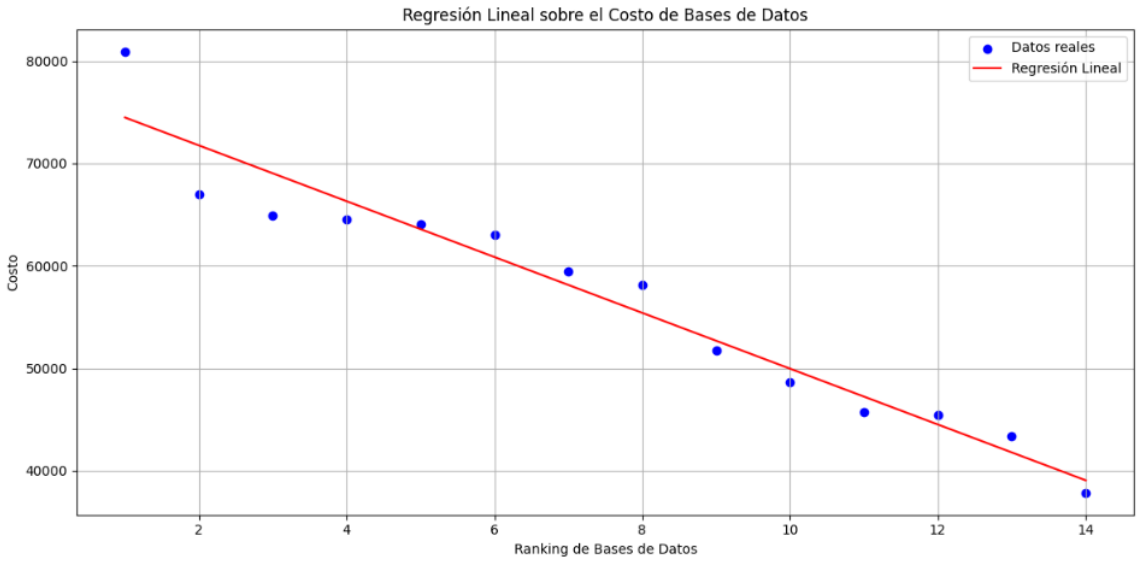
1. Costo base de datos

Código:

```python
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.linear_model import LinearRegression
import numpy as np

# Cargar los datos desde el archivo CSV
archivo_csv = 'base_de_datos.csv'  # Asegúrate de tener el archivo con este nombre
df = pd.read_csv(archivo_csv, encoding='latin1')  # Usamos 'latin1' para manejar caracteres incorrectos

# Mostrar los primeros registros para asegurarnos de que se cargó correctamente
print(df.head())

# Limpiar los nombres de las columnas (eliminar espacios extras)
df.columns = df.columns.str.strip()

# Convertir los costos a formato numérico
df['Costo'] = df['Costo'].astype(float)

# Crear un índice ficticio como variable independiente
df = df.sort_values(by="Costo", ascending=False).reset_index(drop=True)
df["Ranking"] = np.arange(1, len(df) + 1)

# Separar las variables independiente (Ranking) y dependiente (Costo)
X = df[["Ranking"]].values  # Variable independiente
y = df["Costo"].values  # Variable dependiente

# Crear y ajustar el modelo de regresión lineal
model = LinearRegression()
model.fit(X, y)

# Generar predicciones
y_pred = model.predict(X)

# Mostrar los coeficientes del modelo
print("Pendiente (coeficiente):", model.coef_[0])
print("Intercepto:", model.intercept_)

# Graficar los datos reales y la regresión lineal
plt.figure(figsize=(12, 6))
plt.scatter(df["Ranking"], df["Costo"], color="blue", label="Datos reales")
plt.plot(df["Ranking"], y_pred, color="red", label="Regresión Lineal")

# Etiquetas y título
plt.xlabel("Ranking de Bases de Datos")
plt.ylabel("Costo")
plt.title("Regresión Lineal sobre el Costo de Bases de Datos")
plt.legend()
plt.grid(True)
plt.tight_layout()

# Mostrar el gráfico
plt.show()
```

Grafica:



Hipótesis:

A medida que el ranking de las bases de datos aumenta (es decir, las bases de datos en posiciones más altas del ranking tienen un costo mayor), se puede observar que las bases de datos más costosas tienden a ser más utilizadas o demandadas en entornos profesionales. Este comportamiento podría estar asociado con características como escalabilidad, soporte empresarial, rendimiento y características avanzadas que justifican un costo más elevado.

Es posible que las bases de datos con un costo más alto, como DynamoDB y Elasticsearch, tengan características que las hacen atractivas para aplicaciones de gran escala o críticas, mientras que las opciones más asequibles, como SQLite o Firebase, podrían estar dirigidas a aplicaciones más pequeñas o de menor complejidad. La regresión lineal, por lo tanto, podría reflejar una tendencia en la que las bases de datos con costos más altos tienden a estar en el "top" del ranking debido a su adopción generalizada y características avanzadas.