Notebook resumen del proyecto

1. Importación de librerias y dataset

In [117…
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import os
```

In [118…
```
df = pd.read_csv('./data/Student_performance_data.csv')
```

2. Descripción de las variables

Información del estudiante

- **StudentID**: Número de identificación único de cada estudiante (1001 hasta 3392) *(Categórica, nominal, politómica)*

Información demográfica

- **Age**: Edad de los estudiantes en años enteros entre 15 y 18 *(Númerica, discreta, razón)*
- **Gender**: Genero *(Categórica, nominal, dicotómica)*
  - 0 -> hombre
  - 1 -> mujer
- **Ethnicity**: Étnia de los estudiantes *(Categórica, nominal, politómica)*
  - 0 -> Caucasian
  - 1 -> African American
  - 2 -> Asian
  - 3 -> Other
- **ParentalEducation**: El nivel de educación de los padres *(Categórica, ordinal, politómica)*
  - 0 -> None
  - 1 -> High School
  - 2 -> Some College
  - 3 -> Bachelor's
  - 4 -> Higher

Hábitos de estudio

- **StudyTimeWeekly**: Horas de estudio semanal en horas con decimales entre 0 y 20 *(Númerica, continua, intervalo)*
- **Absences**: Numero de absencias en dias enteros entre 0 y 30 *(Númerica, discreta, razón)*
- **Tutoring**: El alumno tiene tutor *(Categórica, nominal, dicotómica)*
  - 0 -> No
  - 1 -> Si

Participación de los padres

- **ParentalSupport**: Nivel de participación de los padres *(Categórica, ordinal, politómica)*
  - 0 -> None
  - 1 -> Low
  - 2 -> Moderate
  - 3 -> High
  - 4 -> Very High

Actividades extracurriculares

- **Extracurricular**: Participación en actividades extracurriculares *(Categórica, nominal, dicotómica)*
  - 0 -> No
  - 1 -> Si
- **Sports**: Práctica de deporte *(Categórica, nominal, dicotómica)*
  - 0 -> No
  - 1 -> Si
- **Music**: Práctica de música *(Categórica, nominal, dicotómica)*
  - 0 -> No
  - 1 -> Si
- **Volunteering**: Práctica de voluntariado *(Categórica, nominal, dicotómica)*
  - 0 -> No

- 1 -> Si

Variable objetivo

- **GPA**: Siglas para 'Grade Point Average' que es la media de una puntuación entre 2.0 y 4.0, influenciada por los hábitos de estudio, la participación de los padres y las actividades extracurriculares *(Númerica, continua, intervalo)*

Variable objetivo

- **GradeClass**: Clasificación de los estudiantes basada en el GPA *(Categórica, ordinal, politómica)*
  - 0 -> 'A' (GPA >= 3.5)
  - 1 -> 'B' (3.0 <= GPA < 3.5)
  - 2 -> 'C' (2.5 <= GPA < 3.0)
  - 3 -> 'D' (2.0 <= GPA < 2.5)
  - 4 -> 'F' (GPA < 2.0)

3. Exploración de los datos

In [119…
```
df.head(10)
```

Out[119…

	StudentID	Age	Gender	Ethnicity	ParentalEducation	StudyTimeWeekly	Absences	Tutoring	ParentalSupport	Extracurricular
0	1001	17	1	0	2	19.833723	7	1	2	0
1	1002	18	0	0	1	15.408756	0	0	1	0
2	1003	15	0	2	3	4.210570	26	0	2	0
3	1004	17	1	0	3	10.028829	14	0	3	1
4	1005	17	1	0	2	4.672495	17	1	3	0
5	1006	18	0	0	1	8.191219	0	0	1	1
6	1007	15	0	1	1	15.601680	10	0	3	0
7	1008	15	1	1	4	15.424496	22	1	1	1
8	1009	17	0	0	0	4.562008	1	0	2	0
9	1010	16	1	0	1	18.444466	0	0	3	1

In [120…
```
df.shape
```

Out[120… (2392, 15)

In [121…
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2392 entries, 0 to 2391
Data columns (total 15 columns):
 #   Column             Non-Null Count  Dtype  
---  ------             --------------  -----  
 0   StudentID          2392 non-null   int64  
 1   Age                2392 non-null   int64  
 2   Gender             2392 non-null   int64  
 3   Ethnicity          2392 non-null   int64  
 4   ParentalEducation  2392 non-null   int64  
 5   StudyTimeWeekly    2392 non-null   float64
 6   Absences           2392 non-null   int64  
 7   Tutoring           2392 non-null   int64  
 8   ParentalSupport    2392 non-null   int64  
 9   Extracurricular    2392 non-null   int64  
 10  Sports             2392 non-null   int64  
 11  Music              2392 non-null   int64  
 12  Volunteering       2392 non-null   int64  
 13  GPA                2392 non-null   float64
 14  GradeClass         2392 non-null   float64
dtypes: float64(3), int64(12)
memory usage: 280.4 KB
```

In [122…
```
df.isnull().sum()
```

Out[122…
```
StudentID            0
Age                  0
Gender               0
Ethnicity            0
ParentalEducation    0
StudyTimeWeekly      0
Absences             0
Tutoring             0
ParentalSupport      0
Extracurricular      0
Sports               0
Music                0
Volunteering         0
GPA                  0
GradeClass           0
dtype: int64
```

In [123…
```
df.duplicated().sum()
```

Out[123…
```
0
```

In [124…
```
df.nunique()
```

Out[124…
```
StudentID            2392
Age                     4
Gender                  2
Ethnicity               4
ParentalEducation       5
StudyTimeWeekly      2392
Absences               30
Tutoring                2
ParentalSupport         5
Extracurricular         2
Sports                  2
Music                   2
Volunteering            2
GPA                  2371
GradeClass              5
dtype: int64
```

In [125…
```
df.describe()
```

Out[125…

	StudentID	Age	Gender	Ethnicity	ParentalEducation	StudyTimeWeekly	Absences	Tutoring	Paren
count	2392.000000	2392.000000	2392.000000	2392.000000	2392.000000	2392.000000	2392.000000	2392.000000	23
mean	2196.500000	16.468645	0.510870	0.877508	1.746237	9.771992	14.541388	0.301421	
std	690.655244	1.123798	0.499986	1.028476	1.000411	5.652774	8.467417	0.458971	
min	1001.000000	15.000000	0.000000	0.000000	0.000000	0.001057	0.000000	0.000000	
25%	1598.750000	15.000000	0.000000	0.000000	1.000000	5.043079	7.000000	0.000000	
50%	2196.500000	16.000000	1.000000	0.000000	2.000000	9.705363	15.000000	0.000000	
75%	2794.250000	17.000000	1.000000	2.000000	2.000000	14.408410	22.000000	1.000000	
max	3392.000000	18.000000	1.000000	3.000000	4.000000	19.978094	29.000000	1.000000	

In [126…
```
# Las variables StudentID y GradeClass no son necesarias, ya que la primera es un indice y la segunda es deriva
df.drop(['GradeClass','StudentID'], axis=1, inplace=True)
df.head(5)
```

Out[126…

	Age	Gender	Ethnicity	ParentalEducation	StudyTimeWeekly	Absences	Tutoring	ParentalSupport	Extracurricular	Sports	Mu
0	17	1	0	2	19.833723	7	1	2	0	0	
1	18	0	0	1	15.408756	0	0	1	0	0	
2	15	0	2	3	4.210570	26	0	2	0	0	
3	17	1	0	3	10.028829	14	0	3	1	0	
4	17	1	0	2	4.672495	17	1	3	0	0	

Conclusiones de la exploración:

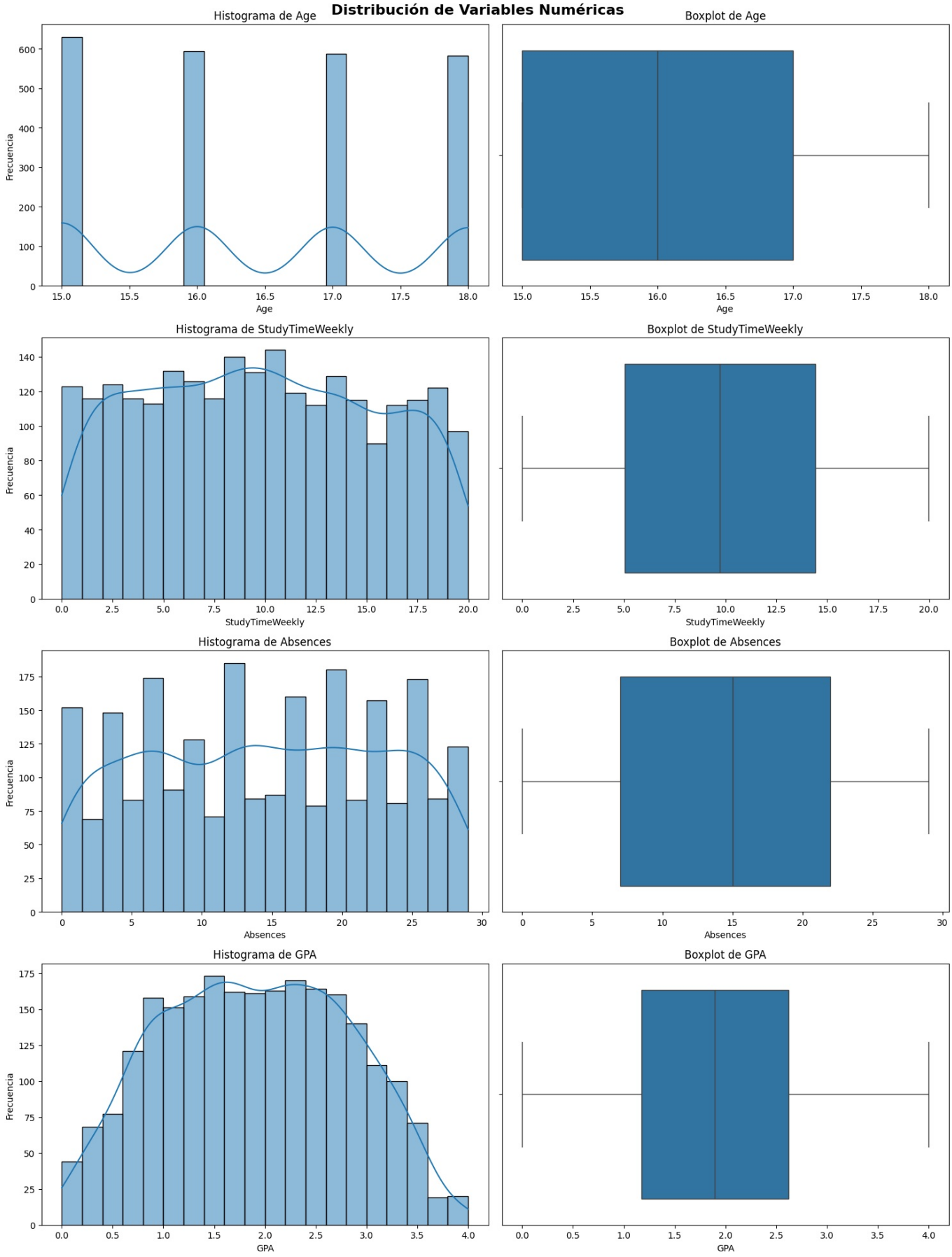
- No hay datos nulos y todos son numéricos
- La variable objetivo es GPA

3. Analisis de los datos - EDA

In [127…
```
# Creación de dos variables para separar las variables númericas y las categóricas
variables_numericas = ['Age', 'StudyTimeWeekly', 'Absences', 'GPA']
df_numericas = df[variables_numericas]

variables_categoricas = ['Gender', 'Ethnicity', 'ParentalEducation', 'Tutoring', 'ParentalSupport', 'Extracurri
df_categoricas = df[variables_categoricas]
```

Variables numéricas

In [128…
```
# Histogramas y boxplots para la distribución de las variables numericas
def generar_graficos_numericos(df, variables):
  fig, ax = plt.subplots(len(variables), 2, figsize=(15, 5 * len(variables)))
  fig.suptitle('Distribución de Variables Numéricas', fontweight='bold', fontsize=16)

  for i, variable in enumerate(variables):
    # Histogramas
    sns.histplot(df[variable], bins=20, kde=True, ax=ax[i, 0])
    ax[i, 0].set_title(f'Histograma de {variable}')
    ax[i, 0].set_xlabel(variable)
    ax[i, 0].set_ylabel('Frecuencia')

    # Boxplots
    sns.boxplot(x=df[variable], ax=ax[i, 1])
    ax[i, 1].set_title(f'Boxplot de {variable}')
    ax[i, 1].set_xlabel(variable)

  dir = str(f'./graph/EDA/')
  os.makedirs(dir, exist_ok=True)
  file = str(f'Distribución de variables numéricas.png')
  plt.savefig(dir + file)

  plt.tight_layout()
  plt.show()

generar_graficos_numericos(df_numericas, variables_numericas)
```



Variables categóricas

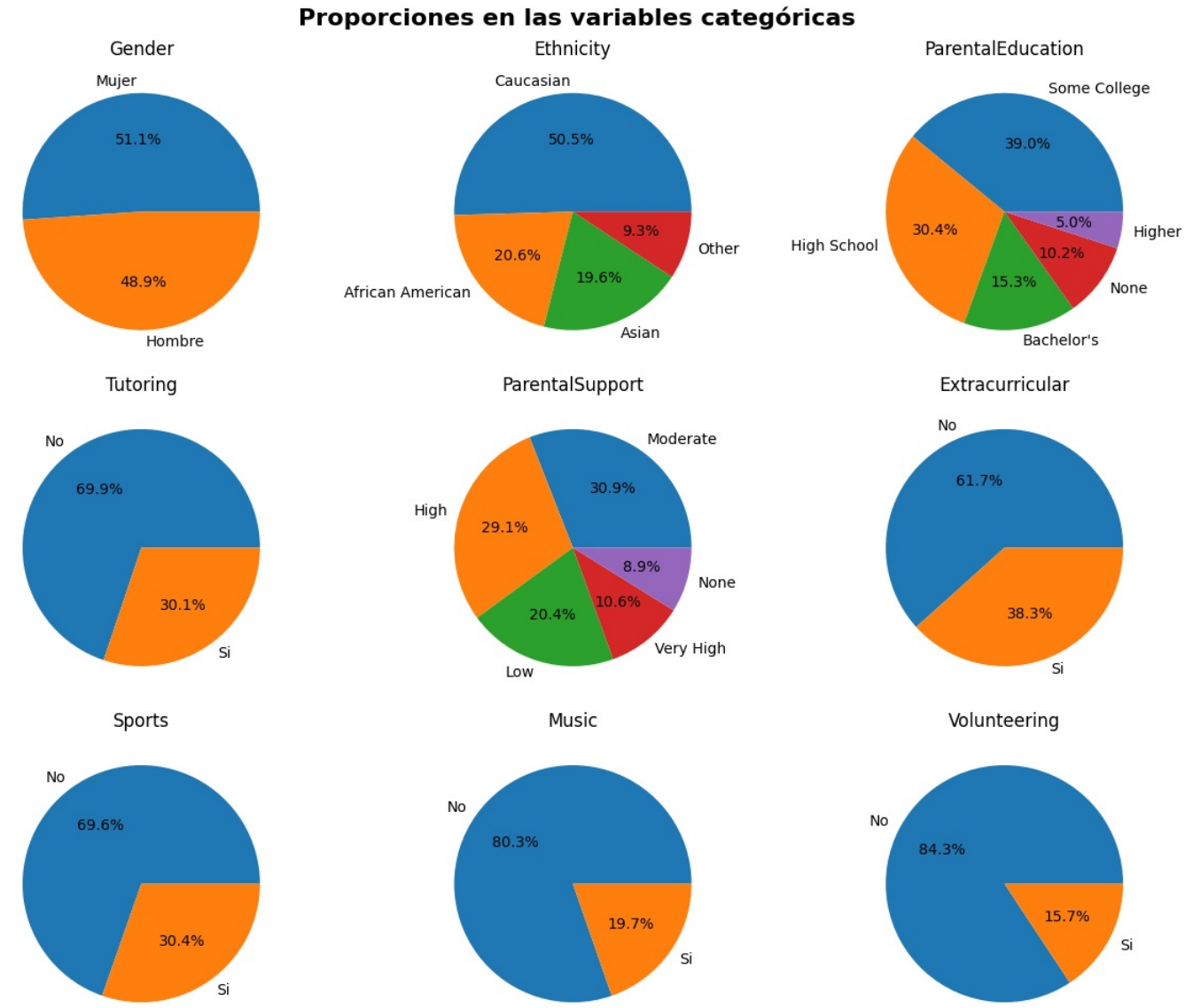
In [129…

```python
# Crear labels de las variables categoricas para mostrar en los gráficos

# Definir los diccionarios de mapeo
gender_map = {0: 'Hombre', 1: 'Mujer'}
ethnicity_map = {0: 'Caucasian', 1: 'African American', 2: 'Asian', 3: 'Other'}
parental_education_map = {0: 'None', 1: 'High School', 2: 'Some College', 3: "Bachelor's", 4: 'Higher'}
tutoring_map = {0: 'No', 1: 'Si'}
parental_support_map = {0: 'None', 1: 'Low', 2: 'Moderate', 3: 'High', 4: 'Very High'}
extracurricular_map = {0: 'No', 1: 'Si'}
sports_map = {0: 'No', 1: 'Si'}
```

```
music_map = {0: 'No', 1: 'Si'}
volunteering_map = {0: 'No', 1: 'Si'}

# Copiar el df para no modificar el original
df_graficos = df.copy()

# Mapear las variables categóricas
df_graficos['Gender'] = df_graficos['Gender'].map(gender_map)
df_graficos['Ethnicity'] = df_graficos['Ethnicity'].map(ethnicity_map)
df_graficos['ParentalEducation'] = df_graficos['ParentalEducation'].map(parental_education_map)
df_graficos['Tutoring'] = df_graficos['Tutoring'].map(tutoring_map)
df_graficos['ParentalSupport'] = df_graficos['ParentalSupport'].map(parental_support_map)
df_graficos['Extracurricular'] = df_graficos['Extracurricular'].map(extracurricular_map)
df_graficos['Sports'] = df_graficos['Sports'].map(sports_map)
df_graficos['Music'] = df_graficos['Music'].map(music_map)
df_graficos['Volunteering'] = df_graficos['Volunteering'].map(volunteering_map)

# Crear la lista de labels para las variables categóricas
labels_categoricas = ['Gender', 'Ethnicity', 'ParentalEducation', 'Tutoring', 'ParentalSupport', 'Extracurricula
```

In [130…

```
# Crear gráficos de pastel para ver las proporciones de las variables categóricas

fig, ax = plt.subplots(3,3, figsize=(12,10))
fig.suptitle('Proporciones en las variables categóricas', fontweight='bold', fontsize=16)

# Recorrer la lista para ir creando los gráficos
for i, var in enumerate(labels_categoricas):
  fila = i // 3
  col = i % 3
  data = df_graficos[var].value_counts()
  ax[fila, col].pie(data, labels=data.index, autopct='%1.1f%%')
  ax[fila, col].set_title(var)

dir = str(f'./graph/EDA/')
os.makedirs(dir, exist_ok=True)
file = str(f'Proporciones de variables categóricas.png')
plt.savefig(dir + file)

plt.tight_layout()
plt.show()
```



Conclusiones de la exploración inicial:

- La distribución de Age es bastante uniforme entre 15 y 18.
- A simple vista las variables numéricas no parecen seguir una distribución concreta, aunque muestran cierta uniformidad.
- Hay proporcion semejante entre hombres y mujeres.
- La etnia mas representada es la caucasica.
- La mayoria de los padres tienen estudios de College o secundaria
- La mayoria de alumnos no tienen tutor
- La mayoria de alumnos no hacen actividades extracurriculares
- La mayoria de alumnos no practican deporte
- La mayoria de alumnos no practican musica
- La mayoria de alumnos no hacen voluntariado
- Los boxplot muestran que no hay valores outlier

Limpieza de valores missing y outliers

In [131…

```python
# Función para identificar outliers con el método del rango intercuartilico

def identificar_outliers(df, col_categorica, col_cuantitativa):
  outliers = pd.DataFrame()

  for categoria in df[col_categorica].unique():
    data_categoria = df[df[col_categorica] == categoria][col_cuantitativa]
    Q1 = data_categoria.quantile(0.25)
```

```
    Q3 = data_categoria.quantile(0.75)
    IQR = Q3 - Q1
    limite_inferior = Q1 - 1.5 * IQR
    limite_superior = Q3 + 1.5 * IQR
    outliers_categoria = data_categoria[(data_categoria < limite_inferior) | (data_categoria > limite_superior)
    outliers = pd.concat([outliers, outliers_categoria])
  return outliers

# Identificar outliers en la variable 'Absences'
outliers = identificar_outliers(df, 'GPA', 'Absences')
print(f"Outliers identificados:\n{outliers}")
```

```
Outliers identificados:
      Absences
1278       3.0
474       24.0
```

In [132…

```
# Identificar los indices de los valores outlier y eliminarlos del dataframe

indices_outliers = outliers.index
df = df.drop(indices_outliers)
```

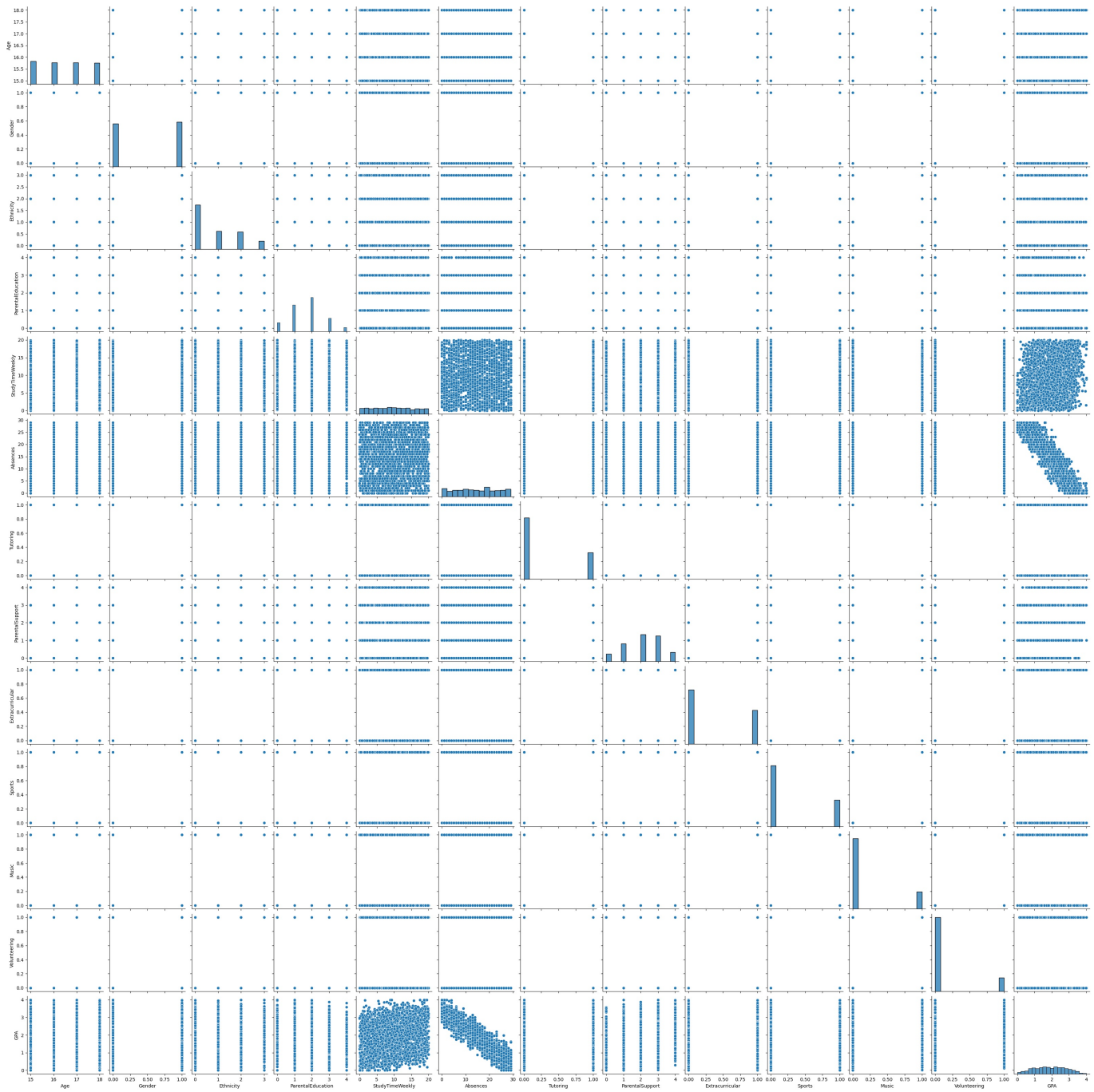
Conclusión de la limpieza:

- No hay valores missing
- Solamente hay dos valores outlier en la variable Absences que se eliminan

Correlación de las variables

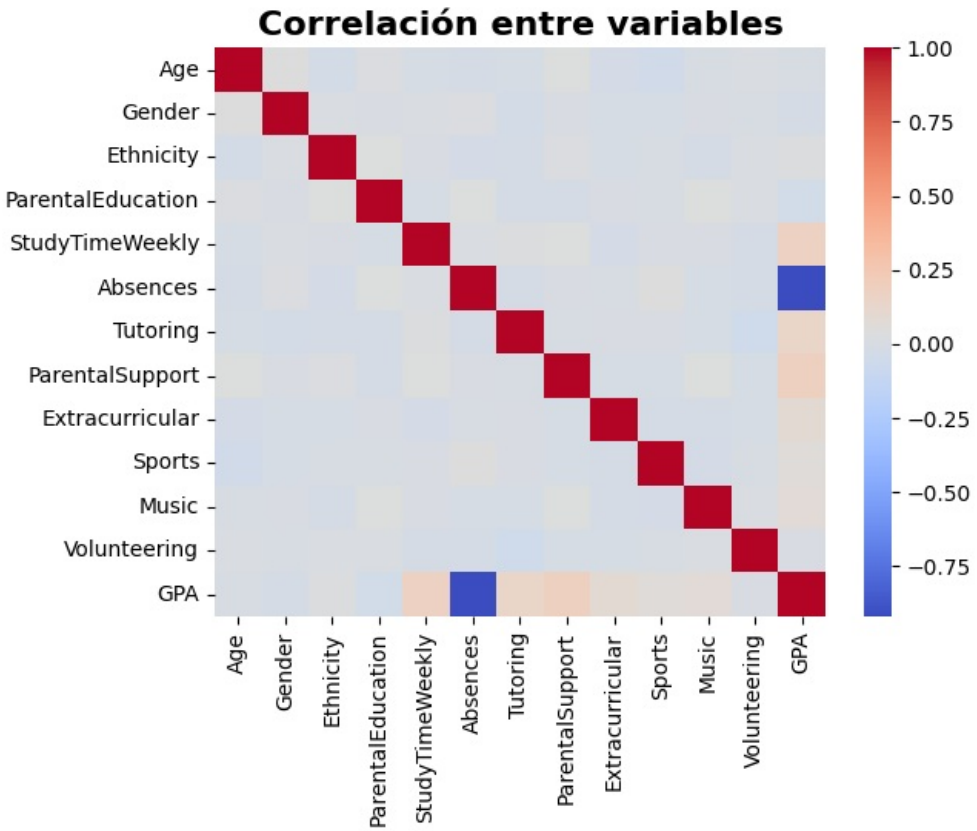
In [114…

```
sns.pairplot(df)
plt.show()
```

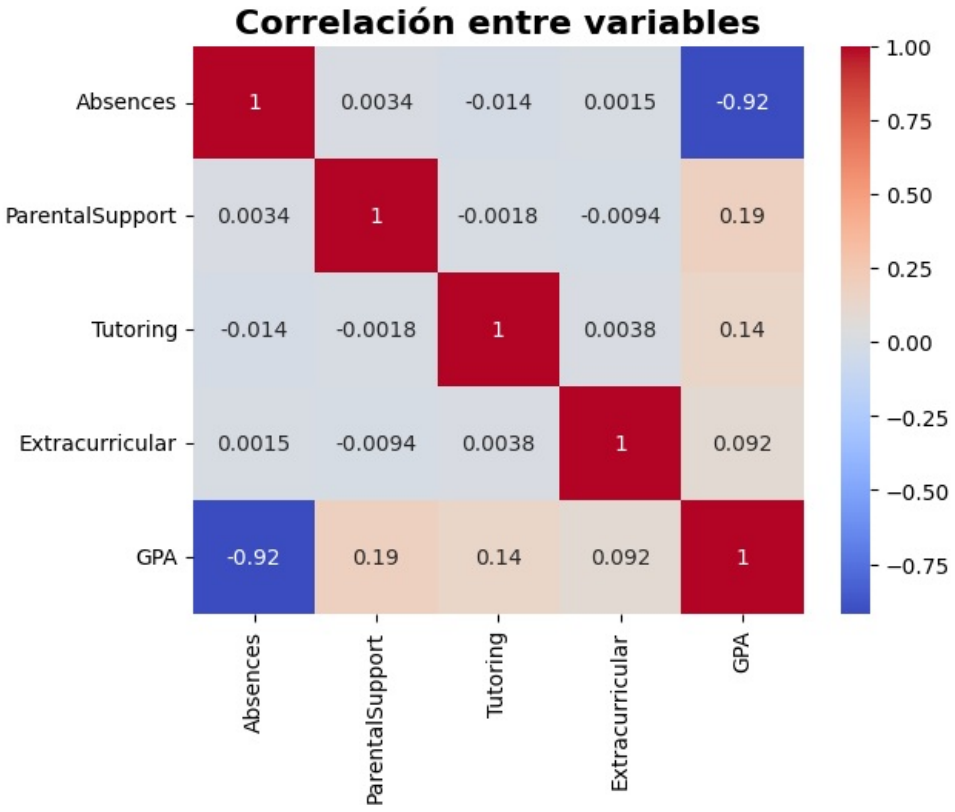


In [115…

```
sns.heatmap(df.corr(), cmap='coolwarm')
plt.title('Correlación entre variables', fontweight='bold', fontsize=16)

dir = str(f'./graph/EDA/')
os.makedirs(dir, exist_ok=True)
file = str(f'Correlación entre variables.png')
plt.savefig(dir + file)

plt.show()
```



In [116…
```
sns.heatmap(df[['Absences', 'ParentalSupport', 'Tutoring', 'Extracurricular', 'GPA']].corr(), annot=True, cmap=
plt.title('Correlación entre variables', fontweight='bold', fontsize=16)
plt.show()
```



Conclusiones de la correclación:

- Hay una fuerte correlación inversa entre Absences y GPA
- Hay una correlación directa entre ParentalSupport, StudyTimeWeekly y Tutoring y GPA
- Hay una correlación directa débil entre Extracurricular y GPA
- No se observan correlaciones cruzadas entre las variables independientes