# Clase 026 — Pandas: MultiIndex

Parte: 0 — Prerrequisitos · Fuente: VanderPlas, cap. 3 § 3.6 Hierarchical Indexing. · Duración estimada: 75 min.

# Clase 026 — Pandas: MultiIndex

> *Parte: 0 — Prerrequisitos · Fuente: VanderPlas, cap. 3 § 3.6 Hierarchical Indexing. · Duración estimada: 75 min.*

## Objetivo

Que el alumno use índices jerárquicos (MultiIndex) cuando hay estructura natural en los datos (país × ciudad, año × mes, sector × empresa). Saber cuándo aporta vs cuándo complica — el 80% del tiempo en data science aplanado es mejor.

## Resultados de aprendizaje

Al finalizar la clase, el alumno podrá:

1. Crear MultiIndex desde tuplas, arrays, producto cartesiano (from_product).
2. Indexar con .loc[(nivel1, nivel2)] y .loc[:, ('grupo', 'col')].
3. Aplanar y reconstruir con unstack(), stack(), reset_index().
4. Decidir cuándo MultiIndex aporta (groupby con múltiples claves devuelve uno automáticamente) y cuándo es más legible aplanar.
5. Renombrar niveles con rename(level=...) y reordenarlos con swaplevel.

## Temas

| # | Tema | Por qué importa |
|---|---|---|
| 1 | MultiIndex: motivación | Datos con jerarquía natural. |
| 2 | Construcción: tuples, arrays, from_product | 3 formas comunes. |
| 3 | Indexación: tuple selector | .loc[('A', 2024)]. |
| 4 | stack / unstack — pivot rápido | Mover niveles entre filas y columnas. |
| 5 | groupby + multiindex resultado | groupby con 2+ claves devuelve MultiIndex. |
| 6 | Cuándo aplanar | Para CSV de salida, plot, scikit-learn. |

## Definiciones y características

MultiIndex

: Índice jerárquico con N niveles. Cada fila identificada por tupla de N labels (('España', 2024)). Útil cuando los datos tienen estructura natural (país→ciudad, año→mes).

Nivel (level)

: Cada "capa" del MultiIndex. Se referencia por nombre (level='año') o posición (level=0). Útil en operaciones como unstack(level=...).

stack / unstack

: Mueven niveles entre filas y columnas. unstack sube un nivel del index a columnas (long→wide). stack baja un nivel de columnas al index (wide→long). Reversibles.

xs (cross-section)

: Slice por un valor en un nivel: df.xs(2024, level='año'). Más limpio que indexar con tuplas parciales.

Aplanar (flatten)

: Convertir MultiIndex a Index plano: df.reset_index() (vuelve a default 0..N) o df.index = ['_'.join(map(str, t)) for t in df.index] (concatena niveles).

## Dataset / recursos

Sintético: ventas por país/año.

## Ejercicios

1. Construye desde tuplas. Crea DataFrame con index [(España, 2023), (España, 2024), (Chile, 2023), (Chile, 2024)] y 2 cols ventas/clientes.

2. from_product. Mismo con pd.MultiIndex.from_product([paises, años]).

3. Acceso jerárquico. df.loc['España'], df.loc[('España', 2024)]. Compara con df.xs(2024, level=1) para slice por nivel.

4. unstack y stack. Convierte tu MultiIndex en wide (años como columnas) y de vuelta.

5. groupby produce MultiIndex. Carga penguins, agrupa por (species, sex) y agrega mean(). Aplana con reset_index().

## Homework verificable

Notebook con ventas trimestre×región sintéticas (4 trimestres × 3 regiones × 2 años): (a) construir con from_product; (b) acceso a un trimestre específico; (c) total por región (unstack); (d) groupby penguins por (species, sex) → MultiIndex → aplanar.

Criterio de aceptación: MultiIndex con shape correcto; unstack/stack reversibles.

## Errores comunes

| Síntoma / mensaje | Causa y cómo arreglar |
|---|---|
| KeyError al acceder df.loc['España', 2024] | loc con MultiIndex requiere tupla: df.loc[ |
| unstack() lanza ValueError: Index contains | Tienes filas duplicadas en (index, columns |
| groupby([a, b]).sum() devuelve cosa extrañ | Es correcto: groupby con N keys devuelve M |
| Plot ignora niveles del MultiIndex | matplotlib/seaborn esperan columnas planas |
| sort_index() ordena raro con MultiIndex | Default ordena por todos los niveles. Para |

## Preguntas frecuentes

¿Cuándo MultiIndex aporta vs cuándo complica?

Aporta en análisis interactivo con slicing por nivel frecuente. Complica para export a CSV, plot, sklearn — aplana ahí.

¿set_index([a, b]) vs groupby([a, b])?

set_index solo mueve cols al index (sin agregar). groupby colapsa filas por las cols (con sum/mean/agg). Diferentes operaciones.

¿Cómo evito MultiIndex en groupby?

groupby([a, b], as_index=False) devuelve DataFrame plano directamente. O .reset_index() después.

¿stack(future_stack=True) qué significa?

Es el comportamiento del nuevo stack (default en pandas 3+). Maneja NaN distinto al legacy. Mejor pasarlo siempre explícito para suprimir warnings.

¿Performance MultiIndex vs Index plano?

MultiIndex tiene overhead. Para datasets grandes (>1M filas) con acceso intenso, aplana al final del pipeline.

## Referencias

- VanderPlas, cap. 3 § 3.6.
- pandas MultiIndex user guide

## Siguiente clase

Clase 027 — Pandas: concat, merge, join

## Apéndice: notebook (primer bloque)

Primera celda ejecutable del notebook de la clase.

```
import numpy as np
import pandas as pd
rng = np.random.default_rng(42)
```

## Archivos complementarios

- notebook.ipynb