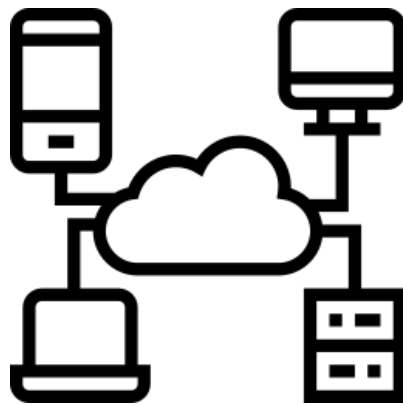


Sistemas distribuidos

PROYECTO YOUTUBE-DL



Crhistian Beteta Navarro & Jorge Muñoz Martínez

Índice

Manual del usuario	1
Inicio rápido.....	1
Entorno de trabajo.....	1
Ejecución de la práctica.....	1
Ejecución del Icegridgui	1
Ejecución del servidor.....	1
Ejecución del cliente.....	4
Comandos del cliente	4
Descargar una canción	4
Listar las canciones descargadas	5
Descargar el local una canción.	5
Cancelar una descarga.....	5
Estado de las canciones	5
Crear servidor de descarga	5
Listar los servidores disponibles	5
Eliminar un servidor de descarga.....	6
Desconectarse del servidor.....	6
Salir del programa	6
Archivo "Makefile"	7
Bibliografía.....	8
Agradecimiento.....	9

Manual del usuario

Inicio rápido

<https://youtu.be/rYBKeKEtPmA>

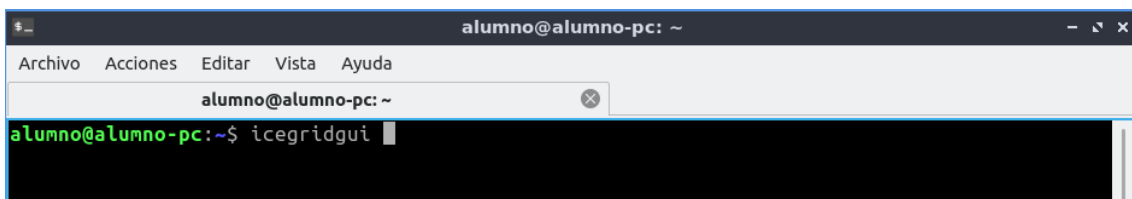
Entorno de trabajo

- Terminal 1: ejecución del servidor.
- Terminal 2: ejecución del cliente.
- Icegridgui: interfaz gráfica de monitorización de servidores.

Ejecución de la práctica

Ejecución del Icegridgui

En una terminal, sin importar la ubicación de esta, se escribe **“icegridgui”** y se presiona Enter.

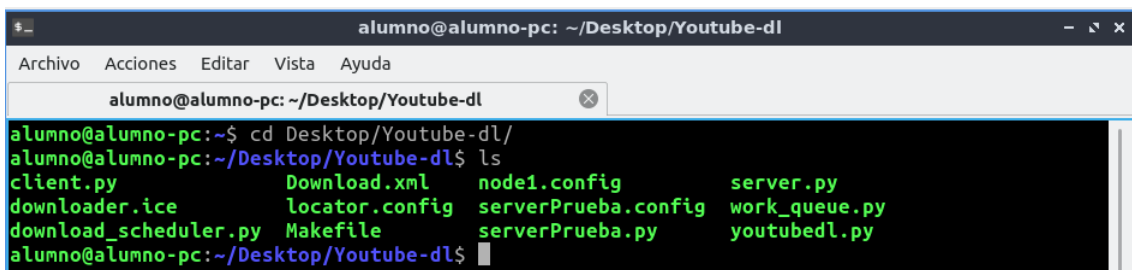


```

alumno@alumno-pc: ~
Archivo Acciones Editar Vista Ayuda
alumno@alumno-pc: ~
alumno@alumno-pc:~$ icegridgui
  
```

Ejecución del servidor.

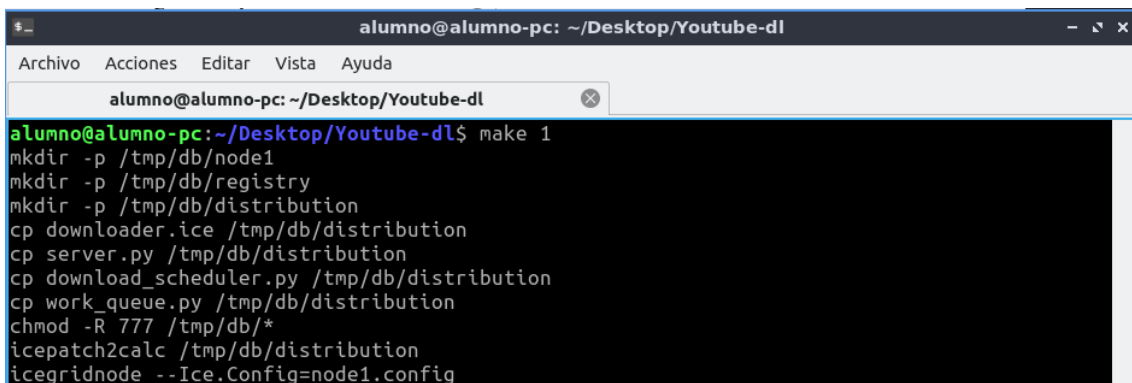
Se debe abrir una terminal localizada en los archivos de la práctica.



```

alumno@alumno-pc: ~/Desktop/Youtube-dl
Archivo Acciones Editar Vista Ayuda
alumno@alumno-pc: ~/Desktop/Youtube-dl
alumno@alumno-pc:~$ cd Desktop/Youtube-dl/
alumno@alumno-pc:~/Desktop/Youtube-dl$ ls
client.py          Download.xml      node1.config      server.py
downloader.ice     locator.config   serverPrueba.config  work_queue.py
download_scheduler.py Makefile         serverPrueba.py    youtubedl.py
alumno@alumno-pc:~/Desktop/Youtube-dl$
  
```

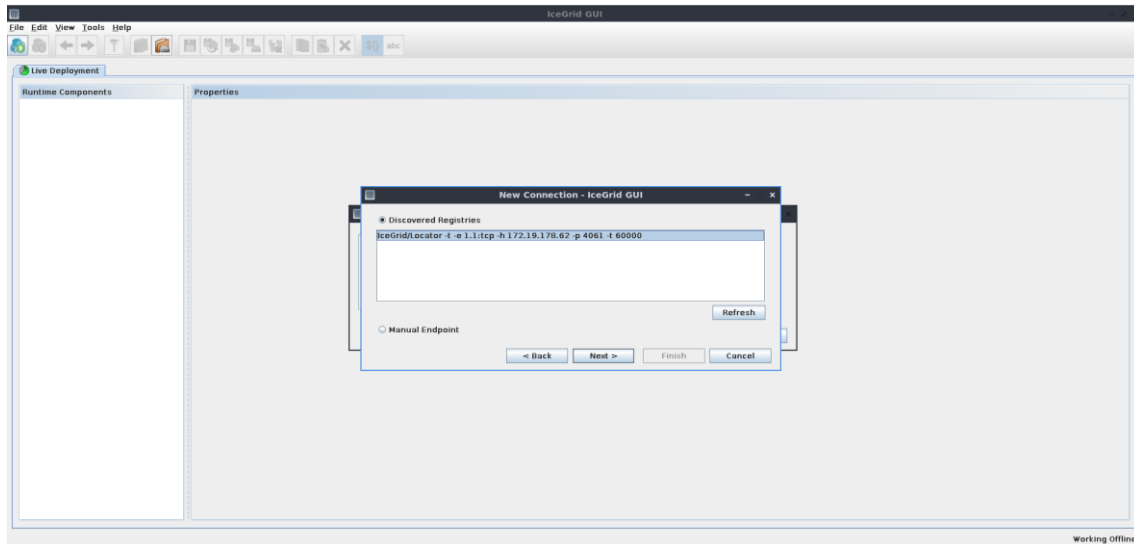
Escribir en el terminal **“make 1”**, el contenido del archivo **“Makefile”** se explica en [“Archivo Makefile”](#), como resultado se obtiene:



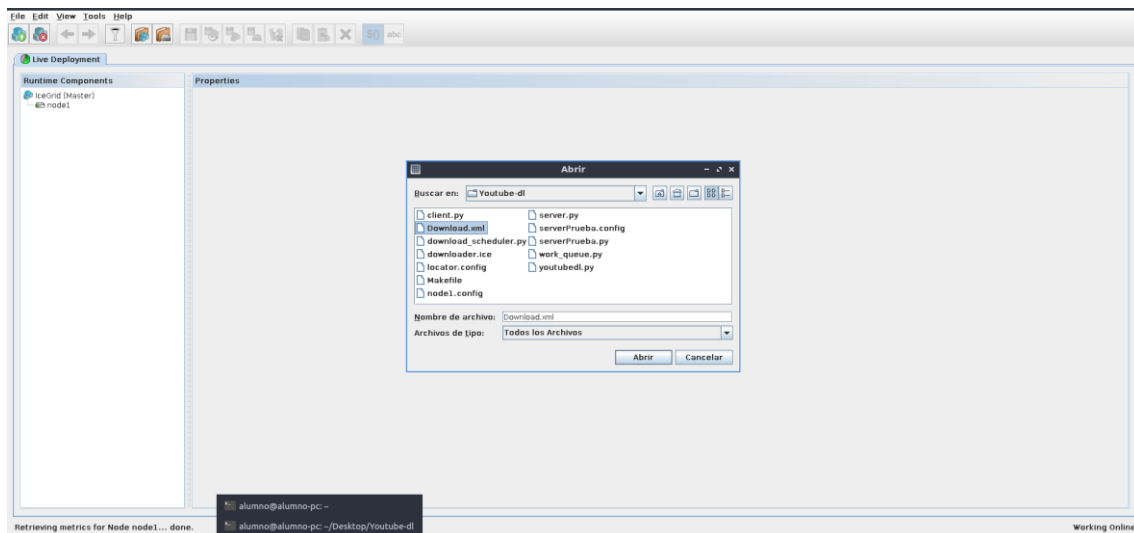
```

alumno@alumno-pc: ~/Desktop/Youtube-dl
Archivo Acciones Editar Vista Ayuda
alumno@alumno-pc: ~/Desktop/Youtube-dl
alumno@alumno-pc:~/Desktop/Youtube-dl$ make 1
mkdir -p /tmp/db/node1
mkdir -p /tmp/db/registry
mkdir -p /tmp/db/distribution
cp downloader.ice /tmp/db/distribution
cp server.py /tmp/db/distribution
cp download_scheduler.py /tmp/db/distribution
cp work_queue.py /tmp/db/distribution
chmod -R 777 /tmp/db/*
icepatch2calc /tmp/db/distribution
icegridnode --Ice.Config=node1.config
  
```

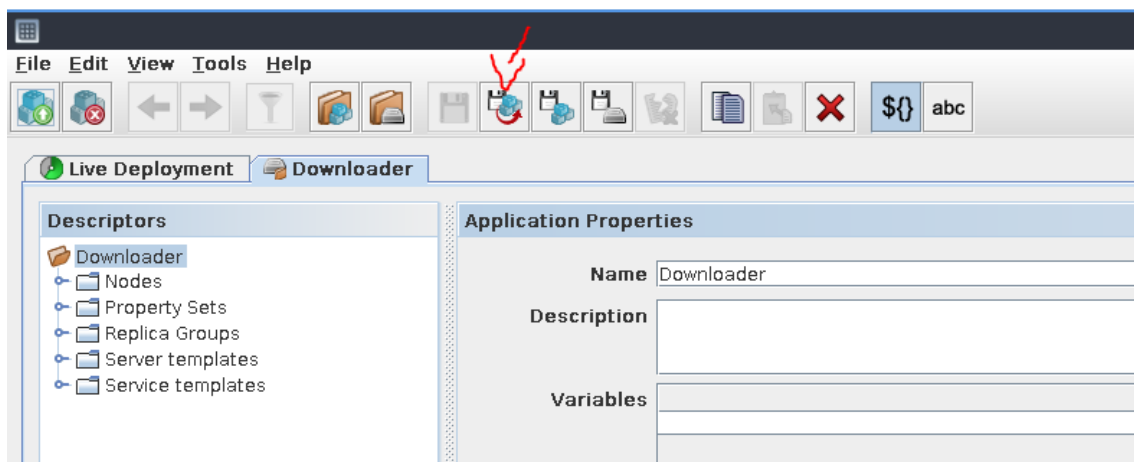
Conexión al registro creado tras la ejecución del icegridgui.

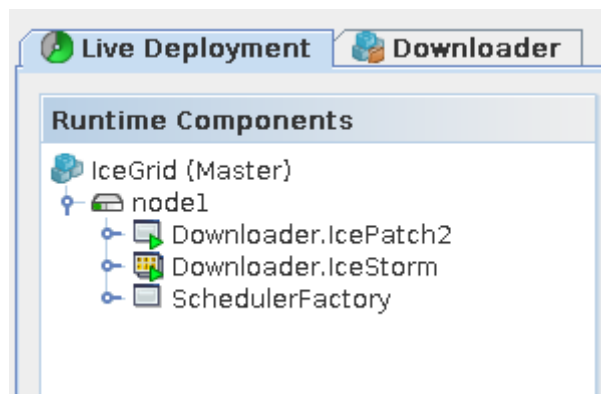
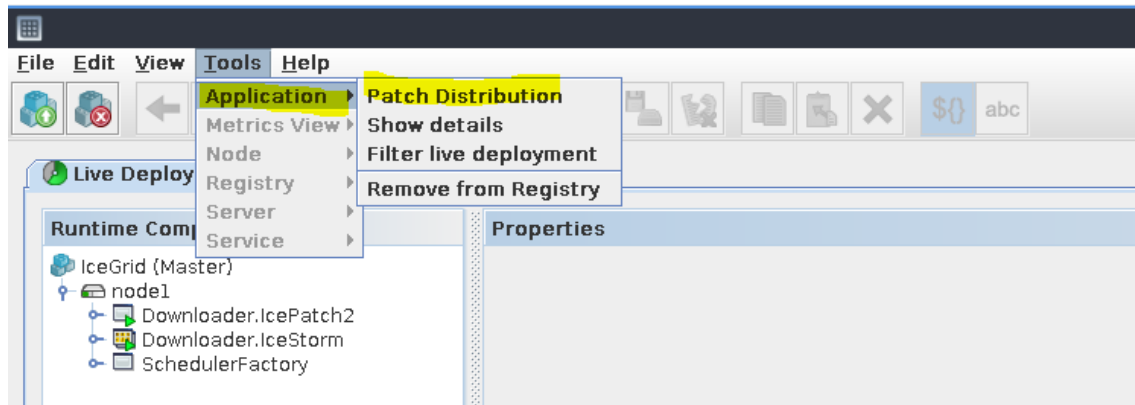


Con la conexión establecida, cargaremos el archivo xml **“Download.xml”** que contiene la configuración del programa.

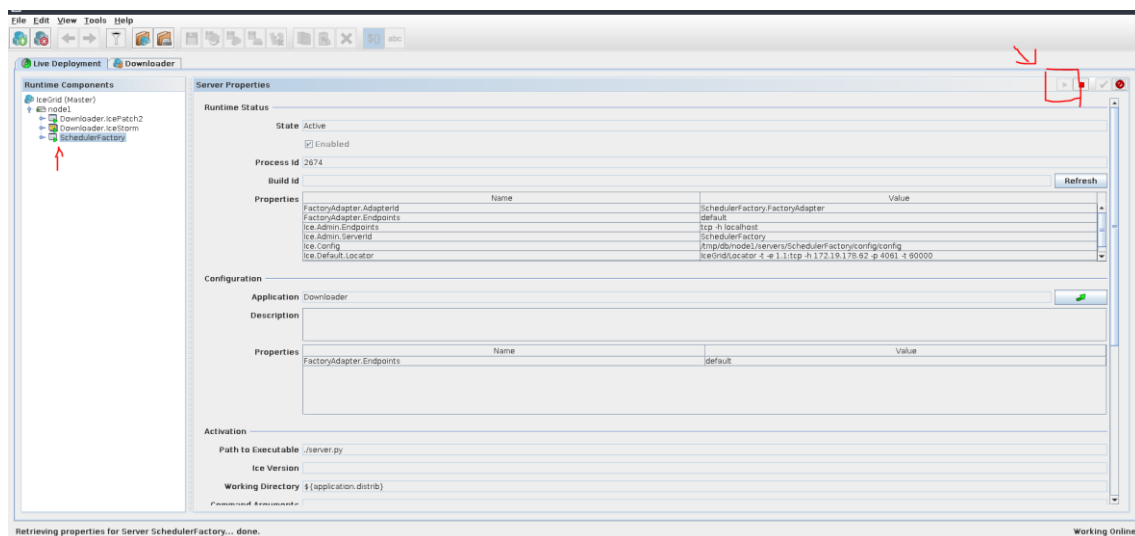


Como levantar los nodos.





Tanto el “**Downloader.IcePatch2**” y el “**Downloader.IceStorm**” se ejecutan directamente cuando se ejecuta la distribución, pero el “**SchedulerFactory**” hay que levantarlo a mano.



Si el levantamiento de los nodos termino de manera satisfactoria, en la terminal donde se ejecutó el “**make 1**” aparecerá una ristra de caracteres, esos caracteres son la dirección del proxy donde el cliente debe realizar la conexión para poder realizar operaciones.

```
alumno@alumno-pc:~/Desktop/VFYoutube$ make 1
mkdir -p /tmp/db/node1
mkdir -p /tmp/db/registry
mkdir -p /tmp/db/distribution
cp downloader.ice /tmp/db/distribution
cp server.py /tmp/db/distribution
cp download_scheduler.py /tmp/db/distribution
cp work_queue.py /tmp/db/distribution
chmod -R 777 /tmp/db/*
icepatch2calc /tmp/db/distribution
icegridnode --Ice.Config=node1.config
5E4DCE7B-29BF-4202-AC6A-FCBB92E98B9B -t -e 1.1 @ SchedulerFactory.FactoryAdapter
```

Tenemos ya los nodos levantados y la dirección del proxy donde debe conectarse, ahora tenemos que trabajar en el cliente.

Ejecución del cliente

En la terminal 2 escribimos “**./client.py --Ice.Config=locator.config**”, en una terminal alojada en la dirección de los archivos, con ese comando estamos ejecutando el cliente con una configuración almacenada en el archivo “**locator.config**”.

```
alumno@alumno-pc:~/Desktop/VFYoutube$ ./client.py --Ice.Config=locator.config
Client> █
```

Podemos ver como se nos abrió una terminal Shell, podemos ver los comandos disponibles si escribimos “**help**”, en el apartado [“Comandos del cliente”](#) se explica la funcionalidad de cada comando.

```
alumno@alumno-pc:~/Desktop/VFYoutube$ ./client.py --Ice.Config=locator.config
Client> help

Documented commands (type help <topic>):
=====
add_download_song      del_scheduler          get_download_song      list_schedulers
cancell_download_song  disconnect             help                   new_scheduler
connect               exit                   list_download_songs    status_song

Client> █
```

Para que el cliente pueda trabajar con el servidor es necesario que este se conecte con el servidor para ellos tendremos que escribir en el terminal del Cmd “**connect (+ dirección del Proxy obtenido en el servidor)**”.

```
Client> connect 5E4DCE7B-29BF-4202-AC6A-FCBB92E98B9B -t -e 1.1 @ SchedulerFactory.FactoryAd
apter
using icestorm in: 'Downloader.IceStorm/TopicManager'
Client (connected)> █
```

Si en el Shell aparece “**Client (connected)**” el cliente está conectado con el servidor, en ese momento cuando se pueden realizar operaciones con el servidor.

Comandos del cliente

Descargar una canción

Descarga una canción dada una url.

```
Client (connected)> add_download_song https://www.youtube.com/watch?v=BBGEG21CGo0
***Downloading song***
***New Scheduler created***
Scheduler name: 26f39991-3bb5-40b1-9fa5-26c52b21e872
Scheduler proxy: 26f39991-3bb5-40b1-9fa5-26c52b21e872 -t -e 1.1 @ SchedulerFactory.Fac
toryAdapter
```

Comando “***add_download_song + (url del video).***”

Listar las canciones descargadas

Muestra las canciones descargadas.

```
Client (connected)> list_download_songs
Song: ./HD Epic Sax Gandalf.mp3
Client (connected)> █
```

Comando: “***list_download_songs***”.

Descargar el local una canción.

Mueve a la carpeta del programa una canción descargada dada una url.

```
Client (connected)> get_download_song ./HD Epic Sax Gandalf.mp3
Client (connected)> █
```



Comando: “***get_download_song + (nombre de la canción).***”

Cancelar una descarga

Cancela una descarga dada una url.

Comando: “***cancell_download_song + (url descarga).***”

Estado de las canciones

Muestra el estado de las canciones.

```
Client (connected)> status_song
Song: https://www.youtube.com/watch?v=BBGEG21CGo0 -> DONE
Client (connected)> █
```

```
Client (connected)> add_download_song https://www.youtube.com/watch?v=QH2-TGUlwu4
***Downloading song***
Client (connected)> status_song
Song: https://www.youtube.com/watch?v=BBGEG21CGo0 -> DONE
Song: https://www.youtube.com/watch?v=QH2-TGUlwu4 -> INPROGRESS
Client (connected)> █
```

Comando: “***status_song***”.

Crear servidor de descarga

Crea un nuevo servidor de descarga.

```
Client (connected)> new_scheduler
***New Scheduler created***
Scheduler name: 9fdcec09-4a53-4087-b1ae-f53766f7332a
Scheduler proxy: 9fdcec09-4a53-4087-b1ae-f53766f7332a -t -e 1.1 @ SchedulerFactory.FactoryAdapter
Client (connected)> █
```

Comando: “***new_scheduler***”.

Listar los servidores disponibles

Lista los servidores de descarga disponibles.

```
Client (connected)> list_schedulers
***Schedulers created***
    Scheduler name: 26f39991-3bb5-40b1-9fa5-26c52b21e872
    Scheduler name: 9fdcec09-4a53-4087-b1ae-f53766f7332a
Client (connected)> █
```

Comando: ***"list_schedulers"***.

Eliminar un servidor de descarga

Elimina un servidor de descarga dado nombre de servidor.

```
Client (connected)> list_schedulers
***Schedulers created***
    Scheduler name: 26f39991-3bb5-40b1-9fa5-26c52b21e872
    Scheduler name: 9fdcec09-4a53-4087-b1ae-f53766f7332a
Client (connected)> del_scheduler 9fdcec09-4a53-4087-b1ae-f53766f7332a
    Trying to delete: 9fdcec09-4a53-4087-b1ae-f53766f7332a
    Scheduler delete
Client (connected)> list_schedulers
***Schedulers created***
    Scheduler name: 26f39991-3bb5-40b1-9fa5-26c52b21e872
```

Comando: ***"del_scheduler + (nombre del scheduler)"***.

Desconectarse del servidor.

El cliente se desconecta del servidor.

```
Client (connected)> disconnect
Client> █
```

Comando: ***"disconnect"***.

Salir del programa

Finalizar la ejecución del cliente.

```
Client> exit
alumno@alumno-pc:~/Desktop/VFYoutube$ █
```

```
alumno@alumno-pc:~/Desktop/VFYoutube$ ./client.py --Ice.Config=locator.config
Client> connect 5E4DCE7B-29BF-4202-AC6A-FCBB92E98B9B -t -e 1.1 @ SchedulerFactory.FactoryAdapter
using icestorm in: 'Downloader.IceStorm/TopicManager'
Client (connected)> exit
alumno@alumno-pc:~/Desktop/VFYoutube$ █
```

Comando: ***"exit"***.

Archivo “Makefile”

```
clean:
    rm -rf /tmp/db/node1
    rm -rf /tmp/db/registry
    rm -rf /tmp/db/distribution

stop:
    sudo systemctl stop icegridregistry
    sudo systemctl stop icegridnode

1:
    mkdir -p /tmp/db/node1
    mkdir -p /tmp/db/registry
    mkdir -p /tmp/db/distribution
    cp downloader.ice /tmp/db/distribution
    cp server.py /tmp/db/distribution
    cp download_scheduler.py /tmp/db/distribution
    cp work_queue.py /tmp/db/distribution
    chmod -R 777 /tmp/db/*
    icepatch2calc /tmp/db/distribution
    icegridnode --Ice.Config=node1.config
```

- **Make 1:** Este método realiza varias funciones:
 - Crea los directorios “node1”, “registry” y “distribution” en el directorio “/tmp/db”.
 - Copia los archivos contenidos en el directorio actual en el directorio “/tmp/db/distribution”.
 - Da permisos de ejecución, lectura y escritura a los archivos en el directorio “/tmp/db/”.
 - Prepara el directorio “/tmp/db/distribution” para ser transmitido por el servidor.
 - Inicio del servidor “icegridnode” con el archivo de configuración “node1.config”
- **Make stop:** Para los procesos “icegridregistry” e “icegridnode”.
- **Make clean:** Elimina las carpetas creadas “node1”, “registry” y “distribution” en el directorio “/tmp/db”.

Bibliografía

<https://doc.zeroc.com/ice/3.7/ice-services/icegrid/icegrid-gui-tool>

<https://doc.zeroc.com/ice/3.7/ice-services/icepatch2/using-icepatch2calc>

<https://doc.zeroc.com/ice/3.7/ice-services/icegrid/icegrid-server-reference/icegridnode>

Agradecimiento

En primer lugar, agradecer a los profesores que imparten las prácticas por su rápida respuesta en el foro y por su ayuda por hangouts.

En segundo lugar, al grupo de Francisco de la Mata por ayudarnos mutuamente a la hora de realizar la práctica.