

1. ¿Cómo se llama por convención el archivo de requisitos?

En el ecosistema pip/Python, el nombre convencional es requirements.txt. Es un archivo de texto plano que lista dependencias y (opcionalmente) versiones/pines para reconstruir el entorno con `pip install -r requirements.txt`. Variantes comunes: requirements-dev.txt (deps de desarrollo), requirements-prod.txt (producción), constraints.txt (congelar versiones sin instalar por sí mismo) y requirements.in cuando se usan herramientas de resolución como pip-tools. Para paquetes instalables, las dependencias “de runtime” viven en pyproject.toml (o antes setup.cfg/setup.py), mientras que requirements.txt captura el entorno de trabajo/ejecución reproducible.

2. ¿Se puede definir un archivo de requisitos para Conda?

Sí. En Conda el análogo se define como environment.yml (YAML). A diferencia de requirements.txt, environment.yml permite declarar el nombre del entorno, canales (p. ej. conda-forge), paquetes no-Python y versiones; además puede incluir una sección pip: para mezclar dependencias de pip. Ejemplo mínimo:

```
name: mlops-taller4
channels: [conda-forge]
dependencies:
- python=3.10
- numpy
- pandas
- scikit-learn
- pip
- pip:
  - joblib
```

Se crea con `conda env create -f environment.yml`. Variaciones clave frente a requirements.txt: formato declarativo, soporte de canales/paquetes compilados, y posibilidad de exportar un “lock” reproducible (`conda env export --no-builds > environment-lock.yml`). En proyectos mixtos, se suele mantener ambos: environment.yml para usuarios de Conda y requirements.txt para pip/venv.

3. Conclusiones: usos de ambientes virtuales en ML

Los ambientes virtuales son esenciales para aislar dependencias, garantizar reproducibilidad y evitar “dependency hell” en ML, donde pequeñas diferencias de versiones (ej., numpy, pandas, scikit-learn) cambian resultados o rompen pipelines. Permiten:

- Onboarding rápido del equipo (un comando reconstruye el stack)
- Integración con CI/CD y experiments tracking al versionar los requisitos
- Coexistencia de proyectos con librerías conflictivas (incluso bindings nativos y GPU)
- Portabilidad entre máquinas/servidores

- Empaquetar y desplegar entrenamientos/servicios con garantías de entorno. En suma, capturar el entorno con requirements.txt/environment.yml y usar venv/Conda reduce fricción, acota riesgos y acelera el ciclo de vida MLOps de datos → modelo → producción.

Ambiente en Python

Instalación de los requirements en el ambiente de python

```
e, nbclient, jupyter-events, nbconvert, jupyter-server, notebook-shim, jupyterlab-server, jupyter-lsp, jupyterlab, notebook, jupyter
Successfully installed MarkupSafe-3.0.2 anyio-4.10.0 argon2-cffi-25.1.0 argon2-cffi-bindings-25.1.0 arrow-1.3.0 asttokens-3.0.0 async-lru-2.0.5 attrs-25.3.0 bab
el-2.17.0 beautifulsoup4-4.13.5 bleach-6.2.0 certifi-2025.8.3 cffi-1.17.1 charset-normalizer-3.4.3 colorama-0.4.6 comm-0.2.3 contourpy-1.3.3 cycler-0.12.1 debug
py-1.8.16 decorator-5.2.1 defusedxml-0.7.1 executing-2.2.1 fastjsonschema-2.21.2 fonttools-4.59.2 fqdn-1.5.1 h11-0.16.0 httpcore-1.0.9 httpx-0.28.1 idna-3.10 ip
ykernel-6.30.1 ipython-9.5.0 ipython-pygments-lexers-1.1.1 ipywidgets-8.1.7 isoduration-20.11.0 jedi-0.19.2 jinja2-3.1.6 joblib-1.5.2 json5-0.12.1 jsonpointer-3
.0.0 jsonschema-4.25.1 jsonschema-specifications-2025.4.1 jupyter-1.1.1 jupyter-client-8.6.3 jupyter-console-6.6.3 jupyter-core-5.8.1 jupyter-events-0.12.0 jupy
ter-lsp-2.3.0 jupyter-server-2.17.0 jupyter-server-terminals-0.5.3 jupyterlab-4.4.7 jupyterlab-pygments-0.3.0 jupyterlab-server-2.27.3 jupyterlab-widgets-3.0.15
kiwisolver-1.4.9 lark-1.2.2 matplotlib-3.10.6 matplotlib-inline-0.1.7 mistune-3.1.4 nbclient-0.10.2 nbconvert-7.16.6 nbformat-5.10.4 nest-asyncio-1.6.0 notebook
k-7.4.5 notebook-shim-0.2.4 numpy-2.3.2 overrides-7.7.0 packaging-25.0 pandas-2.3.2 pandocfilters-1.5.1 parso-0.8.5 pillow-11.3.0 platformdirs-4.4.0 prometheus-
client-0.22.1 prompt-toolkit-3.0.52 psutil-7.0.0 pure-eval-0.2.3 pycparser-2.22 pygments-2.19.2 pyparsing-3.2.3 python-dateutil-2.9.0.post0 python-json-logger-3
.3.0 pytz-2025.2 pywin32-311 pywinpty-3.0.0 pyyaml-6.0.2 pyzmq-27.0.2 referencing-0.36.2 requests-2.32.5 rfc3339-validator-0.1.4 rfc3986-validator-0.1.1 rfc3987
-syntax-1.1.0 rpds-py-0.27.1 scikit-learn-1.7.1 scipy-1.16.1 seaborn-0.13.2 send2trash-1.8.3 six-1.17.0 sniffio-1.3.1 soupsieve-2.8 stack-data-0.6.3 terminado-0
.18.1 threadpoolctl-3.6.0 tinycss2-1.4.0 tornado-6.5.2 traitlets-5.14.3 types-python-dateutil-2.9.0.20250822 typing_extensions-4.15.0 tzdata-2025.2 uri-template
-1.3.0 urllib3-2.5.0 wcwidth-0.2.13 webcolors-24.11.1 webencodings-0.5.1 websocket-client-1.8.0 widgetsnextension-4.0.14
(.venv) PS C:\Users\caste\OneDrive\Documentos\Universidad\semestre10\MLOPS\Taller-4-MLOPS>

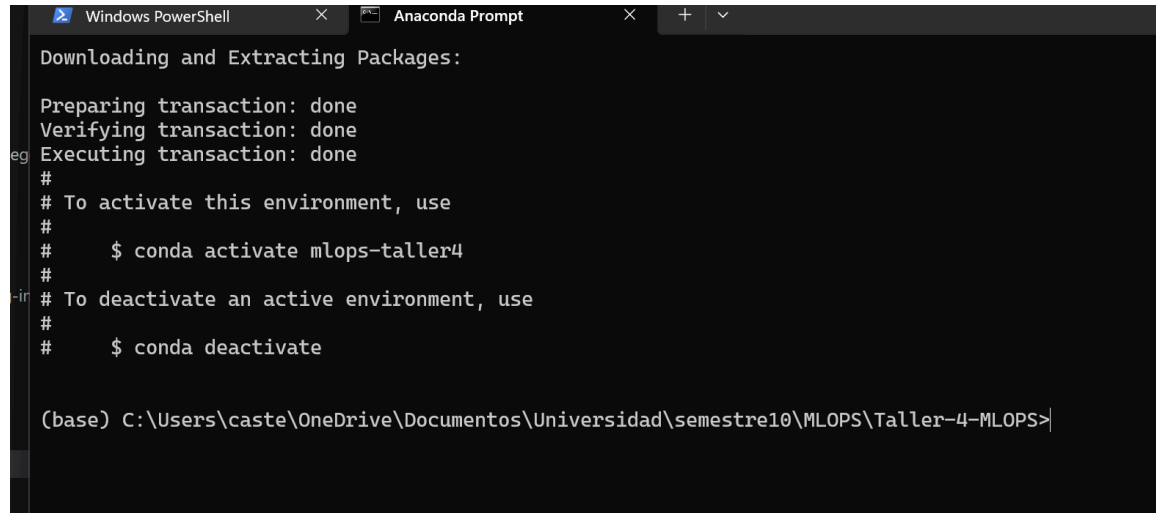
• PS C:\Users\caste\OneDrive\Documentos\Universidad\semestre10\MLOPS\Taller-4-MLOPS> python -m venv .venv
• PS C:\Users\caste\OneDrive\Documentos\Universidad\semestre10\MLOPS\Taller-4-MLOPS> . .\venv\Scripts\Activate.ps1
• (.venv) PS C:\Users\caste\OneDrive\Documentos\Universidad\semestre10\MLOPS\Taller-4-MLOPS> python -m pip install --upgrade pip
Requirement already satisfied: pip in c:\Users\caste\onedrive\documentos\universidad\semestre10\mlops\taller-4-mlops\.venv\lib\site-packages (23.1.2)
Collecting pip
  Downloading pip-25.2-py3-none-any.whl (1.8 MB)
Installing collected packages: pip
  Attempting uninstall: pip
    Found existing installation: pip 23.1.2
    Uninstalling pip-23.1.2:
      Successfully uninstalled pip-23.1.2
  Successfully installed pip-25.2
• (.venv) PS C:\Users\caste\OneDrive\Documentos\Universidad\semestre10\MLOPS\Taller-4-MLOPS> pip install -r requirements.txt
>>
Collecting numpy==1.26.4 from -r requirements.txt (1.1 KB)
  Stored in directory: C:\Users\caste\AppData\Local\Temp\pip-ephem-wheel-cache-wfascvqv\wheels\88\af\3d\601630d34e36528
Successfully built pipeline-ml
Installing collected packages: click, pipeline-ml
  Attempting uninstall: pipeline-ml
    Found existing installation: pipeline-ml 0.1.0
    Uninstalling pipeline-ml-0.1.0:
      Successfully uninstalled pipeline-ml-0.1.0
  Successfully installed click-8.2.1 pipeline-ml-0.1.0
• (.venv) PS C:\Users\caste\OneDrive\Documentos\Universidad\semestre10\MLOPS\Taller-4-MLOPS>
• (.venv) PS C:\Users\caste\OneDrive\Documentos\Universidad\semestre10\MLOPS\Taller-4-MLOPS> pipeline -o model.joblib
Accuracy: 1.000
Modelo guardado en: model.joblib
• (.venv) PS C:\Users\caste\OneDrive\Documentos\Universidad\semestre10\MLOPS\Taller-4-MLOPS>
```

Ejecución del pipeline con el ambiente

```
X_norms = np.sqrt(row_norms(X.T, squared=True) - n_samples * X_means**2)
>> Mejores parámetros: {'model': RandomForestRegressor(n_jobs=1, random_state=42), 'model_max_depth': 20, 'model_min_samples_leaf': 1, 'model_min_samples
_split': 2, 'model_n_estimators': 400}
C:\Users\caste\OneDrive\Documentos\Universidad\semestre10\MLOPS\Taller-4-MLOPS\.venv\lib\site-packages\sklearn\impute\_base.py:637: UserWarning: Skipping fe
atures without any observed values: ['TIDAL Popularity']. At least one non-missing value is needed for imputation with strategy='median'.
  warnings.warn(
C:\Users\caste\OneDrive\Documentos\Universidad\semestre10\MLOPS\Taller-4-MLOPS\.venv\lib\site-packages\sklearn\impute\_base.py:637: UserWarning: Skipping fe
atures without any observed values: ['TIDAL Popularity']. At least one non-missing value is needed for imputation with strategy='median'.
  warnings.warn(
Train R2: 0.9091 | MAE: 3.0295
Test R2: 0.3707 | MAE: 7.7853
>> Modelo guardado en best.joblib
• (.venv) PS C:\Users\caste\OneDrive\Documentos\Universidad\semestre10\MLOPS\Taller-4-MLOPS>
```

Ambiente con Conda

Creación



```

Windows PowerShell
Anaconda Prompt

Downloading and Extracting Packages:

Preparing transaction: done
Verifying transaction: done
Executing transaction: done
#
# To activate this environment, use
#
#     $ conda activate mlops-taller4
#
# To deactivate an active environment, use
#
#     $ conda deactivate

(base) C:\Users\caste\OneDrive\Documentos\Universidad\semestre10\MLOPS\Taller-4-MLOPS>

```

Instalación



```

pipeline==0.1.0) (2.3.2)
Requirement already satisfied: pandas>=2.2 in c:\users\caste\anaconda3\envs\mlops-taller4\lib\site-packages (from mlops-pipeline==0.1.0) (2.3.2)
Requirement already satisfied: scikit-learn>=1.5 in c:\users\caste\anaconda3\envs\mlops-taller4\lib\site-packages (from mlops-pipeline==0.1.0) (1.7.2)
Requirement already satisfied: joblib>=1.4 in c:\users\caste\anaconda3\envs\mlops-taller4\lib\site-packages (from mlops-pipeline==0.1.0) (1.5.2)
Requirement already satisfied: python-dateutil>=2.8.2 in c:\users\caste\anaconda3\envs\mlops-taller4\lib\site-packages (from pandas>=2.2->mlops-pipeline==0.1.0) (2.9.0.post0)
Requirement already satisfied: pytz>=2020.1 in c:\users\caste\anaconda3\envs\mlops-taller4\lib\site-packages (from pandas>=2.2->mlops-pipeline==0.1.0) (2025.2)
Requirement already satisfied: tzdata>=2022.7 in c:\users\caste\anaconda3\envs\mlops-taller4\lib\site-packages (from pandas>=2.2->mlops-pipeline==0.1.0) (2025.2)
Requirement already satisfied: six>=1.5 in c:\users\caste\anaconda3\envs\mlops-taller4\lib\site-packages (from python-dateutil>=2.8.2->pandas>=2.2->mlops-pipeline==0.1.0) (1.17.0)
Requirement already satisfied: scipy>=1.8.0 in c:\users\caste\anaconda3\envs\mlops-taller4\lib\site-packages (from scikit-learn>=1.5->mlops-pipeline==0.1.0) (1.16.1)
Requirement already satisfied: threadpoolctl>=3.1.0 in c:\users\caste\anaconda3\envs\mlops-taller4\lib\site-packages (from scikit-learn>=1.5->mlops-pipeline==0.1.0) (3.6.0)
Building wheels for collected packages: mlops-pipeline
  Building editable for mlops-pipeline (pyproject.toml) ... done
  Created wheel for mlops-pipeline: filename=mlops_pipeline-0.1.0-0.editable-py3-none-any.whl size=1698 sha256=28503590a9625e6f12f9b2a107bb06e50ccc992cfca15f684e86bb430bc5277
  Stored in directory: C:\Users\caste\AppData\Local\Temp\pip-ephem-wheel-cache-is62703e\wheels\88\30\86\b08d7ab4a315e8756e6971f206d3e142d8261bf8b30cbc2e
Successfully built mlops-pipeline
Installing collected packages: mlops-pipeline
Successfully installed mlops-pipeline-0.1.0

(mlops-taller4) C:\Users\caste\OneDrive\Documentos\Universidad\semestre10\MLOPS\Taller-4-MLOPS>

```

Ejecución

```

gy='median'.
warnings.warn(
[CV] END model=StackingRegressor(estimators=[('rf',
                                             RandomForestRegressor(n_jobs=1,
                                                                     random_state=42)),
                                             ('gb',
                                              GradientBoostingRegressor(random_state=42))],
                                final_estimator=GradientBoostingRegressor(), n_jobs=1); total time= 23.6s
C:\Users\caste\anaconda3\envs\mlops-taller4\Lib\site-packages\sklearn\impute\_base.py:653: UserWarning: Skipping feature
s without any observed values: ['TIDAL Popularity']. At least one non-missing value is needed for imputation with strate
gy='median'.
warnings.warn(
C:\Users\caste\anaconda3\envs\mlops-taller4\Lib\site-packages\sklearn\feature_selection\_univariate_selection.py:380: Ru
ntimeWarning: invalid value encountered in sqrt
X_norms = np.sqrt(row_norms(X.T, squared=True) - n_samples * X_means**2)
>> Mejores parámetros: {'model': RandomForestRegressor(n_jobs=1, random_state=42), 'model__max_depth': 20, 'model__min_s
amples_leaf': 1, 'model__min_samples_split': 2, 'model__n_estimators': 400}
C:\Users\caste\anaconda3\envs\mlops-taller4\Lib\site-packages\sklearn\impute\_base.py:653: UserWarning: Skipping feature
s without any observed values: ['TIDAL Popularity']. At least one non-missing value is needed for imputation with strate
gy='median'.
warnings.warn(
C:\Users\caste\anaconda3\envs\mlops-taller4\Lib\site-packages\sklearn\impute\_base.py:653: UserWarning: Skipping feature
s without any observed values: ['TIDAL Popularity']. At least one non-missing value is needed for imputation with strate
gy='median'.
warnings.warn(
Train R2: 0.9091 | MAE: 3.0295
Test R2: 0.3707 | MAE: 7.7853
>> Modelo guardado en best.joblib

(mlops-taller4) C:\Users\caste\OneDrive\Documentos\Universidad\semestre10\MLOPS\Taller-4-MLOPS>

```