

Ingeniería de Computación y Sistemas

# INGENIERÍA DE DATOS

Mg. Ing. Carlos Edwin Julca Castillo

**UPAO**  
UNIVERSIDAD PRIVADA ANTENOR ORREGO

# UNIDAD I

## SESIÓN 6: Librería PANDAS

### Limpieza de Datos

**UPAO**

UNIVERSIDAD PRIVADA ANTONOR ORREGO

# CASO PRÁCTICO: Cargar Datos CSV

```
import pandas as pd
```

```
df = pd.read_csv('../Sesion_06/Datos/Empleado.csv')  
df
```

	Edad	Renuncia	ViajesNegocios	TarifaDiaria	Departamento	DistanciaDesdeCasa
0	41	Yes	Travel_Rarely	1102	Sales	1
1	49	No	Travel_Frequently	279	Research & Development	8
2	37	Yes	Travel_Rarely	1373	Research & Development	2
3	33	No	Travel_Frequently	1392	Research & Development	3
4	37	Yes	Travel_Rarely	1373	NaN	2
...	...	...	...	...	...	...
1475	47	No	Non-Travel	1162	Research & Development	1
1476	35	No	NaN	1490	Research & Development	11
1477	22	No	NaN	581	Research & Development	1
1478	35	No	NaN	1395	Research & Development	9
1479	33	No	NaN	501	Research & Development	15

1480 rows × 35 columns



# Método `info()`:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1480 entries, 0 to 1479
Data columns (total 35 columns):
#   Column                                     Non-Null Count  Dtype
---  -
0   Edad                                     1480 non-null   int64
1   Renuncia                               1479 non-null   object
2   ViajesNegocios                         1476 non-null   object
3   TarifaDiaria                           1480 non-null   int64
4   Departamento                           1479 non-null   object
5   DistanciaDesdeCasa                    1480 non-null   int64
6   NivelEducacion                        1480 non-null   int64
7   Profesion                             1476 non-null   object
8   RecuentoEmpleados                     1480 non-null   int64
9   NumeroEmpleado                        1480 non-null   int64
10  SatisfaccionAmbienteTrabajo            1480 non-null   int64
11  Genero                                 1476 non-null   object
12  TarifaPorHora                          1480 non-null   int64
13  ParticipacionTrabajo                  1480 non-null   int64
14  NivelTrabajo                          1480 non-null   int64
15  RolTrabajador                         1478 non-null   object
16  SatisfaccionLaboral                   1480 non-null   int64
17  EstadoCivil                           1474 non-null   object
18  IngresosMensuales                     1480 non-null   int64
19  TarifaMensual                         1480 non-null   int64
20  NumeroEmpresasTrabajo                 1480 non-null   int64
21  MayorDe18                             1480 non-null   object
22  TiempoExtra                           1475 non-null   object
23  PorcentajeAumentoSalarial            1480 non-null   int64
24  CalificacionRendimiento                1480 non-null   int64
25  SatisfaccionRelacionLaboral           1480 non-null   int64
26  HorasEstandar                         1480 non-null   int64
27  NivelParticipacionAcciones            1480 non-null   int64
28  AñosLaboralesTotales                  1480 non-null   int64
29  NroCapacitacionUltimoAño              1480 non-null   int64
30  EquilibrioVidaLaboral                 1480 non-null   int64
31  AñosEmpresa                           1480 non-null   int64
32  AñosRolActual                         1479 non-null   float64
33  AñosDesdeUltimaPromocion              1479 non-null   float64
34  AñosComoJefe                           1480 non-null   int64
dtypes: float64(2), int64(24), object(9)
memory usage: 404.8+ KB
```

# Método `describe()`:

```
df.describe()
```

	Edad	TarifaDiaria	DistanciaDesdeCasa	NivelEducacion
<b>count</b>	1480.000000	1480.000000	1480.000000	1480.000000
<b>mean</b>	36.915541	802.883784	9.171622	2.914189
<b>std</b>	9.121168	403.785308	8.092201	1.025075
<b>min</b>	18.000000	102.000000	1.000000	1.000000
<b>25%</b>	30.000000	465.000000	2.000000	2.000000
<b>50%</b>	36.000000	801.500000	7.000000	3.000000
<b>75%</b>	43.000000	1159.000000	14.000000	4.000000
<b>max</b>	60.000000	1499.000000	29.000000	5.000000

8 rows × 26 columns



# CASO PRÁCTICO: Valores Nulos

# Método `isnull()`:

`DataFrame.isnull()`:

```
df.isnull()
```

	Edad	Renuncia	ViajesNegocios	TarifaDiaria	Departamento	DistanciaDesdeCasa
0	False	False	False	False	False	False
1	False	False	False	False	False	False
2	False	False	False	False	False	False
3	False	False	False	False	False	False
4	False	False	False	False	True	False
...	...	...	...	...	...	...
1475	False	False	False	False	False	False
1476	False	False	True	False	False	False
1477	False	False	True	False	False	False
1478	False	False	True	False	False	False
1479	False	False	True	False	False	False

1480 rows × 35 columns



# Método `isnull().sum()`:

`DataFrame.isnull().sum()`: Muestra el total de las columnas sin datos (nulos)

Mostrar el total de los valores nulos

```
df.isnull().sum().sum()
```

55

```
df.isnull().sum()
```

Edad	0
Renuncia	1
ViajesNegocios	4
TarifaDiaria	0
Departamento	1
DistanciaDesdeCasa	0
NivelEducacion	0
Profesion	4
RecuentoEmpleados	0
NumeroEmpleado	0
SatisfaccionAmbienteTrabajo	0
Genero	4
TarifaPorHora	0
ParticipacionTrabajo	0
NivelTrabajo	0
RolTrabajador	2
Satisfaccionlaboral	0
EstadoCivil	6
IngresosMensuales	0
TarifaMensual	0
NumeroEmpresasTrabajo	0
MayorDe18	0
TiempoExtra	5
PorcentajeAumentoSalarial	0
CalificacionRendimiento	0
SatisfaccionRelacionLaboral	0
HorasEstandar	0
NivelParticipacionAcciones	0
AñosLaboralesTotales	0
NroCapacitacionUltimoAño	0
EquilibrioVidaLaboral	0
AñosEmpresa	0
AñosRolActual	1
AñosDesdeUltimaPromocion	1
AñosComoJefe	0
dtype: int64	

# VALORES VACIOS (NULOS)

- 
- Eliminar las filas o columnas con valores nulos

- 
- Reemplazar los valores nulos con datos (imputación)

# Método `dropna()`:

- Eliminar las filas o columnas con valores nulos

## `DataFrame.dropna()`:

Elimina cualquier fila con al menos un valor nulo, pero devolverá un nuevo DataFrame sin alterar el original.

	Edad	Renuncia	ViajesNegocios	TarifaDiaria	Departamento
0	41	Yes	Travel_Rarely	1102	Sales
1	49	No	Travel_Frequently	279	Research & Development
2	37	Yes	Travel_Rarely	1373	Research & Development
3	33	No	Travel_Frequently	1392	Research & Development
4	37	Yes	Travel_Rarely	1373	NaN
...	...	...	...	...	...
1475	47	No	Non-Travel	1162	Research & Development
1476	35	No	NaN	1490	Research & Development
1477	22	No	NaN	581	Research & Development
1478	35	No	NaN	1395	Research & Development
1479	33	No	NaN	501	Research & Development

1480 rows × 35 columns

	Edad	Renuncia	ViajesNegocios	TarifaDiaria	Departamento
0	41	Yes	Travel_Rarely	1102	Sales
1	49	No	Travel_Frequently	279	Research & Development
2	37	Yes	Travel_Rarely	1373	Research & Development
3	33	No	Travel_Frequently	1392	Research & Development
5	27	No	Travel_Rarely	591	Research & Development
...	...	...	...	...	...
1466	36	No	Travel_Frequently	884	Research & Development
1467	39	No	Travel_Rarely	613	Research & Development
1468	27	No	Travel_Rarely	155	Research & Development
1469	49	No	Travel_Frequently	1023	Sales
1470	34	No	Travel_Rarely	628	Research & Development

1470 rows × 35 columns

# Método `dropna(axis=1)`:

```
df.dropna(axis=1)
```

	Edad	TarifaDiaria	DistanciaDesdeCasa	NivelEducacion	RecuentoEmpleados
0	41	1102	1	2	1
1	49	279	8	1	1
2	37	1373	2	2	1
3	33	1392	3	4	1
4	37	1373	2	2	1
...	...	...	...	...	...
1475	47	1162	1	1	1
1476	35	1490	11	4	1
1477	22	581	1	2	1
1478	35	1395	9	4	1
1479	33	501	15	2	1

1480 rows × 25 columns

4

# Método `fillna()`:

- Reemplazar los valores nulos con datos (imputación)

Extraemos los valores de las columnas en una variable

```
vColumnaDistanciaCasa = df['DistanciaDesdeCasa']
```

```
vColumnaDistanciaCasa.head
```

```
<bound method NDFrame.head of 0      1.0>
1      8.0
2      2.0
3      3.0
4      2.0
...
1475    1.0
1476   11.0
1477    1.0
1478    9.0
1479    NaN
Name: DistanciaDesdeCasa, Length: 1480, dtype: float64>
```

Imputaremos asignando la media, para ello calculamos la media.

```
ColumnaDistanciaCasaMedia = vColumnaDistanciaCasa.mean()
ColumnaDistanciaCasaMedia
```

```
9.16508152173913
```

Imputaremos asignando la media.

```
vColumnaDistanciaCasa.fillna(ColumnaDistanciaCasaMedia, inplace=True)
```

```
df.isnull().sum()
Edad      0
Renuncia  1
ViajesNegocios  4
TarifaDiaria  0
Departamento  1
DistanciaDesdeCasa  0
NivelEducacion  0
Profesion  4
RecuentoEmpleados  1
NumeroEmpleado  0
SatisfaccionAmbienteTrabajo  0
Genero  4
TarifaPorHora  0
ParticipacionTrabajo  0
NivelTrabajo  0
RolTrabajador  2
Satisfaccionlaboral  0
EstadoCivil  6
IngresosMensuales  0
TarifaMensual  0
NumeroEmpresasTrabajo  0
MayorDe18  0
TiempoExtra  5
PorcentajeAumentoSalarial  0
CalificacionRendimiento  0
SatisfaccionRelacionLaboral  0
HorasEstandar  0
NivelParticipacionAcciones  0
AñosLaboralesTotales  0
NroCapacitacionUltimoAño  0
EquilibrioVidaLaboral  0
AñosEmpresa  0
AñosRolActual  1
AñosDesdeUltimaPromocion  1
AñosComoJefe  0
dtype: int64
```

# Método `fillna()`:

- Reemplazar los valores nulos con datos (imputación)

Extraemos los valores de las columnas en una variable

```
vcolumna= df['TiempoExtra']
```

```
vcolumna.head()
```

```
0    Yes
1    No
2    Yes
3    Yes
4    Yes
Name: TiempoExtra, dtype: object
```

Asignamos valores a los datos vacíos.

```
vcolumna.fillna("Yes", inplace=True)
```

```
df.isnull().sum()
```

```
Edad                0
Renuncia            1
ViajesNegocios      4
TarifaDiaria        0
Departamento       1
DistanciaDesdeCasa  0
NivelEducacion      0
Profesion           4
RecuentoEmpleados   0
NumeroEmpleado      0
SatisfaccionAmbienteTrabajo  0
Genero              4
TarifaPorHora        0
ParticipacionTrabajo  0
NivelTrabajo         0
RolTrabajador        2
Satisfaccionlaboral  0
EstadoCivil          6
IngresosMensuales    0
TarifaMensual        0
NumeroEmpresasTrabajo  0
MayorDe18           0
TiempoExtra         0
PorcentajeAumentoSalarial  0
CalificacionRendimiento  0
SatisfaccionRelacionLaboral  0
HorasEstandar        0
NivelParticipacionAcciones  0
AñosLaboralesTotales  0
NroCapacitacionUltimoAño  0
EquilibrioVidaLaboral  0
AñosEmpresa          0
AñosRolActual         1
AñosDesdeUltimaPromocion  1
AñosComoJefe          0
dtype: int64
```



# Método `bfill()`:

**DataFrame.bfill(inplace=True):**

Rellena los valores de NA/NaN utilizando el siguiente (hacia adelante) valor válido para rellenar el hueco.

```
df.bfill(inplace = True)
df
```

	Edad	Renuncia	ViajesNegocios	TarifaDiaria	Departamento	DistanciaDesdeCasa	NivelEducacion	Profesion
0	41	Yes	Travel_Rarely	1102	Sales	1.0	2	Life Sciences
1	49	No	Travel_Frequently	279	Research & Development	8.0	1	Life Sciences
2	37	Yes	Travel_Rarely	1373	Research & Development	2.0	2	Other
3	33	No	Travel_Frequently	1392	Research & Development	3.0	4	Life Sciences
4	37	Yes	Travel_Rarely	1373	Research & Development	2.0	2	Other
...	...	...	...	...	...	...	...	...
1480	47	No	Non-Travel	1162	Research & Development	1.0	1	Medical
1481	35	No	NaN	1490	Research & Development	11.0	4	Medical
1482	22	No	NaN	581	Research & Development	1.0	2	Life Sciences
1483	35	No	NaN	1395	Research & Development	9.0	4	Medical
1484	33	No	NaN	501	Research & Development	NaN	2	Medical

# Método `ffill()`:

**DataFrame.bfill(inplace=True):**

Rellena los valores de NA/NaN utilizando el último (hacia atrás) valor válido para rellenar el hueco.

```
df.ffmpeg(inplace = True)
df
```

	Edad	Renuncia	ViajesNegocios	TarifaDiaria	Departamento	DistanciaDesdeCasa	NivelEducacion	Profesion
0	41	Yes	Travel_Rarely	1102	Sales	1.0	2	Life Sciences
1	49	No	Travel_Frequently	279	Research & Development	8.0	1	Life Sciences
2	37	Yes	Travel_Rarely	1373	Research & Development	2.0	2	Other
3	33	No	Travel_Frequently	1392	Research & Development	3.0	4	Life Sciences
4	37	Yes	Travel_Rarely	1373	Research & Development	2.0	2	Other
...	...	...	...	...	...	...	...	...
1480	47	No	Non-Travel	1162	Research & Development	1.0	1	Medical
1481	35	No	Non-Travel	1490	Research & Development	11.0	4	Medical
1482	22	No	Non-Travel	581	Research & Development	1.0	2	Life Sciences
1483	35	No	Non-Travel	1395	Research & Development	9.0	4	Medical
1484	33	No	Non-Travel	501	Research & Development	9.0	2	Medical



# CASO PRÁCTICO: Eliminar Duplicados

Método `df.duplicated()`:

Método `df.drop_duplicates()`:

```
df.duplicated()
```

```
0      False
1      False
2      False
3      False
4      False
...
1480    True
1481    True
1482    True
1483    True
1484    True
Length: 1485, dtype: bool
```

```
df.duplicated().sum()
```

5

```
df = df.drop_duplicates()
df
```

```
df.duplicated()
```

```
0      False
1      False
2      False
3      False
4      False
...
1475    False
1476    False
1477    False
1478    False
1479    False
Length: 1480, dtype: bool
```

```
df.duplicated().sum()
```

0

# CASO PRÁCTICO: Selección de Datos

## Mostrar los Empleados que hicieron horas Extras

```
busqueda = (df['TiempoExtra'] == "Yes")  
busqueda.head()
```

```
0    True  
1    False  
2     True  
3     True  
4     True  
Name: TiempoExtra, dtype: bool
```

```
df[df['TiempoExtra'] == "Yes"]
```

	Edad	Renuncia	ViajesNegocios	TarifaDiaria	Departamento
0	41	Yes	Travel_Rarely	1102	Sales
2	37	Yes	Travel_Rarely	1373	Research & Development
3	33	No	Travel_Frequently	1392	Research & Development
4	37	Yes	Travel_Rarely	1373	NaN
7	59	No	Travel_Rarely	1324	Research & Development
...	...	...	...	...	...
1451	35	No	Travel_Rarely	1146	Human Resources
1457	35	No	Travel_Frequently	1199	Research & Development
1460	29	No	Travel_Rarely	1378	Research & Development
1462	50	Yes	Travel_Rarely	410	Sales
1468	27	No	Travel_Rarely	155	Research & Development

417 rows × 35 columns

# Práctica



Seleccionar los empleados que hicieron horas extras y que tienen una distancia mayor a 20 km. :

```
df[(df['TiempoExtra'] == "Yes") & (df['DistanciaDesdeCasa'] > 20)]
```

	Edad	Renuncia	ViajesNegocios	TarifaDiaria	Departamento	DistanciaDesdeCasa
<b>15</b>	28	Yes	Travel_Rarely	103	Research & Development	24.0
<b>55</b>	26	No	Travel_Rarely	1443	Sales	23.0
<b>58</b>	35	No	Travel_Rarely	1142	Research & Development	23.0
<b>92</b>	51	No	Travel_Rarely	632	Sales	21.0
<b>113</b>	54	No	Non-Travel	142	Human Resources	26.0
...	...	...	...	...	...	...
<b>1396</b>	31	Yes	Travel_Frequently	754	Sales	26.0
<b>1397</b>	53	Yes	Travel_Rarely	1168	Sales	24.0
<b>1402</b>	55	No	Travel_Rarely	189	Human Resources	26.0
<b>1451</b>	35	No	Travel_Rarely	1146	Human Resources	26.0
<b>1462</b>	50	Yes	Travel_Rarely	410	Sales	28.0

61 rows × 35 columns

# Práctica



Seleccionar los empleados que trabajan en el departamento de Ventas o Recursos Humanos:

```
df[(df['Departamento'] == "Sales") | (df['Departamento'] == 'Human Resources')]
```

	Edad	Renuncia	ViajesNegocios	TarifaDiaria	Departamento
0	41	Yes	Travel_Rarely	1102	Sales
19	53	No	Travel_Rarely	1219	Sales
22	36	Yes	Travel_Rarely	1218	Sales
28	42	No	Travel_Rarely	691	Sales
30	46	No	Travel_Rarely	705	Sales
...	...	...	...	...	...
1462	50	Yes	Travel_Rarely	410	Sales
1463	39	No	Travel_Rarely	722	Sales
1465	26	No	Travel_Rarely	1167	Sales
1469	49	No	Travel_Frequently	1023	Sales
1472	35	No	Travel_Rarely	776	Sales

510 rows × 35 columns



# Método `isin()`:

```
df[df['Departamento'].isin(['Sales', 'Human Resources'])]
```

	Edad	Renuncia	ViajesNegocios	TarifaDiaria	Departamento
0	41	Yes	Travel_Rarely	1102	Sales
19	53	No	Travel_Rarely	1219	Sales
22	36	Yes	Travel_Rarely	1218	Sales
28	42	No	Travel_Rarely	691	Sales
30	46	No	Travel_Rarely	705	Sales
...	...	...	...	...	...
1462	50	Yes	Travel_Rarely	410	Sales
1463	39	No	Travel_Rarely	722	Sales
1465	26	No	Travel_Rarely	1167	Sales
1469	49	No	Travel_Frequently	1023	Sales
1472	35	No	Travel_Rarely	776	Sales

510 rows × 35 columns

# CASO PRÁCTICO: Agregar Columnas

# Método `assign()`:

```
totalPagoPorHorasTrabajadas = df["TarifaPorHora"] * df["HorasEstandar"]  
df = df.assign(TotalPagoPorHorasTrabajadas = totalPagoPorHorasTrabajadas)  
df
```

AñosDesdeUltimaPromocion	AñosComoJefe	TotalPagoPorHorasTrabajadas
0.0	5	7520
1.0	7	4880
0.0	0	7360
3.0	0	4480
0.0	0	7360
...	...	...
5.0	12	7840
2.0	2	3440

# Método `insert()`:

```
totalPagoPorHorasTrabajadas = df["TarifaPorHora"] * df["HorasEstandar"]  
df.insert(1,"TotalPagoPorHorasTrabajadas2",totalPagoPorHorasTrabajadas)  
df
```

	Edad	TotalPagoPorHorasTrabajadas2	R
0	41	7520	
1	49	4880	
2	37	7360	
3	33	4480	
4	37	7360	
...	...	...	
1480	47	7840	
1481	35	3440	

# Práctica



Agregue una Columna “AumentoSalario”, para aquellos empleados que tienen más de 3 capacitaciones durante el último año y que los años trabajados en la empresa sean mayores a 30.

Para el cálculo considere los ingresos mensuales y el porcentaje de aumento.

# CASO PRÁCTICO: Cambiar el Nombre de las Columnas

# Método `rename()`:

```
df.rename(columns = {'ViajesNegocios': 'ViajesPorNegocios', 'Departamento': 'AreaEmpresa'}, inplace = True)  
df
```

	Edad	Renuncia	ViajesNegocios	TarifaDiaria	Departamento	DistanciaDesdeCasa	NivelEducacion	Profesion
412	60	No	Travel_Rarely	422	Research & Development	7.0	3	Life Sciences
428	60	No	Travel_Frequently	1499	Sales	28.0	3	Marketing
537	60	No	Travel_Rarely	1179	Sales	16.0	4	Marketing
880	60	No	Travel_Rarely	696	Sales	7.0	4	Marketing
1210	60	No	Travel_Rarely	370	Research & Development	1.0	4	Medical

	Edad	Renuncia	ViajesPorNegocios	TarifaDiaria	AreaEmpresa	DistanciaDesdeCasa	NivelEducacion	Profesion
412	60	No	Travel_Rarely	422	Research & Development	7.0	3	Life Sciences
428	60	No	Travel_Frequently	1499	Sales	28.0	3	Marketing
537	60	No	Travel_Rarely	1179	Sales	16.0	4	Marketing
880	60	No	Travel_Rarely	696	Sales	7.0	4	Marketing
1210	60	No	Travel_Rarely	370	Research & Development	1.0	4	Medical

# CASO PRÁCTICO: Reemplazar Valores de los Datos



Cambiar los valores de la columna “Nivel de Educación”:

- 1: Bachiller
- 2: Titulado
- 3: Magister
- 4: Doctor

```
df['NivelEducacion'] = df['NivelEducacion'].replace([1,2,3,4],["Bachiller", "Titulado", "Maestría", "Doctorado"])
df
```

	Edad	Renuncia	ViajesNegocios	TarifaDiaria	Departamento	DistanciaDesdeCasa	NivelEducacion	Profesion
0	41	Yes	Travel_Rarely	1102	Sales	1.0	Titulado	Life Sciences
1	49	No	Travel_Frequently	279	Research & Development	8.0	Bachiller	Life Sciences
2	37	Yes	Travel_Rarely	1373	Research & Development	2.0	Titulado	Other
3	33	No	Travel_Frequently	1392	Research & Development	3.0	Doctorado	Life Sciences
4	37	Yes	Travel_Rarely	1373	NaN	2.0	Titulado	Other
...	...	...	...	...	...	...	...	...
1480	47	No	Non-Travel	1162	Research & Development	1.0	Bachiller	NaN
1481	35	No	NaN	1490	Research & Development	11.0	Doctorado	Medical

# Práctica



Reemplace los valores de la columna “Nivel de Participación Acciones”:

0: Ninguna

1: Baja

2: Media

3: Alta



# ¡Gracias!

**UPAO**  
UNIVERSIDAD PRIVADA ANTENOR ORREGO