Ingeniería de Computación y Sistemas

INGENIERÍA DE DATOS

Mg. Ing. Carlos Edwin Julca Castillo

UPAO
UNIVERSIDAD PRIVADA ANTENOR ORREGO

UNIDAD I

SESIÓN 5: Librería PANDAS

UPAO

UNIVERSIDAD PRIVADA ANTENOR ORREGO

PANDAS

Es una biblioteca para **limpieza** y **análisis** de datos construida sobre el lenguaje de programación Python. Proporciona estructuras de datos rápidas, expresivas y flexibles para trabajar de forma fácil (e intuitiva) con datos estructurados (tabulares, multidimensionales, potencialmente heterogéneos) y de series temporales.

UPAO
UNIVERSIDAD PRIVADA ANTENOR ORREGO

PANDAS

Generalmente proporcionan dos estructuras de datos para manipular datos:

DataFrame

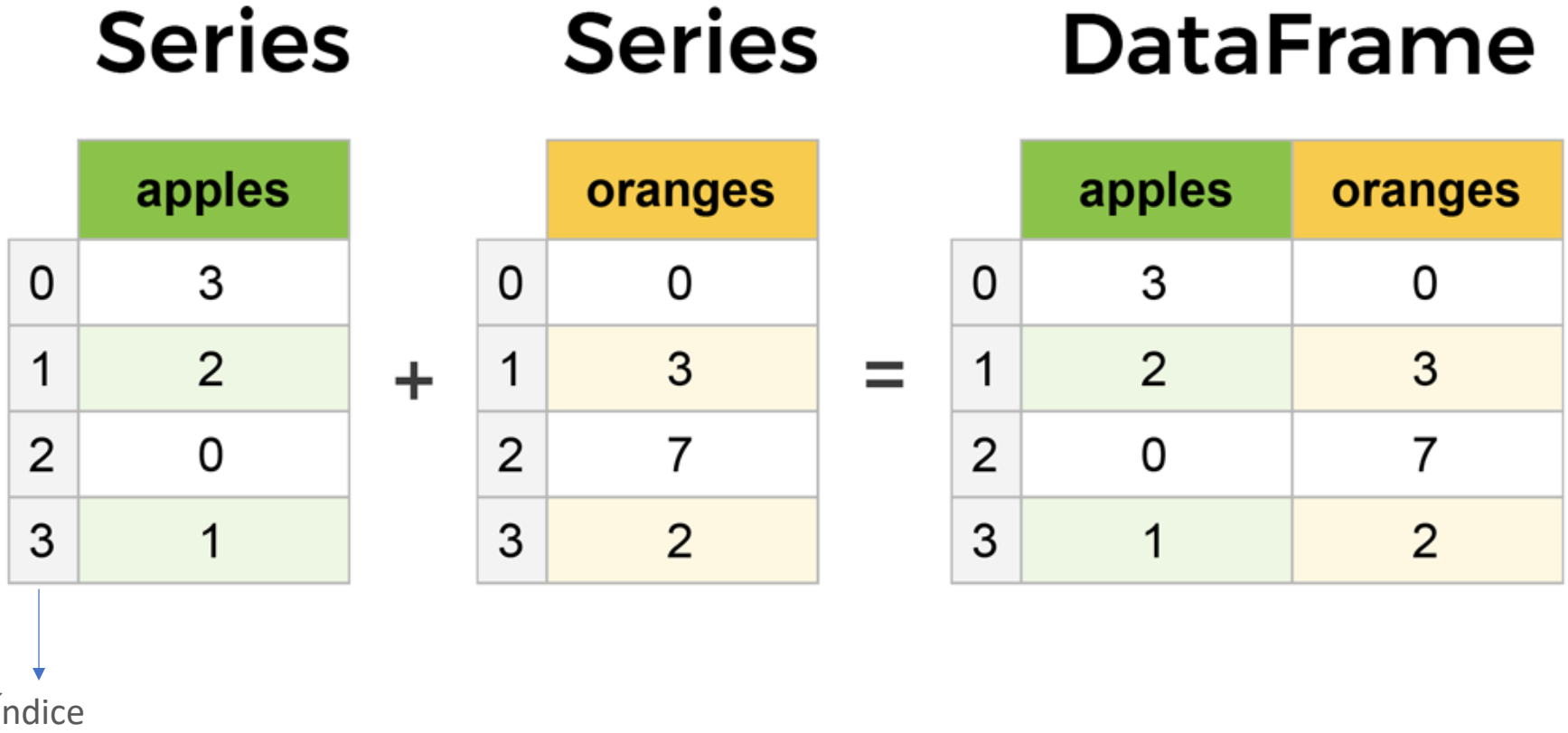
- Es una estructura de datos bidimensional, es decir, los datos están alineados en forma de tabla en filas y columnas.
- Un Pandas DataFrame se crea cargando los conjuntos de datos desde un almacenamiento existente. El almacenamiento puede ser una base de datos SQL, un archivo CSV, un archivo Excel, entre otros.
- También se puede crear a partir de listas, diccionario y de una lista de diccionarios.

Series

- Representa una matriz unidimensional de datos indexados.
- Tiene dos componentes principales:
  - ✓Una **serie** de datos.
  - ✓Una matriz asociada por **índices** o etiquetas de datos. El índice se utiliza para acceder a valores de datos individuales. Tambien se puede obtener una columna de un DataFrame como una serie.

UPAO
UNIVERSIDAD PRIVADA ANTENOR ORREGO

PANDAS


Series
Series
DataFrame
apples
oranges
apples
oranges
0 3
1 2
2 0
3 1
+
0 0
1 3
2 7
3 2
=
0 3 0
1 2 3
2 0 7
3 1 2
Índice

PANDAS.DATAFRAME

- ✓Es una estructura de datos etiquetada bidimensional como una tabla con filas y columnas. El tamaño y los valores del DataFrame son **mutables**, es decir, se pueden modificar.
- ✓DataFrame se utiliza principalmente en **análisis** y **manipulación** de datos. Le permite almacenar datos en forma tabular, como una base de datos SQL, MS Excel o Google Sheets, lo que facilita la **realización** de **operaciones aritméticas** con los datos.
- ✓Es el objeto Pandas más utilizado. La **función** DataFrame() se utiliza para crear un DataFrame en Pandas.

UPAO
UNIVERSIDAD PRIVADA ANTENOR ORREGO

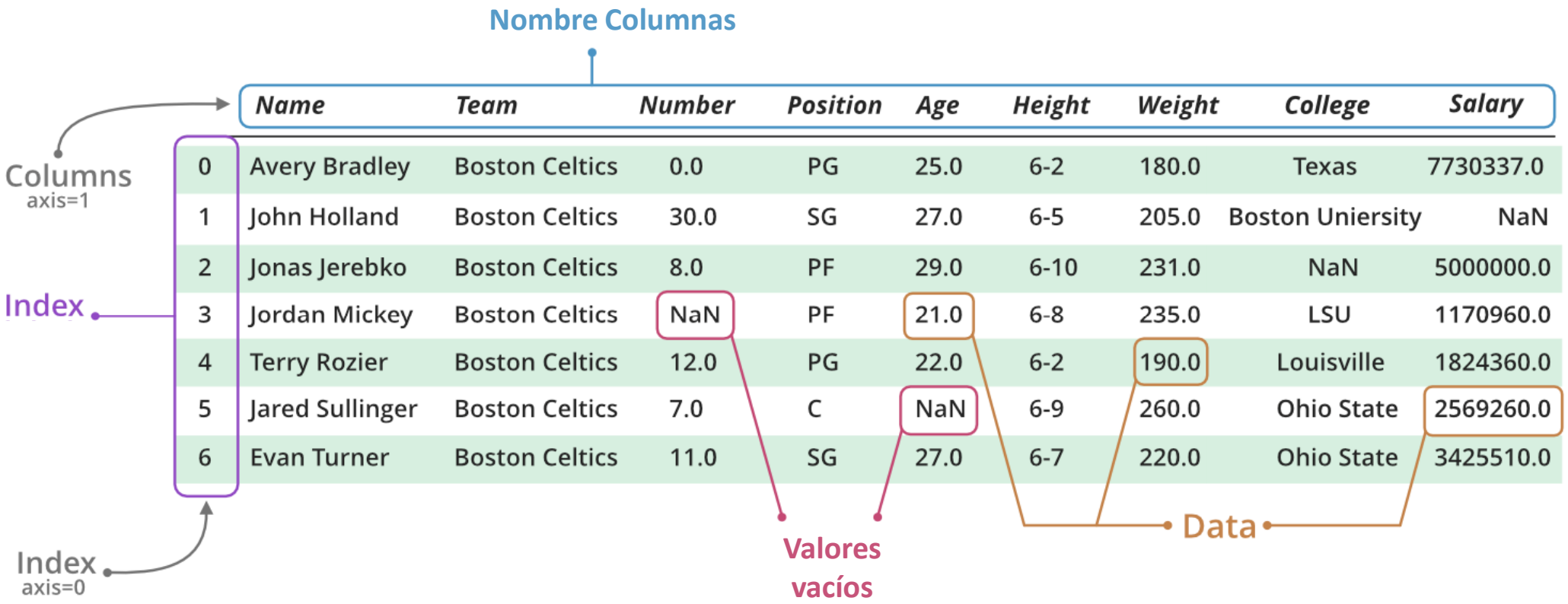
PANDAS.DATAFRAME

SINTAXIS:

```
pandas.DataFrame(datos, índice, columnas)
```

datos	Es un conjunto de datos a partir del cual se creará un DataFrame. Puede ser una lista, un diccionario, un valor escalar, una serie y una matriz, etc.
índice	Es opcional, de forma predeterminada el índice del DataFrame comienza desde 0 y termina en el último valor de datos (n-1). Define explícitamente la etiqueta de la fila.
columnas	Este parámetro se utiliza para proporcionar nombres de columnas en el DataFrame. Si el nombre de la columna no está definido de forma predeterminada, tomará un valor de 0 a n-1.

PANDAS.DATAFRAME



	Name	Team	Number	Position	Age	Height	Weight	College	Salary
0	Avery Bradley	Boston Celtics	0.0	PG	25.0	6-2	180.0	Texas	7730337.0
1	John Holland	Boston Celtics	30.0	SG	27.0	6-5	205.0	Boston Uniersity	NaN
2	Jonas Jerebko	Boston Celtics	8.0	PF	29.0	6-10	231.0	NaN	5000000.0
3	Jordan Mickey	Boston Celtics	NaN	PF	21.0	6-8	235.0	LSU	1170960.0
4	Terry Rozier	Boston Celtics	12.0	PG	22.0	6-2	190.0	Louisville	1824360.0
5	Jared Sullinger	Boston Celtics	7.0	C	NaN	6-9	260.0	Ohio State	2569260.0
6	Evan Turner	Boston Celtics	11.0	SG	27.0	6-7	220.0	Ohio State	3425510.0

CASO PRÁCTICO: Crear un DataFrame

UPAO
UNIVERSIDAD PRIVADA ANTENOR ORREGO

jupyter Ejemplo_Panda Last Checkpoint: hace 8 horas (autosaved)

Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3 (ipykernel)

Run Markdown

Crear un DataFrame a partir de un Dicionario

In [4]:
```
# Importamos la librería Pandas

import pandas as pd
```

In [6]:
```
# Definimos el diccionario d

d = { 'Nombre':['Rosa','Juan', 'María', 'Pedro'],
     'ID': [1, 2, 3, 4],
     'Departamento': ['Ventas', 'Logística','Contabilidad','Sistemas'],
     'Sueldo': [1500, 2000, 2500, 3000]
}

# Convertimos el Diccionario a un DataFrame

DF = pd.DataFrame(d)

# Mostramos los resultados

DF
```

Out[6]:

	Nombre	ID	Departamento	Sueldo
0	Rosa	1	Ventas	1500
1	Juan	2	Logística	2000
2	María	3	Contabilidad	2500
3	Pedro	4	Sistemas	3000

CASO PRÁCTICO: Seleccionar Columnas

UPAO
UNIVERSIDAD PRIVADA ANTENOR ORREGO

Seleccionar Columnas

Para seleccionar una columna del DataFrame, accedemos a las columnas llamándolas por su nombre.

Recuperemos los datos presentes en la columna ID.

In [11]:
```
# Recuperamos la Columna ID y le asignamos a la variable "d"

d = DF[['ID']]

d
```

Out[11]:

	ID
0	1
1	2
2	3
3	4

Usamos la función type() para verifiquemos el tipo de variable.

In [12]:
```
#Verificamos el tipo de variable de d

type (d)
```

Out[12]:
```
pandas.core.frame.DataFrame
```

El resultado nos muestra que el tipo de variable es un objeto DataFrame.

Seleccionar dos columnas

```
DF[['Nombre','Sueldo']]
```

	Nombre	Sueldo
0	Rosa	1500
1	Juan	2000
2	María	2500
3	Pedro	3000

Seleccionar de la Columna 2 hasta la columna 4 (un rango)

```
df[df.columns[1:4]]
```

	Renuncia	ViajesNegocios	TarifaDiaria
0	Yes	Travel_Rarely	1102
1	No	Travel_Frequently	279
2	Yes	Travel_Rarely	1373
3	No	Travel_Frequently	1392
4	Yes	Travel_Rarely	1373
...	...	...	...
1475	No	Non-Travel	1162
1476	No	NaN	1490
1477	No	NaN	581
1478	No	NaN	1395
1479	No	NaN	501

UPAO
UNIVERSIDAD PRIVADA ANTENOR ORREGO

Acceso a Múltiples Columnas

Recuperemos los datos de las columnas: ID, Departamento y Sueldo

In [13]:
```
# Recuperamos las columnas ID, Departamento y Sueldo y las asignamos a la variable "x"
```

In [14]:
```
x = DF[['ID','Departamento','Sueldo']]
x
```

Out[14]:

	ID	Departamento	Sueldo
0	1	Ventas	1500
1	2	Logística	2000
2	3	Contabilidad	2500
3	4	Sistemas	3000

In [ ]:
```

```

CASO PRÁCTICO: Configuración de Índices de un DataFrame (Indexación)

UPAO
UNIVERSIDAD PRIVADA ANTENOR ORREGO

El índice de un DataFrame se crea con números desde 0, pero se puede asignar etiquetas o propios valores.

```
dfi = DF.set_index('Nombre')

dfi
```

	Nombre	ID	Departamento	Sueldo
0	Rosa	1	Ventas	1500
1	Juan	2	Logística	2000
2	María	3	Contabilidad	2500
3	Pedro	4	Sistemas	3000

Nombre	ID	Departamento	Sueldo
Rosa	1	Ventas	1500
Juan	2	Logística	2000
María	3	Contabilidad	2500
Pedro	4	Sistemas	3000

```
DF.index = ['Uno','Dos','Tres','Cuatro']
DF
```

	Nombre	ID	Departamento	Sueldo
0	Rosa	1	Ventas	1500
1	Juan	2	Logística	2000
2	María	3	Contabilidad	2500
3	Pedro	4	Sistemas	3000

	Nombre	ID	Departamento	Sueldo
Uno	Rosa	1	Ventas	1500
Dos	Juan	2	Logística	2000
Tres	María	3	Contabilidad	2500
Cuatro	Pedro	4	Sistemas	3000

```
DFi = pd.DataFrame(d,index = ['Ventas','Logística', 'Contabilidad', 'Sistemas'])

DFi
```

	Nombre	ID	Departamento	Sueldo
0	Rosa	1	Ventas	1500
1	Juan	2	Logística	2000
2	María	3	Contabilidad	2500
3	Pedro	4	Sistemas	3000

	Nombre	ID	Departamento	Sueldo
Ventas	Rosa	1	Ventas	1500
Logística	Juan	2	Logística	2000
Contabilidad	María	3	Contabilidad	2500
Sistemas	Pedro	4	Sistemas	3000

CASO PRÁCTICO: Segmentación de un DataFrame

UPAO
UNIVERSIDAD PRIVADA ANTENOR ORREGO

SEGMENTACIÓN DE UN DATAFRAME

Es posible segmentar un DataFrame para recuperar filas de él.

```
DF [0:2]
```

	Nombre	ID	Departamento	Sueldo
0	Rosa	1	Ventas	1500
1	Juan	2	Logística	2000
2	María	3	Contabilidad	2500
3	Pedro	4	Sistemas	3000

	Nombre	ID	Departamento	Sueldo
0	Rosa	1	Ventas	1500
1	Juan	2	Logística	2000

UPAO
UNIVERSIDAD PRIVADA ANTENOR ORREGO

SEGMENTACIÓN DE UN DATAFRAME

Loc()

Es un método de selección de datos basado en etiquetas, lo que significa que tenemos que pasar el nombre de la fila o columna que queremos seleccionar.

Este método incluye el último elemento del rango pasado.

Sintaxis simple para su comprensión:

```
loc[etiqueta_fila, etiqueta_columna]
```

iLoc()

Es un método de selección basado en índices, lo que significa que tenemos que pasar un índice entero en el método para seleccionar una fila/columna específica..

Este método no incluye el último elemento del rango pasado.

Sintaxis simple para su comprensión:

```
iloc[índice_fila, índice_columna]
```

Métodos Loc() e iLoc()

In [38]:
```
# Accedemos a la columna utilizando el nombre

DF.loc[0,'Sueldo']
```

Out[38]:
```
1500
```

In [43]:
```
# Accedemos al valor de la Primera Fila y la Primera Columna

DF.iloc[0,0]
```

Out[43]:
```
'Rosa'
```

In [40]:
```
# Accedemos a los valores del Primer Objeto

DF.iloc[0]
```

Out[40]:
```
Nombre            Rosa
ID                   1
Departamento    Ventas
Sueldo            1500
Name: 0, dtype: object
```

In [44]:
```
# Accedemos al valor de la Primera Fila y de la Tercera Columna

DF.iloc[0,2]
```

Out[44]:
```
'Ventas'
```

Nombre	ID	Departamento	Sueldo
Rosa	1	Ventas	1500
Juan	2	Logística	2000
María	3	Contabilidad	2500
Pedro	4	Sistemas	3000

Index

Para el DataFrame dfi mostrar el Departamento por el nombre del Empleado

```
dfi.loc['Pedro', 'Departamento']
```

```
'Sistemas'
```

UPAO
UNIVERSIDAD PRIVADA ANTENOR ORREGO

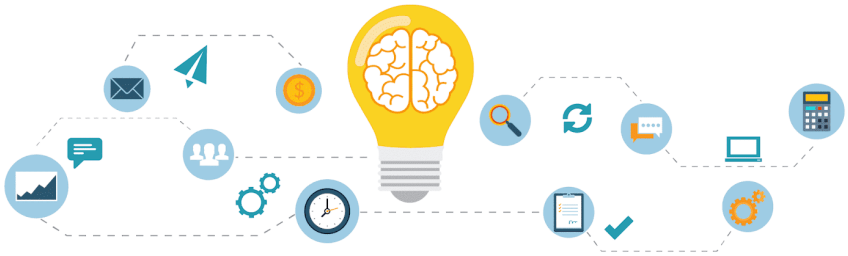
Seleccionar las dos primeras filas desde la columna Nombre hasta la columna Departamento

```
DF.loc[0:1, 'Nombre':'Departamento']
```

	Nombre	ID	Departamento
0	Rosa	1	Ventas
1	Juan	2	Logística

UPAO
UNIVERSIDAD PRIVADA ANTENOR ORREGO

Práctica



Crear un DataFrame para obtener el siguiente resultado:

	Estudiante	Edad	Promedio	Ciclo	Curso
0	Juan	19	15	V	Ingeniería de Datos
1	Carlos	20	12	VI	Base de Datos
2	Rosa	18	8	V	Algoritmos
3	Jean	18	18	IV	Tesis
4	Lourdes	20	10	IX	Proyecto de Tesis
5	Mario	18	17	V	Programación

- Recuperar la Columna Promedio y asignala a una variable “n”.
- Recuperar las Columnas Curso y Ciclo y asignarlas a una variable “c”.
- Configure Índices.
- Ejecute Segmentación de DataFrame.

UPAO
UNIVERSIDAD PRIVADA ANTENOR ORREGO

CASO PRÁCTICO: Métodos del DataFrame

UPAO
UNIVERSIDAD PRIVADA ANTENOR ORREGO

Método head():

DataFrame.head(): Devuelve las primeras filas del DataFrame.

```
DF.head()
```

	Estudiante	Edad	Promedio	Ciclo	Curso
0	Juan	19	15	V	Ingeniería de Datos
1	Carlos	20	12	VI	Base de Datos
2	Rosa	18	8	V	Algoritmos
3	Jean	18	18	IV	Tesis
4	Lourdes	20	10	IX	Proyecto de Tesis
5	Mario	18	17	V	Programación

	Estudiante	Edad	Promedio	Ciclo	Curso
0	Juan	19	15	V	Ingeniería de Datos
1	Carlos	20	12	VI	Base de Datos
2	Rosa	18	8	V	Algoritmos
3	Jean	18	18	IV	Tesis
4	Lourdes	20	10	IX	Proyecto de Tesis

```
DF.head(2)
```

	Estudiante	Edad	Promedio	Ciclo	Curso
0	Juan	19	15	V	Ingeniería de Datos
1	Carlos	20	12	VI	Base de Datos

Método tail():

DataFrame.tail(): Devuelve las ultimas filas del DataFrame.

```
DF.tail()
```

	Estudiante	Edad	Promedio	Ciclo	Curso
0	Juan	19	15	V	Ingeniería de Datos
1	Carlos	20	12	VI	Base de Datos
2	Rosa	18	8	V	Algoritmos
3	Jean	18	18	IV	Tesis
4	Lourdes	20	10	IX	Proyecto de Tesis
5	Mario	18	17	V	Programación

	Estudiante	Edad	Promedio	Ciclo	Curso
1	Carlos	20	12	VI	Base de Datos
2	Rosa	18	8	V	Algoritmos
3	Jean	18	18	IV	Tesis
4	Lourdes	20	10	IX	Proyecto de Tesis
5	Mario	18	17	V	Programación

```
DF.tail(3)
```

	Estudiante	Edad	Promedio	Ciclo	Curso
3	Jean	18	18	IV	Tesis
4	Lourdes	20	10	IX	Proyecto de Tesis
5	Mario	18	17	V	Programación

UPAO
UNIVERSIDAD PRIVADA ANTENOR ORREGO

Método sample():

DataFrame.sample(): Muestra valores aleatorios del DataFrame.

```
DF.sample(3)
```

	Estudiante	Edad	Promedio	Ciclo	Curso
0	Juan	19	15	V	Ingeniería de Datos
1	Carlos	20	12	VI	Base de Datos
2	Rosa	18	8	V	Algoritmos
3	Jean	18	18	IV	Tesis
4	Lourdes	20	10	IX	Proyecto de Tesis
5	Mario	18	17	V	Programación

	Estudiante	Edad	Promedio	Ciclo	Curso
3	Jean	18	18	IV	Tesis
4	Lourdes	20	10	IX	Proyecto de Tesis
0	Juan	19	15	V	Ingeniería de Datos

```
DF.sample()
```

	Estudiante	Edad	Promedio	Ciclo	Curso
3	Jean	18	18	IV	Tesis

UPAO
UNIVERSIDAD PRIVADA ANTENOR ORREGO

Método info():

DataFrame.info(): Muestra los nombres de la columnas, los tipos de datos y si tienen valores faltantes.

```
DF.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 6 entries, 0 to 5
Data columns (total 5 columns):
 #   Column      Non-Null Count  Dtype
---  ------      --------------  -----
 0   Estudiante  6 non-null      object
 1   Edad        6 non-null      object
 2   Promedio    6 non-null      int64
 3   Ciclo       6 non-null      object
 4   Curso       6 non-null      object
dtypes: int64(1), object(4)
memory usage: 372.0+ bytes
```

UPAO
UNIVERSIDAD PRIVADA ANTENOR ORREGO

Método describe():

DataFrame.describe(): Calcula algunas estadísticas de resumen para las columnas numéricas, como la media, el valor máximo, el mínimo.

```
DF.describe()
```

	Promedio
count	6.000000
mean	13.333333
std	3.983298
min	8.000000
25%	10.500000
50%	13.500000
75%	16.500000
max	18.000000

Muestra algunas estadísticas para la columnas Promedio

	ID	Sueldo
count	4.000000	4.000000
mean	2.500000	2250.000000
std	1.290994	645.497224
min	1.000000	1500.000000
25%	1.750000	1875.000000
50%	2.500000	2250.000000
75%	3.250000	2625.000000
max	4.000000	3000.000000

Muestra algunas estadísticas para la columnas numéricas Sueldo y ID

UPAO
UNIVERSIDAD PRIVADA ANTENOR ORREGO

Método describe():

Utilizando **DataFrame['Nombre'].describe()** en serie de datos (columna) con datos de texto (String).

```
df['Departamento'].describe()
```

```
count                         1484
unique                           3
top       Research & Development
freq                           974
Name: Departamento, dtype: object
```

Muestra cantidad total, valores únicos, el valor que más veces aparece en la serie y la cantidad de veces

UPAO
UNIVERSIDAD PRIVADA ANTENOR ORREGO

Atributo shape:

DataFrame.shape: Es un atributo que devuelve el número de Filas y el número de Columnas.

```
DF.shape
```

```
(6, 5)
```

UPAO
UNIVERSIDAD PRIVADA ANTENOR ORREGO

.columns:

DataFrame.columns: Devuelve el nombre de las columnas.

```
DF.columns
```

```
Index(['Estudiante', 'Edad', 'Promedio', 'Ciclo', 'Curso'], dtype='object')
```

UPAO
UNIVERSIDAD PRIVADA ANTENOR ORREGO

.index:

DataFrame.index: Devuelve el índice de las filas ya sea números o nombres de las filas.

```
DF.index
```

	Estudiante	Edad	Promedio	Ciclo	Curso
0	Juan	19	15	V	Ingeniería de Datos
1	Carlos	20	12	VI	Base de Datos
2	Rosa	18	8	V	Algoritmos
3	Jean	18	18	IV	Tesis
4	Lourdes	20	10	IX	Proyecto de Tesis
5	Mario	18	17	V	Programación

```
RangeIndex(start=0, stop=6, step=1)
```

```
DFi.index
```

	Nombre	ID	Departamento	Sueldo
Ventas	Rosa	1	Ventas	1500
Logística	Juan	2	Logística	2000
Contabilidad	María	3	Contabilidad	2500
Sistemas	Pedro	4	Sistemas	3000

```
Index(['Ventas', 'Logística', 'Contabilidad', 'Sistemas'], dtype='object')
```

UPAO
UNIVERSIDAD PRIVADA ANTENOR ORREGO

.values:

DataFrame.values: Devuelve una matriz bidimensional (Numpy 2D) de los valores.

```
1 DF.values
```

```
array([['Juan', '19', 15, 'V', 'Ingeniería de Datos'],
       ['Carlos', '20', 12, 'VI', 'Base de Datos'],
       ['Rosa', '18', 8, 'V', 'Algoritmos'],
       ['Jean', '18', 18, 'IV', 'Tesis'],
       ['Lourdes', '20', 10, 'IX', 'Proyecto de Tesis'],
       ['Mario', '18', 17, 'V', 'Programación']], dtype=object)
```

```
DF.to_numpy
```

```
<bound method DataFrame.to_numpy of   Estudiante Edad  Promedio Ciclo
Curso
0       Juan   19        15     V  Ingeniería de Datos
1     Carlos   20        12    VI        Base de Datos
2       Rosa   18         8     V           Algoritmos
3       Jean   18        18    IV                Tesis
4    Lourdes   20        10    IX    Proyecto de Tesis
5      Mario   18        17     V         Programación>
```

UPAO
UNIVERSIDAD PRIVADA ANTENOR ORREGO

CASO PRÁCTICO: Eliminar Filas y Columnas

UPAO
UNIVERSIDAD PRIVADA ANTENOR ORREGO

Método drop():

DataFrame.drop(): Elimina filas o columnas usando una etiqueta de índice o un nombre de columna.

Eliminar Filas por etiqueta de filas

```
dfi.drop(['Juan','Pedro', inplace = True])
dfi
```

Nombre	ID	Departamento	Sueldo
Rosa	1	Ventas	1500
Juan	2	Logística	2000
María	3	Contabilidad	2500
Pedro	4	Sistemas	3000

Nombre	ID	Departamento	Sueldo
Rosa	1	Ventas	1500
María	3	Contabilidad	2500

Método drop():

Eliminar Columnas por el Nombre de la Columna

```
DF.drop(['ID','Departamento'], axis = 1, inplace = True)
```

	Nombre	ID	Departamento	Sueldo
0	Rosa	1	Ventas	1500
1	Juan	2	Logística	2000
2	María	3	Contabilidad	2500
3	Pedro	4	Sistemas	3000

	Nombre	Sueldo
0	Rosa	1500
1	Juan	2000
2	María	2500
3	Pedro	3000

UPAO
UNIVERSIDAD PRIVADA ANTENOR ORREGO

CASO PRÁCTICO: Iteracion

UPAO
UNIVERSIDAD PRIVADA ANTENOR ORREGO

Método iterrows():

DataFrame.iterrows(): Iteración es **tomar** cada elemento del DataFrame, uno tras otro.

Pandas DataFrame consta de filas y columnas, por lo que, para iterar sobre el marco de datos, tenemos que iterar un marco de datos como un diccionario. En un diccionario, iteramos sobre las claves del objeto de la misma manera que iteramos en el marco de datos.

```
for i,j in DF.iterrows():
    print (i,j)
    print()
```

```
0 Nombre            Rosa
ID                     1
Departamento     Ventas
Sueldo             1500
Name: 0, dtype: object

1 Nombre              Juan
ID                       2
Departamento     Logística
Sueldo                2000
Name: 1, dtype: object

2 Nombre                 María
ID                           3
Departamento     Contabilidad
Sueldo                   2500
Name: 2, dtype: object

3 Nombre             Pedro
ID                       4
Departamento     Sistemas
Sueldo               3000
Name: 3, dtype: object
```

UPAO
UNIVERSIDAD PRIVADA ANTENOR ORREGO

UPAO
UNIVERSIDAD PRIVADA ANTENOR ORREGO

¡Gracias!

UPAO

UNIVERSIDAD PRIVADA ANTENOR ORREGO