

Software disponible en los clusters TeideHPC y AnagaGPU

Antes de pasar a leer la siguiente sección, le recomendamos echar un vistazo a la sección "[Herramienta de módulos](#)"

El sistema de Environment Modules utiliza un conjunto de comandos que incluyen:

- `module overview` : Muestra una lista general resumida de los módulos disponibles.
- `module avail` : Muestra todos los módulos disponibles en el sistema para cargar.
- `module avail [módulo]` : Muestra los módulos disponibles con ese nombre.
- `module spider [módulo]` : Realiza una búsqueda de un módulo concreto.
- `module load [módulo]` : Carga un módulo específico, lo que establece las variables de entorno necesarias para utilizar el software que el módulo representa.
- `module unload [módulo]` : Descarga un módulo, eliminando las variables de entorno asociadas con el software de la sesión del usuario.
- `module swap [módulo_actual] [módulo_nuevo]` : Intercambia un módulo cargado por otro, lo que es útil para cambiar rápidamente de versiones de software.
- `module purge` : Elimina todos los módulos cargados, limpiando el entorno del usuario de cualquier configuración realizada por módulos.
- `module help [módulo]` : Proporciona información de ayuda para un módulo específico, lo que puede incluir información sobre cómo usar el software, sus variables de entorno, etc.
- `module whatis [módulo]` : Muestra una breve descripción de lo que hace un módulo.
- `module show [módulo]` : Muestra información sobre lo que hace el módulo al entorno del usuario; es decir, qué variables de entorno ajusta y cómo.

■ En TeideHPC existen 2 clusters, TeideHPC y AnagaGPU

Por este motivo, el conjunto de módulos disponibles **depende del Cluster que vaya a usar.**

Cada cluster tiene sus propios nodos de login y **no disponen del mismo software instalado.**

■ La ejecución de software está prohibida en los nodos de login

En los nodos de login TeideHPC y AnagaGPU está prohibido ejecutar cualquier aplicación salvo comandos comunes de linux.

■ *ml* es una abreviatura de *modules*

Para evitar errores comunes al escribir el comandos *modules* tales como *modles, moduls, mdules, etc* es posible usar la abreviatura **ml**.

```
ml spider ... ml load ... ml purge ...
```

Significado de algunos módulos significativos.

- **GCCcore**: conjunto básico de compiladores de la GNU Compiler Collection: C, C++, Objective-C, Fortran.
- **GCC**: colección de compiladores de GCCcore y librerías (GDB, binutils, glibc)
- **foss**: el módulo foss consiste en su totalidad en el término común *FOOS*, que es la abreviatura de *Free and Open source software*.
- **gompic**: toolchain que agrupa herramientas compiladas con GCC y OpenMPI
- **gompic**: toolchain que agrupa herramientas compiladas con GCC+CUDA y OpenMPI
- **intel-compilers**: Compiladores de Intel C, C++ & Fortran clasicos y oneAPI.
- **iimpi**: Compilador Intel C/C++ y Fortran con soporte MPI de intel.

Ejemplo de uso de modules

```
module overview  
or ml overview
```

```
-----/share/easybuild/software/common/modules/all -----
EasyBuild (2) Go (1) Mamba (1) Miniconda3 (2) Singularity (1) Squashfs (1) slurm
(1)
```

```
-----/share/easybuild/software/x86_64/modules/all -----
ADMIXTURE          (1) Imath          (1) UCX            (2) jupyter-server (1)
ATK                (1) Infernal      (1) UDUNITS        (2) libGLU          (2)
ATLAS              (1) JasPer          (4) USEARCH        (1) libarchive        (3)
AdmixTools         (2) Java            (5) UnZip           (2) libcerf           (1)
Armadillo          (2) JsonCpp         (1) WPS             (2) libdeflate        (1)
Arrow              (2) JupyterLab     (2) WRF             (2) libdrm            (2)
Autoconf           (4) LAME           (1) X11             (2) libepoxy          (1)
Automake           (4) LAMPLD         (2) XALT            (1) libevent          (1)
Autotools          (5) LAPACK         (1) XML-LibXML      (2) libfabric         (2)
BCFtools           (1) LERC            (1) XZ              (2) libffi            (2)
BEDTools           (1) LLVM           (2) Xerces-C++     (1) libgd             (1)
```

module avail
or ml av

```
----- /share/easybuild/software/common/modules/all -----
EasyBuild/4.7.0    Go/1.18.3        Miniconda3/22.11.1-1    Singularity/3.11.0    slurm/
teide
EasyBuild/4.8.2 (D)  Mamba/4.14.0-0  Miniconda3/23.5.2-0 (D)  Squashfs/4.3
```

```
----- /share/easybuild/software/x86_64/modules/all -----
ADMIXTURE/1.3.0          UCC/1.1.0-GCCcore-12.2.0
ATK/2.38.0-GCCcore-12.2.0 UCX/1.11.2-GCCcore-11.2.0
ATLAS/0.9.9-foss-2022b   UCX/1.13.1-GCCcore-12.2.0 (D)
AdmixTools/7.0.2-foss-2021b UDUNITS/2.2.28-GCCcore-11.2.0
AdmixTools/7.0.2-foss-2022b (D) UDUNITS/2.2.28-
GCCcore-12.2.0 (D)
Armadillo/10.5.3-foss-2022b USEARCH/11.0.667-i86linux32
Armadillo/11.4.3-foss-2022b (D) UnZip/6.0-GCCcore-11.2.0
Arrow/6.0.0-foss-2021b   UnZip/6.0-GCCcore-12.2.0 (D)
Arrow/11.0.0-gfbf-2022b (D) WPS/3.9.1-intel-2021b-dmpar
Autoconf/2.69-GCCcore-11.2.0 WPS/4.1-intel-2021b-dmpar
(D)
Autoconf/2.71-GCCcore-11.2.0 WRF/3.9.1.1-intel-2021b-dmpar
Autoconf/2.71-GCCcore-12.2.0 WRF/4.4.1-foss-2022b-dmpar
(D)
Autoconf/2.71          (D) X11/20210802-GCCcore-11.2.0
Automake/1.16.2-GCCcore-11.2.0 X11/20221110-
GCCcore-12.2.0 (D)
Automake/1.16.4-GCCcore-11.2.0 XALT/3.0.1
...
```

module spider wrf
or ml spider wrf

```
-----
WRF:
-----
```

Description:

The Weather Research and Forecasting (WRF) Model is a next-generation mesoscale numerical weather prediction system designed to serve both operational forecasting and atmospheric research needs.

Versions:

WRF/3.9.1.1-intel-2021b-dmpar
WRF/4.4.1-foss-2022b-dmpar

For detailed information about a specific "WRF" package (including how to load the modules) use the module's full name.

Note that names that have a trailing (E) are extensions provided by other modules. For example:

```
$ module spider WRF/4.4.1-foss-2022b-dmpar
```

```
module load WRF/3.9.1.1-intel-2021b-dmpar  
module list
```

Currently Loaded Modules:

1) GCCcore/11.2.0	11) intel/2021b
2) zlib/1.2.11-GCCcore-11.2.0	12) libtirpc/1.3.1-GCCcore-11.2.0
3) binutils/2.37-GCCcore-11.2.0	13) JasPer/2.0.24-GCCcore-11.2.0
4) intel-compilers/2021.4.0	14) Szip/2.1.1-GCCcore-11.2.0
5) numactl/2.0.14-GCCcore-11.2.0	15) HDF5/1.10.7-iimpi-2021b
6) UCX/1.11.2-GCCcore-11.2.0	16) OpenSSL/1.1
7) impi/2021.4.0-intel-compilers-2021.4.0	17) cURL/7.78.0-GCCcore-11.2.0
8) imkl/2021.4.0	18) netCDF/4.7.4-iimpi-2021b
9) iimpi/2021b	19) netCDF-Fortran/4.5.3-iimpi-2021b
10) imkl-FFTW/2021.4.0-iimpi-2021b	20) WRF/3.9.1.1-intel-2021b-dmpar

• **foss**

El módulo foss consiste en su totalidad en el término común *FOOS*, que es la abreviatura de *Free and Open source software*. 2021a, 2021b, 2022b, etc indica la generación.

Este módulo consta de los siguientes paquetes y se encarga de precargarlos.

- binutils
- Compilador GNU GCC (C), g++ (C++) and gfortran (Fortran)
- Librería Open MPI
- Librería FlexiBLAS con OpenBLAS + LAPACK como backend
- Librería ScaLAPACK
- Librería FFTW (Fourier transform (DFT))

```
module spider foss
module load foss/2021b
module list
```

Currently Loaded Modules:

```
1) GCCcore/11.2.0      8) libpciaccess/0.16-GCCcore-11.2.0 15) OpenMPI/4.1.1-
GCC-11.2.0
2) zlib/1.2.11-GCCcore-11.2.0 9) hwloc/2.5.0-GCCcore-11.2.0 16) OpenBLAS/0.3.18-
GCC-11.2.0
3) binutils/2.37-GCCcore-11.2.0 10) OpenSSL/1.1          17) FlexiBLAS/3.0.4-
GCC-11.2.0
4) GCC/11.2.0          11) libevent/2.1.12-GCCcore-11.2.0 18) gomp/2021b
5) numactl/2.0.14-GCCcore-11.2.0 12) UCX/1.11.2-GCCcore-11.2.0 19) FFTW/3.3.10-
gomp-2021b
6) XZ/5.2.5-GCCcore-11.2.0 13) libfabric/1.13.2-GCCcore-11.2.0 20) ScaLAPACK/
2.1.0-gomp-2021b-fb
7) libxml2/2.9.10-GCCcore-11.2.0 14) PMIx/4.1.0-GCCcore-11.2.0 21) foss/2021b
```

• **gomp:**

El módulo *gomp* precarga todo el software compilado con GNU GCC y OpenMPI. 2021a, 2021b, 2022b, etc indica la generación.

```
$ module load gomp/2021b
$ module list
```

Currently Loaded Modules:

```
1) GCCcore/11.2.0      7) libxml2/2.9.10-GCCcore-11.2.0 13) libfabric/1.13.2-
GCCcore-11.2.0
2) zlib/1.2.11-GCCcore-11.2.0 8) libpciaccess/0.16-GCCcore-11.2.0 14) PMIx/4.1.0-
GCCcore-11.2.0
3) binutils/2.37-GCCcore-11.2.0 9) hwloc/2.5.0-GCCcore-11.2.0 15) OpenMPI/4.1.1-
GCC-11.2.0
4) GCC/11.2.0          10) OpenSSL/1.1          16) gomp/2021b
5) numactl/2.0.14-GCCcore-11.2.0 11) libevent/2.1.12-GCCcore-11.2.0
6) XZ/5.2.5-GCCcore-11.2.0 12) UCX/1.11.2-GCCcore-11.2.0
```

• **intel:**

Este módulo común precarga los compiladores y librerías de Intel. 2021a, 2021b, 2022b, etc indica la generación.

- Compiladores Intel C/C++/Fortran (icc, icpc and ifort)
- binutils
- GCC, el cual sirve como base de los compiladores de intel.
- Librería Intel MPI
- Librería Intel Math Kernel Library (MKL) para funcionalidades de BLAS/LAPACK/FFT

```
module load intel/2022b
module list
```

Currently Loaded Modules:

```
1) GCCcore/12.2.0          5) numactl/2.0.16-GCCcore-12.2.0    9) iimpi/2022b
2) zlib/1.2.12-GCCcore-12.2.0  6) UCX/1.13.1-GCCcore-12.2.0      10) imkl-FFTW/
2022.2.1-iimpi-2022b
3) binutils/2.39-GCCcore-12.2.0  7) impi/2021.7.1-intel-compilers-2022.2.1  11) intel/
2022b
4) intel-compilers/2022.2.1    8) imkl/2022.2.1
```