

Determinantes del Rendimiento Académico: Un Análisis Integral de Factores Estudiantiles

Objetivo del análisis:

Identificar los factores más influyentes en el rendimiento académico de los estudiantes, medido por la calificación del examen final, para ofrecer recomendaciones específicas que mejoren los resultados educativos.

Usuario Final:

María González, Directora del Departamento de Educación y Bienestar Estudiantil en una institución educativa privada. María busca comprender qué factores impactan más en el rendimiento académico para diseñar estrategias efectivas que incrementen las tasas de éxito estudiantil. Estas estrategias podrían incluir programas de tutorías, acceso a recursos educativos, mejoras en la infraestructura escolar o iniciativas que promuevan un equilibrio entre actividades extracurriculares y académicas.

Acerca del Conjunto de Datos

Contiene datos sobre factores de rendimiento de estudiantes:

- **Horas Estudiadas:** Número de horas dedicadas al estudio por semana.
- **Asistencia:** Porcentaje de clases asistidas.
- **Participación Parental:** Nivel de participación de los padres en la educación del estudiante (Bajo, Medio, Alto).
- **Acceso a Recursos:** Disponibilidad de recursos educativos (Bajo, Medio, Alto).
- **Actividades Extracurriculares:** Participación en actividades extracurriculares (Sí, No).
- **Horas Dormir:** Número promedio de horas de sueño por noche.
- **Puntuaciones Anteriores:** Puntuaciones de exámenes anteriores.
- **Nivel Motivación:** Nivel de motivación del estudiante (Bajo, Medio, Alto).
- **Acceso Internet:** Disponibilidad de acceso a internet (Sí, No).
- **Tutorías:** Número de tutorías asistidas por mes.
- **Ingresos Familiares:** Nivel de ingresos familiares (Bajo, Medio, Alto).
- **Calidad Profesor:** Calidad de los profesores (Bajo, Medio, Alto).

- **Tipo Escuela:** Tipo de escuela a la que asiste (Pública, Privada).
- **Influencia de los Pares:** Influencia de los pares en el rendimiento académico (Positiva, Neutral, Negativa).
- **Actividad Física:** Número promedio de horas de actividad física por semana.
- **Discapacidades de Aprendizaje:** Presencia de discapacidades de aprendizaje (Sí, No).
- **Nivel Educativo de los Padres:** Nivel educativo más alto de los padres (Secundaria, Universidad, Posgrado).
- **Distancia desde el Hogar:** Distancia desde el hogar hasta la escuela (Cerca, Moderada, Lejana).
- **Género:** Género del estudiante (Hombre, Mujer).
- **Calificación del Examen:** Calificación del examen final.

[Accede al dataset](#)

Importación de librerías

```
In [299... import pandas as pd
import matplotlib.pyplot as plt
```

```
In [4]: #ESTA CONFIGURACIÓN ES PARA PODER VISUALIZAR TODAS LAS FILAS DEL DATASET EN CASO DE DESEARLO
pd.set_option('display.max.rows', 6700)
```

Lectura de los datos

```
In [7]: df = pd.read_csv(r"C:\Users\Tiki\OneDrive\Documentos\Datasets\StudentPerformanceFactors.csv")
```

Análisis exploratorio

Con el objetivo de verificar la integridad de los datos.

Visualizo las primeras filas del dataset

```
In [11]: df.head(10)
```

Out[11]:

	Hours_Studied	Attendance	Parental_Involvement	Access_to_Resources	Extracurricular_Activities	Sleep_Hours	Previous_Scores	Motivation_Level	Internet_A
0	23	84	Low	High	No	7	73	Low	
1	19	64	Low	Medium	No	8	59	Low	
2	24	98	Medium	Medium	Yes	7	91	Medium	
3	29	89	Low	Medium	Yes	8	98	Medium	
4	19	92	Medium	Medium	Yes	6	65	Medium	
5	19	88	Medium	Medium	Yes	8	89	Medium	
6	29	84	Medium	Low	Yes	7	68	Low	
7	25	78	Low	High	Yes	6	50	Medium	
8	17	94	Medium	High	No	6	80	High	
9	23	98	Medium	Medium	Yes	8	71	Medium	

◀

▶

Reviso que el tipo de dato de las columnas sea el correcto

In [14]:

df.dtypes

```
Out[14]: Hours_Studied          int64
Attendance                    int64
Parental_Involvement          object
Access_to_Resources           object
Extracurricular_Activities    object
Sleep_Hours                   int64
Previous_Scores               int64
Motivation_Level              object
Internet_Access               object
Tutoring_Sessions             int64
Family_Income                 object
Teacher_Quality               object
School_Type                   object
Peer_Influence                object
Physical_Activity              int64
Learning_Disabilities          object
Parental_Education_Level      object
Distance_from_Home            object
Gender                        object
Exam_Score                    int64
dtype: object
```

Reviso los valores nulos en las columnas. En este caso hay valores no numéricos vacíos que serán tratados en la limpieza.

```
In [17]: df.isnull().sum()
```

```
Out[17]: Hours_Studied      0
Attendance      0
Parental_Involvement  0
Access_to_Resources  0
Extracurricular_Activities  0
Sleep_Hours     0
Previous_Scores  0
Motivation_Level  0
Internet_Access  0
Tutoring_Sessions  0
Family_Income    0
Teacher_Quality  78
School_Type      0
Peer_Influence   0
Physical_Activity  0
Learning_Disabilities  0
Parental_Education_Level  90
Distance_from_Home  67
Gender           0
Exam_Score       0
dtype: int64
```

Obtengo medidas estadísticas para analizar valores atípicos. Dado el resultado no encuentro valores que me parezcan fuera de lugar o anormales

```
In [20]: df.describe()
```

Out[20]:

	Hours_Studied	Attendance	Sleep_Hours	Previous_Scores	Tutoring_Sessions	Physical_Activity	Exam_Score
count	6607.000000	6607.000000	6607.000000	6607.000000	6607.000000	6607.000000	6607.000000
mean	19.975329	79.977448	7.02906	75.070531	1.493719	2.967610	67.235659
std	5.990594	11.547475	1.46812	14.399784	1.230570	1.031231	3.890456
min	1.000000	60.000000	4.00000	50.000000	0.000000	0.000000	55.000000
25%	16.000000	70.000000	6.00000	63.000000	1.000000	2.000000	65.000000
50%	20.000000	80.000000	7.00000	75.000000	1.000000	3.000000	67.000000
75%	24.000000	90.000000	8.00000	88.000000	2.000000	4.000000	69.000000
max	44.000000	100.000000	10.00000	100.000000	8.000000	6.000000	101.000000

Verifico que la cantidad de valores únicos que debe haber en cada columna no numérica sea la correcta

```
In [23]: df.nunique()
```

```
Out[23]: Hours_Studied          41
Attendance                    41
Parental_Involvement          3
Access_to_Resources           3
Extracurricular_Activities     2
Sleep_Hours                   7
Previous_Scores               51
Motivation_Level              3
Internet_Access               2
Tutoring_Sessions             9
Family_Income                 3
Teacher_Quality               3
School_Type                   2
Peer_Influence                3
Physical_Activity             7
Learning_Disabilities          2
Parental_Education_Level      3
Distance_from_Home            3
Gender                        2
Exam_Score                    45
dtype: int64
```

Selecciono solamente las columnas numéricas y analizo si existe alguna correlación entre las variables. En este caso no se observan correlaciones positivas o negativas perfectas, pero no significa que las variables no tengan incidencia entre ellas, como veremos más adelante

```
In [26]: #PRIMERO SELECCIONO SOLAMENTE LAS COLUMNAS NUMÉRICAS
df_numerico = df.select_dtypes(include=['number'])
#LUEGO APLICO LA FUNCIÓN CORR PARA OBTENER LA CORRELACIÓN ENTRE LAS VARIABLES
df_numerico.corr()
```

Out[26]:

	Hours_Studied	Attendance	Sleep_Hours	Previous_Scores	Tutoring_Sessions	Physical_Activity	Exam_Score
Hours_Studied	1.000000	-0.009908	0.010977	0.024846	-0.014282	0.004624	0.445455
Attendance	-0.009908	1.000000	-0.015918	-0.020186	0.014324	-0.022435	0.581072
Sleep_Hours	0.010977	-0.015918	1.000000	-0.021750	-0.012216	-0.000378	-0.017022
Previous_Scores	0.024846	-0.020186	-0.021750	1.000000	-0.013122	-0.011274	0.175079
Tutoring_Sessions	-0.014282	0.014324	-0.012216	-0.013122	1.000000	0.017733	0.156525
Physical_Activity	0.004624	-0.022435	-0.000378	-0.011274	0.017733	1.000000	0.027824
Exam_Score	0.445455	0.581072	-0.017022	0.175079	0.156525	0.027824	1.000000

Limpieza de los datos

Luego del análisis exploratorio puedo decir que los datos se encuentran limpios y poseen la integridad necesaria para un correcto análisis, voy a proceder a eliminar aquellos registros que incluyan datos de alumnos de colegios públicos para centrar el análisis exclusivamente en los estudiantes del ámbito de interés de mi usuario final.

Por último voy a cambiar los nombres de las columnas y sus valores al español para estandarizarlos, y reemplazaré los valores nulos por una cadena vacía, ya que al ser éstos valores nulos no numéricos, no podría completarlos con un promedio o moda.

Cambio los nombres de las columnas del inglés al español

```
In [31]: df.columns = ['Horas Estudiadas', 'Asistencia', 'Participación Parental', 'Acceso a Recursos', 'Actividades Extracurriculares', 'Horas Dormir',  
    'Puntuaciones Anteriores', 'Nivel Motivación', 'Acceso Internet', 'Tutorías', 'Ingresos Familiares', 'Calidad Profesor', 'Tipo Escuela',  
    'Influencia de los pares', 'Actividad Física', 'Discapacidades de Aprendizaje', 'Nivel Educativo de los Padres', 'Distancia desde el Hogar',  
    'Género', 'Calificación del Examen']
```

Cambio los valores del inglés al español

```
In [34]: df['Participación Parental'] = df['Participación Parental'].replace({'Low':'Baja', 'Medium':'Media', 'High':'Alta'})  
df['Acceso a Recursos'] = df['Acceso a Recursos'].replace({'Low':'Baja', 'Medium':'Media', 'High':'Alta'})  
df['Actividades Extracurriculares'] = df['Actividades Extracurriculares'].replace('Yes','Si')  
df['Nivel Motivación'] = df['Nivel Motivación'].replace({'Low':'Baja', 'Medium':'Media', 'High':'Alta'})  
df['Acceso Internet'] = df['Acceso Internet'].replace('Yes','Si')  
df['Ingresos Familiares'] = df['Ingresos Familiares'].replace({'Low':'Baja', 'Medium':'Media', 'High':'Alta'})
```

```
df['Calidad Profesor'] = df['Calidad Profesor'].replace({'Low':'Baja', 'Medium':'Media', 'High':'Alta'})
df['Tipo Escuela'] = df['Tipo Escuela'].replace({'Private':'Privada', 'Public':'Pública'})
df['Influencia de los pares'] = df['Influencia de los pares'].replace({'Positive':'Positiva', 'Negative':'Negativa'})
df['Discapacidades de Aprendizaje'] = df['Discapacidades de Aprendizaje'].replace('Yes', 'Si')
df['Nivel Educativo de los Padres'] = df['Nivel Educativo de los Padres'].replace({'High School':'Secundaria', 'College':'Universidad', 'Postgraduate':'Posgrado'})
df['Distancia desde el Hogar'] = df['Distancia desde el Hogar'].replace({'Near':'Cerca', 'Moderate':'Moderada', 'Far':'Lejos'})
df['Género'] = df['Género'].replace({'Female':'Femenino', 'Male':'Masculino'})
```

Elimino los registros donde el tipo de institución es pública

```
In [37]: df = df[df['Tipo Escuela'] != 'Pública']
```

Reemplazo los valores nulos por una cadena de texto vacía

```
In [40]: df = df.fillna('')
```

```
In [42]: #YA NO HAY VALORES NULOS
df.isnull().sum()
```

```
Out[42]: Horas Estudiadas      0
Asistencia                    0
Participación Parental       0
Acceso a Recursos             0
Actividades Extracurriculares 0
Horas Dormir                  0
Puntuaciones Anteriores      0
Nivel Motivación              0
Acceso Internet               0
Tutorías                      0
Ingresos Familiares           0
Calidad Profesor              0
Tipo Escuela                  0
Influencia de los pares       0
Actividad Física              0
Discapacidades de Aprendizaje 0
Nivel Educativo de los Padres  0
Distancia desde el Hogar      0
Género                        0
Calificación del Examen       0
dtype: int64
```


Breve visualización de los datos estandarizados en español

```
In [45]: df.head(15)
```

Out[45]:

	Horas Estudiadas	Asistencia	Participación Parental	Acceso a Recursos	Actividades Extracurriculares	Horas Dormir	Puntuaciones Anteriores	Nivel Motivación	Acceso Internet	Tutorías	Ingresos Familiares	Calidad Profesor	Tipo Escuela	Infl
6	29	84	Media	Baja	Si	7	68	Baja	Si	1	Baja	Media	Privada	I
8	17	94	Media	Alta	No	6	80	Alta	Si	0	Media	Baja	Privada	I
10	17	80	Baja	Alta	No	8	88	Media	No	4	Media	Alta	Privada	I
11	17	97	Media	Alta	Si	6	87	Baja	Si	2	Baja	Alta	Privada	I
13	9	82	Media	Media	Si	8	72	Media	Si	2	Media	Media	Privada	F
14	10	78	Media	Alta	Si	8	74	Media	Si	1	Baja	Media	Privada	I
15	17	68	Media	Media	No	8	70	Media	Si	2	Media	Media	Privada	F
16	14	60	Media	Baja	Si	10	65	Baja	Si	0	Alta	Media	Privada	F
19	12	75	Media	Alta	Si	7	58	Media	Si	3	Media	Media	Privada	F
29	18	66	Alta	Alta	No	4	51	Baja	Si	2	Baja	Media	Privada	I
42	19	63	Baja	Alta	No	7	94	Media	Si	1	Media	Media	Privada	F
43	18	94	Media	Alta	Si	7	82	Media	Si	3	Media	Media	Privada	No
44	11	98	Media	Media	Si	9	82	Baja	Si	1	Media	Media	Privada	F
46	17	78	Media	Media	Si	6	88	Media	Si	1	Media	Alta	Privada	F
54	26	67	Media	Alta	Si	7	51	Baja	Si	2	Alta	Alta	Privada	I

Análisis de los Datos

Voy a proceder a realizar el análisis, mediante visualizaciones, de aquellos factores sobre los cuáles la institución podría trabajar para mejorar el rendimiento de sus alumnos, y también analizaré las variables que podrían llegar a afectar, positiva o negativamente, la motivación de los estudiantes.

Comienzo con un poco de contexto sobre la cantidad de estudiantes analizados por género

```
In [50]: df.groupby('Género').size()
```

```
Out[50]: Género
Femenino      843
Masculino     1166
dtype: int64
```

Verifico si existe una diferencia significativa en la calificación promedio, para ver si vale la pena realizar un análisis por género, pero la diferencia es casi nula.

```
In [53]: df[['Género', 'Calificación del Examen']].groupby('Género').mean('Calificación del Examen')
```

```
Out[53]:
```

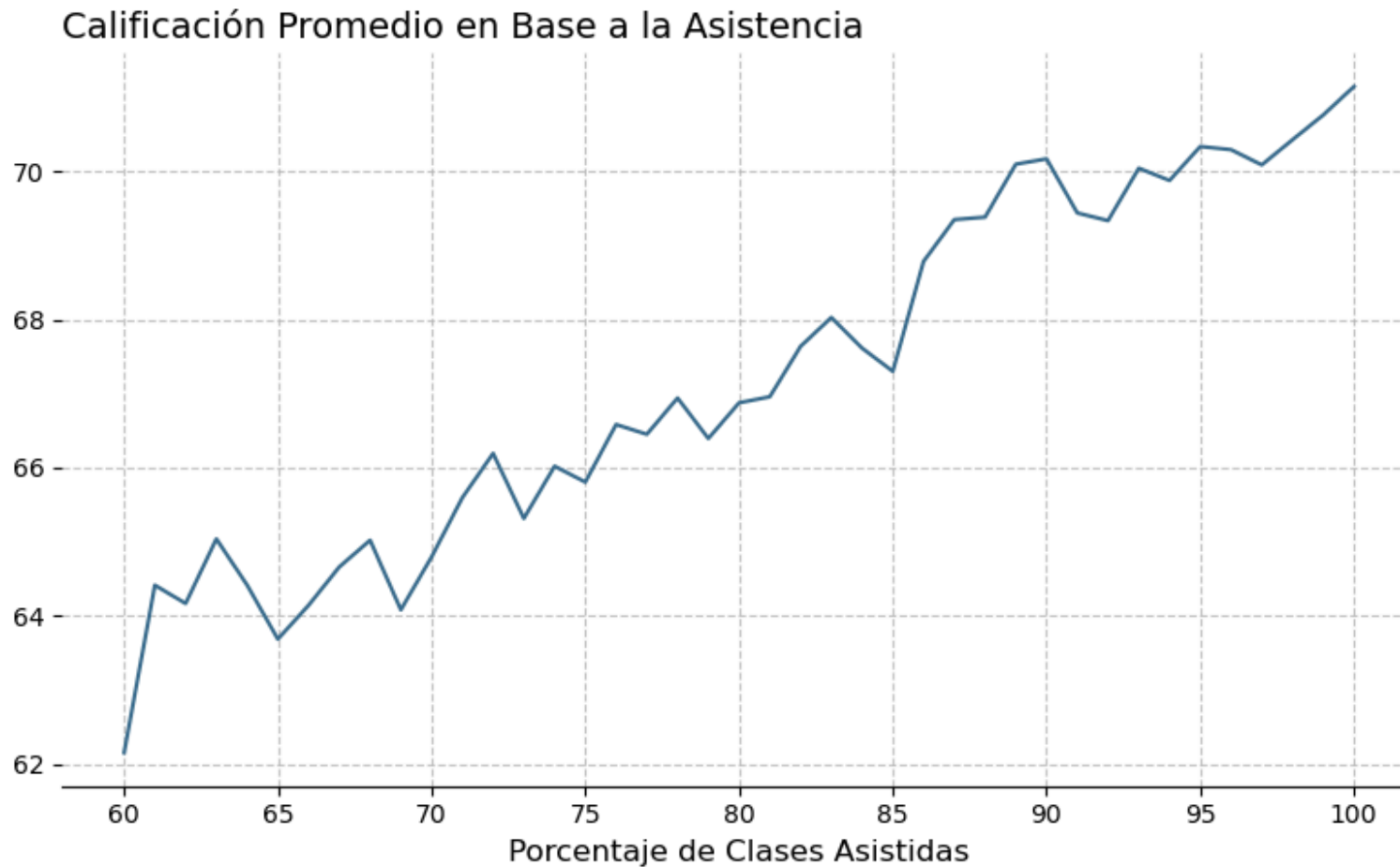
Calificación del Examen	
Género	
Femenino	67.300119
Masculino	67.278731

Comienzo visualizando la relación entre la calificación promedio y porcentaje de las clases asistidas, dando cuenta que a una mayor concurrencia a las clases, una mayor nota en el examen.

```
In [214... #Agrupo Los datos y genero el gráfico
ax=(df[['Asistencia', 'Calificación del Examen']]
     .groupby('Asistencia')
     .mean('Calificación del Examen')
     .plot(color='#3A6D8C', figsize=(8,5), legend=False))
```

```
#Agrego algunas características como el título, la etiqueta del eje x y la cuadrícula
ax.set_title("Calificación Promedio en Base a la Asistencia", fontsize=14, loc='left')
ax.set_xlabel("Porcentaje de Clases Asistidas", fontsize=12)
ax.spines['top'].set_visible(False)
ax.spines['left'].set_visible(False)
ax.spines['right'].set_visible(False)
ax.grid(True, linestyle='--', alpha=0.7)

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



Continuamos analizando cómo afecta a las calificaciones la participación de los padres en la educación de los estudiantes. Aunque la diferencia sea poca, siempre es importante el involucramiento de los padres.

In [216...

```
#Agrupo y ordeno los datos
df_agrupada = (df[['Participación Parental', 'Calificación del Examen']]
               .groupby('Participación Parental')
               .mean()
               .sort_values(by='Calificación del Examen'))

#Defino los colores
colores = ['#B43F3F', '#EAD8B1', '#3A6D8C']

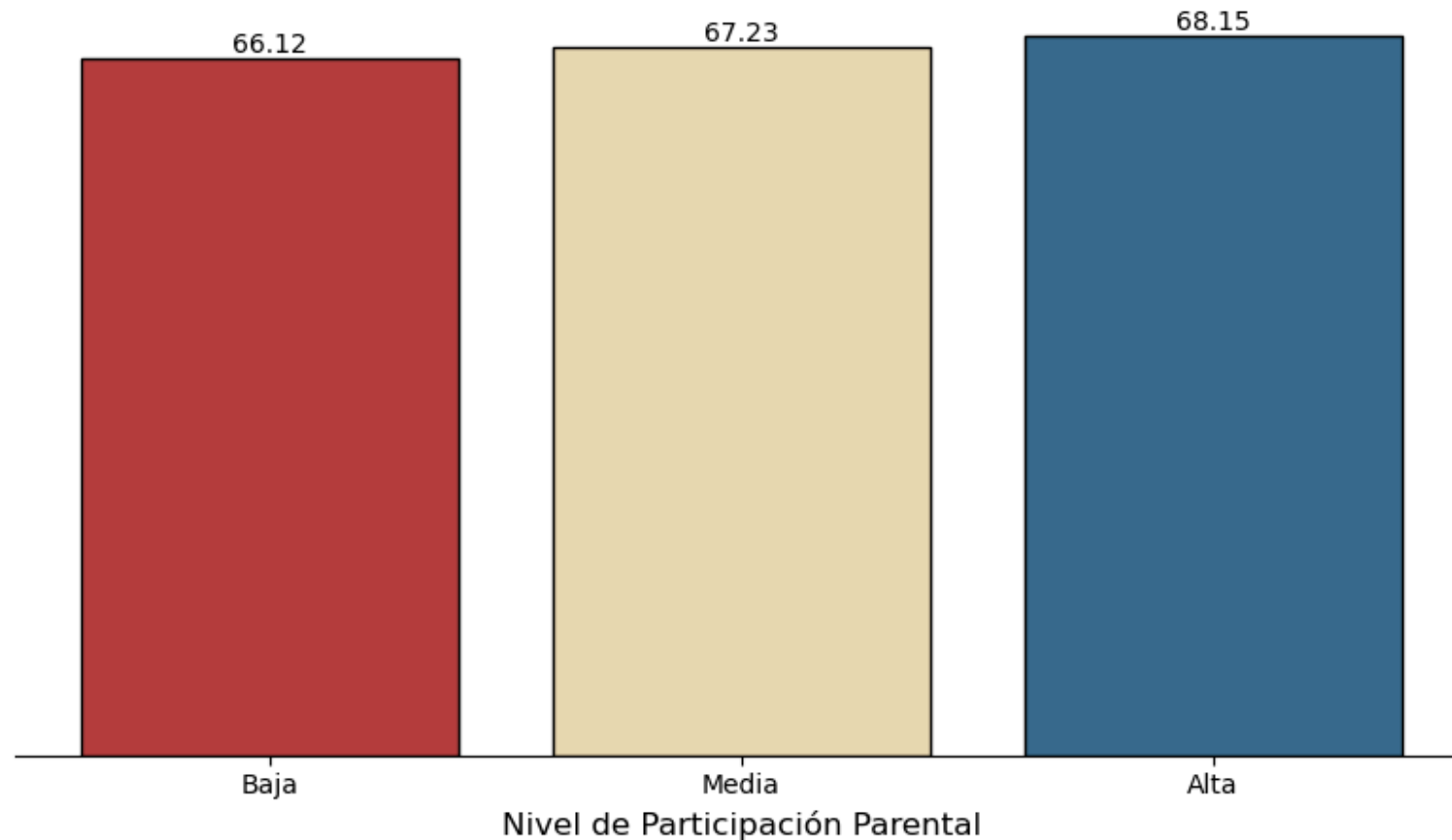
#Creo la figura
fig, ax = plt.subplots(figsize=(8, 5))

#Gráfico manualmente cada barra
for i, (index, row) in enumerate(df_agrupada.iterrows()):
    ax.bar(index, row['Calificación del Examen'], color=colores[i], edgecolor='black')

#modifico características del gráfico
ax.set_title("Calificación Promedio en Base a Participación Parental", fontweight='bold', fontsize=14, loc='left')
ax.set_xlabel("Nivel de Participación Parental", fontsize=12)
ax.spines['top'].set_visible(False)
ax.spines['left'].set_visible(False)
ax.spines['right'].set_visible(False)
ax.yaxis.set_ticks([])
ax.set_xticks(range(len(df_agrupada)))
ax.set_xticklabels(df_agrupada.index, rotation=0)

#Añado los valores encima de las barras
for p in ax.patches:
    ax.annotate(
        f"{p.get_height():.2f}",
        (p.get_x() + p.get_width() / 2, p.get_height()),
        ha='center',
        va='bottom',
        fontsize=10
    )
plt.tight_layout()
plt.show()
```

Calificación Promedio en Base a Participación Parental



Ahora visualizamos cómo se ven afectados los estudiantes respecto del acceso a recursos educativos. Aunque se observe una breve diferencia entre aquellos que poseen un bajo acceso en relación a los que poseen uno alto, es importante garantizar a todos los estudiantes un acceso igualitario a algo tan importante como los recursos educativos.

In [218...

```
#Grupo y ordeno Los datos
data_agrupada = (df[['Acceso a Recursos', 'Calificación del Examen']])
                .groupby('Acceso a Recursos')
                .mean('Calificación del Examen')
                .sort_values(by='Calificación del Examen'))

#Defino Los colores
colores=['#B43F3F', '#EAD8B1', '#3A6D8C']
```

```

#Creo La figura
fig, ax = plt.subplots(figsize=(8,5))

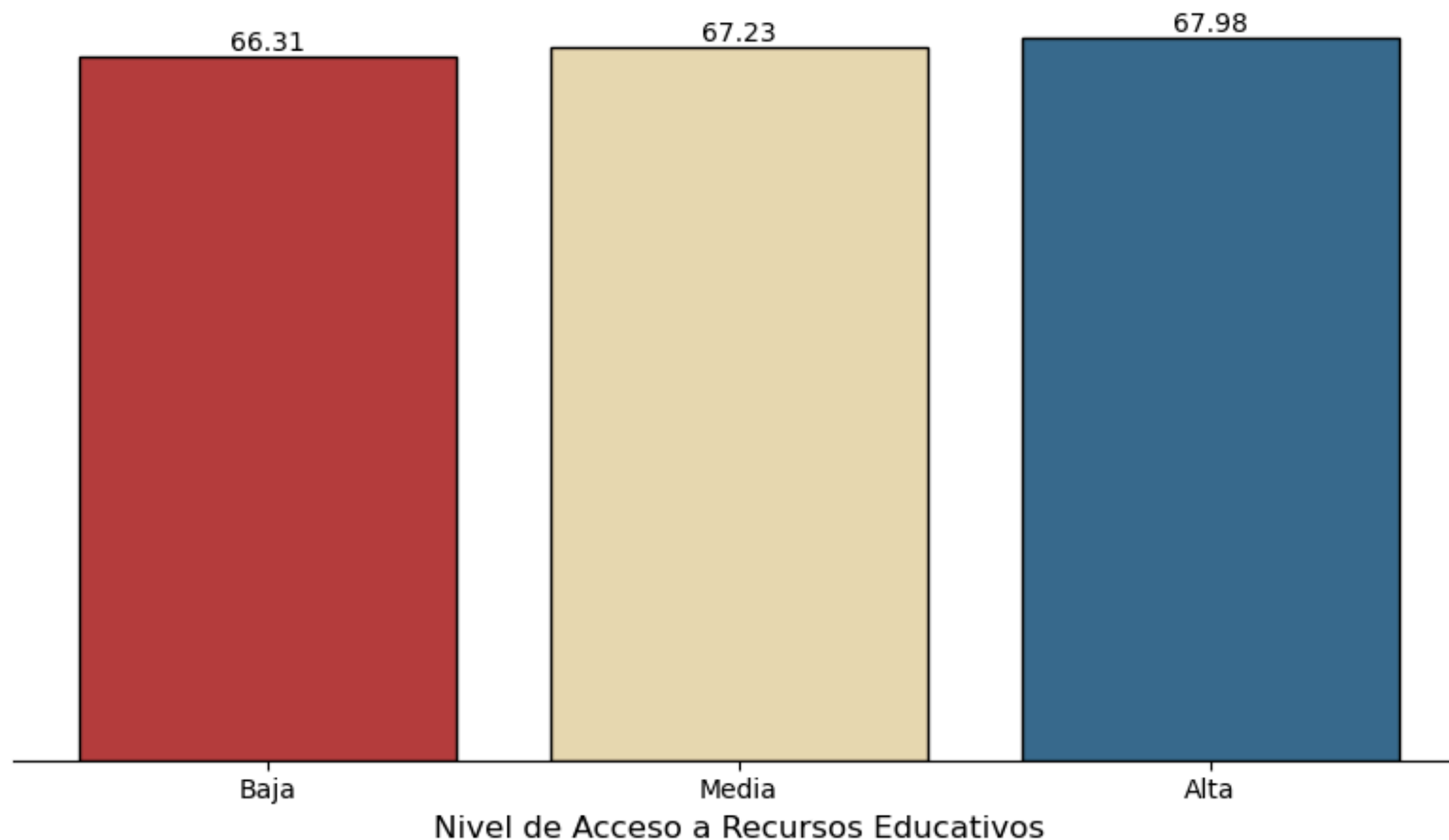
#Gráfico manualmente cada barra
for i, (index,row) in enumerate(data_agrupada.iterrows()):
    ax.bar(index, row['Calificación del Examen'], color=colores[i], edgecolor='black')

#modifico características del gráfico
ax.set_title("Calificación Promedio en base al Acceso a Recursos Educativos", fontsize=14, fontweight='bold', loc='left')
ax.set_xlabel("Nivel de Acceso a Recursos Educativos", fontsize=12)
ax.spines['top'].set_visible(False)
ax.spines['left'].set_visible(False)
ax.spines['right'].set_visible(False)
ax.yaxis.set_ticks([])
ax.set_xticks(range(len(data_agrupada)))
ax.set_xticklabels(data_agrupada.index, rotation=0)

#Añado Los valores encima de Las barras
for p in ax.patches:
    ax.annotate(
        f"{p.get_height():.2f}",
        (p.get_x() + p.get_width() / 2, p.get_height()),
        ha="center",
        va="bottom",
        fontsize=10
    )
plt.tight_layout()
plt.show()

```

Calificación Promedio en base al Acceso a Recursos Educativos



En este gráfico observamos que aquellos estudiantes que participan en actividades extracurriculares obtienen una mayor calificación promedio en el examen, pero es una diferencia menor a un punto, por lo que podría ser una variable no tan importante para la calificación de los exámenes.

In [220...

```
#Defino los datos y creo el gráfico
ax = (df[['Actividades Extracurriculares', 'Calificación del Examen']]
      .groupby('Actividades Extracurriculares')
      .mean('Calificación del Examen')
      .plot.bar(color='#3A6D8C', edgecolor='black', figsize=(8,5), legend=False)
      )

#modifico características del gráfico
ax.set_title("Calificación Promedio en Base a la Realización de Actividades Extracurriculares", fontsize=12, fontweight='bold', loc='left')
ax.set_xlabel("Participación en Actividades Extracurriculares", fontsize=12)
```

```

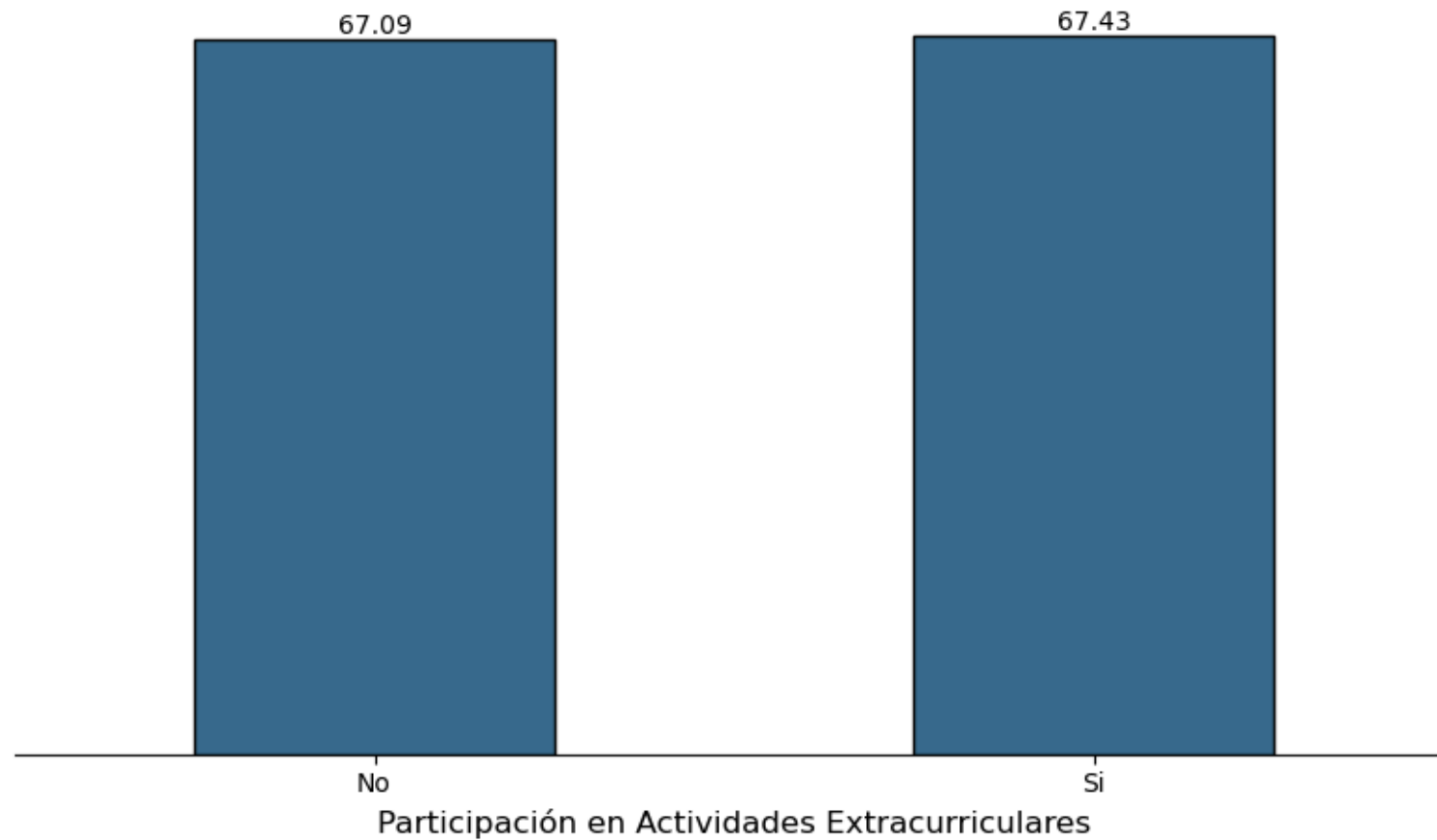
ax.spines['top'].set_visible(False)
ax.spines['left'].set_visible(False)
ax.spines['right'].set_visible(False)
ax.yaxis.set_ticks([])
ax.set_xticklabels(ax.get_xticklabels(), rotation=0)

#Le coloco a cada barra el valor de su dato
for p in ax.patches:
    ax.annotate(
        f"{p.get_height():.2f}",
        (p.get_x() + p.get_width()/2, p.get_height()),
        ha="center",
        va="bottom",
        fontsize=10
    )

plt.tight_layout()
plt.show()

```

Calificación Promedio en Base a la Realización de Actividades Extracurriculares



Siguiendo con la variable de la motivación se observa aquellos estudiantes con un nivel de motivación más alto obtienen una calificación promedio mayor respecto de los que tienen una motivación menor.

In [222...

```
#Agrupo y ordeno Los datos
df_agrupado = (df[['Nivel Motivación', 'Calificación del Examen']]
               .groupby('Nivel Motivación')
               .mean('Calificación del Examen')
               .sort_values(by='Calificación del Examen'))

#defino los colores
colores=['#B43F3F', '#EAD8B1', '#3A6D8C']

#creo la figura
fig, ax = plt.subplots(figsize=(8,5))

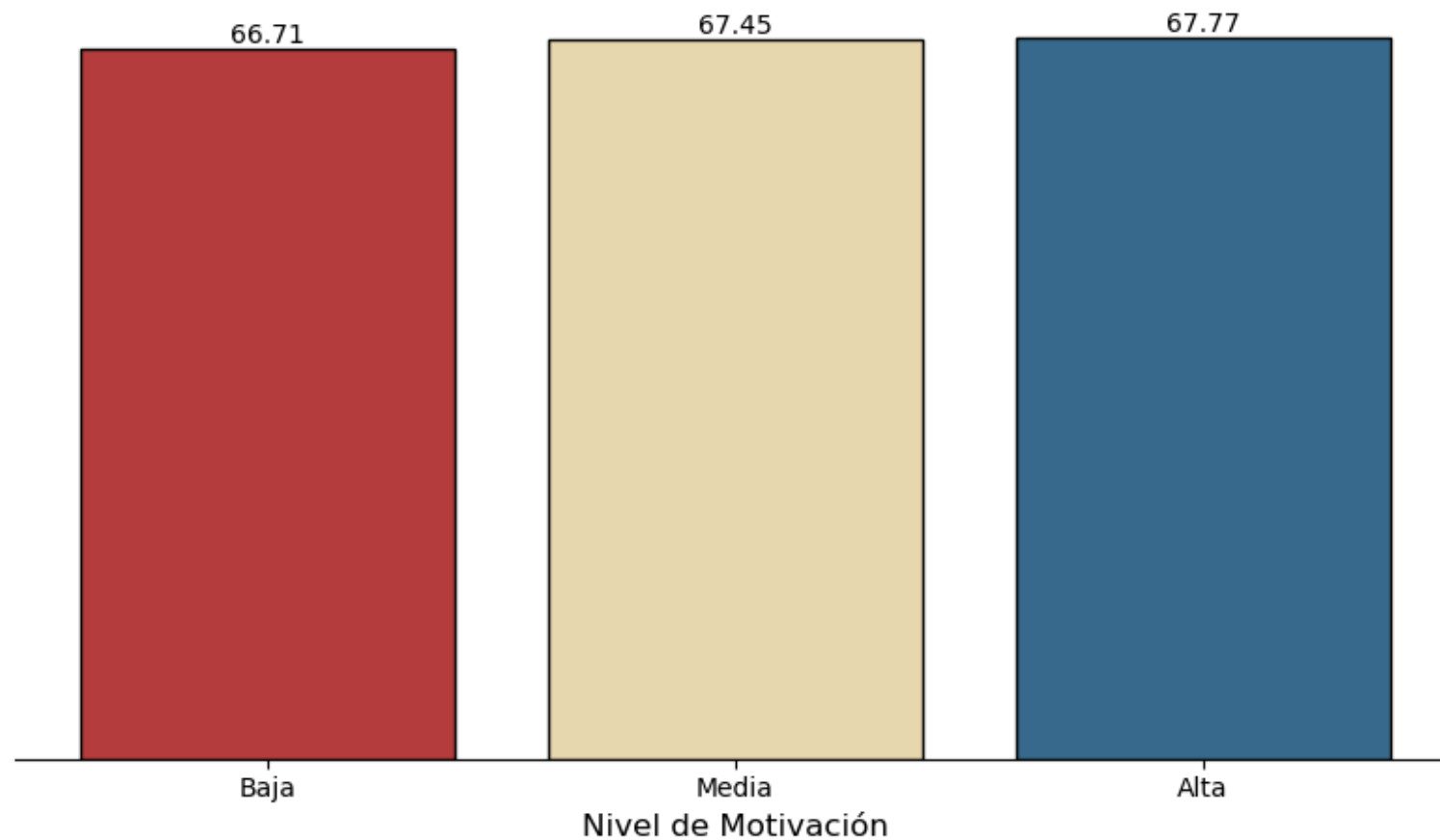
#Gráfico manualmente cada barra
for i, (index, row) in enumerate(df_agrupado.iterrows()):
    ax.bar(index, row['Calificación del Examen'], color=colores[i], edgecolor='black')

#modifico características del gráfico
ax.set_title("Calificación Promedio en Base al Nivel de Motivación", fontsize=12, fontweight='bold', loc='left')
ax.set_xlabel("Nivel de Motivación", fontsize=12)
ax.spines['top'].set_visible(False)
ax.spines['left'].set_visible(False)
ax.spines['right'].set_visible(False)
ax.yaxis.set_ticks([])
ax.set_xticks(range(len(df_agrupado)))
ax.set_xticklabels(df_agrupado.index, rotation=0)

#Le coloco a cada barra el valor de su dato
for p in ax.patches:
    ax.annotate(
        f"{p.get_height():.2f}",
        (p.get_x() + p.get_width()/2, p.get_height()),
        ha="center",
        va="bottom",
        fontsize=10
    )

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

Calificación Promedio en Base al Nivel de Motivación



Acá se observa que aquellos estudiantes que tienen acceso a internet tienen una calificación del examen levemente mayor a la de los que no poseen acceso a internet.

```
In [224... #Defino los datos y creo el gráfico
ax = (df[['Acceso Internet', 'Calificación del Examen']]
      .groupby('Acceso Internet')
      .mean('Calificación del Examen')
      .plot.bar(color='#3A6D8C', edgecolor='black', figsize=(8,5), legend=False)
      )

#Modifico características del gráfico
ax.set_title("Calificación Promedio en Base al Acceso a Internet", fontsize=14, fontweight='bold', loc='left')
ax.set_xlabel("Acceso a Internet", fontsize=12)
ax.spines['top'].set_visible(False)
```

```

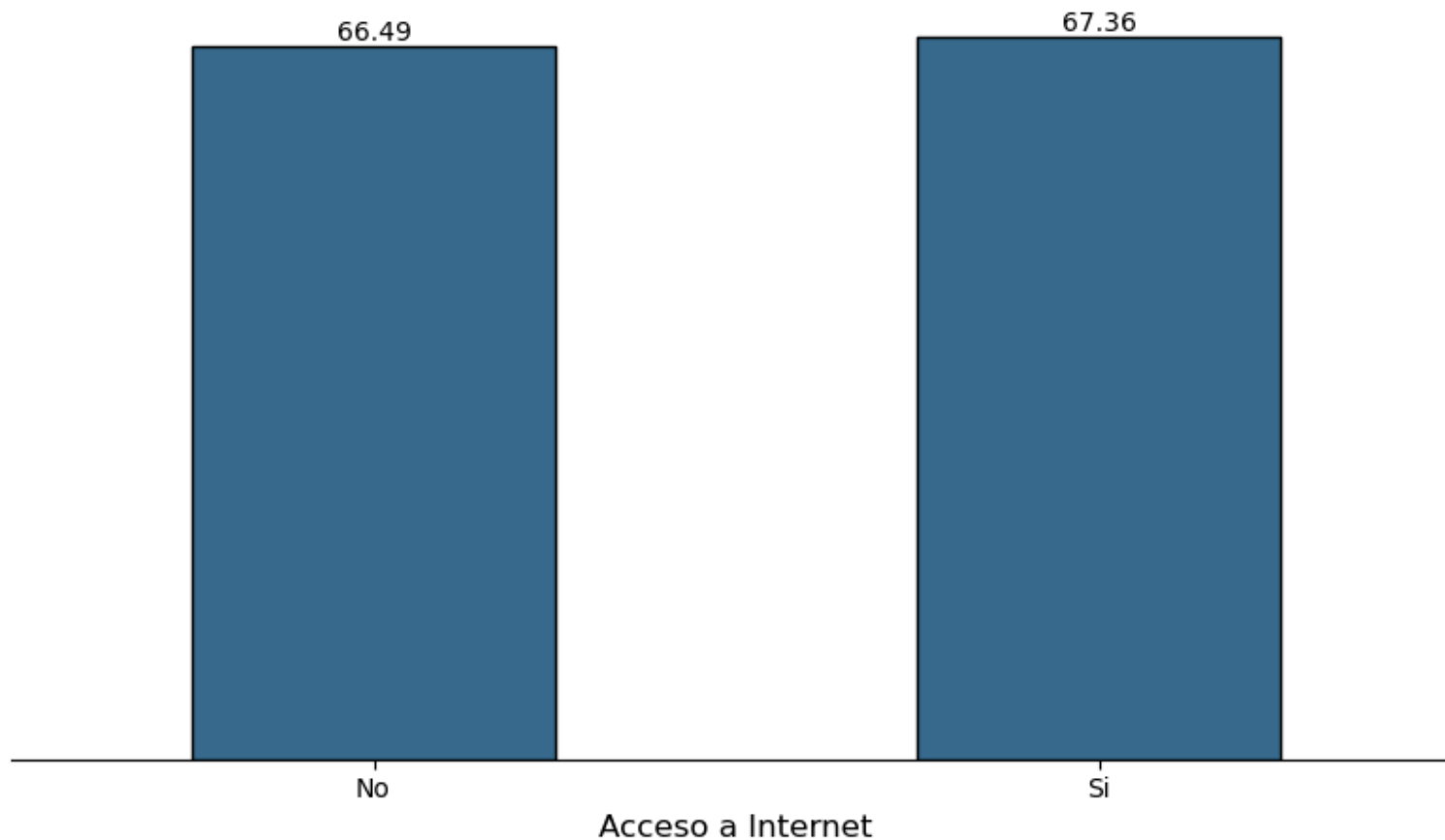
ax.spines['left'].set_visible(False)
ax.spines['right'].set_visible(False)
ax.yaxis.set_ticks([])
ax.set_xticklabels(ax.get_xticklabels(), rotation=0)

#Le coloco a cada barra el valor de su dato
for p in ax.patches:
    ax.annotate(
        f"{p.get_height():.2f}",
        (p.get_x() + p.get_width()/2, p.get_height()),
        ha="center",
        va="bottom",
        fontsize=10
    )

plt.tight_layout()
plt.show()

```

Calificación Promedio en Base al Acceso a Internet



Siguiendo por la asistencia a tutorías podemos observar que los estudiantes que asisten a una mayor cantidad de tutorías mensuales obtienen mejores calificaciones en sus exámenes.

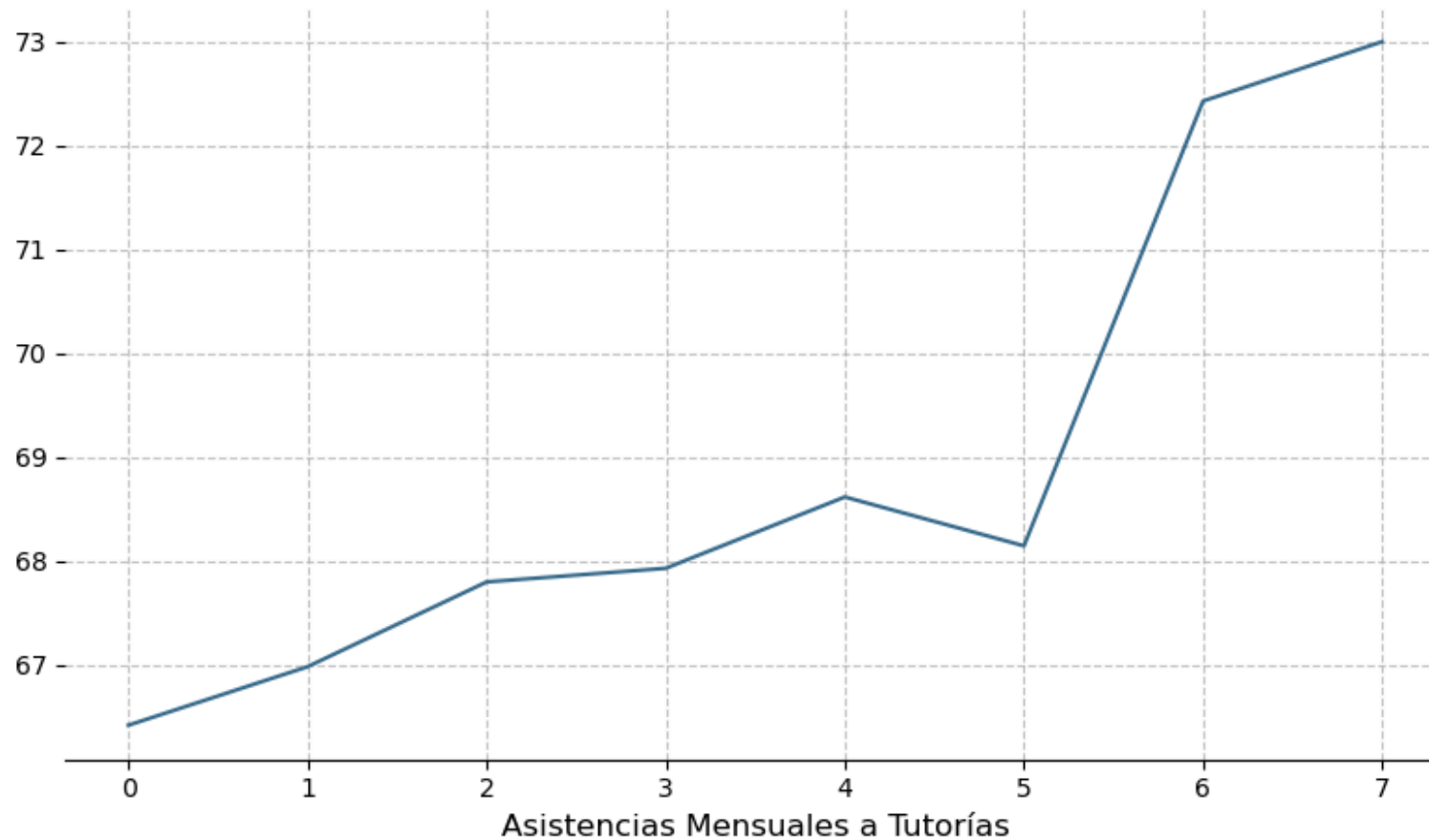
In [194...

```
#Defino los datos y creo el gráfico
ax = (df[['Tutorías','Calificación del Examen']]
      .groupby('Tutorías')
      .mean('Calificación del Examen')
      .plot(color='#3A6D8C', legend=False, figsize=(8,5))
      )

#Modifico características del gráfico
ax.set_title("Calificación Promedio en Base a las Tutorías Asistidas Por Mes", fontsize=12, fontweight='bold')
ax.set_xlabel("Asistencias Mensuales a Tutorías", fontsize=12)
ax.spines['top'].set_visible(False)
ax.spines['left'].set_visible(False)
ax.spines['right'].set_visible(False)
ax.grid(True, linestyle="--", alpha=0.7)

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

Calificación Promedio en Base a las Tutorías Asistidas Por Mes



Analizamos cómo repercute la calidad de los profesores en las notas de los estudiantes. Pero es una variable que en este caso no tendría en cuenta, ya que la calidad del profesor es subjetiva para cada estudiante particular.

```
In [226... #dado que había registros donde calidad profesor no tenía valores, aplico un filtro que los excluya
#Agrupo y ordeno los datos
df_agrupado = (df[df['Calidad Profesor'] != ''][['Calidad Profesor', 'Calificación del Examen']]
               .groupby('Calidad Profesor')
               .mean('Calificación del Examen')
               .sort_values(by='Calificación del Examen'))

#defino los colores
colores=['#B43F3F', '#EAD8B1', '#3A6D8C']

#Creo la figura
```

```

fig, ax = plt.subplots(figsize=(8,5))

#Gráfico manualmente cada barra
for i, (index, row) in enumerate(df_agrupado.iterrows()):
    ax.bar(index, row['Calificación del Examen'], color=colores[i], edgecolor='black')

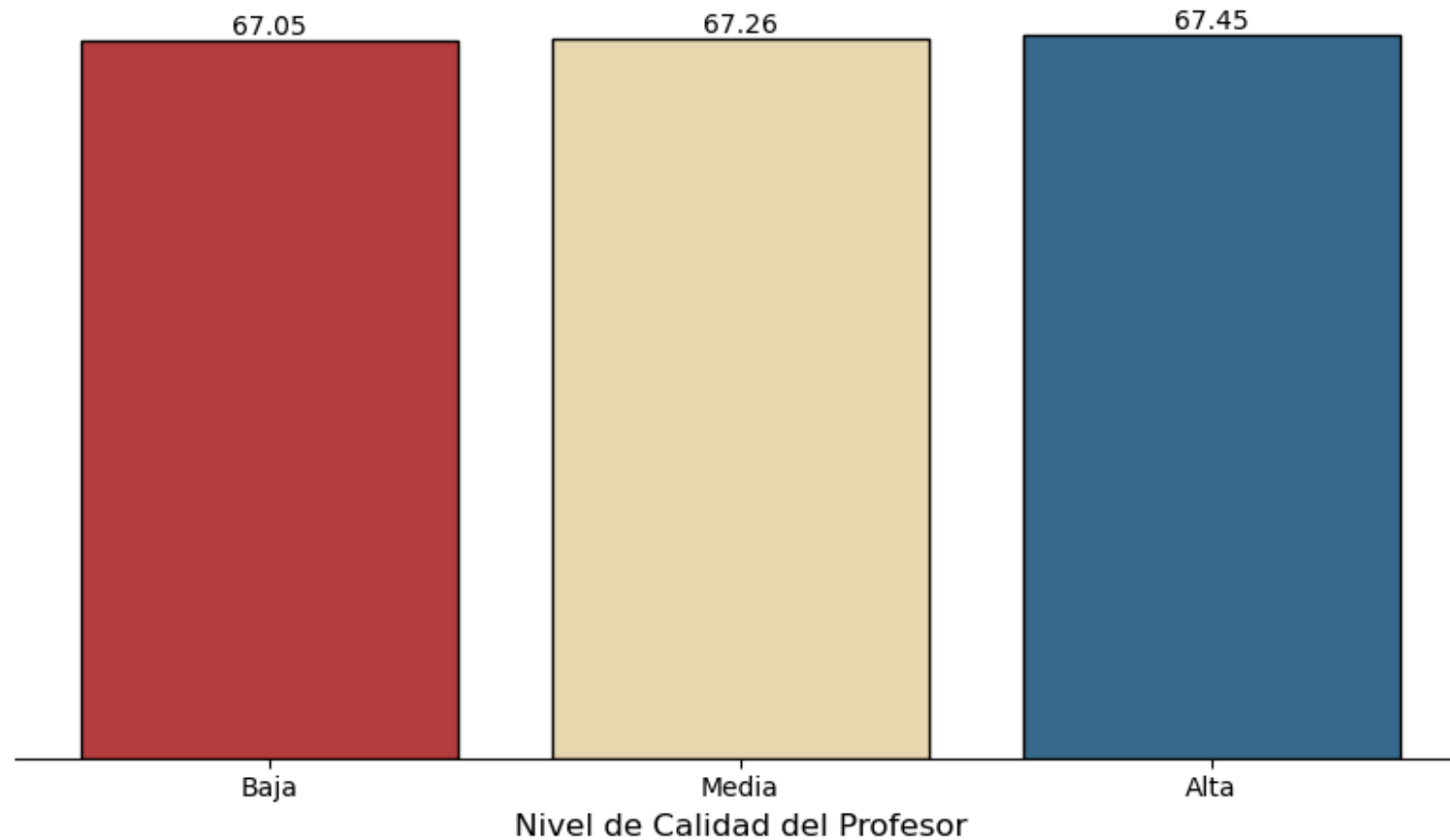
#Modifico características del gráfico
ax.set_title("Calificación Promedio en Base a la Calidad del Profesor", fontsize=14, fontweight='bold', loc='left')
ax.set_xlabel("Nivel de Calidad del Profesor", fontsize=12)
ax.spines['top'].set_visible(False)
ax.spines['left'].set_visible(False)
ax.spines['right'].set_visible(False)
ax.yaxis.set_ticks([])
ax.set_xticks(range(len(df_agrupado)))
ax.set_xticklabels(df_agrupado.index, rotation=0)

#Le coloco a cada barra el valor de su dato
for p in ax.patches:
    ax.annotate(
        f"{p.get_height():.2f}",
        (p.get_x() + p.get_width()/2, p.get_height()),
        ha="center",
        va="bottom",
        fontsize=10
    )

plt.tight_layout()
plt.show()

```

Calificación Promedio en Base a la Calidad del Profesor



Seguimos analizando una variable interesante: las horas de actividad física semanal.

En este caso parece favorecer a la calificación promedio la realización de actividad física semanalmente.

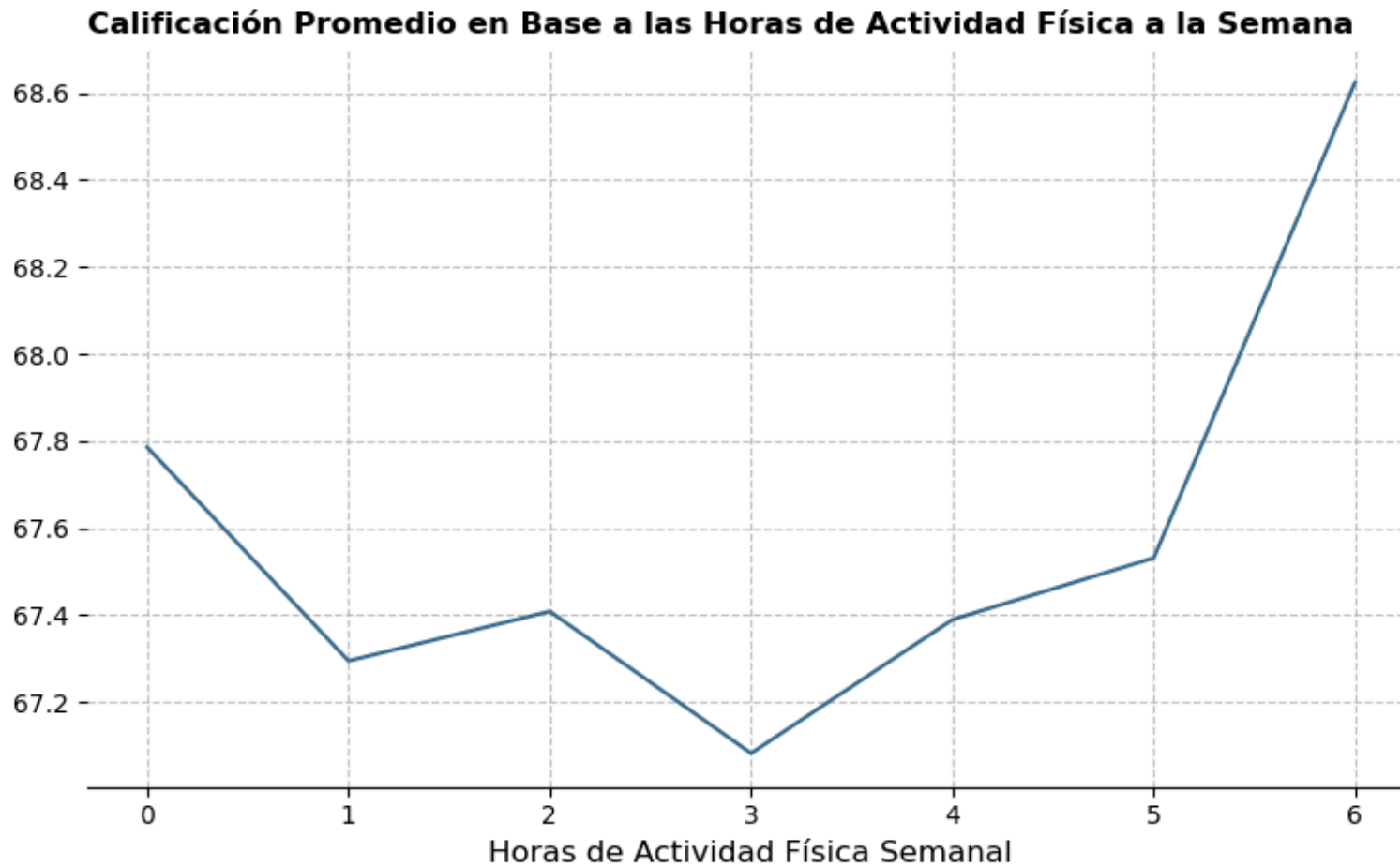
In [190...

```
#Defino los datos y creo el gráfico
ax = (df[['Actividad Física', 'Calificación del Examen']]
      .groupby('Actividad Física')
      .mean('Calificación del Examen')
      .plot(color='#3A6D8C', figsize=(8,5), legend=False)
)

#Modifico características del gráfico
ax.set_title("Calificación Promedio en Base a las Horas de Actividad Física a la Semana", fontsize=12, fontweight='bold', loc='left')
ax.set_xlabel("Horas de Actividad Física Semanal", fontsize=12)
```

```
ax.spines['top'].set_visible(False)
ax.spines['left'].set_visible(False)
ax.spines['right'].set_visible(False)
ax.grid(True, linestyle="--", alpha=0.7)

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



Continuando con la presencia de discapacidades de aprendizaje se observa que aquellos estudiantes que poseen una discapacidad de aprendizaje obtienen en promedio una nota menor en relación a aquellos que no poseen ninguna.

In [228...

```
#Defino Los datos y creo el gráfico
ax = (df[['Discapacidades de Aprendizaje', 'Calificación del Examen']]
      .groupby('Discapacidades de Aprendizaje'))
```



```

        .mean('Calificación del Examen')
    .plot.bar(color='#3A6D8C', edgecolor='black', figsize=(8,5), legend=False)
)

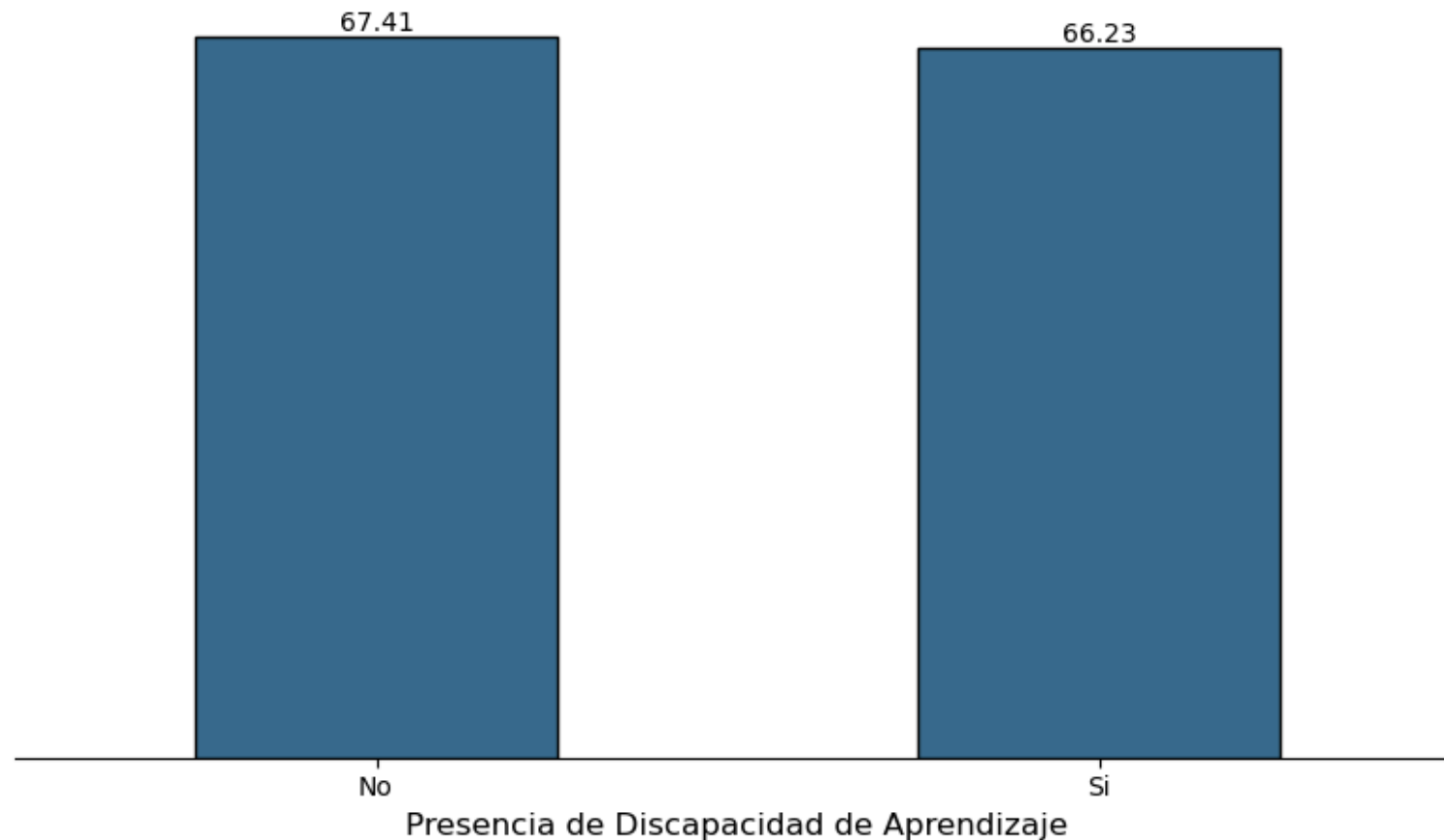
#Modifico características del gráfico
ax.set_title("Calificación Promedio en Base a la Presencia de Discapacidades de Aprendizaje", fontsize=12, fontweight='bold', loc='left')
ax.set_xlabel("Presencia de Discapacidad de Aprendizaje", fontsize=12)
ax.spines['top'].set_visible(False)
ax.spines['left'].set_visible(False)
ax.spines['right'].set_visible(False)
ax.yaxis.set_ticks([])
ax.set_xticklabels(ax.get_xticklabels(),rotation=0)

#Le coloco a cada barra el valor de su dato
for p in ax.patches:
    ax.annotate(
        f"{p.get_height():.2f}",
        (p.get_x() + p.get_width()/2,p.get_height()),
        ha="center",
        va="bottom",
        fontsize=10
    )

plt.tight_layout()
plt.show()

```

Calificación Promedio en Base a la Presencia de Discapacidades de Aprendizaje



Por último, veremos cómo afectan algunas variables a la motivación de los estudiantes.

Empezando con la concurrencia a actividades extracurriculares, podemos observar que los niveles de motivación son más altos para los estudiantes que acceden a actividades fuera del aula.

In [232...

```
#Defino y agrupo los datos
data_agrupada = (df[['Actividades Extracurriculares', 'Nivel Motivación']]
                 .groupby(['Actividades Extracurriculares', 'Nivel Motivación'])
                 .size()
                 .unstack(fill_value=0))

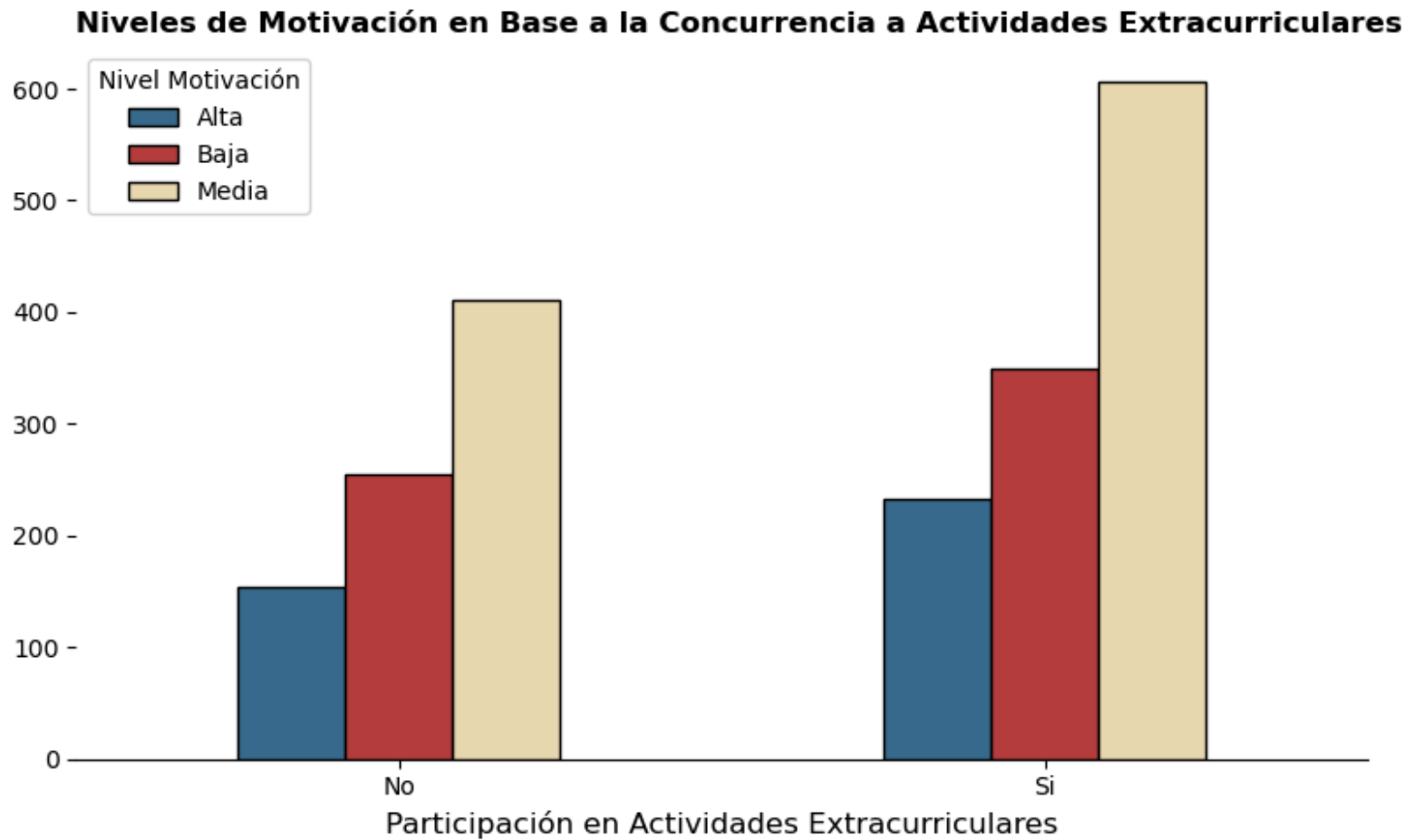
#defino los colores
colores = ['#3A6D8C', '#B43F3F', '#EAD8B1']

#creo el gráfico
```

```
ax = data_agrupada.plot.bar(stacked=False, color=colores, edgecolor='black', figsize=(8,5,))

#Modifico características del gráfico
ax.set_title("Niveles de Motivación en Base a la Concurrencia a Actividades Extracurriculares", fontsize=12, fontweight='bold', loc='left')
ax.set_xlabel("Participación en Actividades Extracurriculares", fontsize=12)
ax.spines['top'].set_visible(False)
ax.spines['left'].set_visible(False)
ax.spines['right'].set_visible(False)
ax.set_xticklabels(ax.get_xticklabels(), rotation=0)

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



Casi terminando con el análisis vemos cómo afecta la participación de los padres en la motivación de sus hijos a la hora del estudio.

Se observa que la motivación más alta se encuentra cuando los padres tienen una participación media, lo cual puede indicar que los estudiantes se sienten más cómodos con este nivel de involucramiento.

In [178...

```
#Defino y agrupo los datos
data_agrupada = (df[['Participación Parental', 'Nivel Motivación']]
                  .groupby(['Participación Parental', 'Nivel Motivación'])
                  .size()
                  .unstack(fill_value=0))

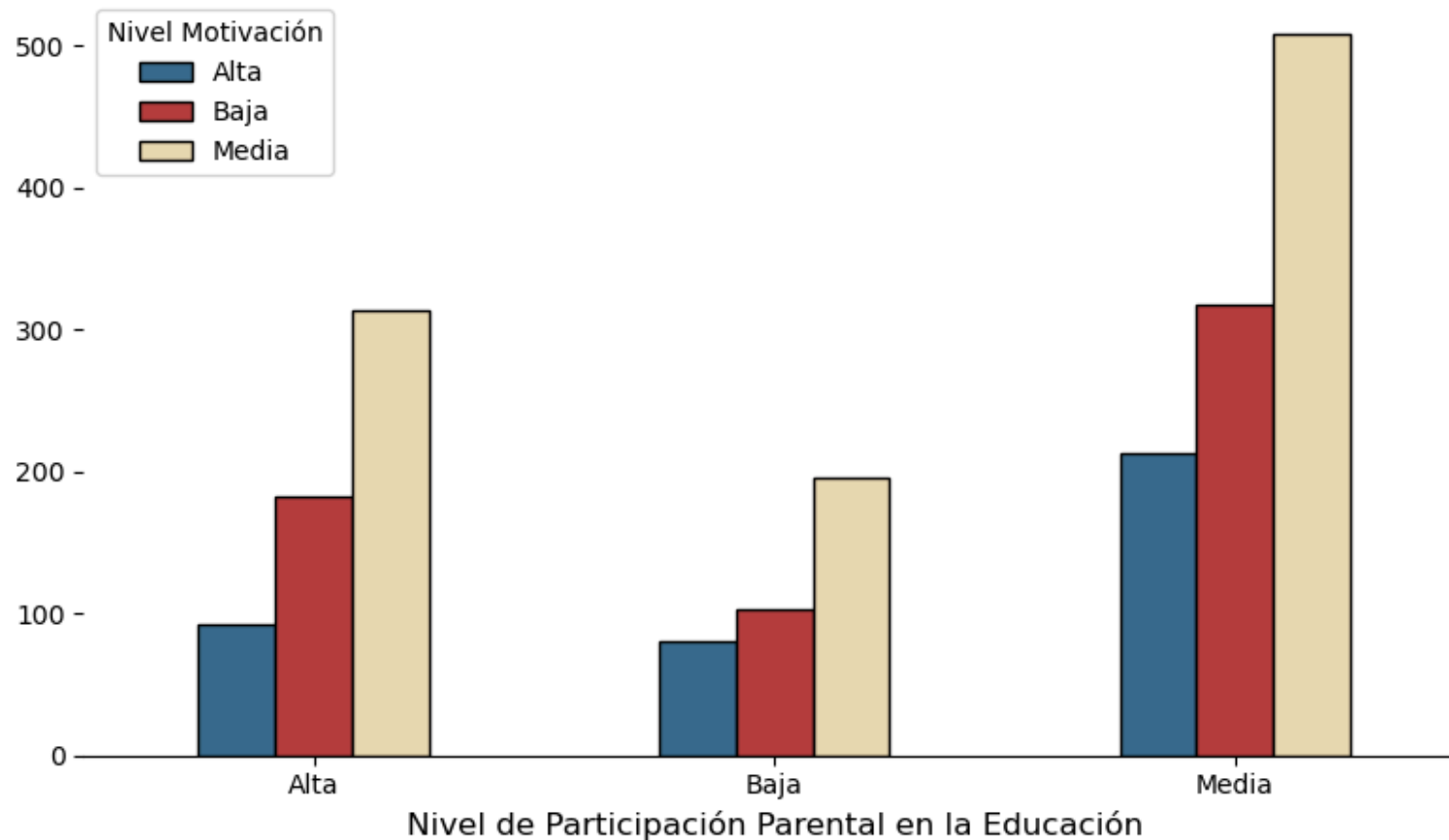
#defino los colores
colores = ['#3A6D8C', '#B43F3F', '#EAD8B1']

#creo el gráfico
ax = data_agrupada.plot.bar(stacked=False, color=colores, edgecolor='black', figsize=(8,5))

#Modifico características del gráfico
ax.set_title("Niveles de Motivación en Base a la Participación Parental en la Educación", fontsize=12, fontweight='bold', loc='left')
ax.set_xlabel("Nivel de Participación Parental en la Educación", fontsize=12)
ax.spines['top'].set_visible(False)
ax.spines['left'].set_visible(False)
ax.spines['right'].set_visible(False)
ax.set_xticklabels(ax.get_xticklabels(), rotation=0)

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

Niveles de Motivación en Base a la Participación Parental en la Educación



Para finalizar analizamos el impacto de la disponibilidad de recursos educativos en la educación en la motivación de los estudiantes.

Podemos observar que en este caso la motivación tiene sus niveles más altos cuando la variable se encuentra en la media. Por lo que puede no ser un factor determinante de la motivación.

In [180...

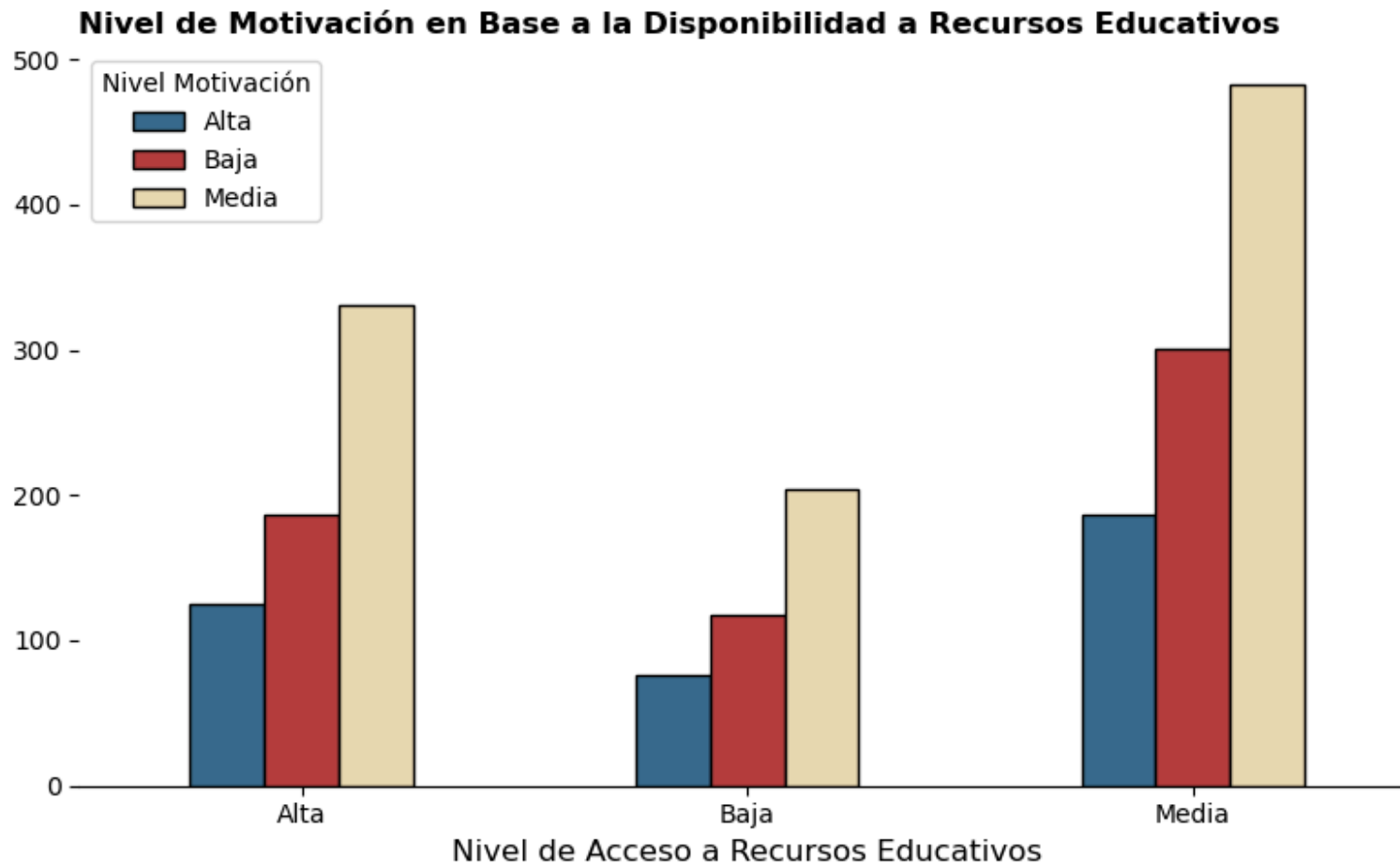
```
#Defino y agrupo los datos
data_agrupada = (df[['Acceso a Recursos', 'Nivel Motivación']]
                 .groupby(['Acceso a Recursos', 'Nivel Motivación'])
                 .size()
                 .unstack(fill_value=0))

#defino los colores
colores = ['#3A6D8C', '#B43F3F', '#EAD8B1']
```

```
#creo el gráfico
ax = data_agrupada.plot.bar(stacked=False, color=colores, edgecolor='black', figsize=(8,5))

#Modifico características del gráfico
ax.set_title("Nivel de Motivación en Base a la Disponibilidad a Recursos Educativos", fontsize=12, fontweight='bold', loc='left')
ax.set_xlabel("Nivel de Acceso a Recursos Educativos", fontsize=12)
ax.spines['top'].set_visible(False)
ax.spines['left'].set_visible(False)
ax.spines['right'].set_visible(False)
ax.set_xticklabels(ax.get_xticklabels(), rotation=0)

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



Conclusiones:

Terminado el análisis realizo las siguientes recomendaciones finales:

Mejorar la accesibilidad a recursos educativos dentro de la institución:

- Ampliar la oferta de información literaria e implementar un sistema para que los estudiantes puedan llevarse libros prestados.
- Garantizar tanto el acceso a internet y computadoras dentro de la institución en espacios de estudio.

Promover un sistema de tutorías:

- Implementar un sistema de tutorías para los estudiantes que necesiten una ayuda extra en alguna materia.
- Acompañar a los estudiantes que posean problemas de aprendizaje y realizar seguimientos personalizados.

Ampliar la oferta de actividad física:

- Realizar campañas para motivar a los estudiantes a realizar actividad física poniendo foco en los beneficios que se pueden obtener a nivel académico gracias al ejercicio.
- Crear espacios recreativos donde los estudiantes puedan ejercitar y realizar actividades físicas en un entorno seguro.

Crear una oferta de actividades extracurriculares:

- Implementar una gama de actividades fuera de las aulas como teatro, ajedrez, música, arte, en donde los estudiantes puedan desarrollar sus habilidades y mejorar su motivación.
- Ver la posibilidad de incluir un sistema de recompensas para aquellos estudiantes que destaquen en sus actividades extracurriculares para incentivarlos a esforzarse.