

# Concatenar DataFrames por filas o columnas

Es una operación corriente.

*Consiste en unir dos DataFrames, poniendo las filas del segundo a continuación de las filas del primero, o las columnas del segundo a continuación de las columnas del primero.*

```
# Preliminares
import pandas as pd
import numpy as np
```

Consideramos los dos `DataFrame` s:

```
df1 = pd.DataFrame(np.arange(9).reshape(3, 3), index=list('ijk'), columns=list('abc'))
df1
```

	a	b	c
i	0	1	2
j	3	4	5
k	6	7	8

```
df2 = pd.DataFrame(np.arange(20, 26).reshape(2, 3), index=list('lm'), columns=list('abc'))
df2
```

	a	b	c
l	20	21	22
m	23	24	25

En su versión más sencilla, pasamos un iterable (lista por ejemplo) de `DataFrame` s a `pd.concat` :

```
pd.concat([df1, df2])
```

	a	b	c
i	0	1	2
j	3	4	5
k	6	7	8
l	20	21	22
m	23	24	25

# Para concatenar por las columnas

Usamos el parámetro `axis` que por defecto es igual a 0 (filas).

```
# Introducimos un nuevo DataFrame para ilustrarlo
df3 = pd.DataFrame(
    np.arange(30, 36).reshape(3, 2),
    index=list('ijk'),
    columns=list('df')
)
df3
```

	d	f
i	30	31
j	32	33
k	34	35

```
pd.concat([df1, df3], axis=1)
```

	a	b	c	d	f
i	0	1	2	30	31
j	3	4	5	32	33
k	6	7	8	34	35

# El método `concat` alinea las filas o las columnas antes de realizar la unión

Para demostrarlo, cambiemos las filas de `df3`, usando el "slicing" con un step negativo.

```
df3 = df3.loc[::-1,:]  
df3
```

	d	f
k	34	35
j	32	33
i	30	31

Al unirlos por las columnas, `pd.concat` alinea las filas basándose en sus etiquetas:

```
pd.concat([df1, df3], axis=1)
```

	a	b	c	d	f
i	0	1	2	30	31
j	3	4	5	32	33
k	6	7	8	34	35

*Obtenemos el mismo resultado que antes de cambiar el orden de las filas de `df3`.*

# Concatenar `DataFrames` por filas o columnas con etiquetas repetidas

*Qué pasa si quiero unir por filas dos `DataFrame`s que tienen etiquetas de filas repetidas?*

Consideramos los dos DataFrame s:

```
df1 = pd.DataFrame(np.arange(9).reshape(3, 3), index=list('ijk'), columns=list('abc'))
df1
```

	a	b	c
i	0	1	2
j	3	4	5
k	6	7	8

```
df4 = pd.DataFrame(np.arange(40,46).reshape(2, 3), index=list('ij'), columns=list('abc'))
df4
```

	a	b	c
i	40	41	42
j	43	44	45

*df1 y df4 tienen las mismas columnas, pero además se repiten sus etiquetas de filas*

Unimos `df1` y `df4` por filas:

```
pd.concat([df1, df4])
```

	a	b	c
i	0	1	2
j	3	4	5
k	6	7	8
i	40	41	42
j	43	44	45

*Se repiten las etiquetas en el resultado.*



# Si no queremos que se repitan las etiquetas

Tenemos dos opciones:

1. Descartar las etiquetas de las filas con `ignore_index=True`
2. Indicar con `keys` las claves que informen de qué `DataFrame` procede cada fila

## Descartando las etiquetas

```
pd.concat([df1, df4], ignore_index=True)
```

	a	b	c
0	0	1	2
1	3	4	5
2	6	7	8
3	40	41	42
4	43	44	45

# Indicando la procedencia de cada fila con **keys**

```
pd.concat([df1, df4], keys=['df1', 'df4'])
```

		a	b	c
df1	i	0	1	2
	j	3	4	5
	k	6	7	8
df4	i	40	41	42
	j	43	44	45

# Qué pasa si quiero unir por filas dos `DataFrames` que no tienen las mismas columnas

Hemos visto que `concat` alinea las columnas usando sus etiquetas, pero en algunas ocasiones, los dos `DataFrames` tienen algunas columnas diferentes

Consideramos los dos DataFrame s:

```
df1 = pd.DataFrame(np.arange(9).reshape(3, 3), index=list('ijk'), columns=list('abc'))
df1
```

	a	b	c
i	0	1	2
j	3	4	5
k	6	7	8

```
df5 = pd.DataFrame(np.arange(50, 56).reshape(2, 3), index=list('lm'), columns=list('abd'))
df5
```

	a	b	d
l	50	51	52
m	53	54	55

*df1 y df5 tienen las columnas a y b en común pero tienen cada una una columna propia*

Unimos `df1` y `df5` por filas:

```
pd.concat([df1, df5])
```

	a	b	c	d
i	0	1	2.0	NaN
j	3	4	5.0	NaN
k	6	7	8.0	NaN
l	50	51	NaN	52.0
m	53	54	NaN	55.0

`concat` ha unido las filas, completando con `NaN` las filas del `DataFrame` que no tienen valor en estas columnas

# Podemos cambiarlo con el parámetro `join`

Por defecto, el parámetro `join` toma el valor 'outer', que implica que se queda con la unión de las columnas, completando con `NaN` donde faltan datos.

*Si indicamos `join='inner'`, sólo se preservan las columnas que ambos `DataFrame`s tienen en común, descartando las que son exclusivas de uno o el otro;*

```
pd.concat([df1, df5], join='inner')
```

	a	b
i	0	1
j	3	4
k	6	7
l	50	51
m	53	54

# Todo funciona de manera similar si unimos por columnas

El comportamiento de `concat` es simétrico por filas o por columnas.

```
# Recordad df1  
df1
```

	a	b	c
i	0	1	2
j	3	4	5
k	6	7	8

```
# Introducimos ahora df6  
df6 = pd.DataFrame(np.arange(60, 66).reshape(3, 2), index=list('ijl'), columns=list('de'))  
df6
```

	d	e
i	60	61
j	62	63
l	64	65

## Unimos por columnas

```
pd.concat([df1, df6], axis=1)
```

	a	b	c	d	e
i	0.0	1.0	2.0	60.0	61.0
j	3.0	4.0	5.0	62.0	63.0
k	6.0	7.0	8.0	NaN	NaN
l	NaN	NaN	NaN	64.0	65.0

Si especificamos `join='inner'`

```
pd.concat([df1, df6], axis=1, join='inner')
```

	a	b	c	d	e
i	0	1	2	60	61
j	3	4	5	62	63