

# la\_clinica

March 11, 2020

```
[5]: import pandas as pd
import glob
```

```
[7]: lista_archivos = glob.glob('*.xlsx')
lista_archivos
```

```
[7]: ['examen1.xlsx',
      'examen2.xlsx',
      'examen3.xlsx',
      'examen4.xlsx',
      'examen5.xlsx',
      'examen6.xlsx',
      'examen7.xlsx']
```

```
[13]: resultados = pd.concat([pd.read_excel(file_name) for file_name in
    ↪ lista_archivos])
resultados
```

```
[13]:
```

	Fecha	Tipo_Examen	Valor	Resultado
0	2001-11-14	Colesterol	200.0	alto
0	2011-05-30	Glucosa	110.0	medio
0	2001-01-10	Rayos_X	NaN	normal
0	2009-10-15	Peso	90.0	sobrepeso
0	2000-10-15	vista	75.0	miopia
0	2010-11-15	glucosa	100.0	medio
0	2020-11-15	glucosa	110.0	medio

```
[18]: # Se tiene diferentes tipos de exámenes. En apariencia 2 para glucosa. Pero por
    ↪ decisión de negocio sólo habrá
# un sólo tipo de examen de glucosa. Pero podría ser mejor pasar todo a upper
    ↪ or lower case suponiendo que fue
# por causa de un typo o error en el dato.
resultados.Tipo_Examen.unique().tolist()
```

```
[18]: ['Colesterol', 'Glucosa', 'Rayos_X', 'Peso', 'vista', 'glucosa']
```

```
[22]: resultados.Tipo_Examen = resultados.Tipo_Examen.str.lower()
resultados
```

```
[22]:
```

	Fecha	Tipo_Examen	Valor	Resultado
0	2001-11-14	colesterol	200.0	alto

```

0 2011-05-30      glucosa  110.0      medio
0 2001-01-10      rayos_x   NaN      normal
0 2009-10-15       peso    90.0  sobrepeso
0 2000-10-15       vista    75.0    miopia
0 2010-11-15      glucosa  100.0      medio
0 2020-11-15      glucosa  110.0      medio

```

```

[23]: # Se revisa y se nota mayor calidad.
resultados.Tipo_Examen.unique().tolist()

```

```

[23]: ['colesterol', 'glucosa', 'rayos_x', 'peso', 'vista']

```

```

[26]: resultados.set_index(['Fecha', 'Tipo_Examen'], inplace=True)

```

```

[27]: resultados

```

```

[27]:
          Valor  Resultado
Fecha      Tipo_Examen
2001-11-14  colesterol  200.0      alto
2011-05-30   glucosa    110.0      medio
2001-01-10   rayos_x      NaN      normal
2009-10-15   peso       90.0  sobrepeso
2000-10-15   vista       75.0    miopia
2010-11-15   glucosa    100.0      medio
2020-11-15   glucosa    110.0      medio

```

```

[38]: # Sabiendo que los campos Valor y Resultado son las métricas y que Tipo_Examen
      → y Fecha las dimensiones
      # Se disponer las dimensiones (Tipo_Examen y Fecha) como columna y indice.
resultados = resultados.unstack(level=-1).swaplevel(axis=1).sort_index(axis=1)
resultados

```

```

[38]: Tipo_Examen  colesterol      glucosa      peso      rayos_x \
          Resultado  Valor Resultado  Valor Resultado  Valor Resultado
Fecha
2000-10-15      NaN    NaN      NaN    NaN      NaN    NaN      NaN
2001-01-10      NaN    NaN      NaN    NaN      NaN    NaN      normal
2001-11-14      alto  200.0      NaN    NaN      NaN    NaN      NaN
2009-10-15      NaN    NaN      NaN    NaN  sobrepeso  90.0      NaN
2010-11-15      NaN    NaN      medio  100.0      NaN    NaN      NaN
2011-05-30      NaN    NaN      medio  110.0      NaN    NaN      NaN
2020-11-15      NaN    NaN      medio  110.0      NaN    NaN      NaN

Tipo_Examen      vista
          Valor Resultado  Valor
Fecha
2000-10-15      NaN    miopia  75.0
2001-01-10      NaN      NaN    NaN
2001-11-14      NaN      NaN    NaN
2009-10-15      NaN      NaN    NaN

```

2010-11-15	NaN	NaN	NaN
2011-05-30	NaN	NaN	NaN
2020-11-15	NaN	NaN	NaN

## 1 Punto clave!!!

```
[41]: # Haciendo ffill forward fill que permite rellenar hacia adelante con los
      # últimos valores válidos.
      # Regla de negocio: El último valor de examen reportado/documentado será
      # oficialmente el actual estado de salud
      # hasta que se diga lo contrario.
      resultados.fillna(method='ffill', inplace=True)
      resultados
```

```
[41]: Tipo_Examen colesterol          glucosa          peso          rayos_x \
      Resultado  Valor Resultado  Valor  Resultado  Valor Resultado
Fecha
2000-10-15      NaN    NaN      NaN    NaN      NaN    NaN      NaN
2001-01-10      NaN    NaN      NaN    NaN      NaN    NaN    normal
2001-11-14      alto  200.0      NaN    NaN      NaN    NaN    normal
2009-10-15      alto  200.0      NaN    NaN  sobrepeso  90.0    normal
2010-11-15      alto  200.0    medio  100.0  sobrepeso  90.0    normal
2011-05-30      alto  200.0    medio  110.0  sobrepeso  90.0    normal
2020-11-15      alto  200.0    medio  110.0  sobrepeso  90.0    normal
```

```
Tipo_Examen          vista
Valor Resultado Valor
Fecha
2000-10-15      NaN    miopia  75.0
2001-01-10      NaN    miopia  75.0
2001-11-14      NaN    miopia  75.0
2009-10-15      NaN    miopia  75.0
2010-11-15      NaN    miopia  75.0
2011-05-30      NaN    miopia  75.0
2020-11-15      NaN    miopia  75.0
```

## 2 último / más reciente valor reportado. Ultimo estado de salud basado en datos

## 3 Diferentes formas / presentaciones

```
[42]: resultados.iloc[-1]
```

```
[42]: Tipo_Examen
colesterol  Resultado      alto
```

	Valor	200
glucosa	Resultado	medio
	Valor	110
peso	Resultado	sobrepeso
	Valor	90
rayos_x	Resultado	normal
	Valor	NaN
vista	Resultado	miopia
	Valor	75

Name: 2020-11-15 00:00:00, dtype: object

```
[44]: resultados.iloc[-1].to_frame().T
```

```
[44]: Tipo_Examen colesterol          glucosa          peso          rayos_x  \
      Resultado Valor Resultado Valor Resultado Valor Resultado Valor
2020-11-15      alto    200      medio    110  sobrepeso    90      normal    NaN

      Tipo_Examen vista
      Resultado Valor
2020-11-15      miopia    75
```

```
[46]: resultados.iloc[-1].unstack()
```

```
[46]:      Resultado Valor
Tipo_Examen
colesterol      alto    200
glucosa          medio    110
peso      sobrepeso    90
rayos_x      normal    NaN
vista      miopia    75
```

```
[ ]:
```