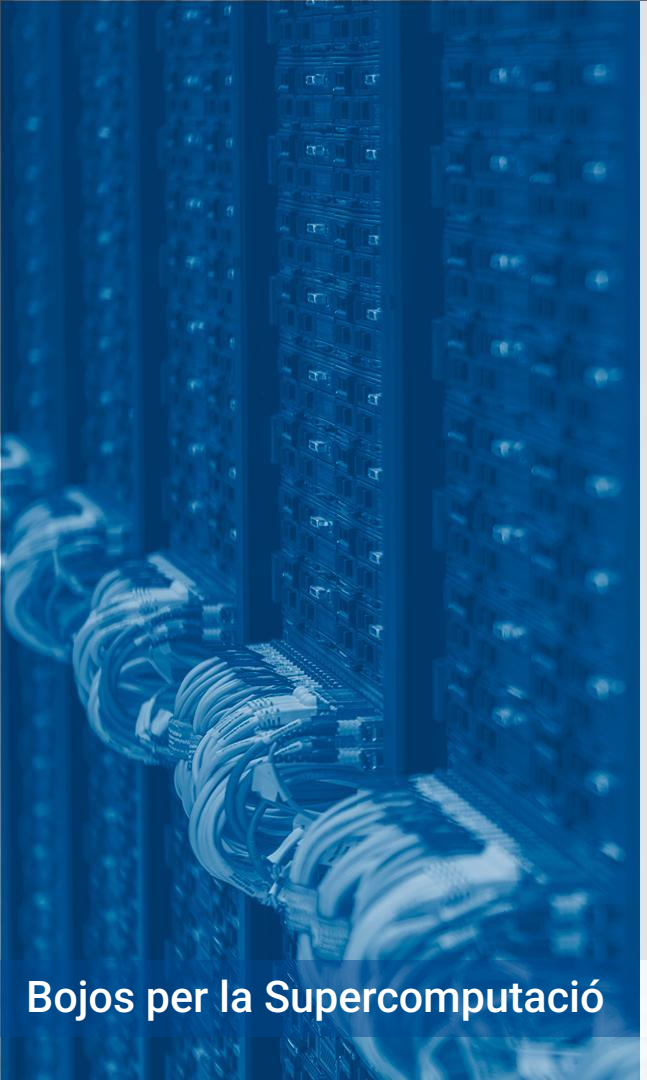


Barcelona Supercomputing Center
Centro Nacional de Supercomputación

Taller 2:

Hands-on con Salamandra en MareNostrum 5.

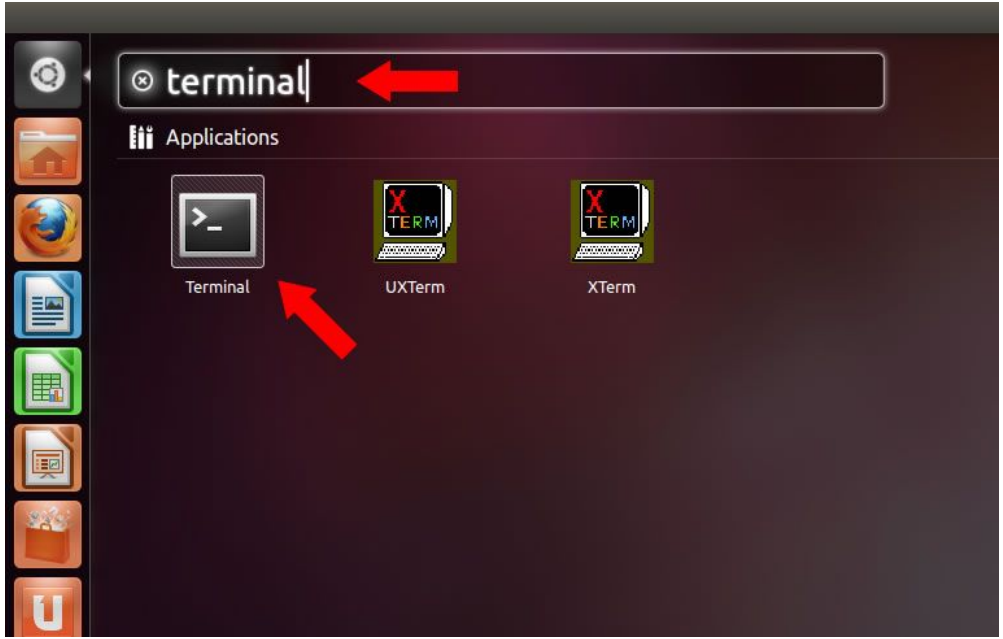
Bojos per la Supercomputació

Valle Ruiz-Fernández - Adrián Rubio Pintado

Setup

Conexión a MareNostrum 5

El primer paso es abrir nuestra terminal:



Conexión a MareNostrum 5

Usamos para ello SSH: Secure Shell

El comando ssh nos permite conectarnos de manera segura una máquina remota.

```
ssh <USER_NAME>@alogin4.bsc.es
```

Cada alumno deberá de disponer de un USER_NAME de la forma nct01XXX

Conexión a MareNostrum 5

El sistema nos pedirá la contraseña:

```
arubio@bsc-848850144:~$ ssh nct00003@acc4
nct00003@alogin4.bsc.es's password:
```

Conexión a MareNostrum 5

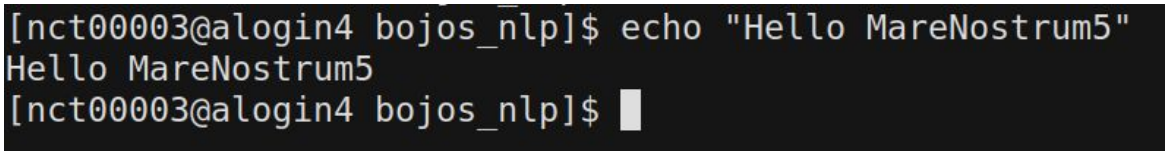
Si la conexión se ha realizado con éxito recibiremos un mensaje de bienvenida del MareNostrum como el que vemos a continuación:

Conexión a MareNostrum 5

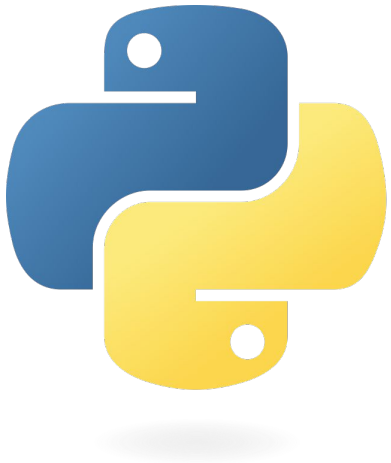
Podemos ver como ya podemos usar la máquina con el siguiente comando:

```
echo "Hello MareNostrum5"
```

Recibiremos una salida como la siguiente:



Creando nuestro entorno Python



Creamos nuestra carpeta para el proyecto:

```
mkdir modelos_nlp

cd modelos_nlp
```

Y creamos un entorno de ejecución mediante:

```
# Creamos el entorno de ejecución:

python -m venv nlp_env

# A continuación activamos el entorno
source nlp_env/bin/activate
```

Veremos que está activado correctamente si precede cada línea de entrada tal que:

```
[nct00003@alogin4 bojos_nlp]$ source nlp_env/bin/activate
(nlp_env) [nct00003@alogin4 bojos_nlp]$
```

Creando nuestro entorno Python

Descargamos el código necesario con el siguiente comando:

```
git clone https://github.com/adriwitek/Bojos_per_la_Supercomputacio.git
```

Nos movemos a la siguiente carpeta:

```
cd Bojos_per_la_Supercomputacio/Taller_2/src
```

Instalación de las librerías necesarias.

Instalamos la librerías necesarias mediante:

```
python -m pip install -r requirements.txt
```

Podemos ver en todo caso si se han instalado correctamente mediante el siguiente comando:

```
python -m pip freeze
```

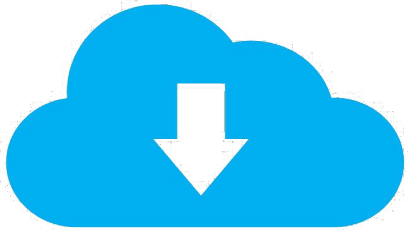
Deberían aparecernos varías líneas indicando librería y versión. Por ejemplo para el caso de la librería de transformers:

```
transformers==4.51.3
```

Escogiendo nuestro modelo

- Salamandra es un conjunto de modelos fundacionales open source basados en la arquitectura Decoder-Only Transformer.
- Ha sido desarrollado por nuestro grupo Language Technologies Laboratory :D
- Salamandra ha sido entrenado usando datos multilingües de 35 idiomas europeos distintos.
- En este caso escogeremos el modelo más pequeño con 2B de parámetros.

Descargamos el modelo Salamandra 2B



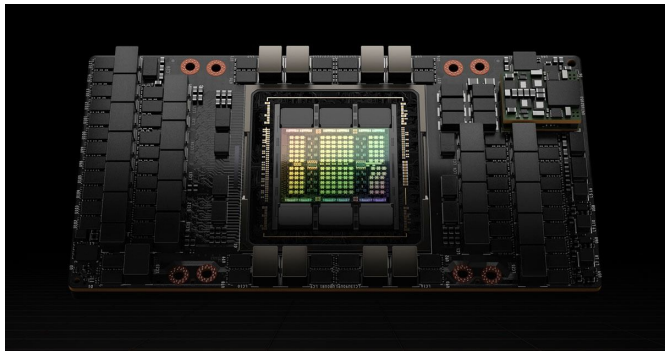
Instalamos la librería Hugging Face Transformers

```
# Cargamos git-lfs para poder descargar archivos grandes

module load git-lfs

git lfs install

git clone https://huggingface.co/BSC-LT/salamandra-2b-instruct
```

Veremos una pantalla de descarga como la siguiente:

```
Clonando en 'salamandra-2b-instruct'...
remote: Enumerating objects: 195, done.
remote: Counting objects: 100% (88/88), done.
remote: Compressing objects: 100% (87/87), done.
remote: Total 195 (delta 43), reused 0 (delta 0), pack-reused 107 (from 1)
Recibiendo objetos: 100% (195/195), 845.81 KiB | 3.92 MiB/s, listo.
Resolviendo deltas: 100% (97/97), listo.
Filtrando contenido:  75% (3/4)
```

Estableciendo los recursos de ejecución necesarios.



¡Listo! Ahora solo necesitamos pedirle recursos al ordenador. Concretamente una gráfica. Lo hacemos mediante el siguiente comando:

```
salloc -N 1 --qos acc_training --partition acc  --cpus-per-task 20 --account nct_320 --gres gpu:1
```

Estableciendo los recursos de ejecución necesarios.

Esperamos a que se nos proporcione los recursos que hemos solicitado. Veremos un mensaje como el siguiente:

```
[nct00005@as01r1b16 ~]$ salloc -N 1 --qos acc_training --partition acc  --cpus-per-task 20 --account nct_320 --gres gpu:1
salloc: Pending job allocation 19998427
salloc: job 19998427 queued and waiting for resources
salloc: job 19998427 has been allocated resources
salloc: Granted job allocation 19998427
salloc: Waiting for resource configuration
salloc: Nodes as01r3b28 are ready for job
unload bsc/1.0 (PATH, MANPATH)
load bsc/1.0 (PATH, MANPATH)
[nct00005@as01r3b28 ~]$
```

Cargamos nuestro modelo

Ya podemos ejecutar nuestro script de inferencia:

```
python inference_salamandra2b.py
```

Si todo ha salido correcto veremos una salida como la siguiente:

```
------------------------------------------------------------
system
I am Salamandra, an AI language model developed at the Barcelona Supercomputing Centre (BSC) by the Language Technologies Unit. My know
ledge base was last updated on August 2023. Today Date: 2025-04-26
Soy Salamandra, un modelo lingüístico de IA desarrollado en el Barcelona Supercomputing Centre (BSC) por la Language Technologies Unit.
 Mi base de conocimientos se actualizó por última vez en agosto de 2023.
Soc Salamandra, un model de llenguatge d'IA desenvolupat al Barcelona Supercomputing Centre (BSC) per la Language Technologies Unit. La
 meva base de coneixement es va actualitzar per última vegada l'agost de 2023.
 user
¿Quién escribió Cien años de soledad y en qué año se publicó por primera vez?
 assistant
Cien años de soledad fue escrito por Gabriel García Márquez y publicado por primera vez en 1967.
(nlp_env) [nct00003@alogin4 bojos_nlp]$
```

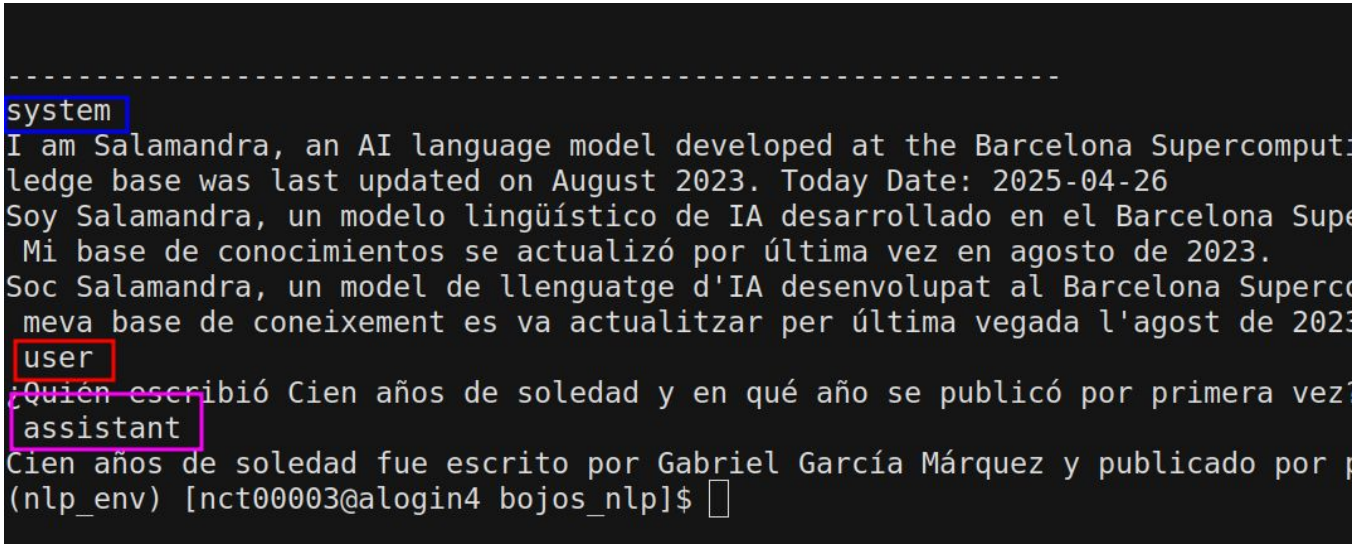
Inferencia con Salamandra-2b Instruct

Aplicando el Chat Template

- ➔ Acabamos de ejecutar un pequeño script para poder “hablar” con el modelo. Para ello lo que hemos hecho ha sido aplicar un Chat Template.
- ➔ Mediante el Chat Template le decimos al modelo de modo estructurado que se comporte como un chatbot.
- ➔ Habitualmente un Chat Template tiene 3 partes (roles) principales:
  - System: Le indicamos al modelo que es un chatbot.
  - User: Contiene el texto que el usuario escribe al modelo.
  - Assistant: Contiene la respuesta del modelo a la pregunta anterior del usuario. Esto es lo que tiene que generar realmente el modelo.

Aplicando el ChatTemplate

Podemos identificar los 3 roles descritos en la ejecución de nuestro script:

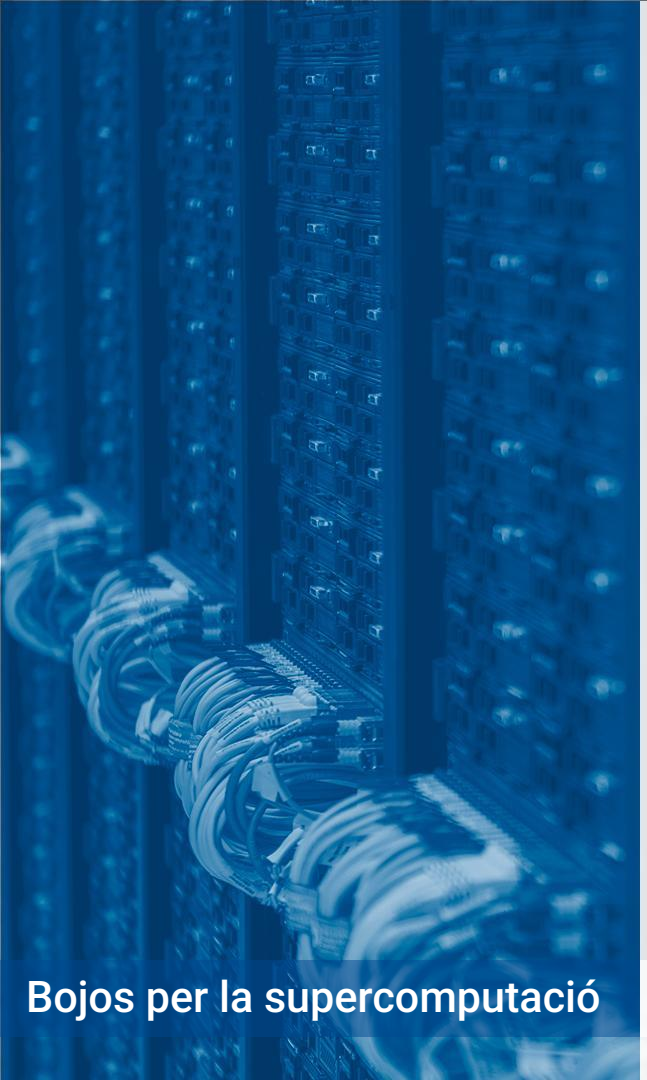



```
------------------------------------------------------------
system
I am Salamandra, an AI language model developed at the Barcelona Supercomputi
ledge base was last updated on August 2023. Today Date: 2025-04-26
Soy Salamandra, un modelo lingüístico de IA desarrollado en el Barcelona Supe
 Mi base de conocimientos se actualizó por última vez en agosto de 2023.
Soc Salamandra, un model de llenguatge d'IA desenvolupat al Barcelona Superc
 meva base de coneixement es va actualitzar per última vegada l'agost de 202
user
¿Quién escribió Cien años de soledad y en qué año se publicó por primera vez
assistant
Cien años de soledad fue escrito por Gabriel García Márquez y publicado por p
(nlp_env) [nct00003@alogin4 bojos_nlp]$
```

Testeando Salamandra-2b Instruct

Evaluando el modelo

- A continuación deberéis rellenar un formulario que os proporcionaremos, para que evaluéis de primera mano el modelo Salamandra.
- Cambiando el mensaje del user, le pediréis al modelo que realice varias tareas:
  - generación de texto, traducción, resumen de texto, etc.
- Se os proporcionará un ejemplo para cada una de ellas. Podéis probar con uno vuestro a continuación.
- Deberéis rellenar el formulario evaluando al modelo en función de las respuestas generadas.



Barcelona Supercomputing Center
Centro Nacional de Supercomputación

La inteligencia artificial aplicada al lenguaje natural

Bojos per la supercomputació

Valle Ruiz-Fernández - Adrián Rubio Pintado