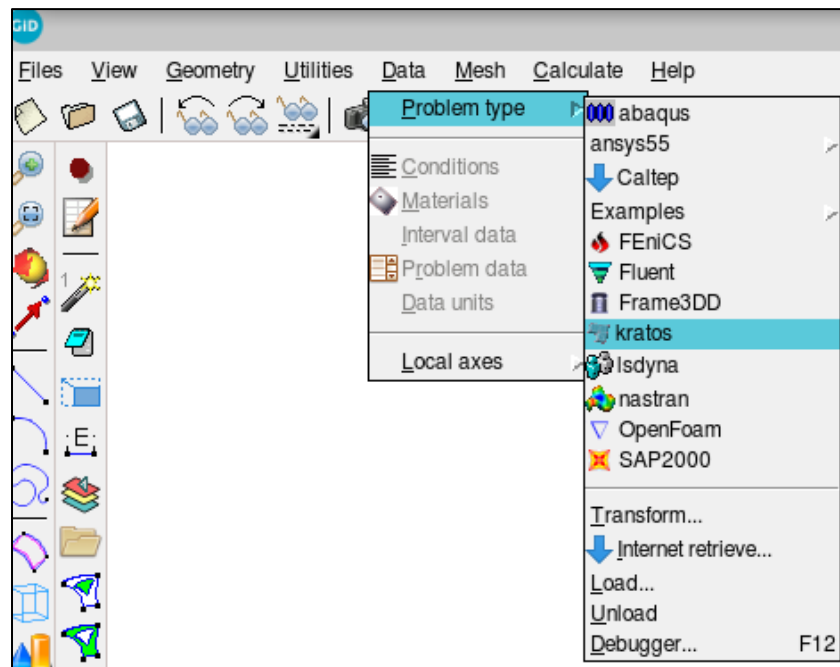
ln -s ~/Kratos/bin/Release Kratos

```
engineer-user@engineer-007:~/GiDInterface/kratos.gid/exec$ ln -s ~/Kratos/bin/Release Kratos
ln: fallo al crear el enlace simbólico 'Kratos/Release': El archivo ya existe
engineer-user@engineer-007:~/GiDInterface/kratos.gid/exec$
```

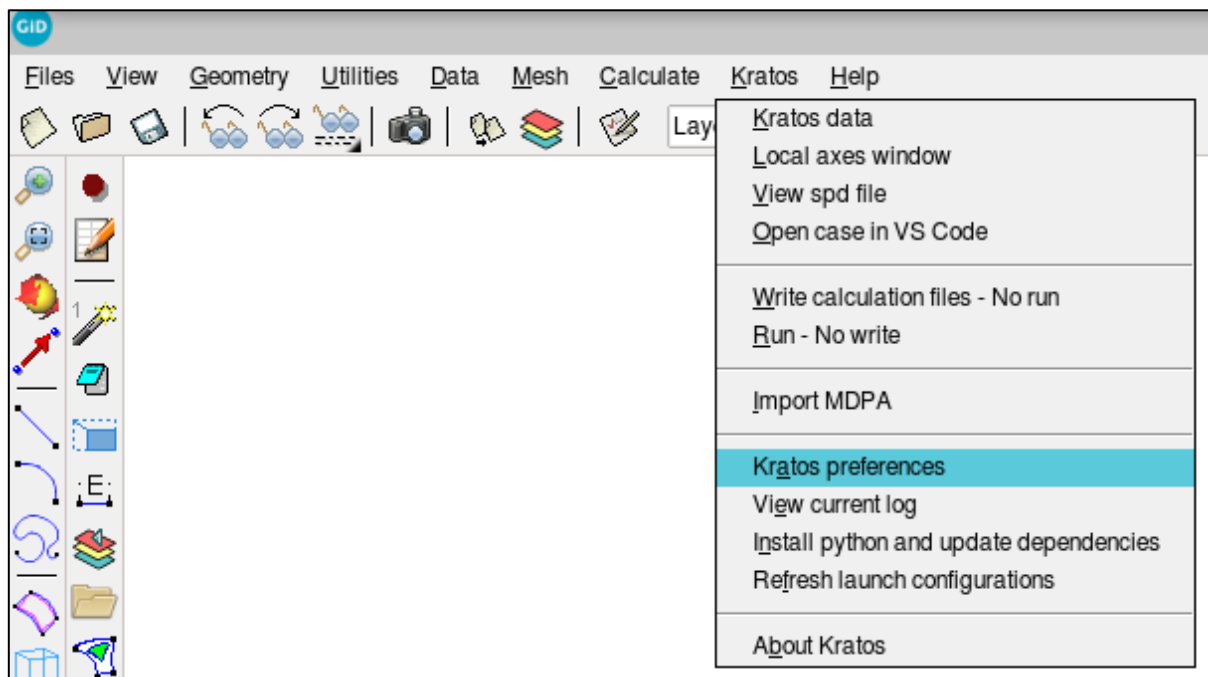
6.-Revisar en la carpeta "**exec**" el acceso directo de Kratos.

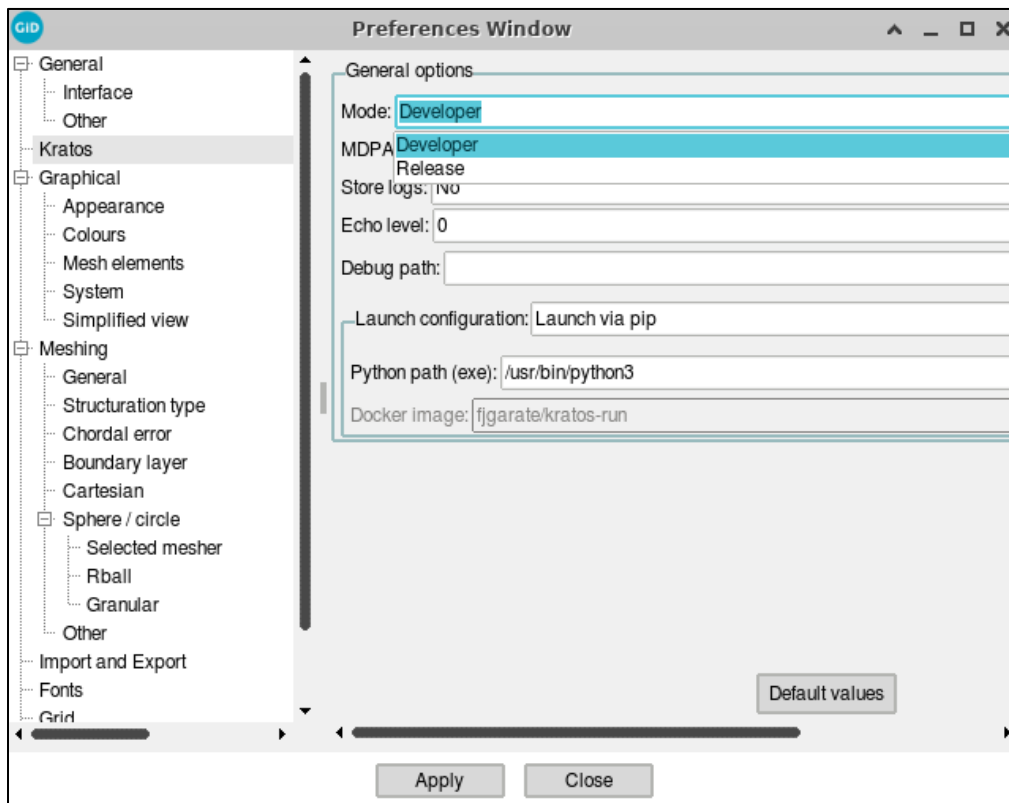
/home/engineer-user/GiDInterface/kratos.gid/exec/			
	Nombre	Tamaño	Tipo
	Kratos	4,0 KiB	Enlace a /home/engineer-user/Kratos/bin/Release

Finalizado el procedimiento, se debe abrir el programa GiD, en las pestañas superiores se debe seleccionar Data > Problem types > Kratos y cerrar la interface previamente abierta.



Posteriormente se debe seleccionar la pestaña Kratos > Kratos preferences y en la ventana que se abrirá hay que encontrar la pestaña “mode” y cambiar a "Developer"





Por último, se debe seleccionar Data > Problem types > Kratos y seleccionar la aplicación que se quiere utilizar a través de la interface abierta.

