ITSRLL
INSTITUTO TECNOLÓGICO SUPERIOR DE LA REGIÓN DE LOS LLANOS

Ingeniería Mecatrónica

PROGRAMACIÓN AVANZADA

Enero – Junio 2025

M.C. Osbaldo Aragón Banderas

Competencia:

1	2	3	4	5

Actividad número: 4

Nombre de actividad:

U2A4. (10%) NOOTEBOOK: Análisis de Datos Aplicables a Regresión Lineal Simple

Actividad realizada por:

Muñiz Galvan Bryam

21030021

Guadalupe Victoria, Durango

Fecha de entrega:

09	03	2025

NOOTEBOOK:

Análisis de Datos Aplicables a Regresión Lineal Simple

Objetivo

El propósito de esta actividad es que los estudiantes busquen, seleccionen y analicen un conjunto de datos en Kaggle u otra fuente confiable para aplicar regresión lineal simple y predecir una variable continua con base en una única característica independiente. Además, deberán presentar los resultados y conclusiones obtenidas, explicando la relación entre las variables y la interpretación del modelo.

Regresión Lineal Simple

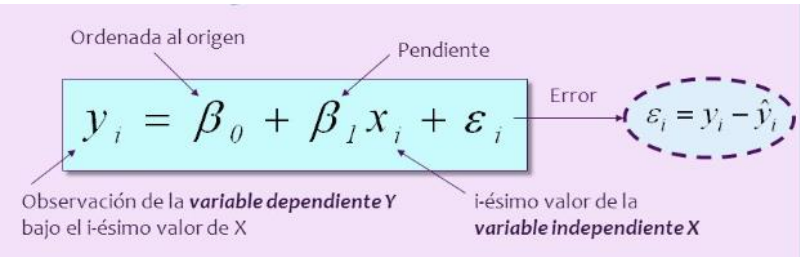
La regresión lineal simple es un modelo estadístico que se utiliza para predecir el valor de una variable dependiente (también llamada variable de respuesta) a partir de una sola variable independiente (también llamada variable predictora). Esta técnica asume que existe una relación lineal entre ambas variables, lo que significa que los datos pueden representarse con una línea recta en un gráfico de dispersión.

Aplicación en problemas de predicción

En términos prácticos, la regresión lineal simple se usa cuando queremos predecir un valor basado en una sola característica. Por ejemplo, si se quiere predecir el precio de una casa basándose en su tamaño (en metros cuadrados), podemos aplicar la regresión lineal simple para encontrar una relación entre ambas variables y luego predecir el precio de una casa con un tamaño dado.

Ecuación matemática

La ecuación matemática de la regresión lineal simple es la siguiente:



Donde:

- y es la variable dependiente (lo que queremos predecir).
- x es la variable independiente (la característica que usamos para hacer la predicción).
- β_0 es el intercepto (el valor de y cuando x es igual a cero).
- β_1 es el pendiente (indica el cambio en y por cada unidad que cambia x).
- ε es el término de error (la diferencia entre el valor predicho por el modelo y el valor real observado).

Determinación de la mejor línea de ajuste: Método de Mínimos Cuadrados

El objetivo de la regresión lineal es encontrar los valores de β_0 y β_1 que mejor ajusten la línea a los datos, minimizando el error. Este error se mide como la diferencia entre los valores observados y los valores predichos por la línea de regresión.

El método de mínimos cuadrados busca minimizar la suma de los cuadrados de estas diferencias (errores). Matemáticamente, esto se expresa como:

$$S = \sum(y_i - (\beta_0 + \beta_1 x_i))^2$$

Donde y_i son los valores observados de la variable dependiente y x_i son los valores correspondientes de la variable independiente.

El proceso para encontrar los coeficientes β_0 y β_1 es el siguiente:

1. Derivamos la función de error con respecto a β_0 y β_1.
2. Igualamos las derivadas a cero para obtener los valores de β_0 y β_1 que minimizan el error.
3. Resolvemos estas ecuaciones para obtener los valores óptimos de β_0 y β_1.

La fórmula para $\boldsymbol{\beta_1}$ es:

$$\boldsymbol{\beta}^1 = \frac{\sum(x_i - \bar{x})(y_i - \bar{y})}{\sum(x_i - \bar{x})^2}$$

Y la fórmula para $\boldsymbol{\beta_0}$ es:

$$\boldsymbol{\beta}^0 = \bar{y} - \boldsymbol{\beta}_1 \bar{x}$$

Donde $\bar{x}$ y $\bar{y}$ son las medias de las variables x y y, respectivamente.

De esta manera, al aplicar el método de mínimos cuadrados, se obtiene la mejor línea recta que ajusta los datos, minimizando el error cuadrado y permitiendo hacer predicciones con base en nuevos valores de x.

Justificación de la Selección del Dataset para Regresión Lineal Simple

Elección de Variables

Para realizar un análisis de regresión lineal simple, necesitamos:

- Variable Independiente (Numérica): Hours Studied
- Variable Dependiente (Continua): Performance Index

La elección de este dataset es adecuada porque cumple con los criterios fundamentales para aplicar regresión lineal simple.

¿Por qué es adecuado para regresión lineal simple?

1. Relación Causal Potencial
- Es lógico suponer que el número de horas de estudio (Hours Studied) influye directamente en el desempeño académico (Performance Index).
- La relación entre estas variables puede modelarse con una línea recta, lo que es ideal para una regresión lineal.
2. Variable Dependiente Continua
- Performance Index es una variable continua que puede tomar valores en un rango determinado, lo cual es un requisito clave para la regresión lineal.
3. Fácil Interpretación
- Un modelo de regresión lineal simple permitirá responder preguntas como:
    - ¿Cuánto aumenta el rendimiento si se estudian más horas?
    - ¿Existe una correlación positiva entre estudiar más y un mejor desempeño?
4. Simplicidad y Eficiencia

- La regresión lineal simple es un modelo fácil de interpretar y eficiente computacionalmente.
- Ideal para análisis preliminares antes de probar modelos más complejos.

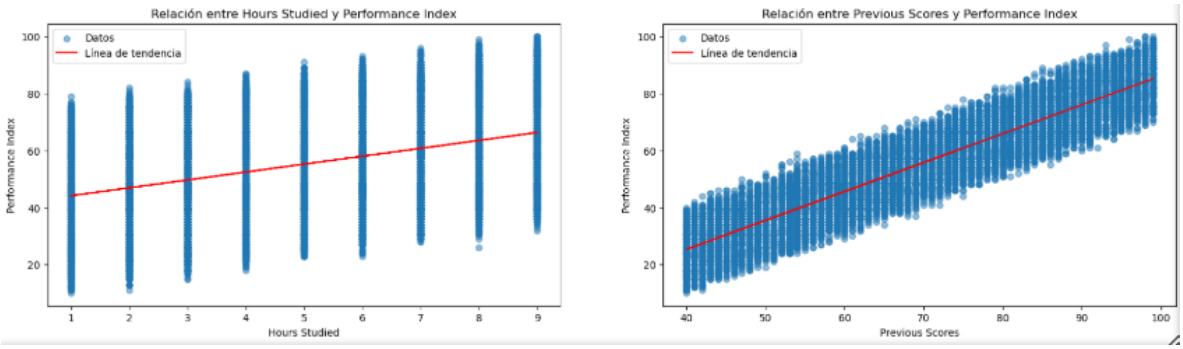
Posibles Aplicaciones

- Predecir el desempeño de un estudiante en función de sus horas de estudio.
- Identificar umbrales óptimos de horas de estudio para maximizar el rendimiento.
- Comparar el impacto del estudio frente a otras variables como el sueño o las actividades extracurriculares.

Consideraciones

- Se debe verificar que la relación entre Hours Studied y Performance Index sea lineal antes de aplicar regresión lineal.
- Es recomendable visualizar los datos con un gráfico de dispersión y calcular el coeficiente de correlación.
- Otros factores no considerados pueden influir en el desempeño (sesgo en los datos).

Visualización de relación entre variables de interés (Gráficos de dispersión)



Análisis de Regresión Lineal Simple

1. **Relación entre "Hours Studied" y "Performance Index":**

- La pendiente de la línea de tendencia sugiere una correlación positiva, aunque relativamente débil.

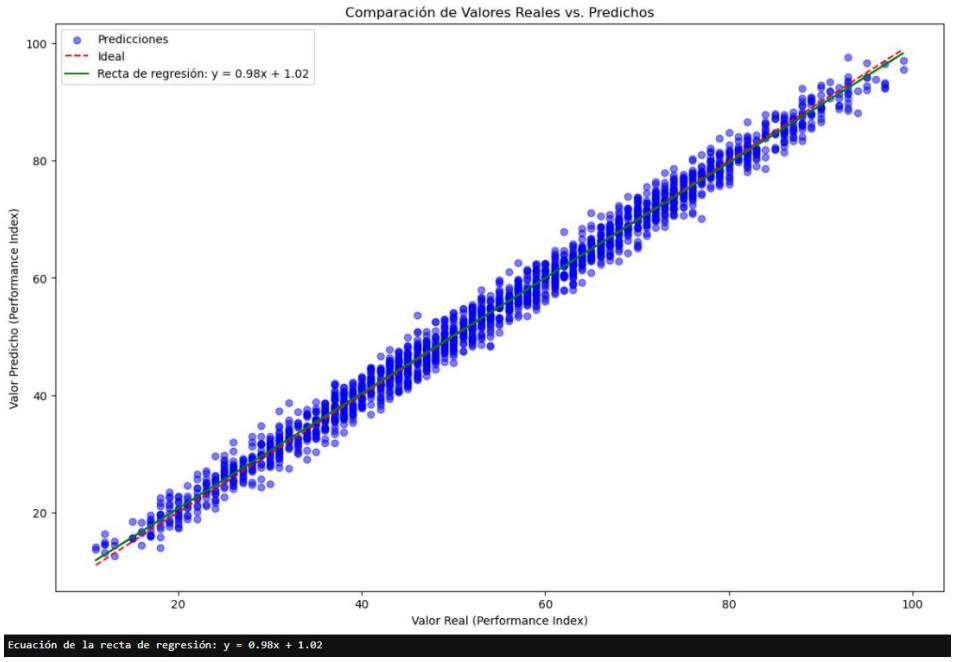
- A mayor cantidad de horas estudiadas, el índice de desempeño tiende a aumentar.
- La dispersión de los datos indica que otros factores pueden influir significativamente en el rendimiento.

2. **Relación entre "Previous Scores" y "Performance Index":**

- La correlación es más fuerte que en el caso anterior.
- La línea de tendencia muestra un crecimiento más marcado y los datos están más alineados con la regresión.
- Esto sugiere que los puntajes previos son un buen predictor del índice de desempeño.

El análisis de regresión lineal sugiere que tanto las horas de estudio como los puntajes previos tienen una relación positiva con el índice de desempeño. Sin embargo, la segunda relación (puntajes previos) parece ser un mejor predictor del rendimiento que las horas de estudio, ya que muestra una tendencia más clara y menor dispersión en los datos.

Graficar la línea de regresión sobre los datos originales para visualizar el ajuste



Análisis del Modelo de Regresión Lineal

1. Interpretación de los Coeficientes

El modelo de regresión lineal se expresa en la forma:

$$y=\beta0+\beta1Xy = \beta_0 + \beta_1 X$$

Donde:

- y: Performance Index (variable dependiente).
- X: Número de Horas de Estudio (variable independiente).
- β_0: Intercepto, representa el valor del Performance Index cuando X = 0.
- β_1: Pendiente, indica cuánto aumenta el Performance Index por cada hora adicional de estudio.

Ejemplo de interpretación:

- Si β1=5, significa que, por cada hora adicional de estudio, el rendimiento aumenta en 5 puntos.
- Si β0=20, sin estudiar, el Performance Index inicial es de 20 puntos.

Si β1es positivo, confirma la correlación positiva entre el estudio y el rendimiento académico.

2. Significado del Coeficiente R2R^2

El coeficiente de determinación R2R^2 mide qué porcentaje de la variabilidad del Performance Index puede explicarse a través de las Horas de Estudio.

$$R^2 = \frac{Variabilidad\ explicada\ por\ el\ modelo}{Variabilidad\ total}$$

Valores típicos de R2R^2:

R^2 ≈1 → El modelo explica casi toda la variabilidad (ajuste excelente).

R^2 >0.7 → Relación fuerte.

0.5< R^2 <0.7 → Relación moderada.

R^2 <0.5 → Relación débil.

Ejemplo de interpretación:

- Si R^2 = 0.85, significa que el 85% de la variabilidad del Performance Index se explica por las horas de estudio.
- Si R^2 = 0.45, solo el 45% del rendimiento académico se explica con este modelo, lo que sugiere que hay otros factores influyentes.

3. ¿Es una relación fuerte, moderada o débil?

- Si R^2 es mayor a 0.7, la relación es fuerte y el modelo es confiable.
- Si R^2 está entre 0.5 y 0.7, la relación es moderada y podría mejorarse.
- Si R^2 es menor a 0.5, la relación es débil y se necesitan más factores.

Ejemplo:

Si R^2 = 0.82, estudiar más horas es un factor clave para el desempeño académico.

Si R^2 = 0.52, otras variables como sueño, práctica de exámenes y actividades extracurriculares también influyen significativamente.

4. Posibles Mejoras o Ajustes al Modelo

1. Agregar más variables predictoras:
    - Horas de sueño
    - Práctica de exámenes
    - Actividades extracurriculares
2. Verificar la distribución de los datos:
    - Si los datos no siguen una relación lineal, podríamos probar un modelo polinómico.
3. Identificar valores atípicos:
    - Si hay datos extremos, pueden estar afectando el ajuste del modelo.
4. Evaluar la multicolinealidad:
    - Si hay otras variables correlacionadas entre sí, pueden distorsionar el análisis.

Conclusión Final

- El coeficiente β1 indica cuánto aumenta el rendimiento por cada hora adicional de estudio.
- El R^2 nos dice qué tan bien el modelo explica la variabilidad del rendimiento académico.
- Si R^2 es alto (>0.7), el modelo es confiable. Si es bajo (<0.5), se deben incluir más factores.
- Para mejorar el modelo, podemos agregar otras variables que influyen en el rendimiento académico.

Notebook Jupyter

Análisis de Datos Aplicables a Regresión Lineal Simple

Descripción del conjunto de datos

Análisis del Conjunto de Datos para Predicción del **Performance Index**

El conjunto de datos contiene información sobre **el desempeño académico de los estudiantes**, con el objetivo de predecir su **Performance Index** (índice de rendimiento). Cada fila representa un estudiante con diversas características relacionadas con su preparación y hábitos de estudio.

Variables (Features y Target)

Variable	Descripción	Tipo
Hours Studied	Número de horas dedicadas al estudio.	Numérico
Previous Scores	Calificación obtenida en evaluaciones anteriores.	Numérico
Extracurricular Activities	Indica si el estudiante participa en actividades extracurriculares (`Yes` = Sí, `No` = No).	Categórica
Sleep Hours	Cantidad de horas de sueño diarias del estudiante.	Numérico
Sample Question Papers Practiced	Número de simulacros o exámenes de práctica realizados.	Numérico
Performance Index	Variable objetivo (**Target**), indica el rendimiento académico del estudiante.	Numérico

Objetivo

El propósito de esta actividad es que los estudiantes busquen, seleccionen y analicen un conjunto de datos en **Kaggle u otra fuente confiable** para aplicar **regresión lineal simple** y predecir una variable continua (**Performance Index**) con base en una única característica independiente.

Metodología

1. **Selección de la Variable Independiente:**
   - Se debe elegir una única variable independiente que tenga una relación significativa con el **Performance Index**.
   - Se puede analizar la correlación entre las variables para determinar la más adecuada.
2. **Aplicación de la Regresión Lineal Simple:**
   - Se ajustará un modelo de regresión lineal para predecir el **Performance Index** en función de la variable seleccionada.
   - Se evaluará la ecuación obtenida y su capacidad de predicción.
3. **Evaluación del Modelo:**
   - Se analizará la **pendiente y el intercepto** de la recta de regresión.
4. **Presentación de Resultados y Conclusiones:**
   - Se interpretarán los resultados obtenidos, explicando la relación entre las variables.
   - Se comentará la validez del modelo y posibles mejoras.

Nota sobre el Preprocesamiento

- La variable **Extracurricular Activities** se puede convertir en una variable binaria:
  - `Yes` → 1
  - `No` → 0

Justificación de la Selección del Dataset para Regresión Lineal Simple

Elección de Variables

Para realizar un análisis de **regresión lineal simple**, necesitamos:

1. **Variable Independiente (Numérica):** `Hours Studied`
2. **Variable Dependiente (Continua):** `Performance Index`

La elección de este dataset es adecuada porque cumple con los criterios fundamentales para aplicar regresión lineal simple.

¿Por qué es adecuado para regresión lineal simple?

1. **Relación Causal Potencial**
   - Es lógico suponer que el número de horas de estudio (`Hours Studied`) influye directamente en el desempeño académico (`Performance Index`).
   - La relación entre estas variables puede modelarse con una línea recta, lo que es ideal para una regresión lineal.
2. **Variable Dependiente Continua**
   - `Performance Index` es una variable continua que puede tomar valores en un rango determinado, lo cual es un requisito clave para la regresión lineal.
3. **Fácil Interpretación**
   - Un modelo de regresión lineal simple permitirá responder preguntas como:
     - ¿Cuánto aumenta el rendimiento si se estudian más horas?
     - ¿Existe una correlación positiva entre estudiar más y un mejor desempeño?
4. **Simplicidad y Eficiencia**
   - La regresión lineal simple es un modelo fácil de interpretar y eficiente computacionalmente.
   - Ideal para análisis preliminares antes de probar modelos más complejos.

Posibles Aplicaciones

- Predecir el desempeño de un estudiante en función de sus horas de estudio.
- Identificar umbrales óptimos de horas de estudio para maximizar el rendimiento.
- Comparar el impacto del estudio frente a otras variables como el sueño o las actividades extracurriculares.

Consideraciones

- Se debe verificar que la relación entre `Hours Studied` y `Performance Index` sea **lineal** antes de aplicar regresión lineal.
- Es recomendable visualizar los datos con un **gráfico de dispersión** y calcular el coeficiente de correlación.
- Otros factores no considerados pueden influir en el desempeño (sesgo en los datos).

In [2]:
```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score
```

In [3]:
```
from sklearn.preprocessing import StandardScaler, MinMaxScaler
from sklearn.model_selection import train_test_split, KFold, cross_val_score
from sklearn.naive_bayes import MultinomialNB
from sklearn import metrics
```

In [4]:
```
df = pd.read_csv('Student_Performance.csv')
print(f"Shape: {df.shape}")
df.head()
```

```
Shape: (10000, 6)
```

Out[4]:

	Hours Studied	Previous Scores	Extracurricular Activities	Sleep Hours	Sample Question Papers Practiced	Performance Index
0	7	99	Yes	9	1	91
1	4	82	No	4	2	65
2	8	51	Yes	7	2	45
3	5	52	Yes	5	2	36
4	7	75	No	8	5	66

In [5]:
```
df = pd.read_csv('Student_Performance.csv')  # Load data from CSV
df
```

Out[5]:

	Hours Studied	Previous Scores	Extracurricular Activities	Sleep Hours	Sample Question Papers Practiced	Performance Index
0	7	99	Yes	9	1	91
1	4	82	No	4	2	65
2	8	51	Yes	7	2	45
3	5	52	Yes	5	2	36
4	7	75	No	8	5	66
...	...	...	...	...	...	...
9995	1	49	Yes	4	2	23
9996	7	64	Yes	8	5	58
9997	6	83	Yes	8	5	74
9998	9	97	Yes	7	0	95
9999	7	74	No	8	1	64

10000 rows × 6 columns

Analisis Exploratorio de datos

Informacion general del Dataset

In [6]:
```
print(df.info())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10000 entries, 0 to 9999
Data columns (total 6 columns):
 #   Column                           Non-Null Count  Dtype
---  ------                           --------------  -----
 0   Hours Studied                    10000 non-null  int64
 1   Previous Scores                  10000 non-null  int64
 2   Extracurricular Activities       10000 non-null  object
 3   Sleep Hours                      10000 non-null  int64
 4   Sample Question Papers Practiced 10000 non-null  int64
 5   Performance Index                10000 non-null  int64
dtypes: int64(5), object(1)
memory usage: 468.9+ KB
None
```

Los datos estan en tipo object, se modificaran para hacerlos numericos

In [7]:
```
df['Extracurricular Activities'] = df['Extracurricular Activities'].replace(['Yes', 'No'], [1, 0])
df = df.infer_objects(copy=False)  # Explicitly infer object types
df
```

```
C:\Users\bryan\AppData\Local\Temp\ipykernel_21708\2978787161.py:1: FutureWarning: Downcasting behavior in `replace` is deprecated and will be removed in a future version. To retain the old behavior, explicitly call `result.infer_objects(copy=False)`. To opt-in to the future behavior, set `pd.set_option('future.no_silent_downcasting', True)`
  df['Extracurricular Activities'] = df['Extracurricular Activities'].replace(['Yes', 'No'], [1, 0])
```

Out[7]:

	Hours Studied	Previous Scores	Extracurricular Activities	Sleep Hours	Sample Question Papers Practiced	Performance Index
0	7	99	1	9	1	91
1	4	82	0	4	2	65
2	8	51	1	7	2	45
3	5	52	1	5	2	36
4	7	75	0	8	5	66

In [8]:
```
print(df.info())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10000 entries, 0 to 9999
Data columns (total 6 columns):
 #   Column                            Non-Null Count  Dtype
---  ------                            --------------  -----
 0   Hours Studied                     10000 non-null  int64
 1   Previous Scores                   10000 non-null  int64
 2   Extracurricular Activities        10000 non-null  int64
 3   Sleep Hours                       10000 non-null  int64
 4   Sample Question Papers Practiced  10000 non-null  int64
 5   Performance Index                 10000 non-null  int64
dtypes: int64(6)
memory usage: 468.9 KB
None
```

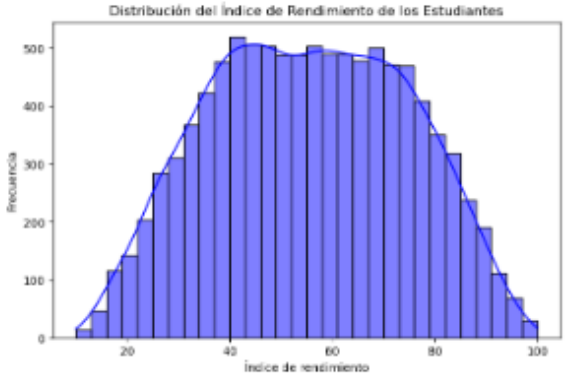
Estadistica descriptivas

In [9]:
```
print(df.describe())
```

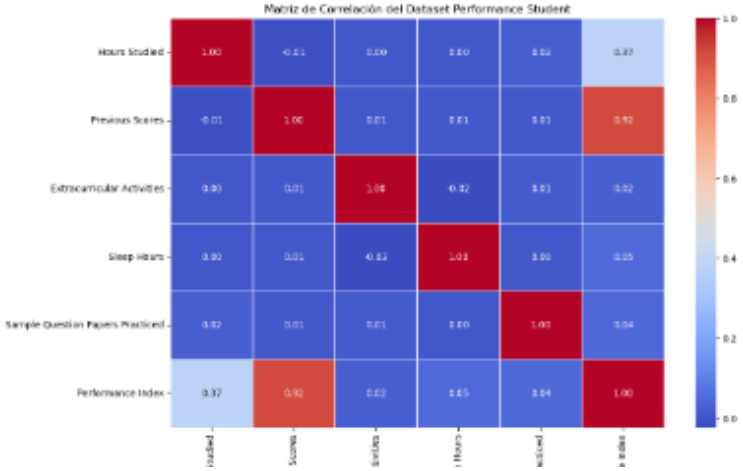
```
       Hours Studied  Previous Scores  Extracurricular Activities  \
count   10000.000000     10000.000000                10000.000000
mean        4.992900        69.445700                    0.494800
std         2.589309        17.343152                    0.499998
min         1.000000        40.000000                    0.000000
25%         3.000000        54.000000                    0.000000
50%         5.000000        69.000000                    0.000000
75%         7.000000        85.000000                    1.000000
max         9.000000        99.000000                    1.000000

       Sleep Hours  Sample Question Papers Practiced  Performance Index
count  10000.000000                      10000.000000       10000.000000
mean       6.530600                          4.583300          55.224800
std        1.695863                          2.867348          19.212558
min        4.000000                          0.000000          10.000000
25%        5.000000                          2.000000          40.000000
50%        7.000000                          5.000000          55.000000
75%        8.000000                          7.000000          71.000000
max        9.000000                          9.000000         100.000000
```

Visualización de la distribución del Perfomance Index

In [10]:
```
plt.figure(figsize=(8, 5))
sns.histplot(df["Performance Index"], bins=30, kde=True, color="blue")
plt.xlabel("Índice de rendimiento")
plt.ylabel("Frecuencia")
plt.title("Distribución del Índice de Rendimiento de los Estudiantes")
plt.show()
```



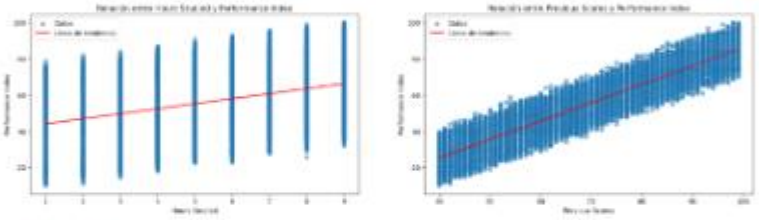
In [11]:
```
# Calcular la matriz de correlación
correlation_matrix = df.corr()

# Visualizar la matriz de correlación con un heatmap
plt.figure(figsize=(12, 8))
sns.heatmap(correlation_matrix, annot=True, cmap="coolwarm", fmt=".2f", linewidths=0.5)
plt.title("Matriz de Correlación del Dataset Performance Student")
plt.show()
```



- **Mayor correlación**: "Previous Scores" tiene una correlación de 0.92 con "Performance Index", indicando que el desempeño pasado predice fuertemente el rendimiento actual.
- **Correlación positiva moderada**: "Hours Studied" muestra una correlación de 0.37 con "Performance Index", sugiriendo que estudiar más horas contribuye al rendimiento, pero no es determinante.
- **Baja o nula correlación**: "Sleep Hours", "Sample Question Papers Practiced" y "Extracurricular Activities" tienen correlaciones cercanas a 0 con "Performance Index", indicando poca o nula influencia en el rendimiento.

Conclusión general: El puntaje previo es el mejor predictor del rendimiento, seguido por las horas de estudio, mientras que otras variables tienen poco impacto.

In [12]:
```
import numpy as np
import matplotlib.pyplot as plt

# Seleccionar las características y la variable objetivo
features = ['Hours Studied', 'Previous Scores']
target = df['Performance Index']

# Crear la figura con dos subgráficos
plt.figure(figsize=(20, 5))

for i, col in enumerate(features):
    plt.subplot(1, len(features), i + 1)
    x = df[col]
    y = target

    # Dibujar los puntos de dispersión
    plt.scatter(x, y, marker='o', alpha=0.5, label='Datos')

    # Ajustar y dibujar la línea de tendencia (regresión lineal)
    coef = np.polyfit(x, y, 1)  # Ajuste de una recta (grado 1)
    poly1d_fn = np.poly1d(coef)  # Función de la recta obtenida
    plt.plot(x, poly1d_fn(x), color="red", label="Línea de tendencia")

    plt.title(f'Relación entre {col} y Performance Index')
    plt.xlabel(col)
    plt.ylabel('Performance Index')
    plt.legend()

plt.show()
```



Análisis de Regresión Lineal Simple

Conclusión

1. **Relación entre "Hours Studied" y "Performance Index"**:
   - La pendiente de la línea de tendencia sugiere una correlación positiva, aunque relativamente débil.
   - A mayor cantidad de horas estudiadas, el índice de desempeño tiende a aumentar.
   - La dispersión de los datos indica que otros factores pueden influir significativamente en el rendimiento.
2. **Relación entre "Previous Scores" y "Performance Index"**:
   - La correlación es más fuerte que en el caso anterior.
   - La línea de tendencia muestra un crecimiento más marcado y los datos están más alineados con la regresión.
   - Esto sugiere que los puntajes previos son un buen predictor del índice de desempeño.

Conclusión General

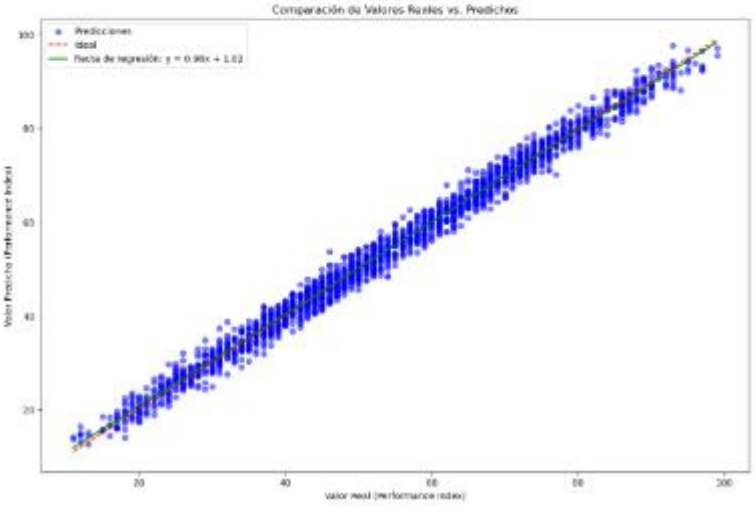
El análisis de regresión lineal sugiere que tanto las horas de estudio como los puntajes previos tienen una relación positiva con el índice de desempeño. Sin embargo, la segunda relación (puntajes previos) parece ser un mejor predictor del rendimiento que las horas de estudio, ya que muestra una tendencia más clara y menor dispersión en los datos.

In [14]:
```
from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score

# Definir variables predictoras y objetivo
X = df.drop(columns=["Performance Index"])  # Todas las columnas excepto Made
y = df["Performance Index"]  # Variable objetivo

# Dividir los datos en conjunto de entrenamiento y prueba
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Crear y entrenar el modelo de regresión lineal
model = LinearRegression()
model.fit(X_train, y_train)

# Realizar predicciones
y_pred = model.predict(X_test)

# Evaluar el modelo
mae = mean_absolute_error(y_test, y_pred)
mse = mean_squared_error(y_test, y_pred)
rmse = np.sqrt(mse)
r2 = r2_score(y_test, y_pred)

# Mostrar métricas de evaluación
metrics = pd.DataFrame({
    "Métrica": ["Error Absoluto Medio (MAE)", "Error Cuadrático Medio (MSE)",
                "Raíz del Error Cuadrático Medio (RMSE)", "Coeficiente de Determinación (R²)"],
    "Valor": [mae, mse, rmse, r2]
})

print("Métricas de Evaluación del Modelo")
print(metrics)
```

```
Metricas de Evaluación del Modelo:
                          Metrica     Valor
0             Error Absoluto Medio (MAE)  3.611121
1          Error Cuadrático Medio (MSE)  5.082628
2  Raíz del Error Cuadrático Medio (RMSE)  2.020352
3       Coeficiente de Determinación (R²)  0.988984
```

In [24]:

```
# Ajustar el modelo de regresión lineal
model = LinearRegression()
model.fit(y_test.values.reshape(-1, 1), y_pred)

# Obtener los coeficientes (pendiente e intercepto)
slope = model.coef_[0]
intercept = model.intercept_

# Crear la figura
plt.figure(figsize=(10, 6))

# Graficar los valores reales vs. predichos
plt.scatter(y_test, y_pred, alpha=0.5, color='blue', label='Predicciones')

# Graficar la línea ideal (y = x)
plt.plot([y_test.min(), y_test.max()], [y_test.min(), y_test.max()], '--', color='red', label='Ideal')

# Graficar la línea de regresión
plt.plot(y_test, slope * y_test + intercept, color='green', label=f'Recta de regresión: y = {slope:.2f}x + {intercept:.2f}')

# Etiquetas y título
plt.xlabel('Valor Real (Performance Index)')
plt.ylabel('Valor Predicho (Performance Index)')
plt.title('Comparación de Valores Reales vs. Predichos')
plt.legend()

# Mostrar la gráfica
plt.show()

# Imprimir la ecuación de la recta
print(f'Ecuación de la recta de regresión: y = {slope:.2f}x + {intercept:.2f}')
```




```
Ecuación de la recta de regresión: y = 0.98x + 1.02
```