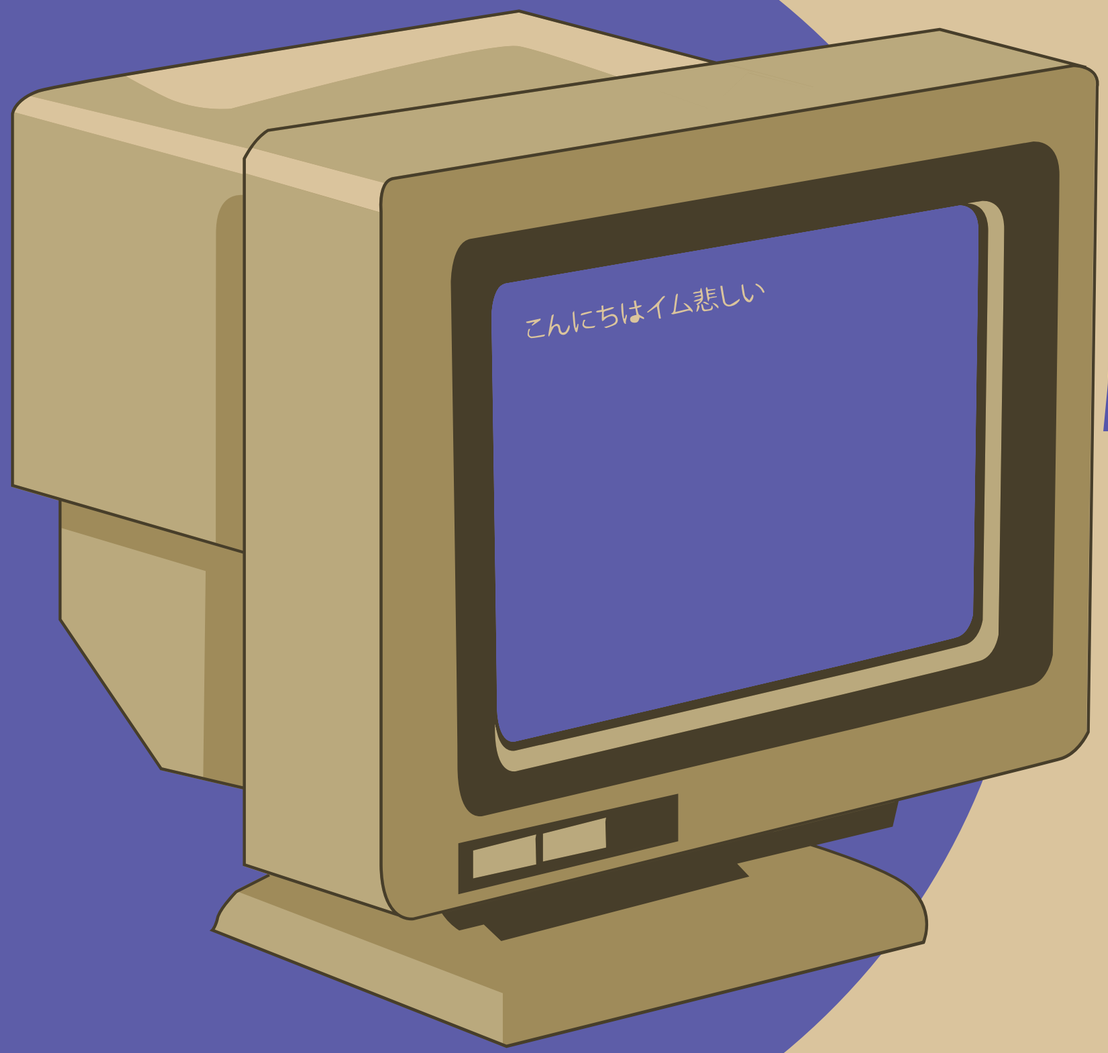
PHYSICS IN SILICO



AMBIENTES DE PROGRAMACIÓN

BARNALD BOCKER - ISAAC FLORES

GESTIÓN DE AMBIENTES CON ANACONDA Y MAMBA

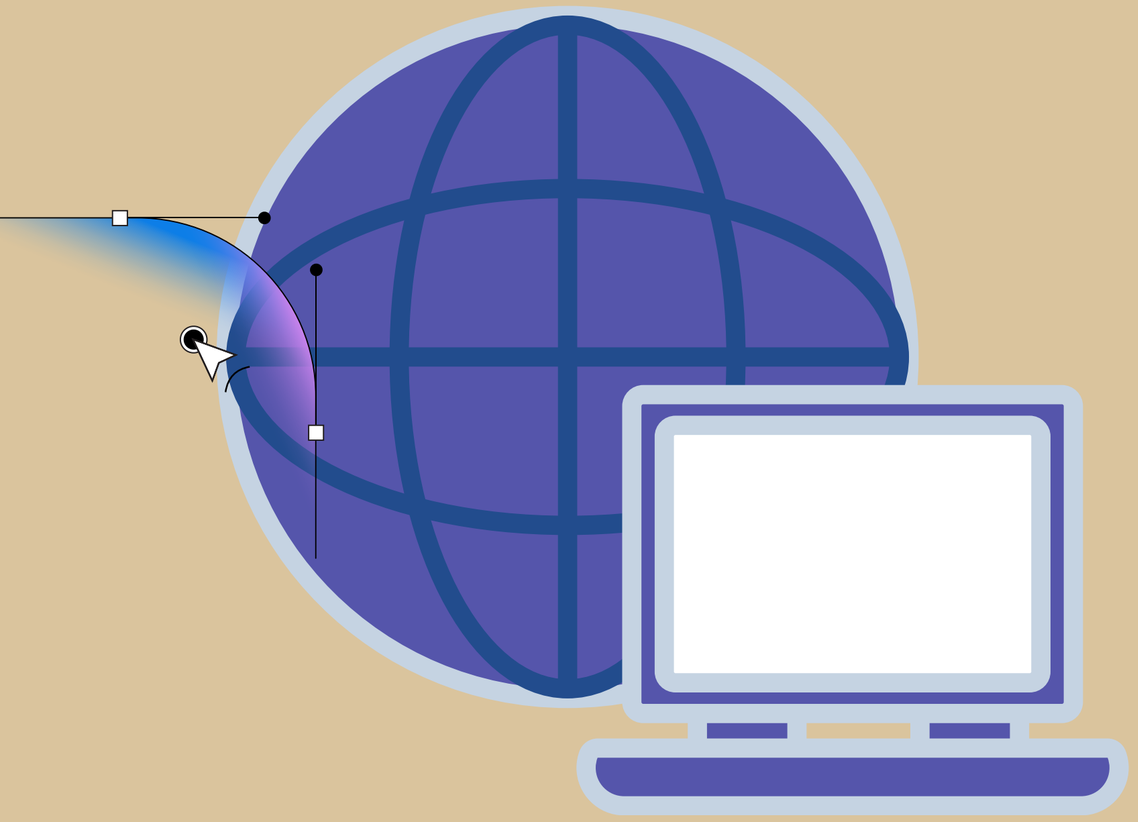
Son ecosistemas completos para diferentes tipos de desarrollo tienen las siguientes divisiones:

- Anaconda Distribution: Corresponde a la “Suite” completa con 250+ paquetes preinstalados
- Anaconda Navigator: Interfaz gráfica para gestión de ambientes
- Conda: Gestor de paquetes y ambientes (Parecido a apt-get)
- Mamba: Alternativa de alto rendimiento (es nativa en C++)

Ventajas:

- Instalación todo en uno
- Permite manejar diferentes ambientes para evitar conflictos con bibliotecas

PHYSICS IN SILICO



INSTALACIÓN DE CONDA

curl -O https://repo.anaconda.com/archive/Anaconda3-2025.12-2-Linux-x86_64.sh

wget https://repo.anaconda.com/archive/Anaconda3-2025.12-2-Linux-x86_64.sh

chmod +x Anaconda3-2025.12-2-Linux-x86_64.sh

source ~/.bashrc

conda --version

PHYSICS IN SILICO

GESTIÓN DE AMBIENTES DE CONDA

conda create -n test

conda create -n ciencia_datos numpy pandas matplotlib scikit-learn jupyter seaborn plotly

conda create --name base_clone --clone base

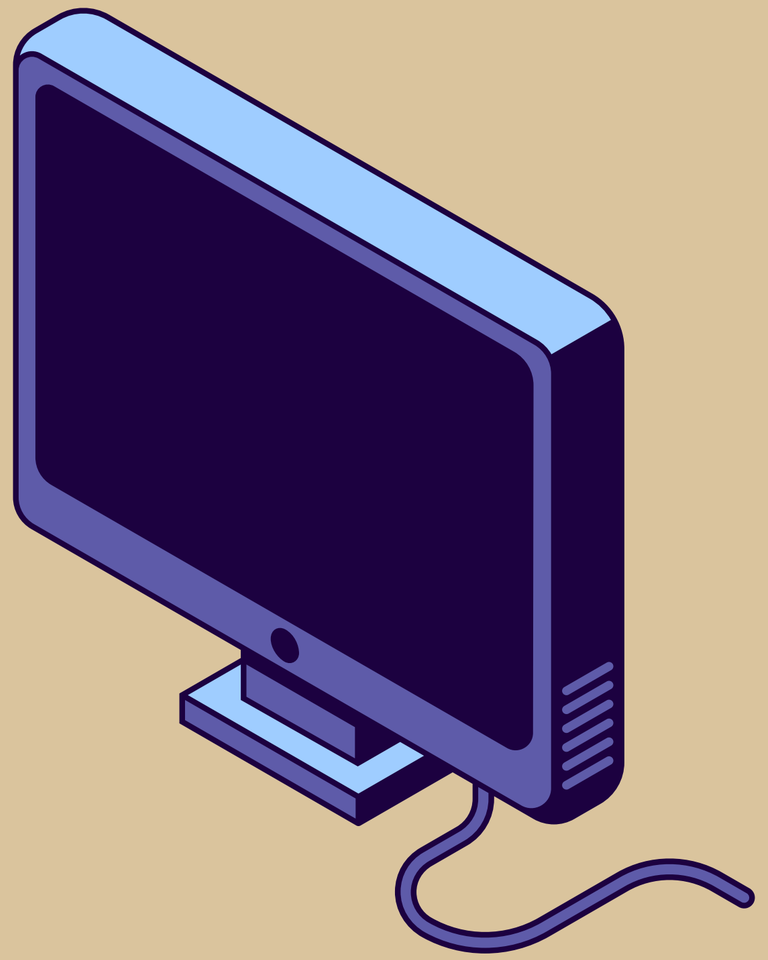
conda env list: Lista todos los ambientes disponibles

conda activate name: activa el ambiente

conda deactivate: Desactiva el ambiente

conda env remove -n name: Elimina el ambiente

MAMBA



Mamba es el hermano de Anaconda, la diferencia es que la implementación de mamba es en C++. Esto último le da una mucha mayor eficiencia para lidiar con paquetes.

Se instala de la siguiente forma:

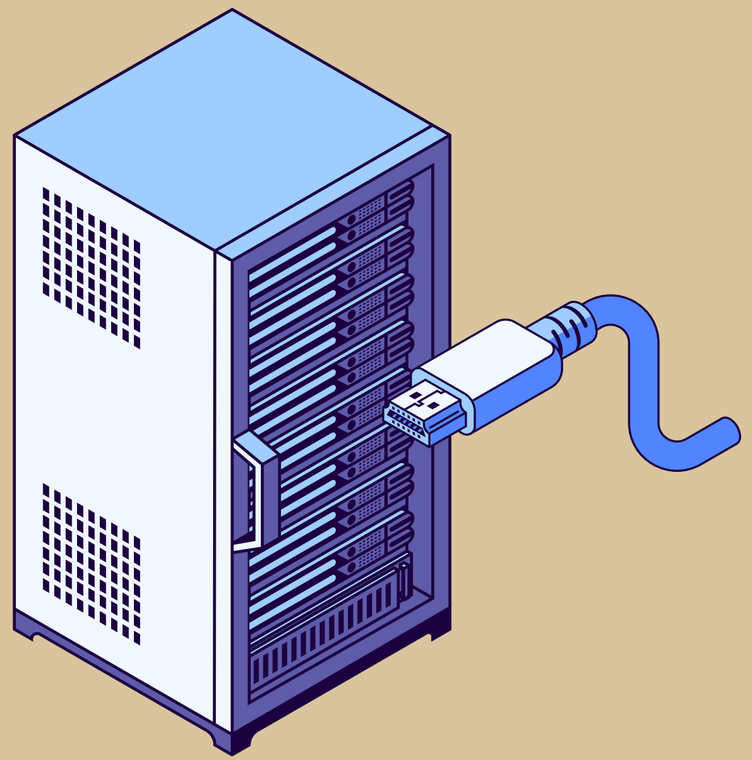
conda install mamba -n base -c conda-forge -y

mamba --version

# CONDA	# MAMBA
conda create -n env	→ mamba create -n env
conda install pkg	→ mamba install pkg
conda update --all	→ mamba update --all
conda remove pkg	→ mamba remove pkg
conda list	→ mamba list

PHYSICS IN SILICO

DEPENDENCIAS



1. **Versiones específicazs**
   conda install numpy=1.24.0 pandas=2.0.0 matplotlib=3.7.0

2. **Verificar paquetes instalados**
   conda list | grep -E “(numpy|pandas|matplotlib)”

3. **Buscar un paquete**
   mamba search nombre_paquete

EXPORTACIÓN REPRODUCIBILIDAD

Un elemento muy importante de los ambientes debe ser la posibilidad de exportarlos para que alguien más pueda instalar los paquetes de la misma forma, a continuación se les va a mostrar el proceso:

conda activate important_env

conda env export > enviroment_detallado.yaml

conda env export --from-history > enviroment_minimo.yaml

Este último excluye datos específicos de instalación cómo el hash de compilación específico
conda env export --no-builds > enviroment_compartir.yaml

PHYSICS IN SILICO

EXPORTACIÓN Y REPRODUCIBILIDAD

Usualmente los archivos .yaml se ven de la siguiente forma:

```
name: proyecto_ciencia_datos
channels:
  - conda-forge
  - defaults
dependencies:
  - python=3.10
  - numpy=1.24.0
  - pandas=2.0.0
  - matplotlib=3.7.0
  - scikit-learn=1.3.0
  - jupyterlab=4.0.0
  - pip
  - pip:
    - streamlit==1.25.0
    - python-dotenv==1.0.0
```

```
conda env create -f enviroment.yaml

# Esta es la opción recomendada debido a dependencias complicadas
mamba env create -f enviroment.yaml
```

PHYSICS IN SILICO

FLUJO DE TRABAJO

1. Crear ambiente para proyecto
mamba create -n proyecto python=3.10 -y
conda activate proyecto

#2. Instalar dependencias
mamba install numpy pandas matplotlib jupýterlab -y

#3. Se programa.....
#4. Congelar el ambiente al finalizar
conda env export > enviroment.yaml

#5. Compartir por git.... (teaser de la siguiente clase)

PHYSICS IN SILICO

BUENAS PRÁCTICAS Y RECURSOS

1. Mantener base limpio: Aquí no se deberían instalar paquetes con dependencias complejas, es para uso general
2. Usar mamba para instalaciones complejas
3. Especificar versiones en archivos .yaml: esto permite mayor portabilidad de los programas que se crean
4. Se puede investigar sobre los canales de descarga, pero usualmente se usa conda-forge
5. Limpiar y documentar

Solución de problemas comunes:

```
# Error de canales mixtos
conda config --set channel_priority strict

# Conflictos de dependencias
mamba install --solver=libmamba  # Usar solver alternativo

# Espacio en disco
conda clean --all --yes

# Recuperar ambiente dañado
conda list --revisions
conda install --revision 2  # Volver a revisión anterior
```

https://docs.anaconda.com/
https://docs.conda.io/projects/conda/en/latest/user-guide/cheatsheet.html
https://mamba.readthedocs.io/

Cheatsheet

Quickstart

Tip: It is recommended to createanewenvironmentfor any new project or workflow.

verify conda install and check version	conda info
update conda in base environment	conda update --name base conda
install latest anaconda distribution	conda install anaconda
create a new environment (tip: name environment descriptively)	conda create --name ENVNAME
activate environment (do this before installing packages)	conda activate ENVNAME

Channels and Packages

Tip: Package dependenciesandplatformspecifics areautomaticallyresolved when using conda.

list installed packages	conda list
list installed packages with source info	conda list --show-channel-urls
update all packages	conda update --all
install a package from specific channel	conda install --channel CHANNELNAME PKGNAME conda install CHANNELNAME::PKGNAME
install package with AND logic	conda install "PKGNAME>2.5,<3.2"
install package with OR logic	conda install "PKGNAME [version='2.5|3.2']"
uninstall package	conda uninstall PKGNAME
view channel sources	conda config --show-sources
add channel	conda config --add channels CHANNELNAME
set default channel for pkg fetching	conda config --set channel_priority strict

Working with Conda Environments

Tip: List environmentsatthe beginningofyoursession. Environmentswithanasterisk are active.

list all environments and locations	conda info --envs
list all packages + source channels	conda list --name ENVNAME --show-channel-urls
install packages in environment	conda install --name ENVNAME PKGNAME1 PKGNAME2
remove package from environment	conda uninstall --name ENVNAME PKGNAME
update all packages in environment	conda update --all --name ENVNAME

Environment Management

Tip: Specifying the environment name confines conda commands to that environment.

create environment with Python version	conda create --name ENVNAME python=3.12
clone environment	conda create --clone ENVNAME --name NEWENV
rename environment	conda rename --name ENVNAME NEWENVNAME
delete environment by name	conda remove --name ENVNAME --all
list revisions made to environment	conda list --name ENVNAME --revisions
restore environment to a revision	conda install --name ENVNAME --revision NUMBER
uninstall package from specific channel	conda remove --name ENVNAME --channel CHANNELNAME PKGNAME

Exporting Environments

Recommendation: Name the export file "environment". Your environment name will be preserved.

cross-platform compatible	conda export --from-history>ENV.yml
platform + package specific	conda export ENVNAME>ENV.yml
platform + package + channel specific	conda list --explicit>ENV.txt

Importing Environments

Tip: When importing an environment, conda resolves platform and package specifics.

from a .yml file	conda env create --name ENVNAME --file ENV.yml
from a .txt file	conda create --name ENVNAME --file ENV.txt

Additional Hints

get help for any command	conda COMMAND --help
get info for any package	conda search PKGNAME --info
run commands w/o user prompteg, installing multiple packages remove all unused files	conda COMMAND ARG --yes conda install PKGNAME1 PKGNAME2 --yes conda clean --all
examine conda configuration	conda config --show

MUCHAS GRACIAS