

**UNIVERSIDAD NACIONAL DANIEL ALCIDES CARRIÓN**  
**FACULTAD DE INGENIERÍA**  
**ESCUELA DE FORMACIÓN PROFESIONAL DE INGENIERÍA**  
**DE SISTEMAS Y COMPUTACIÓN**



**EJERCICIOS EN PHYTON**

**PRESENTADO POR:**

CARHUARICRA MENDOZA, Nelson Daniel

**DOCENTE:**

Ing. MUÑOZ ROBLES, Williams Antonio

**Cerro de Pasco - 2025**

## DATAFRAME

```
import pandas as pd
```

```
import numpy as np
```

```
data = {  
    'ID': range(1, 11),  
    'Nombre': ['Ana', 'Luis', 'Carlos', 'Elena', 'Jorge', 'Lucía', 'Pedro', 'Sofía', 'Miguel', 'María'],  
    'Edad': [23, 35, 29, 40, 31, 27, 22, 36, 41, 25],  
    'Carrera': ['Sistemas', 'Contabilidad', 'Sistemas', 'Derecho', 'Sistemas', 'Contabilidad',  
    'Psicología', 'Derecho', 'Psicología', 'Sistemas'],  
    'Promedio': [14.5, 15.2, 13.7, 16.3, 14.9, 15.5, 12.8, 17.0, 13.0, 15.0],  
    'Ingreso': pd.to_datetime(['2021-01-10', '2020-03-15', '2022-07-22', '2019-05-05', '2021-08-  
13', '2020-09-01', '2022-01-20', '2019-11-30', '2022-12-05', '2021-06-18'])  
}  
  
df = pd.DataFrame(data)
```

```
: import pandas as pd  
import numpy as np  
  
# Simular un DataFrame base para Los ejercicios  
data = {  
    'ID': range(1, 11),  
    'Nombre': ['Ana', 'Luis', 'Carlos', 'Elena', 'Jorge', 'Lucía', 'Pedro', 'Sofía', 'Miguel', 'María'],  
    'Edad': [23, 35, 29, 40, 31, 27, 22, 36, 41, 25],  
    'Carrera': ['Sistemas', 'Contabilidad', 'Sistemas', 'Derecho', 'Sistemas', 'Contabilidad', 'Psicología', 'Derecho', 'Psicología', 'Sistemas'],  
    'Promedio': [14.5, 15.2, 13.7, 16.3, 14.9, 15.5, 12.8, 17.0, 13.0, 15.0],  
    'Ingreso': pd.to_datetime(['2021-01-10', '2020-03-15', '2022-07-22', '2019-05-05', '2021-08-13', '2020-09-01', '2022-01-20', '2019-11-30', '2022-12-05', '2021-06-18'])  
}  
df = pd.DataFrame(data)
```

## Ejercicios

### 1. Filtrar estudiantes con promedio mayor a 15

```
df[df['Promedio'] > 15]
```

```
df[df['Promedio'] > 15]
```

	ID	Nombre	Edad	Carrera	Promedio	Ingreso
1	2	Luis	35	Contabilidad	15.2	2020-03-15
3	4	Elena	40	Derecho	16.3	2019-05-05
5	6	Lucía	27	Contabilidad	15.5	2020-09-01
7	8	Sofía	36	Derecho	17.0	2019-11-30

### 2. Agrupar por carrera y obtener promedio general

```
df.groupby('Carrera')['Promedio'].mean()
```

```
df.groupby('Carrera')['Promedio'].mean()
```

```
Carrera
Contabilidad    15.350
Derecho          16.650
Psicología      12.900
Sistemas        14.525
Name: Promedio, dtype: float64
```

### 3. Agregar una columna "Estado" (Aprobado $\geq 14$ )

```
df['Estado'] = np.where(df['Promedio'] >= 14, 'Aprobado', 'Desaprobado')
```

```
df['Estado'] = np.where(df['Promedio'] >= 14, 'Aprobado', 'Desaprobado')
```

df

	ID	Nombre	Edad	Carrera	Promedio	Ingreso	Estado
0	1	Ana	23	Sistemas	14.5	2021-01-10	Aprobado
1	2	Luis	35	Contabilidad	15.2	2020-03-15	Aprobado
2	3	Carlos	29	Sistemas	13.7	2022-07-22	Desaprobado
3	4	Elena	40	Derecho	16.3	2019-05-05	Aprobado
4	5	Jorge	31	Sistemas	14.9	2021-08-13	Aprobado
5	6	Lucía	27	Contabilidad	15.5	2020-09-01	Aprobado
6	7	Pedro	22	Psicología	12.8	2022-01-20	Desaprobado
7	8	Sofía	36	Derecho	17.0	2019-11-30	Aprobado
8	9	Miguel	41	Psicología	13.0	2022-12-05	Desaprobado
9	10	María	25	Sistemas	15.0	2021-06-18	Aprobado

#### 4. Obtener la edad máxima por carrera

```
df.groupby('Carrera')['Edad'].max()
```

```
df.groupby('Carrera')['Edad'].max()
```

```
Carrera
Contabilidad    35
Derecho          40
Psicología      41
Sistemas        31
Name: Edad, dtype: int64
```

#### 5. Mostrar los tres estudiantes con mayor promedio

```
df.nlargest(3, 'Promedio')
```

```
df.nlargest(3, 'Promedio')
```

	ID	Nombre	Edad	Carrera	Promedio	Ingreso	Estado
7	8	Sofía	36	Derecho	17.0	2019-11-30	Aprobado
3	4	Elena	40	Derecho	16.3	2019-05-05	Aprobado
5	6	Lucía	27	Contabilidad	15.5	2020-09-01	Aprobado

#### 6. Contar cuántos estudiantes hay por carrera

```
df['Carrera'].value_counts()
```

```
df['Carrera'].value_counts()
```

```
Carrera
Sistemas      4
Contabilidad   2
Derecho        2
Psicología     2
Name: count, dtype: int64
```

#### 7. Obtener estudiantes que ingresaron después del 2021

```
df[df['Ingreso'].dt.year > 2021]
```

```
df[df['Ingreso'].dt.year > 2021]
```

	ID	Nombre	Edad	Carrera	Promedio	Ingreso	Estado
2	3	Carlos	29	Sistemas	13.7	2022-07-22	Desaprobado
6	7	Pedro	22	Psicología	12.8	2022-01-20	Desaprobado
8	9	Miguel	41	Psicología	13.0	2022-12-05	Desaprobado

#### 8. Ordenar por promedio descendente y edad ascendente

```
df.sort_values(by=['Promedio', 'Edad'], ascending=[False, True])
```

```
df.sort_values(by=['Promedio', 'Edad'], ascending=[False, True])
```

	ID	Nombre	Edad	Carrera	Promedio	Ingreso	Estado
7	8	Soffá	36	Derecho	17.0	2019-11-30	Aprobado
3	4	Elena	40	Derecho	16.3	2019-05-05	Aprobado
5	6	Lucía	27	Contabilidad	15.5	2020-09-01	Aprobado
1	2	Luis	35	Contabilidad	15.2	2020-03-15	Aprobado
9	10	María	25	Sistemas	15.0	2021-06-18	Aprobado
4	5	Jorge	31	Sistemas	14.9	2021-08-13	Aprobado
0	1	Ana	23	Sistemas	14.5	2021-01-10	Aprobado
2	3	Carlos	29	Sistemas	13.7	2022-07-22	Desaprobado
8	9	Miguel	41	Psicología	13.0	2022-12-05	Desaprobado
6	7	Pedro	22	Psicología	12.8	2022-01-20	Desaprobado

#### 9. Agrupar por carrera y contar cuántos aprobaron

```
df[df['Estado'] == 'Aprobado'].groupby('Carrera').size()
```

```
df[df['Estado'] == 'Aprobado'].groupby('Carrera').size()
```

```
Carrera
Contabilidad    2
Derecho         2
Sistemas        3
dtype: int64
```

#### 10. Agregar columna de año de ingreso

```
df['AñoIngreso'] = df['Ingreso'].dt.year
```

```
df['AñoIngreso'] = df['Ingreso'].dt.year  
df
```

	ID	Nombre	Edad	Carrera	Promedio	Ingreso	AñoIngreso
0	1	Ana	23	Sistemas	14.5	2021-01-10	2021
1	2	Luis	35	Contabilidad	15.2	2020-03-15	2020
2	3	Carlos	29	Sistemas	13.7	2022-07-22	2022
3	4	Elena	40	Derecho	16.3	2019-05-05	2019
4	5	Jorge	31	Sistemas	14.9	2021-08-13	2021
5	6	Lucía	27	Contabilidad	15.5	2020-09-01	2020
6	7	Pedro	22	Psicología	12.8	2022-01-20	2022
7	8	Sofía	36	Derecho	17.0	2019-11-30	2019
8	9	Miguel	41	Psicología	13.0	2022-12-05	2022
9	10	María	25	Sistemas	15.0	2021-06-18	2021

#### 11. Reemplazar 'Sistemas' por 'Ing. Sistemas'

```
df['Carrera'] = df['Carrera'].replace('Sistemas', 'Ing. Sistemas')
```

```
df['Carrera'] = df['Carrera'].replace('Sistemas', 'Ing. Sistemas')
df
```

	ID	Nombre	Edad	Carrera	Promedio	Ingreso	AñoIngreso
0	1	Ana	23	Ing. Sistemas	14.5	2021-01-10	2021
1	2	Luis	35	Contabilidad	15.2	2020-03-15	2020
2	3	Carlos	29	Ing. Sistemas	13.7	2022-07-22	2022
3	4	Elena	40	Derecho	16.3	2019-05-05	2019
4	5	Jorge	31	Ing. Sistemas	14.9	2021-08-13	2021
5	6	Lucía	27	Contabilidad	15.5	2020-09-01	2020
6	7	Pedro	22	Psicología	12.8	2022-01-20	2022
7	8	Sofía	36	Derecho	17.0	2019-11-30	2019
8	9	Miguel	41	Psicología	13.0	2022-12-05	2022
9	10	María	25	Ing. Sistemas	15.0	2021-06-18	2021

## 12. Crear columna “Edad x Promedio”

```
df['EdadPromedio'] = df['Edad'] * df['Promedio']
```

```
df['EdadPromedio'] = df['Edad'] * df['Promedio']
```

```
df.head(2)
```

	ID	Nombre	Edad	Carrera	Promedio	Ingreso	Estado	AñoIngreso	EdadPromedio
0	1	Ana	23	Ing. Sistemas	14.5	2021-01-10	Aprobado	2021	333.5
1	2	Luis	35	Contabilidad	15.2	2020-03-15	Aprobado	2020	532.0

## 13. Crear un DataFrame solo con Aprobados de Derecho

```
df[(df['Carrera'] == 'Derecho') & (df['Estado'] == 'Aprobado')]
```



```
df[(df['Carrera'] == 'Derecho') & (df['Estado'] == 'Aprobado')]
```

	ID	Nombre	Edad	Carrera	Promedio	Ingreso	Estado	AñoIngreso	EdadPromedio
3	4	Elena	40	Derecho	16.3	2019-05-05	Aprobado	2019	652.0
7	8	Sofía	36	Derecho	17.0	2019-11-30	Aprobado	2019	612.0

#### 14. Mostrar promedio general por año de ingreso

```
df.groupby(df['Ingreso'].dt.year)['Promedio'].mean()
```

```
df.groupby(df['Ingreso'].dt.year)['Promedio'].mean()
```

```
Ingreso
2019    16.650000
2020    15.350000
2021    14.800000
2022    13.166667
Name: Promedio, dtype: float64
```

#### 15. Exportar a Excel los estudiantes de Psicología

```
df[df['Carrera'] == 'Psicología'].to_excel('psicologia_aprobados.xlsx', index=False)
```

```
df[df['Carrera'] == 'Psicología'].to_excel('psicologia_aprobados.xlsx', index=False)
```

#### 16. Detectar duplicados en la columna Nombre

```
df[df['Nombre'].duplicated(keep=False)]
```

```
df[df['Nombre'].duplicated(keep=False)]
```

ID	Nombre	Edad	Carrera	Promedio	Ingreso	Estado	AñoIngreso	EdadPromedio
----	--------	------	---------	----------	---------	--------	------------	--------------

#### 17. Eliminar la columna EdadPromedio

```
df.drop(columns='EdadPromedio', inplace=True)
```

```
df.drop(columns='EdadPromedio', inplace=True)
```

```
df.head(2)
```

	ID	Nombre	Edad	Carrera	Promedio	Ingreso	Estado	AñoIngreso
0	1	Ana	23	Ing. Sistemas	14.5	2021-01-10	Aprobado	2021
1	2	Luis	35	Contabilidad	15.2	2020-03-15	Aprobado	2020

#### 18. Filtrar alumnos con nombres que empiezan con "M"

```
df[df['Nombre'].str.startswith('M')]
```

```
df[df['Nombre'].str.startswith('M')]
```

	ID	Nombre	Edad	Carrera	Promedio	Ingreso	Estado	AñoIngreso
8	9	Miguel	41	Psicología	13.0	2022-12-05	Desaprobado	2022
9	10	María	25	Ing. Sistemas	15.0	2021-06-18	Aprobado	2021

#### 19. Promedio de edad por carrera para desaprobados

```
df[df['Estado'] == 'Desaprobado'].groupby('Carrera')['Edad'].mean()
```

```
df[df['Estado'] == 'Desaprobado'].groupby('Carrera')['Edad'].mean()
```

```
Carrera
Ing. Sistemas    29.0
Psicología       31.5
Name: Edad, dtype: float64
```

#### 20. Crear resumen estadístico por carrera

```
df.groupby('Carrera')[['Edad', 'Promedio']].agg(['mean', 'min', 'max'])
```

```
df.groupby('Carrera')[['Edad', 'Promedio']].agg(['mean', 'min', 'max'])
```

	Edad			Promedio		
	mean	min	max	mean	min	max
Carrera						
Contabilidad	31.0	27	35	15.350	15.2	15.5
Derecho	38.0	36	40	16.650	16.3	17.0
Ing. Sistemas	27.0	23	31	14.525	13.7	15.0
Psicología	31.5	22	41	12.900	12.8	13.0