



# Lo que aprendí en el pasaje a MendietaF2

Nicolás Hormann - Data Scientist en fligoo

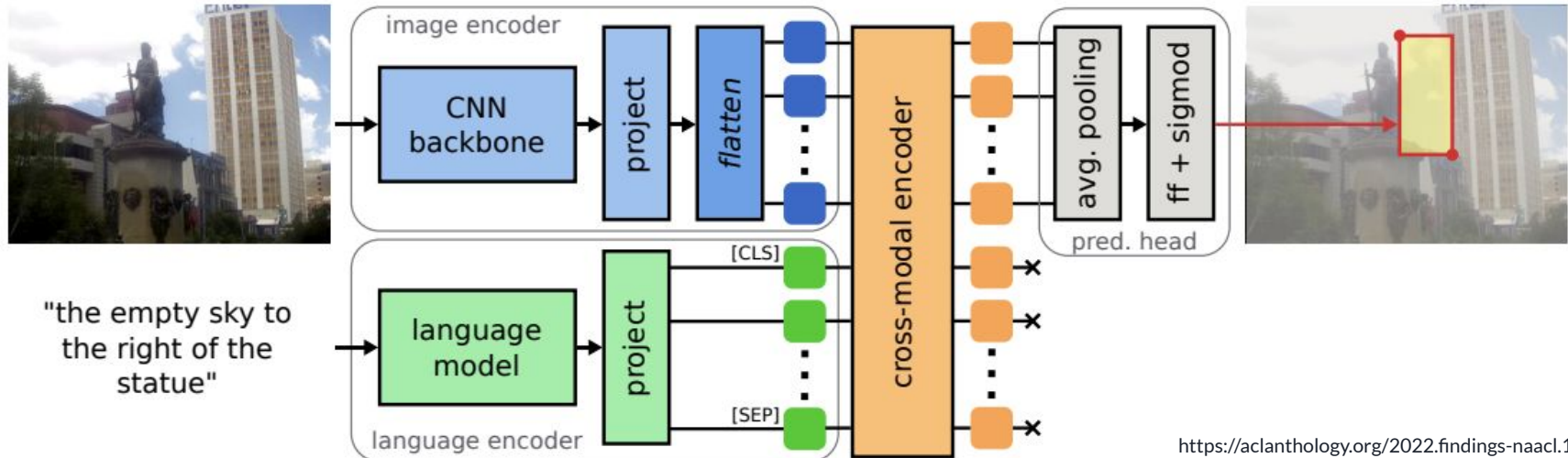


# Agenda

- Contexto y mi historia de Nabu a Mendieta
- El proceso de entender Mendieta.
- Script que uso + minidemo de cómo es el flujo hasta que pongo a correr un job.
- Status actual y algunos tips.
- Resumen de ayudas.

# Contexto

Tesis de Grado usando python, torch y datasets grandes\*. Environment: miniconda.



\*Dado al volúmen de datos y arquitectura usada, es necesario cómputo de GPU para acelerar la experimentación.



# Contexto - Historia en Nabu 1 y 2

## Nabu 1

- 2 GPUS
- Mucha RAM

## Nabu 2

- 3 GPUS
- Menos RAM que nabu 1



## Limitaciones:

- Explota en RAM. Muerte de procesos.
- Compartido entre varios users.



# El proceso - migración a MendietaF2

## Pedido de Cuentas

Para pedir nuevas cuentas en los recursos computacionales del CCAD se deberá completar el siguiente **formulario**.

- Te llega un mail con instrucciones para la conexión y el uso de todas las compus, entre ellas MendietaF2
- Conexión a través de SSH.
- A partir de este correo ya podemos conectarnos a Mendieta y empezar a renegar ;)

# El proceso - Entendiendo cómo corren los procesos

En Nabu:

- Activar environment.
- Instalar dependencias.
- Correr los experimentos.



En Mendieta:

- Preparar script de bash.
- Lanzar script.
- Esperar a que tu tarea empiece a correr.



# El proceso - Entendiendo a Mendieta

En Nabu:

- 3 GPU's
  - Si estaban disponibles, las usaba.
- Mucho kill de procesos cuando no daba más.



En Mendieta:

- 1 nodo tiene 2 GPU's, dedicadas a mi user exclusivamente.
- Máx: 4 nodos, 8 GPU's dedicadas a mi user.
- Nodo cabecera (sin GPU)
  - De acá llamamos a nuestro script usando slrun.

# Mi script para correr en mendietaF2

```
tools > $ run_mendieta_train.sh
```

```
1  #!/bin/bash
2  #SBATCH --job-name=exp_027 # short name for your job
3  #SBATCH --mail-type=ALL
4  #SBATCH --mail-user=nicolas.hormann@mi.unc.edu.ar
5  #SBATCH --output=slurm-%x.%j.out # %j job id, %x job name
6  #SBATCH --error=slurm-%x.%j.err
7  #SBATCH --partition=multi
8  #SBATCH --nodes=1           # node count
9  #SBATCH --ntasks=1        # total number of tasks across all nodes
10 #SBATCH --gres=gpu:2
11 #SBATCH --cpus-per-task=20 # cpu-cores per task (>1 if multi-threaded tasks)
12 #SBATCH --time=1-15:00    # total run time limit (<days>-<hours>:<minutes>)
13
14 . /etc/profile
15 module purge
16 ulimit -c unlimited # core dump
17 ulimit -s unlimited # stack
18
19 source ${HOME}/.bashrc
20 conda activate rec-env
21
22 export TRANSFORMERS_OFFLINE=1
23 export NCCL_DEBUG=INFO
24 export NCCL_IB_DISABLE=1
25 export NCCL_SOCKET_IFNAME=lo
26
27 cd ${HOME}/rec/code/rec/
28
29 srun yaer run -e exp_027
--
```

→ Configuración de los recursos

→ Activar environment.

→ Variables de entorno.

→ Corremos con srun.



Demo - tranqui



# Un error que me hizo renegar

```
30 | self._call_and_handle_interrupt(  
31 | File "/home/nhormann/miniconda3/envs/rec-env/lib/python3.9/site-packages/pytorch_lightning/trainer/trainer.py", line 719, in _call_and_handle_interrupt  
32 |     return self.strategy.launcher.launch(trainer_fn, *args, trainer=self, **kwargs)  
33 | File "/home/nhormann/miniconda3/envs/rec-env/lib/python3.9/site-packages/pytorch_lightning/strategies/launchers/subprocess_script.py", line 93, in launch  
34 |     return function(*args, **kwargs)  
35 | File "/home/nhormann/miniconda3/envs/rec-env/lib/python3.9/site-packages/pytorch_lightning/trainer/trainer.py", line 809, in _fit_impl  
36 |     results = self._run(model, ckpt_path=self.ckpt_path)  
37 | File "/home/nhormann/miniconda3/envs/rec-env/lib/python3.9/site-packages/pytorch_lightning/trainer/trainer.py", line 1170, in _run  
38 |     self._setup_profiler()  
39 | File "/home/nhormann/miniconda3/envs/rec-env/lib/python3.9/site-packages/pytorch_lightning/trainer/trainer.py", line 1795, in __setup_profiler  
40 |     self.profiler.setup(stage=self.state.fn._setup_fn, local_rank=local_rank, log_dir=self.log_dir)  
41 | File "/home/nhormann/miniconda3/envs/rec-env/lib/python3.9/site-packages/pytorch_lightning/trainer/trainer.py", line 2232, in log_dir  
42 |     dirpath = self.strategy.broadcast(dirpath)  
43 | File "/home/nhormann/miniconda3/envs/rec-env/lib/python3.9/site-packages/pytorch_lightning/strategies/ddp.py", line 311, in broadcast  
44 |     torch.distributed.broadcast_object_list(obj, src, group=group.WORLD)  
45 | File "/home/nhormann/miniconda3/envs/rec-env/lib/python3.9/site-packages/torch/distributed/distributed_c10d.py", line 1734, in broadcast_object_list  
46 |     broadcast(object_sizes_tensor, src=src, group=group)  
47 | File "/home/nhormann/miniconda3/envs/rec-env/lib/python3.9/site-packages/torch/distributed/distributed_c10d.py", line 1090, in broadcast  
48 |     work = default_pg.broadcast([tensor], opts)  
49 | RuntimeError: NCCL error in: ../torch/lib/c10d/ProcessGroupNCCL.cpp:911, internal error, NCCL version 2.7.8
```



```
24 export NCCL_IB_DISABLE=1  
25 export NCCL_SOCKET_IFNAME=lo
```

# El hoy - Corriendo exps a lo loco

- Al menos 25 experimentos corrieron **OK** en la primera quincena de octubre.
- Cada experimento usó un nodo con 2 GPU's.

```
REC [SSH: MENDIETA.CCAD...]  
~/rec/models/exp_009  
└─ models  
  > exp_001  
  > exp_003  
  > exp_004  
  > exp_005  
  > exp_006  
  > exp_007  
  > exp_008  
  > exp_009  
  > exp_010  
  > exp_011  
  > exp_012  
  > exp_013  
  > exp_014  
  > exp_015  
  > exp_016  
  > exp_019  
  > exp_020  
  > exp_021  
  > exp_022  
  > exp_023  
  > exp_024  
  > exp_025  
  > exp_026  
  > exp_027  
  ◆ .gitkeep
```





# Algunos tips

- Tener un repositorio con el código versionado.
  - Un README paso a paso
  - Un archivo de requirements con las dependencias y usar conda / micromamba / docker.
  - Ventajas: facilitar la reproducibilidad de tus experimentos en cualquier máquina, no solo la tuya.
- Versionar los experimentos, si usás python te puede interesar usar [YAER](#).
- Asegurarse que tu experimento ocupa todos los recursos solicitados.
  - `ssh ivb<ID>`, ejemplo `ssh ivb14`
  - correr `htop`, `nvidia-smi`. Monitorear uso de los recursos.



# Resumen de fuentes y ayudas

- [Manual de uso de mendieta. Karen Palacio.](#)
- [Wiki del CCAD.](#)
- Usar los canales disponibles (Zulip, lista de usuarios)

Gracias!

