ANALISIS DE LA EDUCACION EN EGIPTO

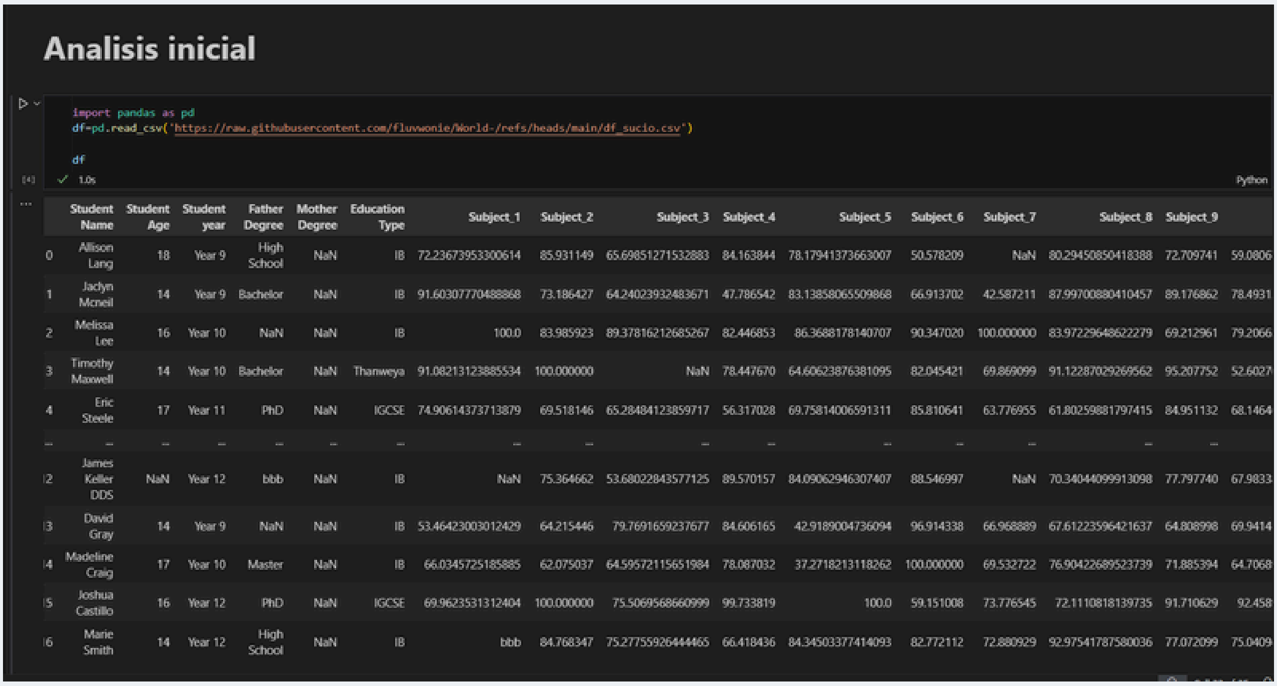
Introducción a las Ciencias de Datos

Fátima Sánchez Domínguez

Profesor: Jaime Alejandro Romero Sierra

Análisis inicial

-Carga de la base de datos

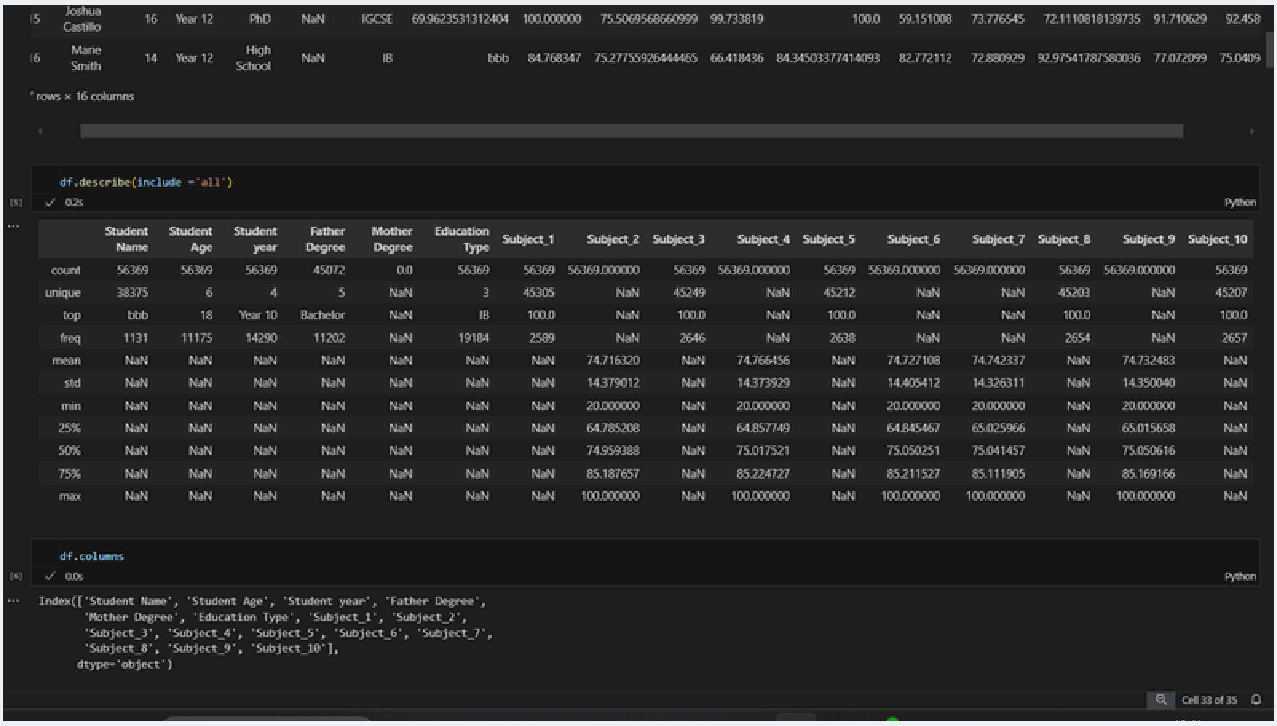


Analisis inicial

```python
import pandas as pd
df=pd.read_csv('https://raw.githubusercontent.com/fluvwonie/World-/refs/heads/main/df_sucio.csv')

df
```

	Student Name	Student Age	Student year	Father Degree	Mother Degree	Education Type	Subject_1	Subject_2	Subject_3	Subject_4	Subject_5	Subject_6	Subject_7	Subject_8	Subject_9	
0	Allison Lang	18	Year 9	High School	NaN	IB	72.23673953300614	85.931149	65.69851271532883	84.163844	78.17941373663007	50.578209	NaN	80.29450850418388	72.709741	59.0806
1	Jaclyn Mcneil	14	Year 9	Bachelor	NaN	IB	91.60307770488868	73.186427	64.24023932483671	47.786542	83.13858065509868	66.913702	42.587211	87.99700880410457	89.176862	78.4931
2	Melissa Lee	16	Year 10	NaN	NaN	IB	100.0	83.985923	89.37816212685267	82.446853	86.3688178140707	90.347020	100.000000	83.97229648622279	69.212961	79.2066
3	Timothy Maxwell	14	Year 10	Bachelor	NaN	Thanweya	91.08213123885534	100.000000	NaN	78.447670	64.60623876381095	82.045421	69.869099	91.12287029269562	95.207752	52.6027
4	Eric Steele	17	Year 11	PhD	NaN	IGCSE	74.90614373713879	69.518146	65.28484123859717	56.317028	69.75814006591311	85.810641	63.776955	61.80259881797415	84.951132	68.1464
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
2	James Keller DDS	NaN	Year 12	bbb	NaN	IB	NaN	75.364662	53.68022843577125	89.570157	84.09062946307407	88.546997	NaN	70.34044099913098	77.797740	67.9833
3	David Gray	14	Year 9	NaN	NaN	IB	53.46423003012429	64.215446	79.7691659237677	84.606165	42.9189004736094	96.914338	66.968889	67.61223596421637	64.808998	69.9414
4	Madeline Craig	17	Year 10	Master	NaN	IB	66.0345725185885	62.075037	64.59572115651984	78.087032	37.2718213118262	100.000000	69.532722	76.90422689523739	71.885394	64.7068
5	Joshua Castillo	16	Year 12	PhD	NaN	IGCSE	69.9623531312404	100.000000	75.5069568660999	99.733819	100.0	59.151008	73.776545	72.1110818139735	91.710629	92.458
6	Marie Smith	14	Year 12	High School	NaN	IB	bbb	84.768347	75.27755926444465	66.418436	84.34503377414093	82.772112	72.880929	92.97541787580036	77.072099	75.0409

-Resumen estadistico antes de la limpieza



	Student Name	Student Age	Student year	Father Degree	Mother Degree	Education Type	Subject_1	Subject_2	Subject_3	Subject_4	Subject_5	Subject_6	Subject_7	Subject_8	Subject_9	
5	Joshua Castillo	16	Year 12	PhD	NaN	IGCSE	69.9623531312404	100.000000	75.5069568660999	99.733819	100.0	59.151008	73.776545	72.1110818139735	91.710629	92.458
6	Marie Smith	14	Year 12	High School	NaN	IB	bbb	84.768347	75.27755926444465	66.418436	84.34503377414093	82.772112	72.880929	92.97541787580036	77.072099	75.0409

rows × 16 columns

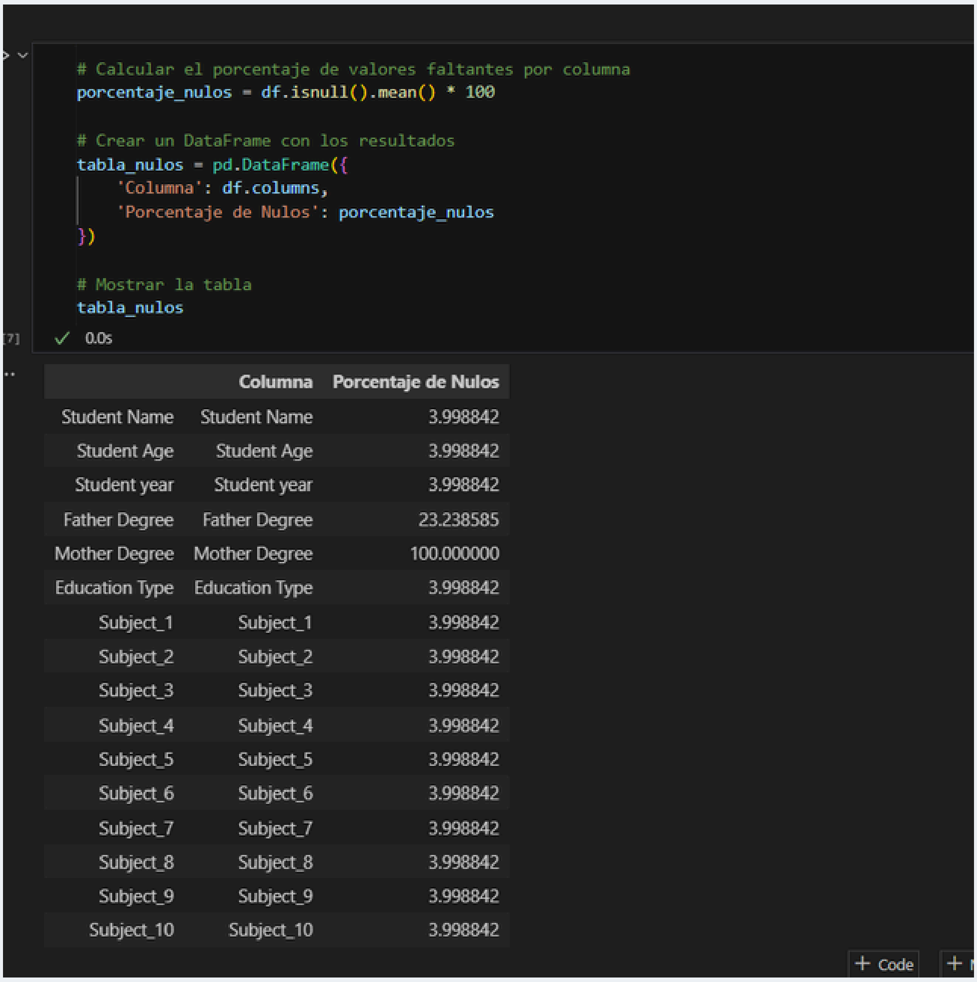
```python
df.describe(include ='all')
```

	Student Name	Student Age	Student year	Father Degree	Mother Degree	Education Type	Subject_1	Subject_2	Subject_3	Subject_4	Subject_5	Subject_6	Subject_7	Subject_8	Subject_9	Subject_10
count	56369	56369	56369	45072	0.0	56369	56369	56369.000000	56369	56369.000000	56369	56369.000000	56369.000000	56369	56369.000000	56369
unique	38375	6	4	5	NaN	3	45305	NaN	45249	NaN	45212	NaN	NaN	45203	NaN	45207
top	bbb	18	Year 10	Bachelor	NaN	IB	100.0	NaN	100.0	NaN	100.0	NaN	NaN	100.0	NaN	100.0
freq	1131	11175	14290	11202	NaN	19184	2589	NaN	2646	NaN	2638	NaN	NaN	2654	NaN	2657
mean	NaN	NaN	NaN	NaN	NaN	NaN	NaN	74.716320	NaN	74.766456	NaN	74.727108	74.742337	NaN	74.732483	NaN
std	NaN	NaN	NaN	NaN	NaN	NaN	NaN	14.379012	NaN	14.373929	NaN	14.405412	14.326311	NaN	14.350040	NaN
min	NaN	NaN	NaN	NaN	NaN	NaN	NaN	20.000000	NaN	20.000000	NaN	20.000000	20.000000	NaN	20.000000	NaN
25%	NaN	NaN	NaN	NaN	NaN	NaN	NaN	64.785208	NaN	64.857749	NaN	64.845467	65.025966	NaN	65.015658	NaN
50%	NaN	NaN	NaN	NaN	NaN	NaN	NaN	74.959388	NaN	75.017521	NaN	75.050251	75.041457	NaN	75.050616	NaN
75%	NaN	NaN	NaN	NaN	NaN	NaN	NaN	85.187657	NaN	85.224727	NaN	85.211527	85.111905	NaN	85.169166	NaN
max	NaN	NaN	NaN	NaN	NaN	NaN	NaN	100.000000	NaN	100.000000	NaN	100.000000	100.000000	NaN	100.000000	NaN

```python
df.columns
```

```
Index(['Student Name', 'Student Age', 'Student year', 'Father Degree',
       'Mother Degree', 'Education Type', 'Subject_1', 'Subject_2',
       'Subject_3', 'Subject_4', 'Subject_5', 'Subject_6', 'Subject_7',
       'Subject_8', 'Subject_9', 'Subject_10'],
      dtype='object')
```

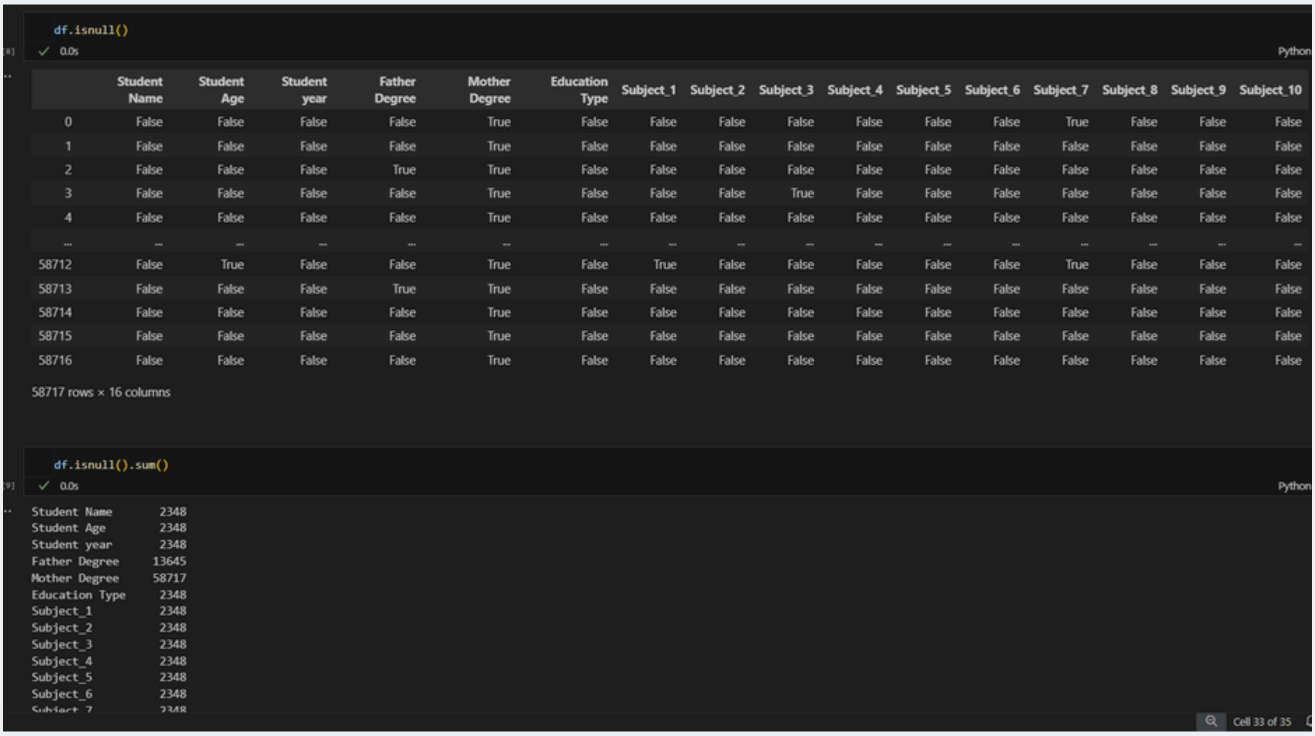
-Valores faltantes en las columnas



```python
# Calcular el porcentaje de valores faltantes por columna
porcentaje_nulos = df.isnull().mean() * 100

# Crear un DataFrame con los resultados
tabla_nulos = pd.DataFrame({
    'Columna': df.columns,
    'Porcentaje de Nulos': porcentaje_nulos
})

# Mostrar la tabla
tabla_nulos
```

✓ 0.0s

	Columna	Porcentaje de Nulos
Student Name	Student Name	3.998842
Student Age	Student Age	3.998842
Student year	Student year	3.998842
Father Degree	Father Degree	23.238585
Mother Degree	Mother Degree	100.000000
Education Type	Education Type	3.998842
Subject_1	Subject_1	3.998842
Subject_2	Subject_2	3.998842
Subject_3	Subject_3	3.998842
Subject_4	Subject_4	3.998842
Subject_5	Subject_5	3.998842
Subject_6	Subject_6	3.998842
Subject_7	Subject_7	3.998842
Subject_8	Subject_8	3.998842
Subject_9	Subject_9	3.998842
Subject_10	Subject_10	3.998842



```python
df.isnull()
```

✓ 0.0s

	Student Name	Student Age	Student year	Father Degree	Mother Degree	Education Type	Subject_1	Subject_2	Subject_3	Subject_4	Subject_5	Subject_6	Subject_7	Subject_8	Subject_9	Subject_10
0	False	False	False	False	True	False	False	False	False	False	False	False	True	False	False	False
1	False	False	False	False	True	False	False	False	False	False	False	False	False	False	False	False
2	False	False	False	True	True	False	False	False	False	False	False	False	False	False	False	False
3	False	False	False	False	True	False	False	False	True	False	False	False	False	False	False	False
4	False	False	False	False	True	False	False	False	False	False	False	False	False	False	False	False
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
58712	False	True	False	False	True	False	True	False	False	False	False	False	True	False	False	False
58713	False	False	False	True	True	False	False	False	False	False	False	False	False	False	False	False
58714	False	False	False	False	True	False	False	False	False	False	False	False	False	False	False	False
58715	False	False	False	False	True	False	False	False	False	False	False	False	False	False	False	False
58716	False	False	False	False	True	False	False	False	False	False	False	False	False	False	False	False

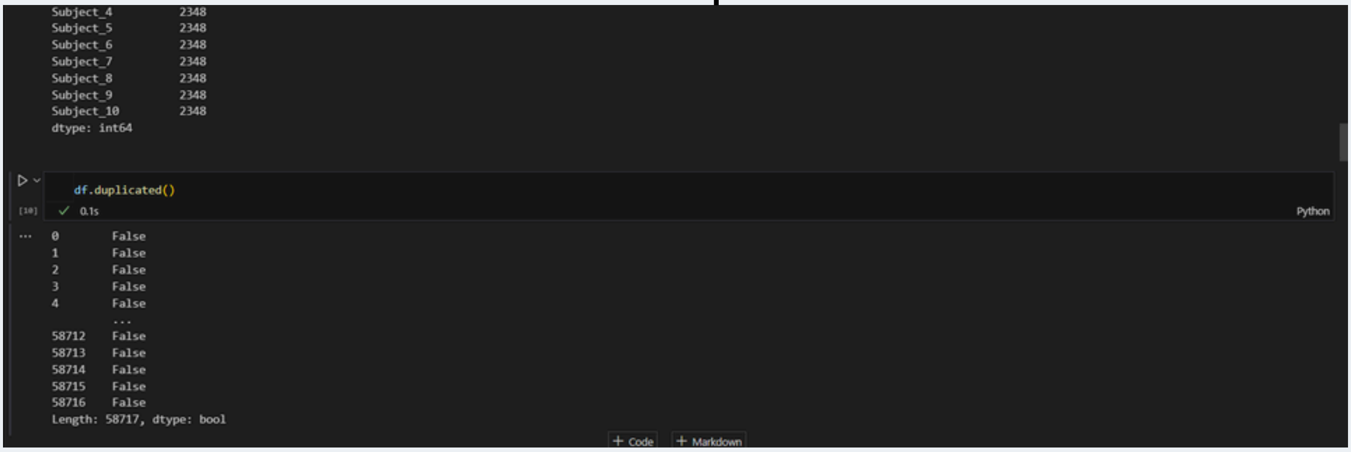
58717 rows × 16 columns

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

✓ 0.0s

```
Student Name       2348
Student Age        2348
Student year       2348
Father Degree     13645
Mother Degree     58717
Education Type     2348
Subject_1          2348
Subject_2          2348
Subject_3          2348
Subject_4          2348
Subject_5          2348
Subject_6          2348
Subject_7          2348
```

-Valores duplicados



```
Subject_4          2348
Subject_5          2348
Subject_6          2348
Subject_7          2348
Subject_8          2348
Subject_9          2348
Subject_10         2348
dtype: int64
```

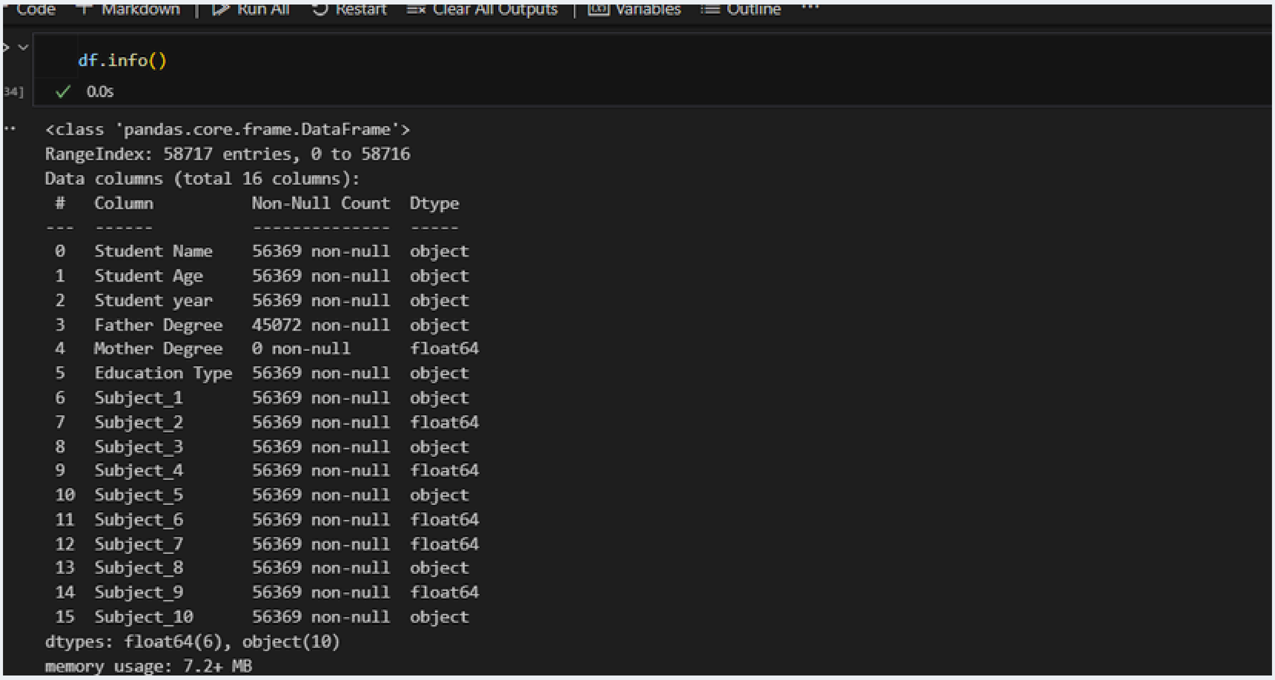
```
df.duplicated()
```

[10] ✓ 0.1s

```
0        False
1        False
2        False
3        False
4        False
         ...
58712    False
58713    False
58714    False
58715    False
58716    False
Length: 58717, dtype: bool
```

-Tipo de dato

Como observamos, en el existen varias incongruencias que seran arregladas mas adelante



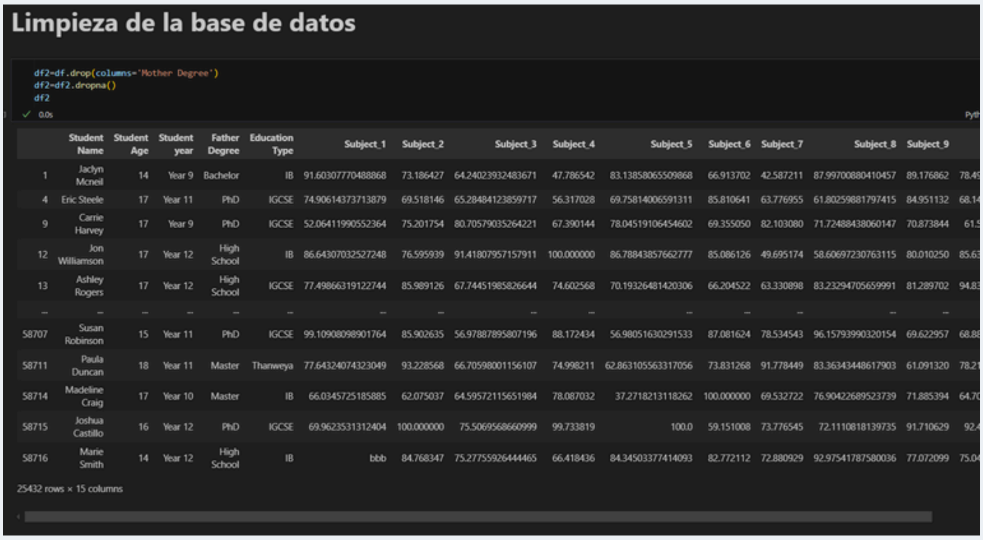
```
df.info()
```

34] ✓ 0.0s

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 58717 entries, 0 to 58716
Data columns (total 16 columns):
 #   Column          Non-Null Count  Dtype  
---  ------          --------------  -----  
 0   Student Name    56369 non-null  object 
 1   Student Age     56369 non-null  object 
 2   Student year    56369 non-null  object 
 3   Father Degree   45072 non-null  object 
 4   Mother Degree   0 non-null      float64
 5   Education Type  56369 non-null  object 
 6   Subject_1       56369 non-null  object 
 7   Subject_2       56369 non-null  float64
 8   Subject_3       56369 non-null  object 
 9   Subject_4       56369 non-null  float64
 10  Subject_5       56369 non-null  object 
 11  Subject_6       56369 non-null  float64
 12  Subject_7       56369 non-null  float64
 13  Subject_8       56369 non-null  object 
 14  Subject_9       56369 non-null  float64
 15  Subject_10      56369 non-null  object 
dtypes: float64(6), object(10)
memory usage: 7.2+ MB
```

Limpieza de la base

Primero eliminamos los valores Nan de la base. Se observa que los datos de ‘Mother Degree’ son todos Nan por lo que mejor eliminamos la columna.



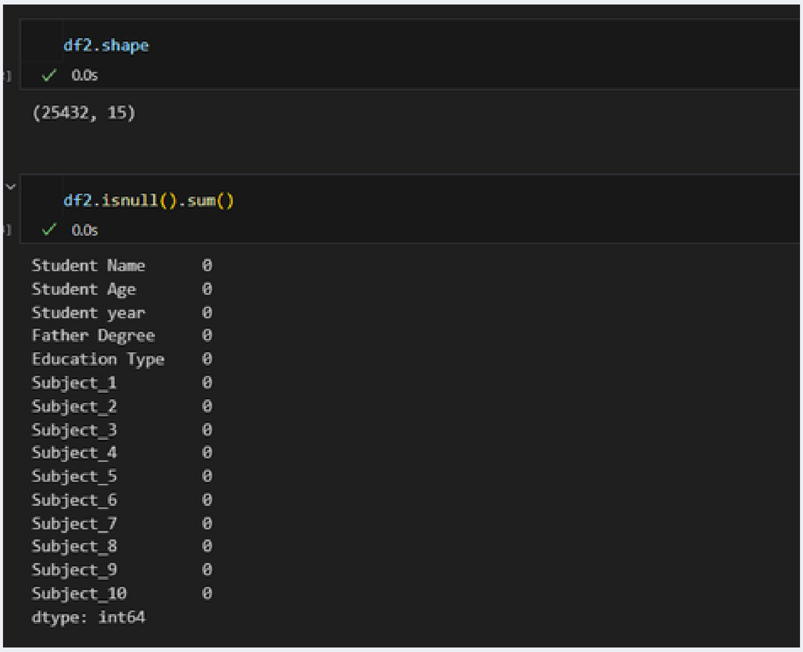
Limpieza de la base de datos

```
df2=df.drop(columns='Mother Degree')
df2=df2.dropna()
df2
```

✓ 0.0s

	Student Name	Student Age	Student year	Father Degree	Education Type	Subject_1	Subject_2	Subject_3	Subject_4	Subject_5	Subject_6	Subject_7	Subject_8	Subject_9	
1	Jaclyn Mcneil	14	Year 9	Bachelor	IB	91.60307770488868	73.186427	64.24023932483671	47.786542	83.13858065509868	66.913702	42.587211	87.99700880410457	89.176862	78.49
4	Eric Steele	17	Year 11	PhD	IGCSE	74.90614373713879	69.518146	65.28484123859717	56.317028	69.75814006591311	85.810641	63.776955	61.80259881797415	84.951132	68.14
9	Carrie Harvey	17	Year 9	PhD	IGCSE	52.06411990552364	75.201754	80.70579035264221	67.390144	78.04519106454602	69.355050	82.103080	71.72488438060147	70.873844	61.5
12	Jon Williamson	17	Year 12	High School	IB	86.64307032527248	76.595939	91.41807957157911	100.000000	86.78843857662777	85.086126	49.695174	58.60697230763115	80.010250	85.63
13	Ashley Rogers	17	Year 12	High School	IGCSE	77.49866319122744	85.989126	67.74451985826644	74.602568	70.19326481420306	66.204522	63.330898	83.23294705659991	81.289702	94.83
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
58707	Susan Robinson	15	Year 11	PhD	IGCSE	99.10908098901764	85.902635	56.97887895807196	88.172434	56.98051630291533	87.081624	78.534543	96.15793990320154	69.622957	68.88
58711	Paula Duncan	18	Year 11	Master	Thanweya	77.64324074323049	93.228568	66.70598001156107	74.998211	62.863105563317056	73.831268	91.778449	83.36343448617903	61.091320	78.21
58714	Madeline Craig	17	Year 10	Master	IB	66.0345725185885	62.075037	64.59572115651984	78.087032	37.2718213118262	100.000000	69.532722	76.90422689523739	71.885394	64.70
58715	Joshua Castillo	16	Year 12	PhD	IGCSE	69.9623531312404	100.000000	75.5069568660999	99.733819	100.0	59.151008	73.776545	72.1110818139735	91.710629	92.4
58716	Marie Smith	14	Year 12	High School	IB	bbb	84.768347	75.27755926444465	66.418436	84.34503377414093	82.772112	72.880929	92.97541787580036	77.072099	75.04

25432 rows × 15 columns



```
df2.shape
```

✓ 0.0s

```
(25432, 15)
```

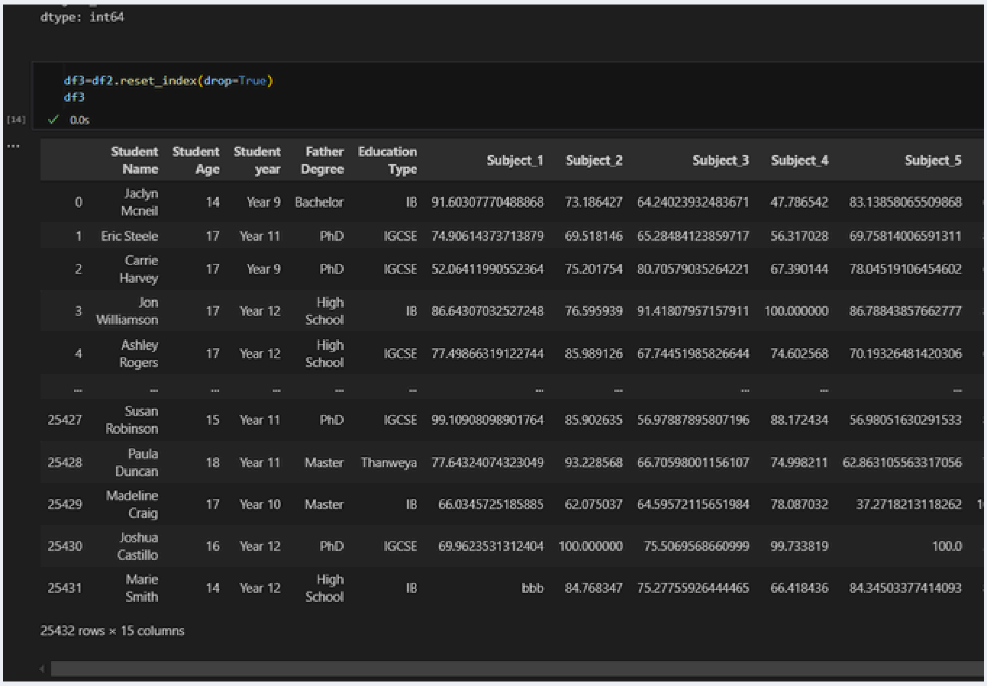
```
df2.isnull().sum()
```

✓ 0.0s

```
Student Name      0
Student Age       0
Student year      0
Father Degree     0
Education Type    0
Subject_1         0
Subject_2         0
Subject_3         0
Subject_4         0
Subject_5         0
Subject_6         0
Subject_7         0
Subject_8         0
Subject_9         0
Subject_10        0
dtype: int64
```

Comprobamos que los nulos se hayan eliminado correctamente

Reseteamos el índice



```
dtype: int64
```

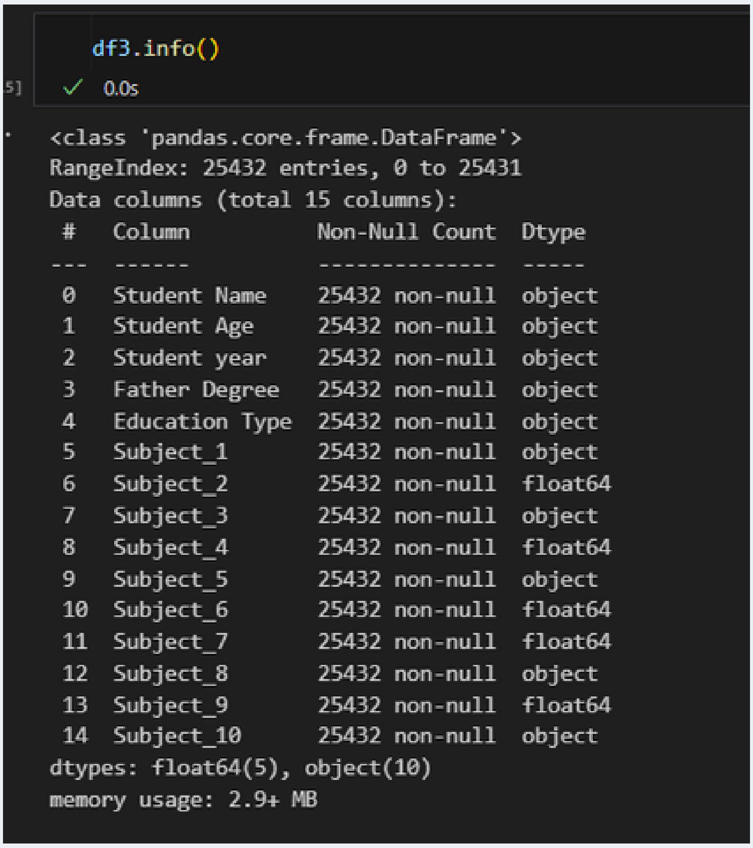
```
df3=df2.reset_index(drop=True)
df3
```

[14] ✓ 0.0s

	Student Name	Student Age	Student year	Father Degree	Education Type	Subject_1	Subject_2	Subject_3	Subject_4	Subject_5
0	Jaclyn Mcneil	14	Year 9	Bachelor	IB	91.60307770488868	73.186427	64.24023932483671	47.786542	83.13858065509868
1	Eric Steele	17	Year 11	PhD	IGCSE	74.90614373713879	69.518146	65.28484123859717	56.317028	69.75814006591311
2	Carrie Harvey	17	Year 9	PhD	IGCSE	52.06411990552364	75.201754	80.70579035264221	67.390144	78.04519106454602
3	Jon Williamson	17	Year 12	High School	IB	86.64307032527248	76.595939	91.41807957157911	100.000000	86.78843857662777
4	Ashley Rogers	17	Year 12	High School	IGCSE	77.49866319122744	85.989126	67.74451985826644	74.602568	70.19326481420306
...	...	...	...	...	...	...	...	...	...	...
25427	Susan Robinson	15	Year 11	PhD	IGCSE	99.10908098901764	85.902635	56.97887895807196	88.172434	56.98051630291533
25428	Paula Duncan	18	Year 11	Master	Thanweya	77.64324074323049	93.228568	66.70598001156107	74.998211	62.863105563317056
25429	Madeline Craig	17	Year 10	Master	IB	66.0345725185885	62.075037	64.59572115651984	78.087032	37.2718213118262
25430	Joshua Castillo	16	Year 12	PhD	IGCSE	69.9623531312404	100.000000	75.5069568660999	99.733819	100.0
25431	Marie Smith	14	Year 12	High School	IB	bbb	84.768347	75.27755926444465	66.418436	84.34503377414093

25432 rows × 15 columns

Volvemos a comprobar el tipo de dato, y observamos que algunos subject son de tipo objeto, cuando en realidad son datos numericos de tipo float.

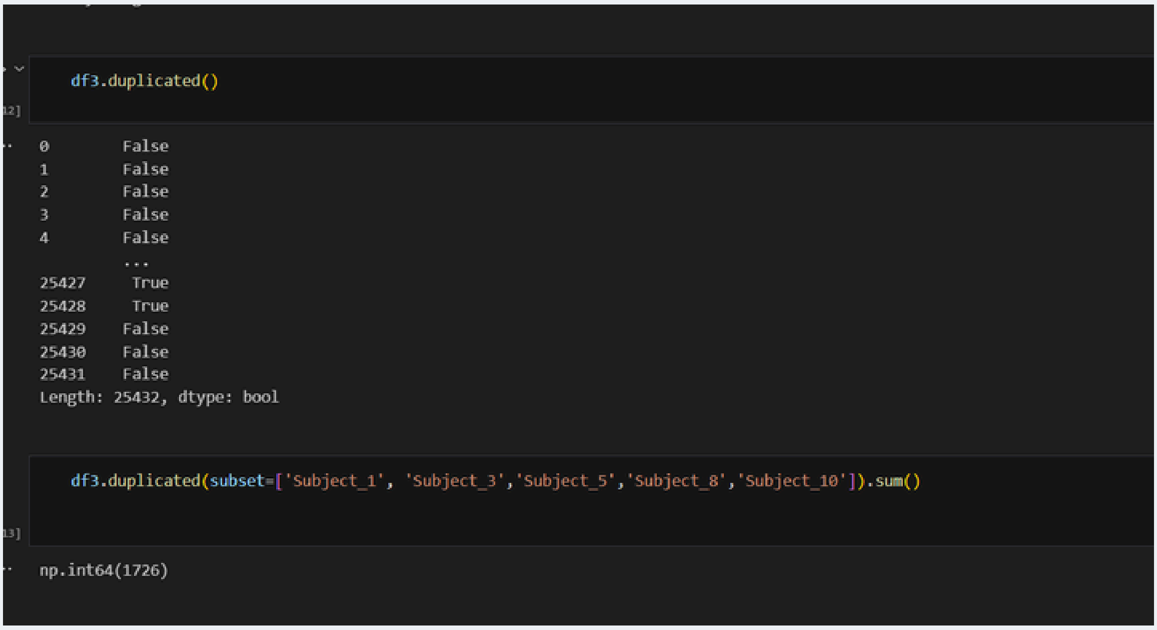


```
df3.info()
```

✓ 0.0s

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 25432 entries, 0 to 25431
Data columns (total 15 columns):
 #   Column          Non-Null Count  Dtype  
---  ------          --------------  -----  
 0   Student Name    25432 non-null  object 
 1   Student Age     25432 non-null  object 
 2   Student year    25432 non-null  object 
 3   Father Degree   25432 non-null  object 
 4   Education Type  25432 non-null  object 
 5   Subject_1       25432 non-null  object 
 6   Subject_2       25432 non-null  float64
 7   Subject_3       25432 non-null  object 
 8   Subject_4       25432 non-null  float64
 9   Subject_5       25432 non-null  object 
 10  Subject_6       25432 non-null  float64
 11  Subject_7       25432 non-null  float64
 12  Subject_8       25432 non-null  object 
 13  Subject_9       25432 non-null  float64
 14  Subject_10      25432 non-null  object 
dtypes: float64(5), object(10)
memory usage: 2.9+ MB
```

Verificamos si hay datos duplicados



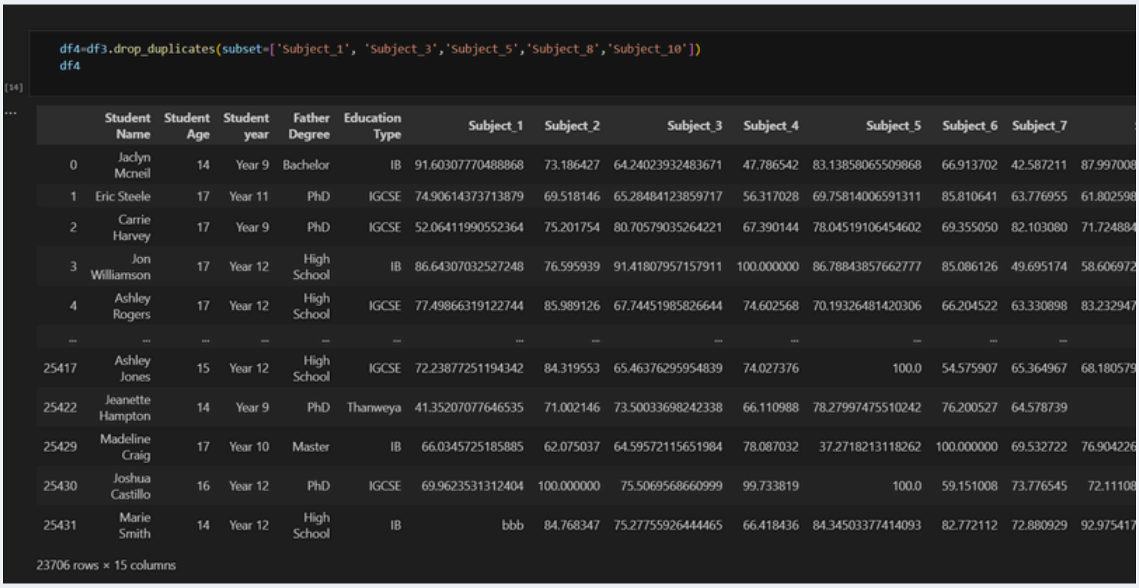
```
df3.duplicated()
```

```
0        False
1        False
2        False
3        False
4        False
         ...  
25427     True
25428     True
25429    False
25430    False
25431    False
Length: 25432, dtype: bool
```

```
df3.duplicated(subset=['Subject_1', 'Subject_3','Subject_5','Subject_8','Subject_10']).sum()
```

```
np.int64(1726)
```

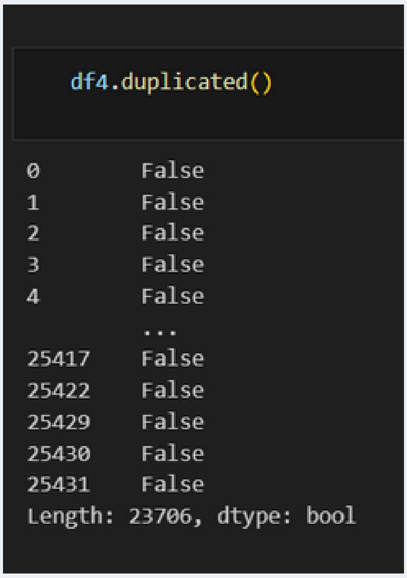
Eliminamos los datos duplicados



```
df4=df3.drop_duplicates(subset=['Subject_1', 'Subject_3','Subject_5','Subject_8','Subject_10'])
df4
```

	Student Name	Student Age	Student year	Father Degree	Education Type	Subject_1	Subject_2	Subject_3	Subject_4	Subject_5	Subject_6	Subject_7	
0	Jaclyn Mcneil	14	Year 9	Bachelor	IB	91.60307770488868	73.186427	64.24023932483671	47.786542	83.13858065509868	66.913702	42.587211	87.997008
1	Eric Steele	17	Year 11	PhD	IGCSE	74.90614373713879	69.518146	65.28484123859717	56.317028	69.75814006591311	85.810641	63.776955	61.802598
2	Carrie Harvey	17	Year 9	PhD	IGCSE	52.06411990552364	75.201754	80.70579035264221	67.390144	78.04519106454602	69.355050	82.103080	71.724884
3	Jon Williamson	17	Year 12	High School	IB	86.64307032527248	76.595939	91.41807957157911	100.000000	86.78843857662777	85.086126	49.695174	58.606972
4	Ashley Rogers	17	Year 12	High School	IGCSE	77.49866319122744	85.989126	67.74451985826644	74.602568	70.19326481420306	66.204522	63.330898	83.232947
...	...	...	...	...	...	...	...	...	...	...	...	...	
25417	Ashley Jones	15	Year 12	High School	IGCSE	72.23877251194342	84.319553	65.46376295954839	74.027376	100.0	54.575907	65.364967	68.180579
25422	Jeanette Hampton	14	Year 9	PhD	Thanweya	41.35207077646535	71.002146	73.50033698242338	66.110988	78.27997475510242	76.200527	64.578739	
25429	Madeline Craig	17	Year 10	Master	IB	66.0345725185885	62.075037	64.59572115651984	78.087032	37.2718213118262	100.000000	69.532722	76.904226
25430	Joshua Castillo	16	Year 12	PhD	IGCSE	69.9623531312404	100.000000	75.5069568660999	99.733819	100.0	59.151008	73.776545	72.11108
25431	Marie Smith	14	Year 12	High School	IB	bbb	84.768347	75.27755926444465	66.418436	84.34503377414093	82.772112	72.880929	92.975417

23706 rows × 15 columns

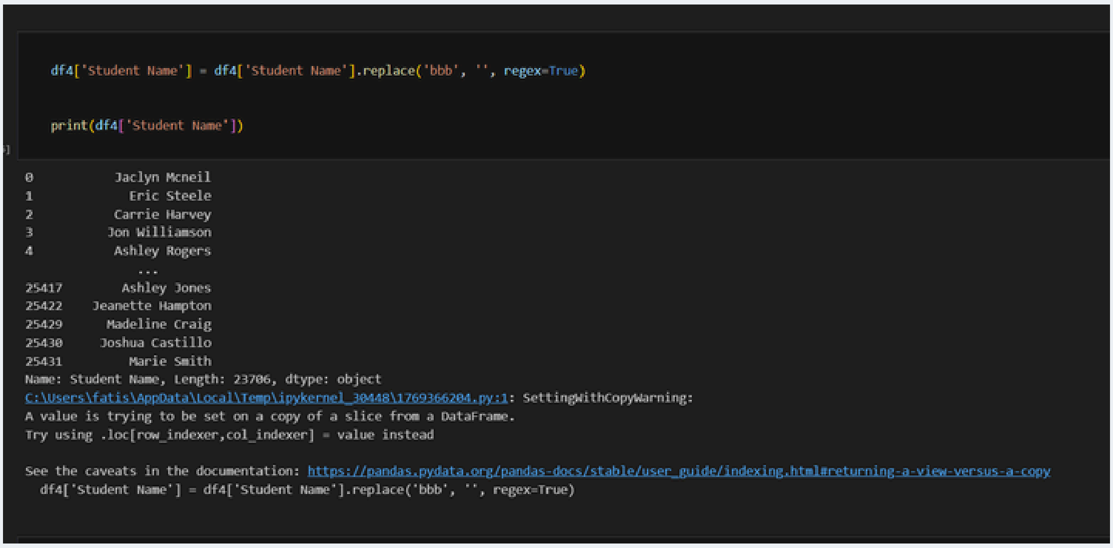


```
df4.duplicated()
```

```
0        False
1        False
2        False
3        False
4        False
         ...  
25417    False
25422    False
25429    False
25430    False
25431    False
Length: 23706, dtype: bool
```

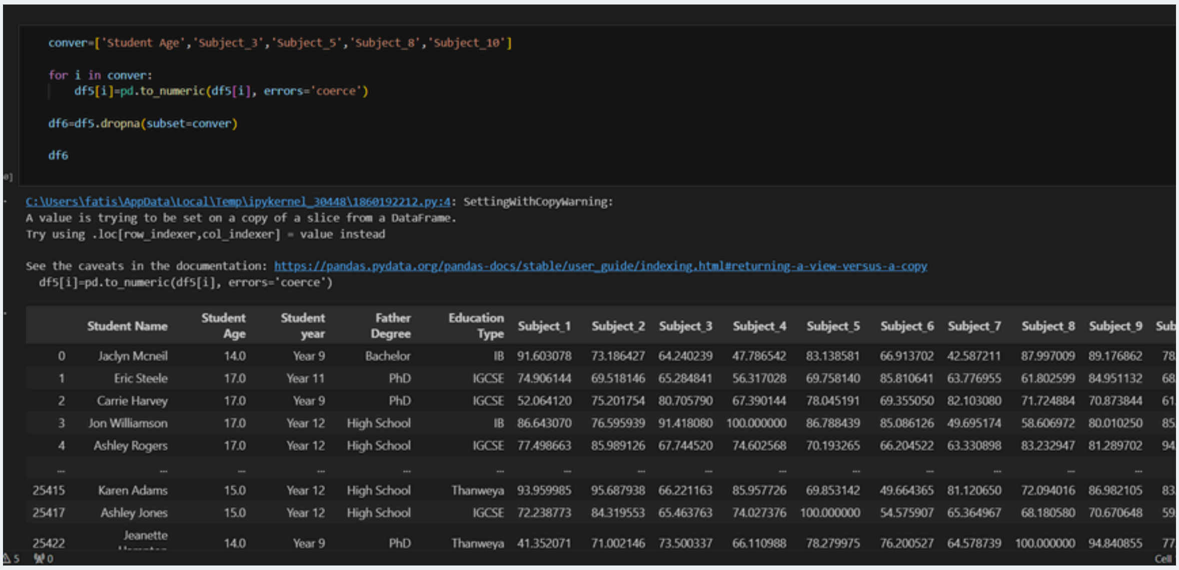
Volvemos a comprobar que los duplicados se hayan eliminado

En la base tambien se encontraron datos nulos como ‘bbb’ por lo que hice una eliminacion de ellos



```
df4['Student Name'] = df4['Student Name'].replace('bbb', '', regex=True)

print(df4['Student Name'])
```

```
0          Jaclyn Mcneil
1            Eric Steele
2          Carrie Harvey
3         Jon Williamson
4          Ashley Rogers
              ...
25417        Ashley Jones
25422    Jeanette Hampton
25429      Madeline Craig
25430     Joshua Castillo
25431         Marie Smith
Name: Student Name, Length: 23706, dtype: object
C:\Users\fatis\AppData\Local\Temp\ipykernel_30448\1769366204.py:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  df4['Student Name'] = df4['Student Name'].replace('bbb', '', regex=True)
```

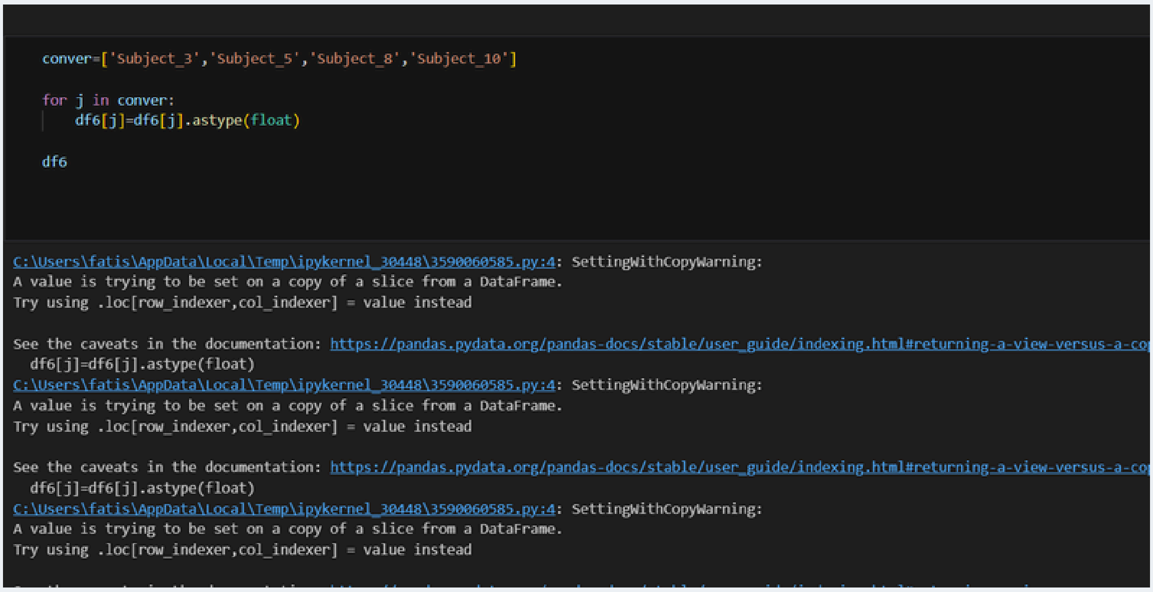


```
conver=['Student Age','Subject_3','Subject_5','Subject_8','Subject_10']

for i in conver:
    df5[i]=pd.to_numeric(df5[i], errors='coerce')

df6=df5.dropna(subset=conver)

df6
```

```
C:\Users\fatis\AppData\Local\Temp\ipykernel_30448\1860192212.py:4: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  df5[i]=pd.to_numeric(df5[i], errors='coerce')
```

	Student Name	Student Age	Student year	Father Degree	Education Type	Subject_1	Subject_2	Subject_3	Subject_4	Subject_5	Subject_6	Subject_7	Subject_8	Subject_9	Sub
0	Jaclyn Mcneil	14.0	Year 9	Bachelor	IB	91.603078	73.186427	64.240239	47.786542	83.138581	66.913702	42.587211	87.997009	89.176862	78
1	Eric Steele	17.0	Year 11	PhD	IGCSE	74.906144	69.518146	65.284841	56.317028	69.758140	85.810641	63.776955	61.802599	84.951132	68
2	Carrie Harvey	17.0	Year 9	PhD	IGCSE	52.064120	75.201754	80.705790	67.390144	78.045191	69.355050	82.103080	71.724884	70.873844	61.
3	Jon Williamson	17.0	Year 12	High School	IB	86.643070	76.595939	91.418080	100.000000	86.788439	85.086126	49.695174	58.606972	80.010250	85
4	Ashley Rogers	17.0	Year 12	High School	IGCSE	77.498663	85.989126	67.744520	74.602568	70.193265	66.204522	63.330898	83.232947	81.289702	94
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
25415	Karen Adams	15.0	Year 12	High School	Thanweya	93.959985	95.687938	66.221163	85.957726	69.853142	49.664365	81.120650	72.094016	86.982105	83
25417	Ashley Jones	15.0	Year 12	High School	IGCSE	72.238773	84.319553	65.463763	74.027376	100.000000	54.575907	65.364967	68.180580	70.670648	59
25422	Jeanette	14.0	Year 9	PhD	Thanweya	41.352071	71.002146	73.500337	66.110988	78.279975	76.200527	64.578739	100.000000	94.840855	77

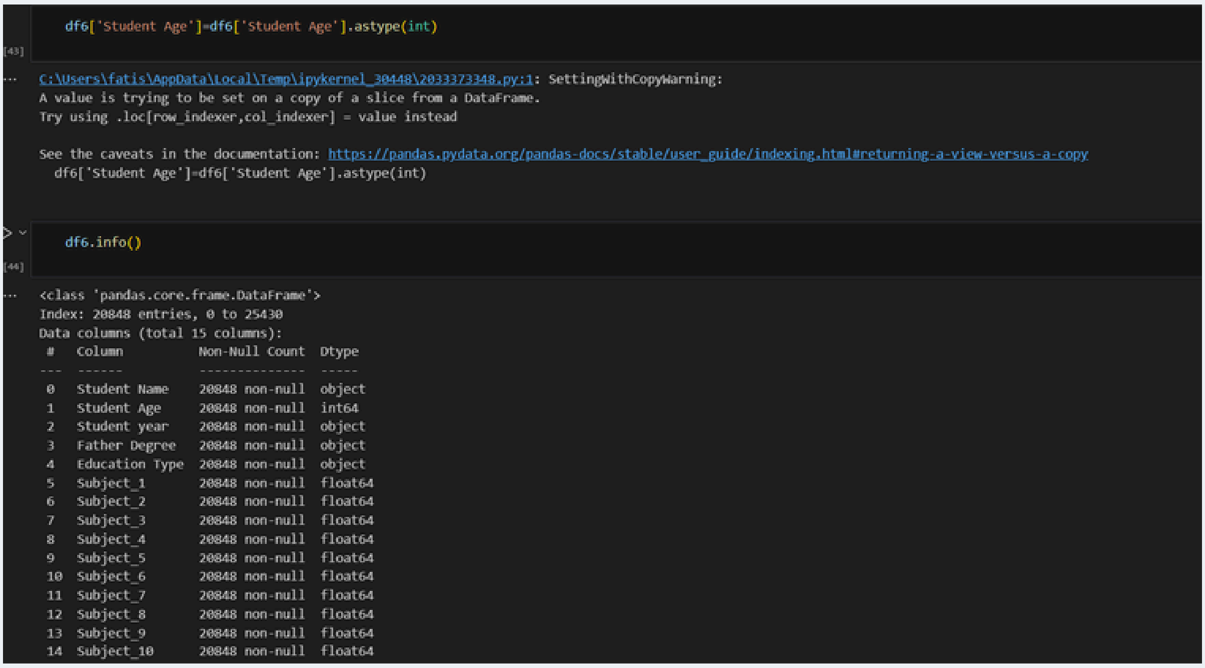
Cambiamos los Subject de tipo objeto a tipo float



```
conver=['Subject_3','Subject_5','Subject_8','Subject_10']

for j in conver:
    df6[j]=df6[j].astype(float)

df6
```

```
C:\Users\fatis\AppData\Local\Temp\ipykernel_30448\3590060585.py:4: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-co
  df6[j]=df6[j].astype(float)
C:\Users\fatis\AppData\Local\Temp\ipykernel_30448\3590060585.py:4: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-co
  df6[j]=df6[j].astype(float)
C:\Users\fatis\AppData\Local\Temp\ipykernel_30448\3590060585.py:4: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead
```

Tambien cambiamos ‘Student Age’ a dato tipo entero y despues vemos si el dato cambio correctamente

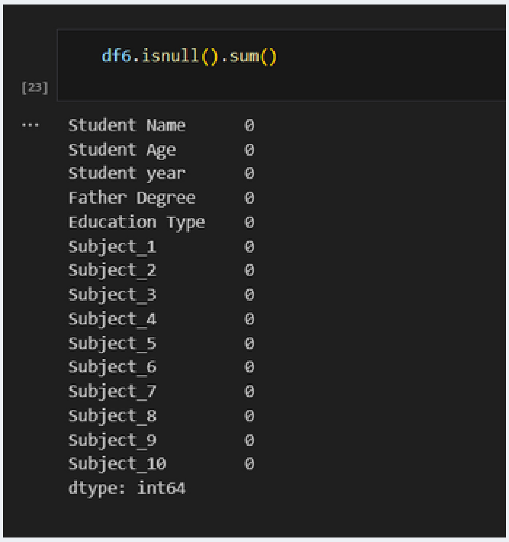


```
df6['Student Age']=df6['Student Age'].astype(int)
[43]
```

```
C:\Users\fatis\AppData\Local\Temp\ipykernel_30448\2033373348.py:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  df6['Student Age']=df6['Student Age'].astype(int)
```

```
df6.info()
[44]
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 20848 entries, 0 to 25430
Data columns (total 15 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   Student Name    20848 non-null  object
 1   Student Age     20848 non-null  int64
 2   Student year    20848 non-null  object
 3   Father Degree   20848 non-null  object
 4   Education Type  20848 non-null  object
 5   Subject_1       20848 non-null  float64
 6   Subject_2       20848 non-null  float64
 7   Subject_3       20848 non-null  float64
 8   Subject_4       20848 non-null  float64
 9   Subject_5       20848 non-null  float64
 10  Subject_6       20848 non-null  float64
 11  Subject_7       20848 non-null  float64
 12  Subject_8       20848 non-null  float64
 13  Subject_9       20848 non-null  float64
 14  Subject_10      20848 non-null  float64
```

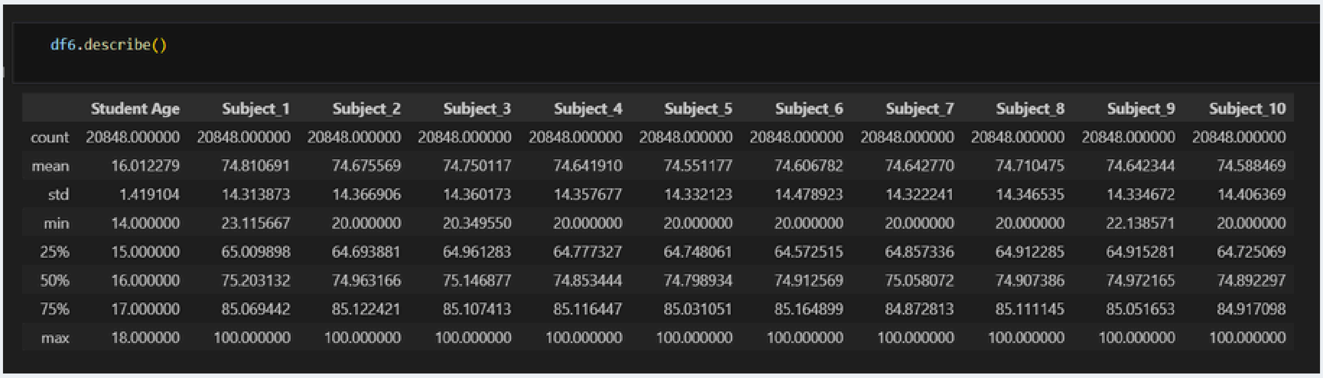


```
df6.isnull().sum()
[23]
```

```
Student Name      0
Student Age       0
Student year      0
Father Degree     0
Education Type    0
Subject_1         0
Subject_2         0
Subject_3         0
Subject_4         0
Subject_5         0
Subject_6         0
Subject_7         0
Subject_8         0
Subject_9         0
Subject_10        0
dtype: int64
```

Comprobamos que no haya mas valores nulos

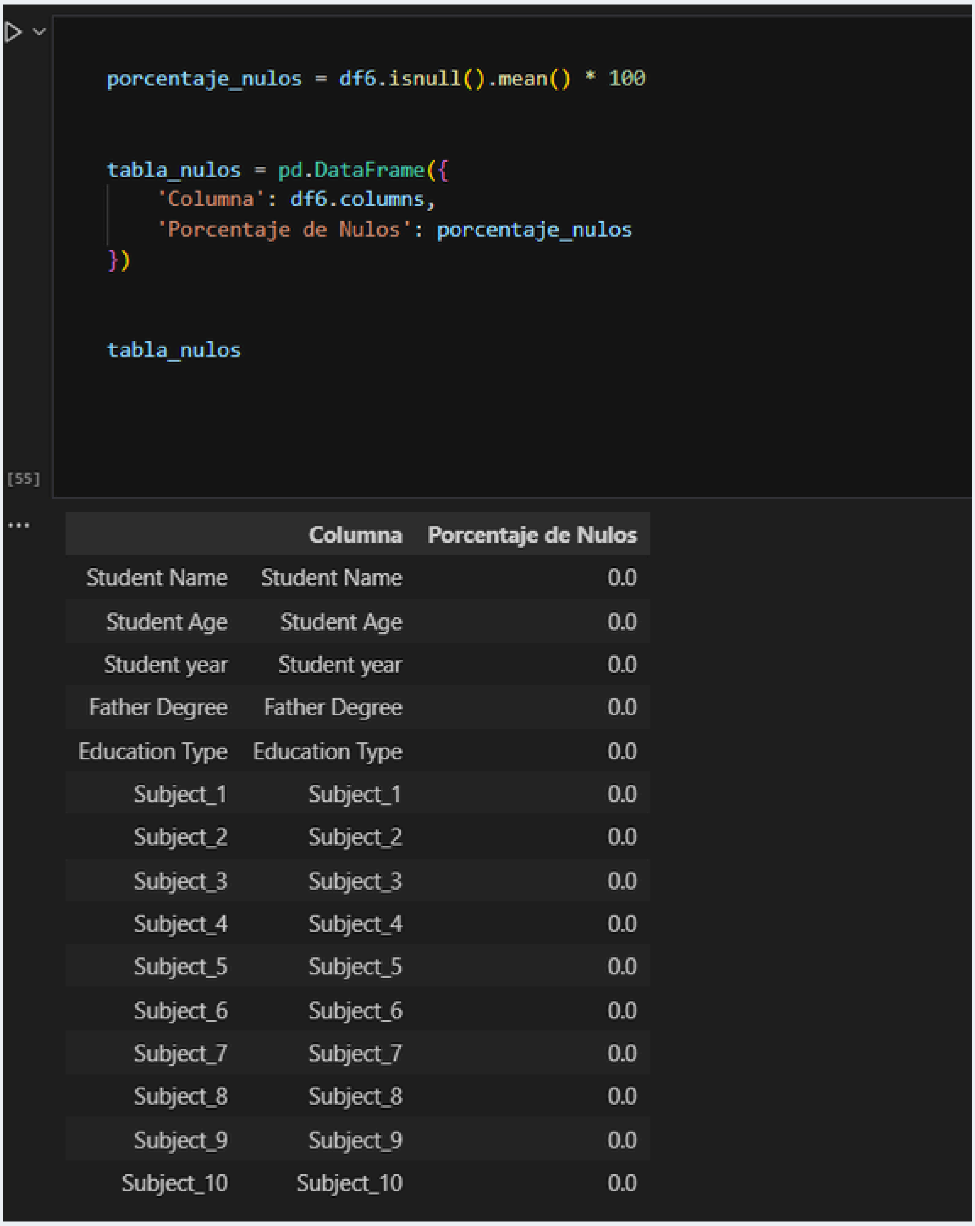
Resumen estadístico de la base de datos ya limpia



```
df6.describe()
```

	Student Age	Subject_1	Subject_2	Subject_3	Subject_4	Subject_5	Subject_6	Subject_7	Subject_8	Subject_9	Subject_10
count	20848.000000	20848.000000	20848.000000	20848.000000	20848.000000	20848.000000	20848.000000	20848.000000	20848.000000	20848.000000	20848.000000
mean	16.012279	74.810691	74.675569	74.750117	74.641910	74.551177	74.606782	74.642770	74.710475	74.642344	74.588469
std	1.419104	14.313873	14.366906	14.360173	14.357677	14.332123	14.478923	14.322241	14.346535	14.334672	14.406369
min	14.000000	23.115667	20.000000	20.349550	20.000000	20.000000	20.000000	20.000000	20.000000	22.138571	20.000000
25%	15.000000	65.009898	64.693881	64.961283	64.777327	64.748061	64.572515	64.857336	64.912285	64.915281	64.725069
50%	16.000000	75.203132	74.963166	75.146877	74.853444	74.798934	74.912569	75.058072	74.907386	74.972165	74.892297
75%	17.000000	85.069442	85.122421	85.107413	85.116447	85.031051	85.164899	84.872813	85.111145	85.051653	84.917098
max	18.000000	100.000000	100.000000	100.000000	100.000000	100.000000	100.000000	100.000000	100.000000	100.000000	100.000000

Porcentaje de nulos en la base ya limpia



```
porcentaje_nulos = df6.isnull().mean() * 100


tabla_nulos = pd.DataFrame({
    'Columna': df6.columns,
    'Porcentaje de Nulos': porcentaje_nulos
})


tabla_nulos
```

	Columna	Porcentaje de Nulos
Student Name	Student Name	0.0
Student Age	Student Age	0.0
Student year	Student year	0.0
Father Degree	Father Degree	0.0
Education Type	Education Type	0.0
Subject_1	Subject_1	0.0
Subject_2	Subject_2	0.0
Subject_3	Subject_3	0.0
Subject_4	Subject_4	0.0
Subject_5	Subject_5	0.0
Subject_6	Subject_6	0.0
Subject_7	Subject_7	0.0
Subject_8	Subject_8	0.0
Subject_9	Subject_9	0.0
Subject_10	Subject_10	0.0