



1. Descripción del Dataset

El conjunto de datos de la diabetes es una recopilación de datos médicos y demográficos de los pacientes, junto con su estado de diabetes (positivo o negativo). Los datos incluyen características como edad, sexo, índice de masa corporal (IMC), hipertensión, enfermedades cardíacas, antecedentes de tabaquismo, nivel de HbA1c y nivel de glucosa en sangre. Este conjunto de datos se puede utilizar para crear modelos de aprendizaje automático para predecir la diabetes en pacientes en función de su historial médico y su información demográfica. Esto puede resultar útil para los profesionales de la salud a la hora de identificar pacientes que pueden estar en riesgo de desarrollar diabetes y desarrollar planes de tratamiento personalizados. Además, los investigadores pueden utilizar el conjunto de datos para explorar las relaciones entre diversos factores médicos y demográficos y la probabilidad de desarrollar diabetes.

El dataset utilizado esta disponible de forma pública en el portal web de Kaggle, la fuente es la siguiente: <https://www.kaggle.com/datasets/iammustafatz/diabetes-prediction-dataset>

Objetivos y Aclaraciones

El presente trabajo, parte de un dataset cuyo propósito principal es la clasificación binaria: predecir si una persona presenta o no diabetes en función de diferentes

factores de riesgo. Variables como la edad, la hipertensión, los niveles de glucosa en sangre, el hábito de fumar o la presencia de cardiopatía se utilizan habitualmente para construir modelos de predicción orientados a esta tarea.

Sin embargo, para efectos de la consigna académica, se tomarán algunas decisiones metodológicas libres que permiten aplicar y ejemplificar distintas técnicas estadísticas, tales como la correlación y la regresión lineal simple. Esto implica que, además de explorar ciertas asociaciones relacionadas con la diabetes, se analizarán también relaciones entre otras variables numéricas del dataset que no necesariamente forman parte directa del problema de clasificación, pero que sirven para ilustrar el uso correcto de dichas técnicas.

De este modo, los objetivos específicos son:

- Mostrar criterios de selección de técnicas estadísticas (paramétricas y no paramétricas) según el tipo de variable objetivo y de predictores.
- Aplicar pruebas de correlación y modelos de regresión lineal simple en pares de variables numéricas, justificando su elección en base a la exploración de datos.
- Presentar resultados clave sobre la diabetes, destacando la importancia de algunos factores de riesgo y su potencial utilidad en el ámbito clínico.

2. Librerías e Importación de Datos

```
In [ ]: # Importar librerías

# Tratamiento de datos
# =====
import numpy as np
import pandas as pd

# Gráficos
# =====
import matplotlib.pyplot as plt
import seaborn as sns

# Preprocesado y análisis
# =====
from scipy import stats
import statsmodels.api as sm
from statsmodels.stats.multicomp import pairwise_tukeyhsd
from statsmodels.stats.diagnostic import het_breuschpagan
from scipy.stats import jarque_bera
```

```

from scipy.stats import pearsonr, spearmanr
from sklearn.model_selection import train_test_split
from sklearn.metrics import root_mean_squared_error
from sklearn.preprocessing import OneHotEncoder, StandardScaler
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import (classification_report, confusion_matrix,
                             roc_auc_score, roc_curve, accuracy_score, precision_score,
                             average_precision_score, RocCurveDisplay, Precision-RecallCurveDisplay)

# Configuración warnings
# =====
import warnings
warnings.filterwarnings("ignore")

```

```

In [ ]: from google.colab import drive
drive.mount('/content/drive')

```

Mounted at /content/drive

```

In [ ]: # Cargamos el archivo 'diabetes_dataset.csv' en un dataframe
df = pd.read_csv('/content/drive/MyDrive/ISPC CIENCIA DE DATOS/diabetes_dataset.csv')
df.head()

```

```

Out[ ]:
   gender  age  hypertension  heart_disease  smoking_history  bmi  HbA1c_level
0  Female  80.0             0              1             never  25.19  6.0
1  Female  54.0             0              0             No Info  27.32  6.0
2   Male   28.0             0              0             never  27.32  5.0
3  Female  36.0             0              0             current  23.45  5.0
4   Male   76.0             1              1             current  20.14  4.0

```

3. Análisis exploratorio (EDA)

Tamaño de los Datos

```

In [ ]: df.shape

```

```

Out[ ]: (100000, 9)

```

Comentario: El dataset consta de 100000 filas y 9 columnas

Columnas

```

In [ ]: df.columns.values

```

```
Out[ ]: array(['gender', 'age', 'hypertension', 'heart_disease',  
             'smoking_history', 'bmi', 'HbA1c_level', 'blood_glucose_level',  
             'diabetes'], dtype=object)
```

Comentarios:

- El género (gender) se refiere al sexo biológico del individuo, que puede tener un impacto en su susceptibilidad a la diabetes. Hay tres categorías: masculino, femenino y otras.
- La edad (age) es un factor importante ya que la diabetes se diagnostica con mayor frecuencia en adultos mayores. La edad oscila entre 0 y 80 años en nuestro conjunto de datos.
- La hipertensión (hypertension) es una afección médica en la que la presión arterial en las arterias está elevada persistentemente. Tiene valores 0 o 1 donde 0 indica que no tiene hipertensión y 1 significa que tiene hipertensión.
- La enfermedad cardíaca (heart_disease) es otra condición médica que se asocia con un mayor riesgo de desarrollar diabetes. Tiene valores 0 o 1 donde 0 indica que no tienen enfermedad cardíaca y 1 significa que tienen enfermedad cardíaca.
- El historial de tabaquismo (smoking_history) también se considera un factor de riesgo para la diabetes y puede exacerbar las complicaciones asociadas con la diabetes. En nuestro conjunto de datos tenemos 6 categorías: no actualmente, anteriormente, sin información, actualmente, nunca y jamás.
- El IMC (índice de masa corporal) (bmi) es una medida de la grasa corporal basada en el peso y la altura. Los valores más altos de IMC están relacionados con un mayor riesgo de diabetes. El rango de IMC en el conjunto de datos es de 10.16 a 71.55. Un IMC inferior a 18.5 indica bajo peso, entre 18.5 y 24.9 es normal, entre 25 y 29.9 indica sobrepeso y 30 o más indica obesidad.
- El nivel de HbA1c (hemoglobina glicosilada) (HbA1c_level) es una prueba de sangre que mide el nivel promedio de glucosa en la sangre durante los últimos 2 o 3 meses. se reporta en porcentaje (%). De acuerdo con los criterios clínicos, un valor de 6.5% o más es consistente con un diagnóstico de diabetes, mientras que valores entre 5.7% y 6.4% se asocian a prediabetes.

- El nivel de glucosa en sangre (blood_glucose_level) se mide en miligramos por decilitro (mg/dL) e indica la cantidad de glucosa presente en el torrente sanguíneo en un momento específico. En condiciones clínicas habituales, valores en ayunas menores a 100 mg/dL se consideran normales, entre 100 y 125 mg/dL corresponden a prediabetes y niveles iguales o superiores a 126 mg/dL sugieren diabetes. Asimismo, una medición aleatoria de 200 mg/dL o más también es indicativa de la enfermedad.
- La diabetes (diabetes) es la variable objetivo que se predice, donde los valores de 1 indican la presencia de diabetes y 0 indican la ausencia de diabetes.

Tipo de Datos

```
In [ ]: # Mostrar la información del dataframe
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 100000 entries, 0 to 99999
Data columns (total 9 columns):
#   Column                Non-Null Count  Dtype
---  -
0   gender                 100000 non-null object
1   age                   100000 non-null float64
2   hypertension           100000 non-null int64
3   heart_disease          100000 non-null int64
4   smoking_history        100000 non-null object
5   bmi                    100000 non-null float64
6   HbA1c_level            100000 non-null float64
7   blood_glucose_level    100000 non-null int64
8   diabetes               100000 non-null int64
dtypes: float64(3), int64(4), object(2)
memory usage: 6.9+ MB
```

Comentarios:

- La data incluye variables numéricas ['age', 'bmi', 'HbA1c_level', 'blood_glucose_level'] y categóricas ['gender', 'hypertension', 'heart_disease', 'smoking_history', 'diabetes']
- Las variables categóricas ['hypertension', 'heart_disease', 'diabetes'], se encuentran codificadas, por lo que las toma como int64.
- Con el conteo de no nulos, se observa que no hay valores faltantes en el conjunto de datos.

Registros Duplicados

```
In [ ]: # cantidad de filas duplicadas
df.duplicated().sum()
```

```
Out[ ]: np.int64(3854)
```

```
In [ ]: # Mostrar filas duplicadas
df[df.duplicated()].head()
```

```
Out[ ]:
```

	gender	age	hypertension	heart_disease	smoking_history	bmi	HbA1c
2756	Male	80.0	0	0	No Info	27.32	
3272	Female	80.0	0	0	No Info	27.32	
3418	Female	19.0	0	0	No Info	27.32	
3939	Female	78.0	1	0	former	27.32	
3960	Male	47.0	0	0	No Info	27.32	

```
In [ ]: #Eliminar Duplicados
df = df.drop_duplicates()

# Validar que se eliminaron
df.duplicated().sum()
```

```
Out[ ]: np.int64(0)
```

```
In [ ]: df.shape
```

```
Out[ ]: (96146, 9)
```

Comentario:

Se encontraron 3854 registros duplicados, los cuales fueron eliminados, para finalmente obtener un dataset con 96146 filas y 9 columnas.

Explorando Variables Categóricas

```
In [ ]: cat_vars = ['gender', 'hypertension', 'heart_disease', 'smoking_history', 'dia

# Diccionario con frecuencias absolutas y relativas
freqs = {}
for col in cat_vars:
    abs_freq = df[col].value_counts(dropna=False)
    rel_freq = df[col].value_counts(normalize=True, dropna=False) * 100
    freqs[col] = pd.DataFrame({"Frecuencia": abs_freq, "Porcentaje": rel_freq.
```

```
# Mostrar tablas individuales
for col, tabla in freqs.items():
    print(f"\n### {col}")
    display(tabla)
```

gender

	Frecuencia	Porcentaje
gender		
Female	56161	58.41
Male	39967	41.57
Other	18	0.02

hypertension

	Frecuencia	Porcentaje
hypertension		
0	88685	92.24
1	7461	7.76

heart_disease

	Frecuencia	Porcentaje
heart_disease		
0	92223	95.92
1	3923	4.08

smoking_history

	Frecuencia	Porcentaje
smoking_history		
never	34398	35.78
No Info	32887	34.21
former	9299	9.67
current	9197	9.57
not current	6367	6.62
ever	3998	4.16

diabetes

	Frecuencia	Porcentaje
diabetes		
0	87664	91.18
1	8482	8.82

```
In [ ]: #Eliminar registros donde gender == "Other"
df = df[df["gender"].isin(["Female", "Male"])]

#Actualizar conteo
conteo_genero = df['gender'].value_counts()
print(conteo_genero)
```

```
gender
Female    56161
Male      39967
Name: count, dtype: int64
```

```
In [ ]: col = "gender"

abs_freq = df[col].value_counts(dropna=False)
rel_freq = df[col].value_counts(normalize=True, dropna=False) * 100

tabla = pd.DataFrame({
    "Frecuencia": abs_freq,
    "Porcentaje": rel_freq.round(2)
})

print(f"\n### {col}")
display(tabla)
```

```
### gender
```

	Frecuencia	Porcentaje
gender		
Female	56161	58.42
Male	39967	41.58

Comentario:

En la variable 'gender' se detecta una categoría 'other' = 'otro', dado que no suma al análisis porque son pocos registros (<0.02%), los eliminamos.

Visualización de Variables Categóricas

```
In [ ]: cat_vars = ['gender', 'hypertension', 'heart_disease', 'smoking_history', 'dia
```



```

fig, axes = plt.subplots(2, 3, figsize=(16, 10))
axes = axes.flatten()

for i, col in enumerate(cat_vars):
    data = df[col].value_counts(normalize=True).mul(100).reset_index()
    data.columns = [col, "Porcentaje"]
    data = data.sort_values("Porcentaje", ascending=False)

    sns.barplot(x=col, y="Porcentaje", data=data, palette="Set2", ax=axes[i])

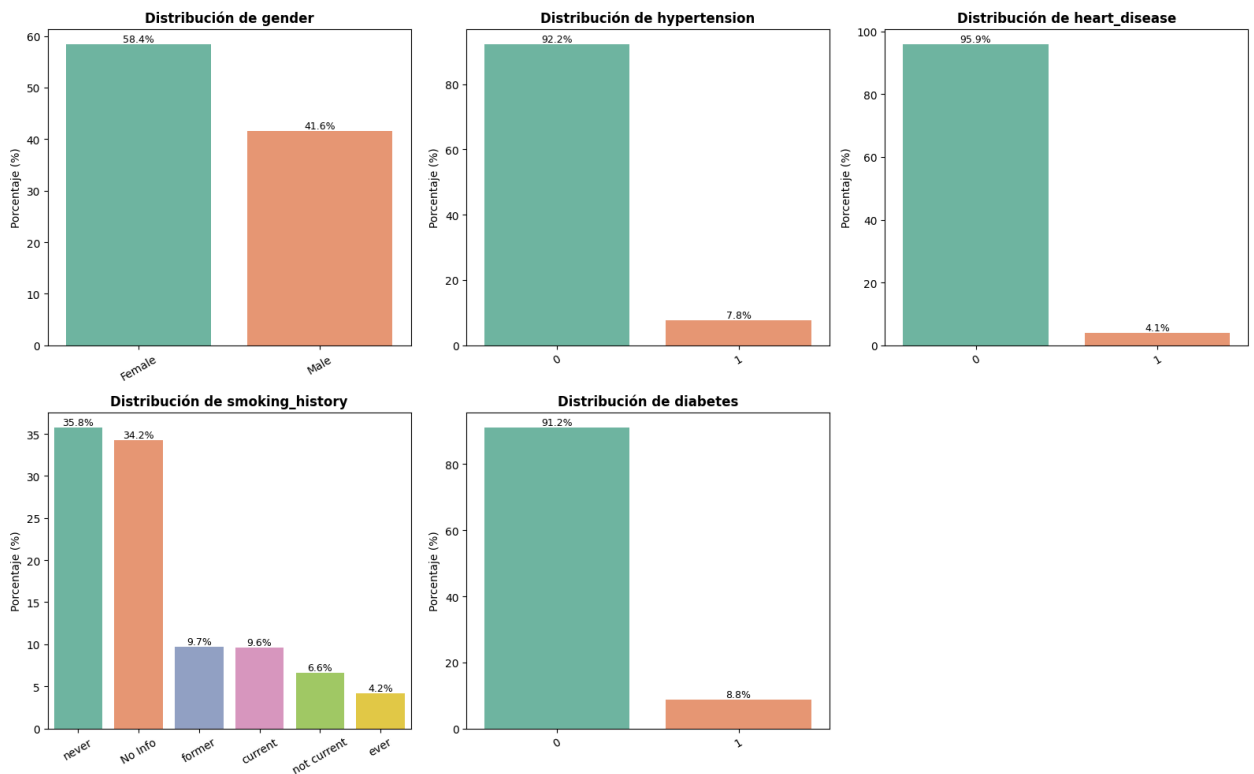
    # Etiquetas en cada barra
    for p in axes[i].patches:
        axes[i].annotate(f"{p.get_height():.1f}%",
                        (p.get_x() + p.get_width() / 2., p.get_height()),
                        ha="center", va="bottom", fontsize=9)

    axes[i].set_title(f"Distribución de {col}", fontsize=12, weight="bold")
    axes[i].set_ylabel("Porcentaje (%)")
    axes[i].set_xlabel("")
    axes[i].tick_params(axis='x', rotation=30)

# Eliminar subplot vacío si sobran casillas
if len(cat_vars) < len(axes):
    fig.delaxes(axes[-1])

plt.tight_layout()
plt.show()

```



Interpretación de variables categóricas

- Género: La muestra incluye 56,161 mujeres (58.4%) y 39,967 hombres (41.6%). Existe un ligero predominio femenino, aunque no supone un desbalance crítico y permite comparaciones válidas entre ambos grupos.
- Hipertensión: El 92.2% de los registros corresponde a personas sin hipertensión y solo el 7.8% a personas con esta condición. Aunque es un grupo minoritario, la hipertensión sigue siendo un factor de riesgo relevante en el desarrollo de diabetes.
- Enfermedad cardíaca: La mayoría (95.9%) no presenta cardiopatía, mientras que el 4.1% sí. Pese a su baja proporción, esta variable resulta importante en el análisis por su asociación con complicaciones metabólicas y cardiovasculares.
- Historial de tabaquismo: La distribución es más heterogénea: un 35.8% nunca fumó, un 9.7% son exfumadores, un 9.6% fumadores actuales, un 6.6% “no actuales” y un 4.2% “ever”. Además, un 34.2% está clasificado como No Info, lo que indica que no se dispone de información sobre el historial de tabaquismo para ese grupo. Aunque no se trata de valores faltantes, esta categoría limita el análisis detallado de la relación entre tabaquismo y diabetes.
- Diabetes (variable objetivo): El dataset está fuertemente desbalanceado: el 91.2% no presenta diabetes frente a un 8.8% con diagnóstico positivo. Este desbalance debe considerarse en los análisis y modelos predictivos, ya que puede sesgar los resultados hacia la clase mayoritaria.

Conclusión

El dataset representa una población mayoritariamente sin hipertensión, cardiopatía ni diabetes, con un ligero predominio de mujeres. El tabaquismo aparece como un factor con categorías variadas y un grupo amplio sin información disponible (No Info). El aspecto más relevante es el desbalance en la variable objetivo (diabetes), que constituye una consideración clave tanto para los análisis estadísticos como para la construcción de modelos de machine learning.

Explorando Variables Numéricas

Resumen Estadístico

```
In [ ]: # Variables cuantitativas
num_vars = ["age", "bmi", "HbA1c_level", "blood_glucose_level"]

# Tabla descriptiva extendida
desc = df[num_vars].describe(percentiles=[0.01,0.1,0.15,0.25,0.5,0.75,0.9,0.99])

# Añadir skewness y kurtosis
desc["skew"] = df[num_vars].skew()
desc["kurtosis"] = df[num_vars].kurtosis()

# Redondear para mejor visualización
desc = desc.round(2)

desc
```

```
Out[ ]:
```

	count	mean	std	min	1%	10%	15%	25%	50%
age	96128.0	41.80	22.46	0.08	1.00	10.0	15.0	24.0	43.0
bmi	96128.0	27.32	6.77	10.01	14.55	19.0	20.8	23.4	27.3
HbA1c_level	96128.0	5.53	1.07	3.50	3.50	4.0	4.0	4.8	5.8
blood_glucose_level	96128.0	138.22	40.91	80.00	80.00	85.0	90.0	100.0	140.0

Interpretación por variable:

♦ Edad (age)

Media: 41.8 años.

Mediana (50%): 43 años, muy cerca de la media, lo que indica simetría.

Mínimo-Máximo: 0.08 a 80 años

Skew = -0.06, prácticamente simétrica.

Kurtosis = -1.0, distribución más “aplanada” que la normal (platicúrtica).

La edad en la muestra está bien distribuida, sin sesgo.

♦ Índice de Masa Corporal (bmi)

Media: 27.3 (categoría sobrepeso, OMS).

Mediana: 27.3, igual a la media, lo que sugiere cierta simetría central.

Percentiles: el 75% está debajo de 29.8, pero el 99% llega a 48.9 y el máximo a 95.7, hay outliers extremos.

Skew = 1.02, distribución sesgada a la derecha.

Curtosis = 3.27, leptocúrtica, con colas más pesadas que la normal.

la mayoría tiene BMI entre 20 y 35, pero existen valores muy altos que empujan la distribución hacia la derecha.

♦ **HbA1c (HbA1c_level)**

Media: 5.53% , dentro de lo normal (diabetes \geq 6.5%).

Mediana: 5.8%, muy próxima a la media.

Rango: 3.5 a 9, cubre desde niveles bajos hasta valores compatibles con diabetes.

Skew = -0.05, simétrica.

Curtosis = 0.24, cercana a la normal.

Variable bien comportada, sin sesgo fuerte; la mayoría de los pacientes no llega al umbral de diabetes por HbA1c.

♦ **Glucosa en sangre (blood_glucose_level)**

Media: 138.2 mg/dL (un poco por encima del valor normal en ayunas < 126 mg/dL).

Mediana: 140 mg/dL, muy próxima a la media.

Rango: 80 a 300, incluye tanto valores normales como casos severos.

Skew = 0.84, distribución sesgada a la derecha (colas largas).

Curtosis = 1.76, leptocúrtica, con colas más pesadas que la normal.

Distribución con valores extremos hacia arriba, consistente con algunos pacientes con hiperglucemia importante.

Conclusiones

- Edad y HbA1c: variables parecen bastante simétricas y normales.

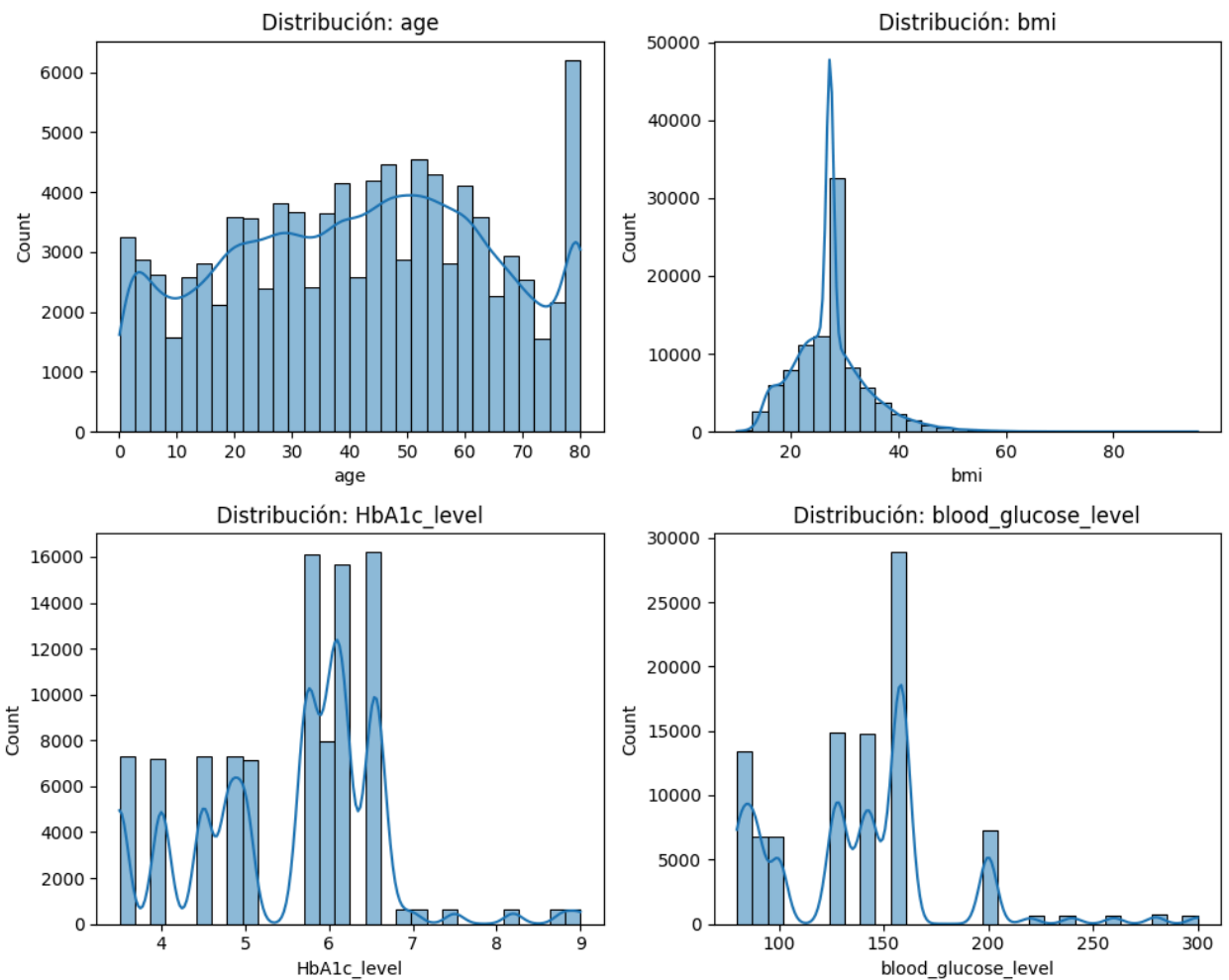
- BMI y Glucosa: variables sesgadas a la derecha, con outliers extremos que pueden influir en correlaciones y regresiones.
- Curtosis positiva en BMI y Glucosa: colas más pesadas, riesgo de outliers que distorsionen el análisis.

Recomendaciones en Análisis Inferencial:

- Probar normalidad formal (Shapiro, Kolmogorov-Smirnov, Anderson-Darling).
- Usar transformaciones (log) o pruebas no paramétricas si los outliers influyen mucho.

Distribución de Variables

```
In [ ]: # Creando las visualizaciones de las distribuciones
fig, axs = plt.subplots(2, 2, figsize=(10, 8))
axs = axs.ravel()
for i, v in enumerate(num_vars):
    sns.histplot(df[v].dropna(), bins=30, kde=True, ax=axs[i])
    axs[i].set_title(f'Distribución: {v}')
plt.tight_layout()
plt.show()
```



Análisis:

En la tabla de estadísticos inicial observamos que edad y HbA1c presentaban valores de asimetría cercanos a cero y curtosis bajas, lo que sugería distribuciones relativamente normales. Esto se corrobora en los histogramas, donde ambas variables muestran una forma simétrica y sin colas extremas relevantes.

Por otro lado, tanto IMC como glucosa en sangre aparecían con asimetría positiva y curtosis elevada en la tabla, lo que anticipaba distribuciones sesgadas con presencia de valores extremos. Los histogramas confirman esta situación: el IMC se concentra en torno a 20-35 pero con colas largas hacia valores muy altos, y la glucosa se centra alrededor de 140 mg/dL con casos severos que alcanzan 300 mg/dL.

En conclusión, la inspección visual de los histogramas respalda lo cuantificado en la tabla: edad y HbA1c son más estables y cercanas a la normalidad, mientras que IMC y glucosa muestran sesgo a la derecha y outliers que deberán considerarse en análisis posteriores.

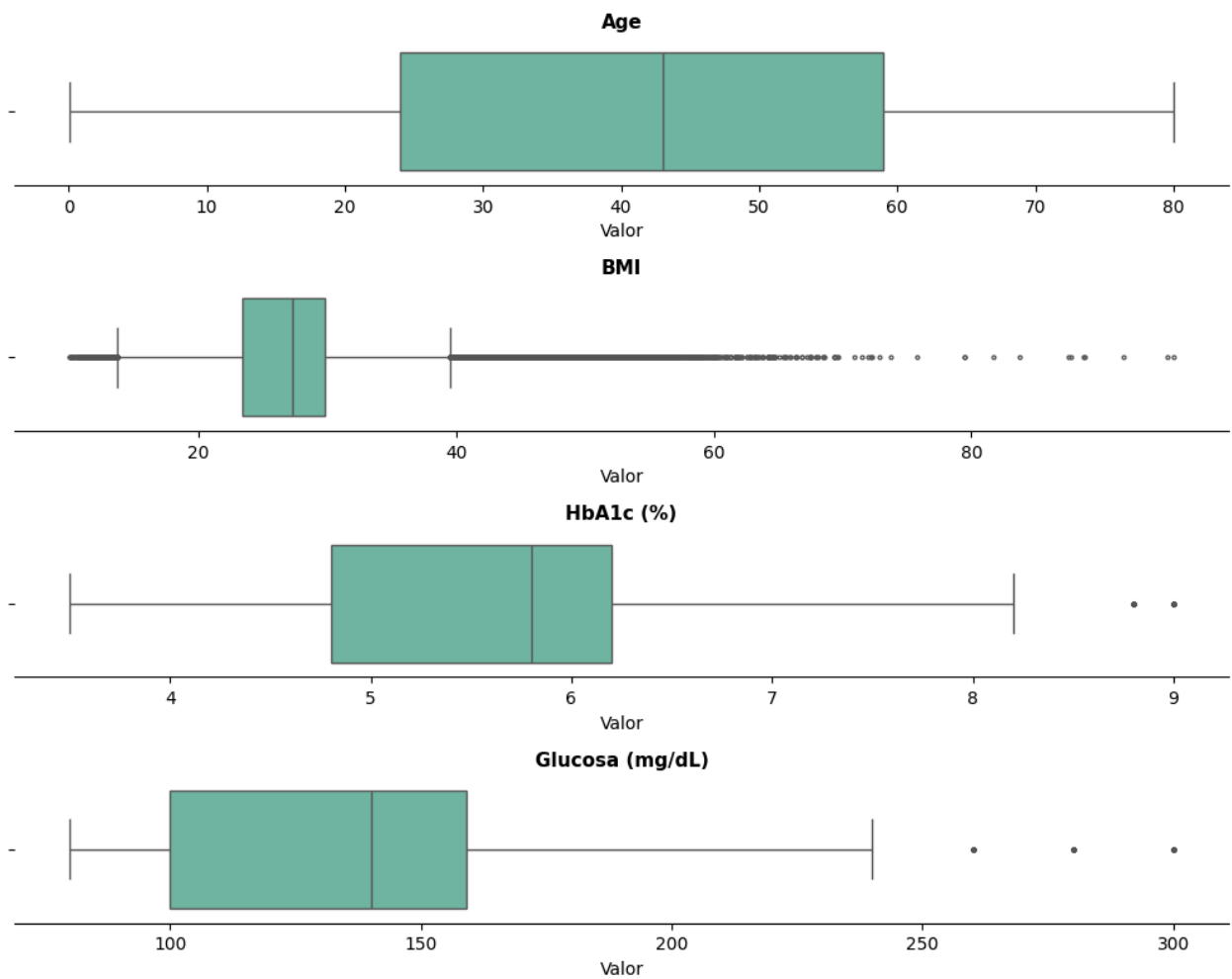
Boxplot de las Variables

```
In [ ]: vars_plot = [
    ("age", "Age"),
    ("bmi", "BMI"),
    ("HbA1c_level", "HbA1c (%)" ),
    ("blood_glucose_level", "Glucosa (mg/dL)" ),
]

fig, axes = plt.subplots(nrows=4, ncols=1, figsize=(10, 8), sharex=False)

for ax, (col, label) in zip(axes, vars_plot):
    sns.boxplot(x=df[col], orient="h", ax=ax, color=sns.color_palette("Set2",
    ax.set_title(label, fontsize=11, fontweight="bold")
    ax.set_ylabel("")
    ax.set_xlabel("Valor")

sns.despine(left=True)
plt.tight_layout()
plt.show()
```



Interpretación por Variable:

♦ Age

La caja está bien centrada, con la mediana hacia el centro, distribución simétrica.

El rango intercuartílico (IQR) es amplio, lo que indica variabilidad moderada.

No se observan outliers relevantes.

La edad es estable y bastante normal

♦ BMI

La caja está comprimida entre 20 y 35 (la mayoría de los valores).

Aparecen muchísimos outliers a la derecha, que llegan hasta 96. Que confirma sesgo positivo y colas largas.

la mayoría está en sobrepeso/obesidad moderada, pero los outliers extremos dominan visualmente y pueden distorsionar análisis paramétricos.

♦ HbA1c_level

Caja más equilibrada, sin sesgo marcado.

Se ven algunos outliers, pero en menor cantidad y no tan extremos.

la variable se ve bastante controlada, pero con ciertos valores elevados que conviene vigilar.

♦ Blood_glucose_level

La caja se concentra entre 100 y 160 mg/dL.

Se destacan varios outliers altos. Esto sugiere una distribución con sesgo positivo y valores atípicos que reflejan casos de glucosa elevada en la muestra.

la mayoría está cerca del rango limítrofe de diabetes (126–140 mg/dL), pero hay casos con glucosa muy elevada que generan colas pesadas.

Conclusiones

- Age y HbA1c: distribuciones más regulares, sin gran presencia de outliers.

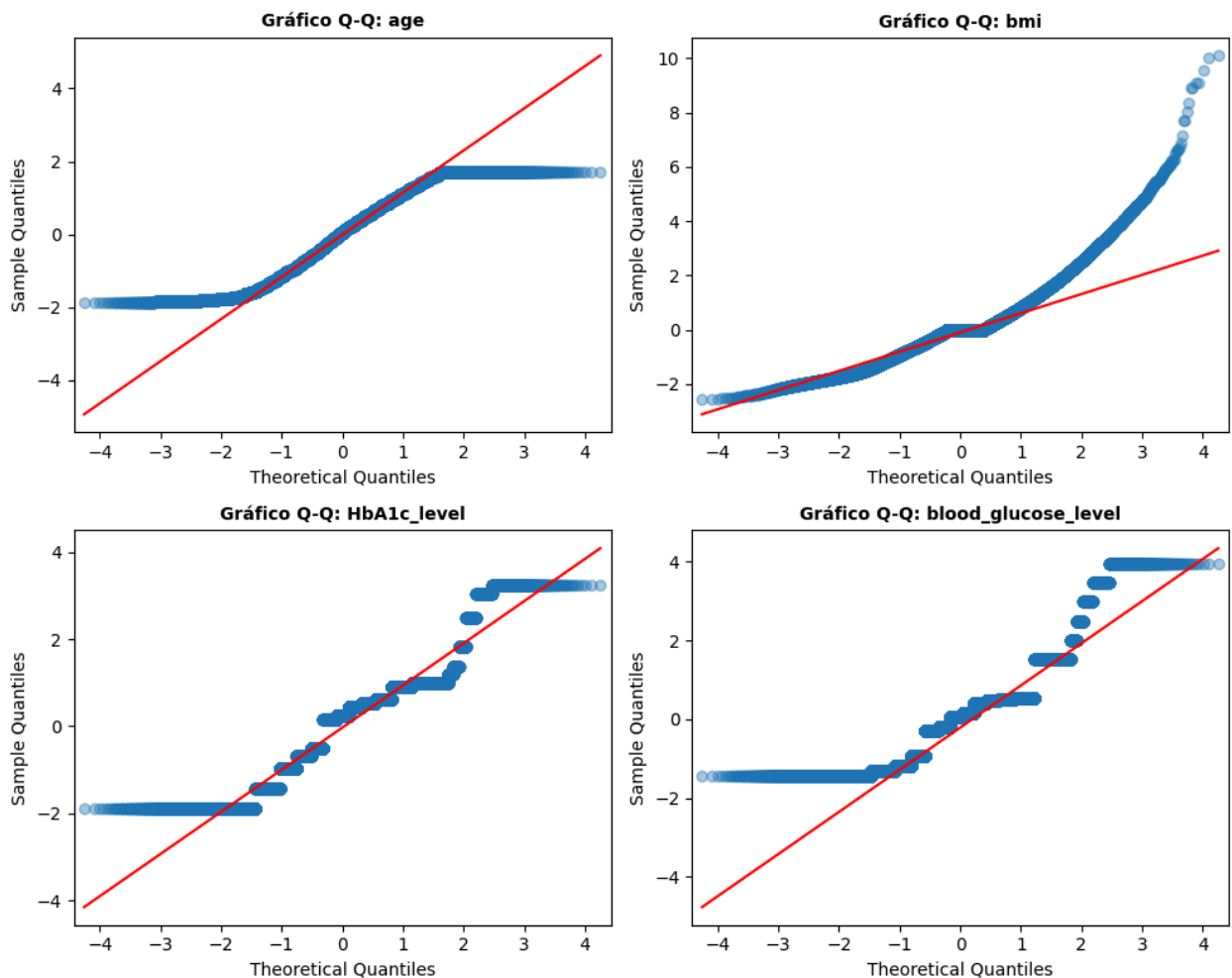
- BMI y Glucosa: variables con clara presencia de outliers y sesgo a la derecha, lo que confirma lo visto en histogramas y en los estadísticos (asimetría positiva, curtosis alta).
- En análisis posteriores (correlaciones, regresión), habrá que evaluar si se tratan los outliers (winsorización, transformaciones logarítmicas) o si se emplean métodos robustos/no paramétricos.

Prueba de Normalidad (Q-Q plots)

```
In [ ]: fig, axes = plt.subplots(2, 2, figsize=(10, 8))

for i, var in enumerate(num_vars):
    sm.qqplot(df[var], line='q', fit=True, ax=axes[i//2, i%2], alpha=0.4, lw=2)
    axes[i//2, i%2].set_title(f"Gráfico Q-Q: {var}", fontsize=10, fontweight="bold")

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



Interpretación de los Q-Q plots:

1. Age

Los puntos siguen bastante bien la línea roja, salvo leves desviaciones en colas. Distribución aproximadamente normal.

2. BMI

Los puntos se separan mucho de la línea en la cola superior (valores altos). No hay normalidad, sesgo positivo y muchos outliers.

3. HbA1c_level

Se ven escalones (efecto de valores discretos) y cierta desviación en colas. Aunque no es perfectamente normal, no tiene el sesgo extremo de BMI.

4. Blood_glucose_level

Igual que HbA1c, escalones y desviaciones marcadas en colas. Muy alejado de la normalidad.

- **Conclusión:** Los gráficos Q-Q corroboran que age y HbA1c tienen distribuciones más estables, mientras que BMI y blood_glucose_level presentan asimetría positiva marcada. Esto condiciona el uso de pruebas paramétricas y justifica considerar transformaciones logarítmicas o correlaciones no paramétricas en los siguientes análisis.

4. Estudio de Correlación

Enfoque para el Análisis de Correlación

1. Volumen de datos:

- Con tamaños muestrales muy grandes, las pruebas formales de normalidad (Shapiro-Wilk, Kolmogorov-Smirnov) suelen rechazar el supuesto incluso ante desviaciones mínimas.
- Por ello, la decisión se apoya principalmente en la inspección visual (histogramas, Q-Q plots, boxplots) y en los estadísticos de asimetría y curtosis.

2. Transformaciones:

- Cuando una variable presenta fuerte sesgo (ejemplo: BMI y glucosa en este dataset), se consideran transformaciones logarítmicas para mejorar simetría y linealidad.
- Se trabaja tanto con variables crudas como transformadas, comparando resultados.

3. **Selección del método de correlación:**

- Pearson → adecuado cuando se busca medir relaciones lineales, aplicable a variables originales o transformadas si mejoran la normalidad.
- Spearman → recomendado en presencia de sesgo, outliers o distribuciones no normales, ya que mide relaciones monótonas.
- Kendall → menos sensible a outliers pero computacionalmente más costoso; se suele reservar para muestras pequeñas o confirmación adicional.

4. **Visualización:**

- Se generan mapas de calor (heatmaps) para comparar matrices de correlación (Pearson vs Spearman).
- Se incluyen scatterplots con línea de tendencia para validar visualmente si la relación observada es efectivamente lineal o sólo monótona.

5. **Selección de Variables:**

- Se priorizan las variables que muestran correlaciones más altas y estables en ambos métodos.
- Si Spearman es claramente mayor que Pearson, se interpreta como una relación monotónica no lineal, y por lo tanto, la regresión lineal simple puede no ser el modelo más apropiado.

Aplicación al Dataset de Diabetes

1. **Volumen:** al contar con más de 90.000 registros, los tests formales de normalidad pierden utilidad práctica.

2. **Distribuciones:** según histogramas, Q-Q plots y boxplots, las variables BMI y

glucosa están sesgadas a la derecha con outliers, mientras que edad y HbA1c son más estables.

3.Método recomendado: Spearman es la opción más robusta y segura para evaluar asociaciones. Pearson se mantiene como referencia, sobre todo después de aplicar transformaciones logarítmicas a BMI y glucosa.

Decisión final: Spearman servirá como punto de partida para determinar pares relevantes, pero se contrastará con Pearson (crudo y transformado) para verificar si existe también evidencia de linealidad suficiente que justifique un modelo de regresión lineal.

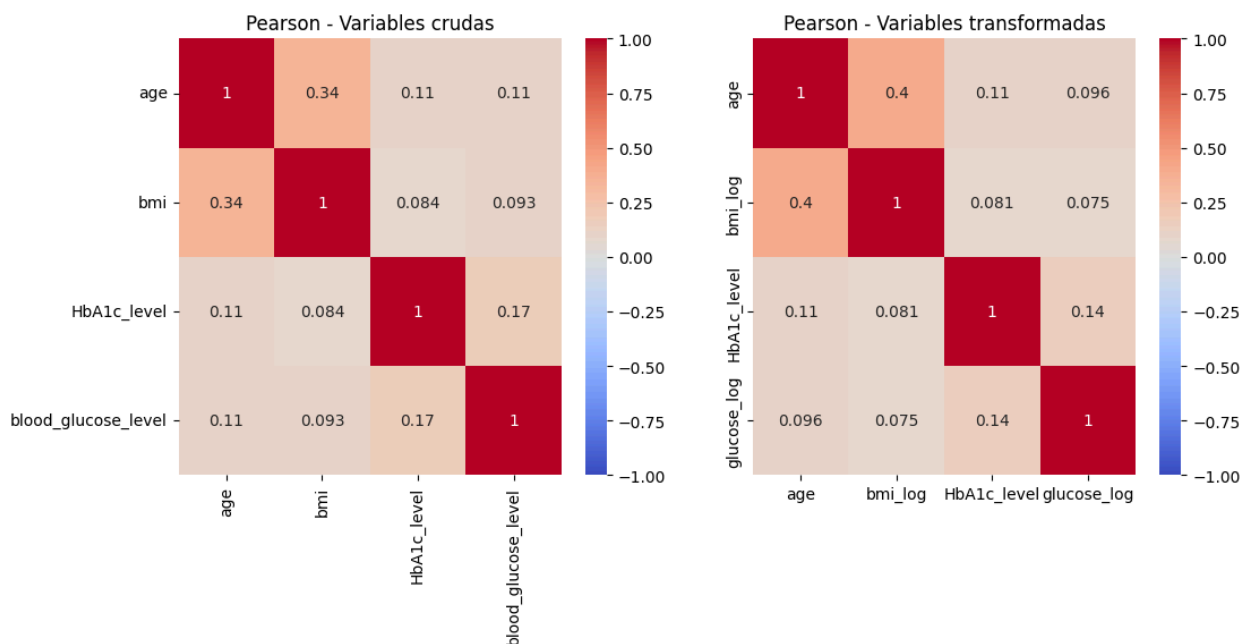
Matrices de Correlación

```
In [ ]: # Matriz Pearson con datos crudos
corr_pearson_raw = df[num_vars].corr(method="pearson")

# Matriz Pearson con variables transformadas (log)
df_trans = df.copy()
df_trans["bmi_log"] = np.log1p(df_trans["bmi"])
df_trans["glucose_log"] = np.log1p(df_trans["blood_glucose_level"])
corr_pearson_log = df_trans[["age", "bmi_log", "HbA1c_level", "glucose_log"]].corr()

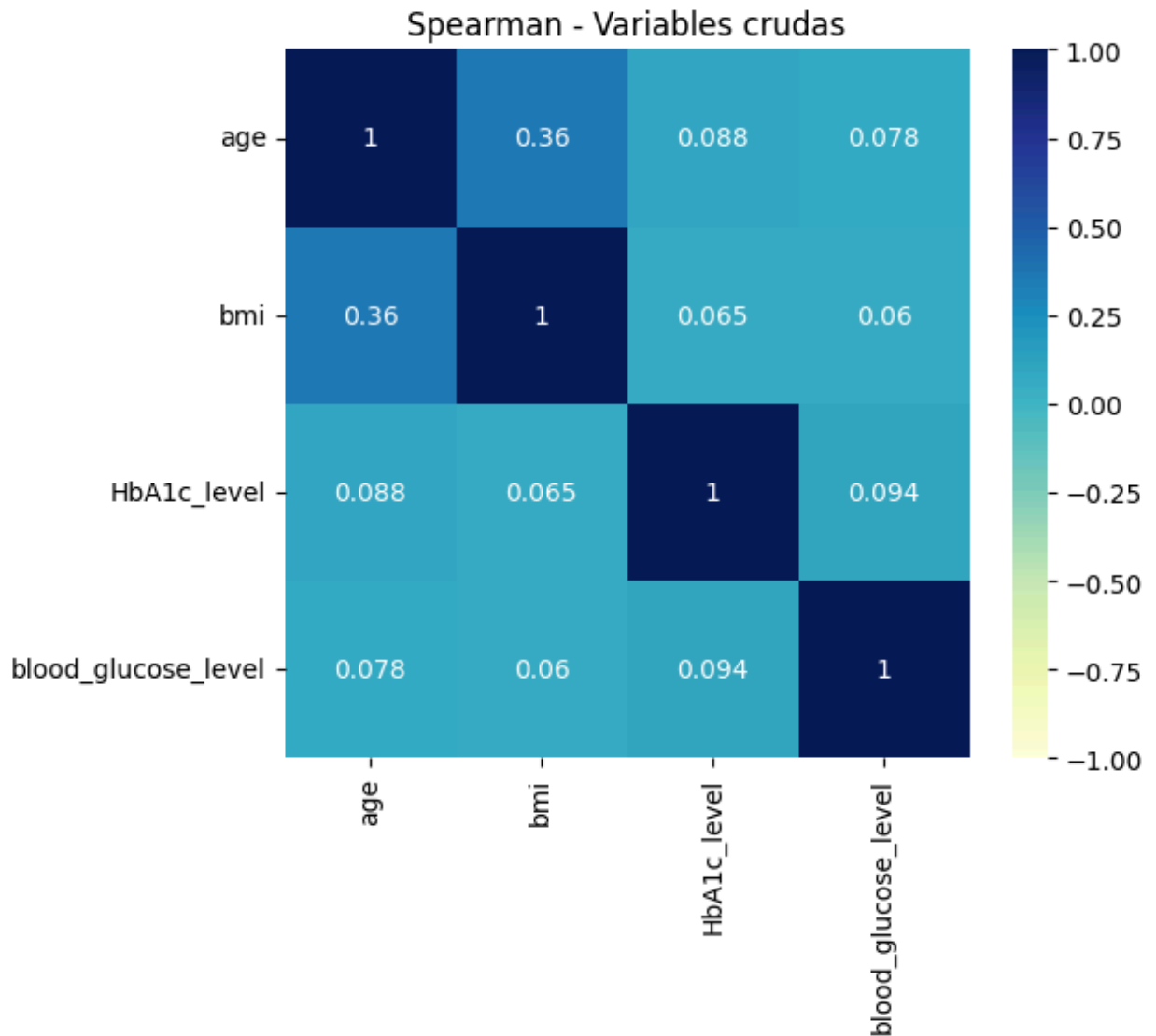
# Graficar
fig, axes = plt.subplots(1, 2, figsize=(12,5))
sns.heatmap(corr_pearson_raw, annot=True, cmap="coolwarm", vmin=-1, vmax=1, ax=axes[0].set_title("Pearson - Variables crudas"))

sns.heatmap(corr_pearson_log, annot=True, cmap="coolwarm", vmin=-1, vmax=1, ax=axes[1].set_title("Pearson - Variables transformadas"))
plt.show()
```



```
In [ ]: corr_spearman = df[num_vars].corr(method="spearman")

plt.figure(figsize=(6,5))
sns.heatmap(corr_spearman, annot=True, cmap="YlGnBu", vmin=-1, vmax=1)
plt.title("Spearman - Variables crudas")
plt.show()
```



Interpretación de Correlaciones

Al comparar los métodos de correlación (Pearson en crudo, Pearson con variables transformadas y Spearman), se observan las siguientes conclusiones clave:

- **Age-BMI:**

Pearson crudo: 0.34

Pearson log-transformado: 0.40

Spearman: 0.36

- ♦ Este par de variables muestra la asociación más consistente. La transformación logarítmica de BMI mejora la linealidad, y Spearman confirma que existe una relación monotónica positiva.
- ♦ Es el par más adecuado para ejemplificar un modelo de regresión lineal.

- **HbA1c-Glucosa:**

Pearson crudo: 0.17

Pearson log-transformado: 0.14

Spearman: 0.094

- ♦ A pesar de su relación clínica conocida, en este dataset la correlación es baja. No se observa una relación fuerte ni estrictamente lineal.

Otros pares (Age-HbA1c, BMI-Glucosa, etc.):

- ♦ Todas presentan correlaciones débiles (<0.15), lo que indica ausencia de asociación relevante en este contexto.

Conclusión

De acuerdo con los resultados:

Age-BMI es el único par que justifica un análisis de regresión lineal, al ser el que muestra mayor fuerza de asociación y mejora con transformaciones.

El resto de las combinaciones presentan correlaciones débiles, por lo que no resultan útiles para construir modelos lineales en este análisis exploratorio.

Diagramas de dispersión con ajuste lineal para las variables seleccionadas

Para los scatterplots se seleccionaron las combinaciones con mayor correlación.

En Age-BMI, se utilizó BMI transformado en logaritmo, ya que la transformación mejoró la linealidad ($r=0.40$).

En Glucosa-HbA1c, se usaron las variables en crudo, dado que la transformación no aportó mejoras y resulta más interpretable en sus unidades originales.

```

In [ ]: # Hacer copia del df original
df_scatter = df.copy()

# Crear variable transformada solo en la copia
df_scatter["bmi_log"] = np.log1p(df_scatter["bmi"]) # log(1+x) para evitar p

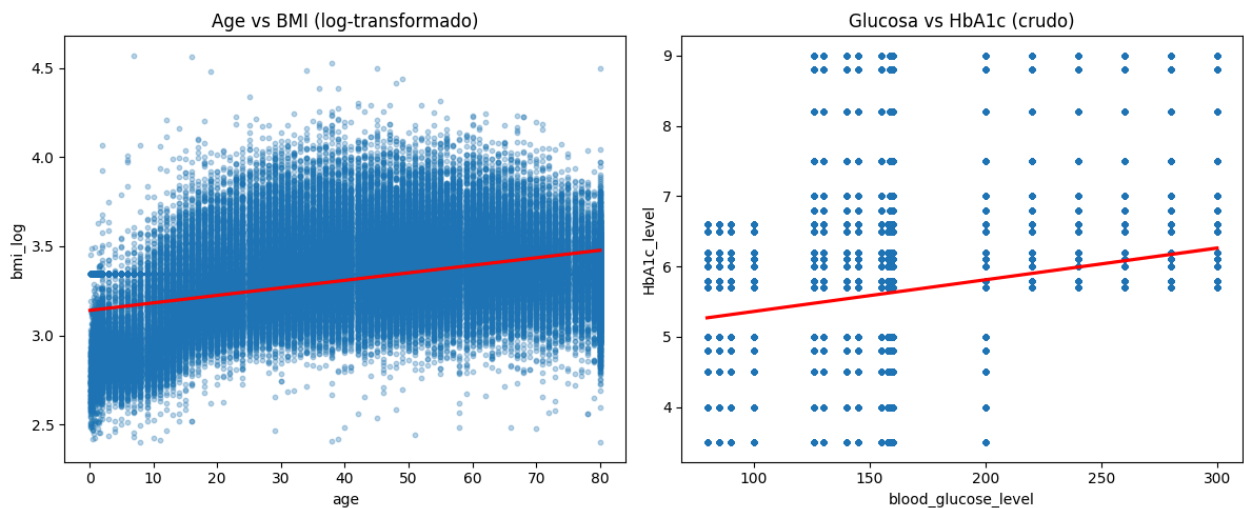
# Pares a graficar: mezclando crudo y transformado
pares = [
    ("age", "bmi_log", "Age vs BMI (log-transformado)"),
    ("blood_glucose_level", "HbA1c_level", "Glucosa vs HbA1c (crudo)")
]

# Generar scatterplots con regresión
fig, axes = plt.subplots(1, 2, figsize=(12,5))

for i, (x, y, titulo) in enumerate(pares):
    sns.regplot(
        data=df_scatter, x=x, y=y,
        scatter_kws={'alpha':0.3, 's':10}, # puntos más pequeños y suaves
        line_kws={'color':'red'},          # línea de tendencia
        ax=axes[i]
    )
    axes[i].set_title(titulo)

plt.tight_layout()
plt.show()

```



Análisis:

Se presentan los scatterplots para los pares de variables con mayor interés según los coeficientes de correlación (Pearson y Spearman):

- **Age vs BMI (log-transformado):** se observa una **tendencia lineal positiva moderada**, donde a mayor edad tiende a aumentar el índice de masa corporal. La transformación logarítmica ayudó a mejorar la

simetría y a estabilizar la relación, lo que respalda su uso en un modelo de regresión lineal.

- **Glucosa (mg/dL) vs HbA1c (%):** existe una **tendencia creciente débil**, lo que coincide con la baja correlación hallada. Aunque clínicamente ambas variables deberían estar asociadas, en este dataset la dispersión de puntos es elevada y limita la capacidad explicativa de un modelo lineal directo.

En conclusión, los scatterplots confirman lo observado en las matrices de correlación:

- La relación **Edad-BMI (log)** es la más consistente para un análisis de regresión.
- La relación **Glucosa-HbA1c**, aunque teóricamente esperable, aparece débil en los datos analizados.

5. Regresión Lineal Simple (OLS)

Modelo: `bmi_log ~ age`

1. Definición y División de Datos

```
In [ ]: # Se utilizan las variables 'age' y 'bmi_log' del DataFrame 'df_scatter'
X = df_scatter['age']
y = df_scatter['bmi_log']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

2. Entrenamiento del Modelo

```
In [ ]: X_train_const = sm.add_constant(X_train, prepend=True)
modelo_bmi = sm.OLS(y_train, X_train_const).fit()
print(modelo_bmi.summary())
```


OLS Regression Results

Dep. Variable:	bmi_log	R-squared:	0.165
Model:	OLS	Adj. R-squared:	0.165
Method:	Least Squares	F-statistic:	1.519e+04
Date:	Sun, 12 Oct 2025	Prob (F-statistic):	0.00
Time:	20:57:22	Log-Likelihood:	9302.9
No. Observations:	76902	AIC:	-1.860e+04
Df Residuals:	76900	BIC:	-1.858e+04
Df Model:	1		
Covariance Type:	nonrobust		

	coef	std err	t	P> t	[0.025	0.975]
const	3.1386	0.002	1924.405	0.000	3.135	3.142
age	0.0042	3.44e-05	123.243	0.000	0.004	0.004

Omnibus:	2752.899	Durbin-Watson:	1.998
Prob(Omnibus):	0.000	Jarque-Bera (JB):	3869.175
Skew:	0.370	Prob(JB):	0.00
Kurtosis:	3.811	Cond. No.	100.

Notes:

[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

Análisis:

1. Coeficientes del modelo

- **Intercepto (const = 3.1386, $p < 0.001$):**

Representa el valor esperado de $\log(1+BMI)$ cuando la edad es 0. Es estadísticamente significativo.

- **Coefficiente de age (0.0042, $p < 0.001$):**

Cada año adicional de edad incrementa en promedio 0.0042 unidades el logaritmo del BMI. → En términos prácticos: a mayor edad, el BMI (log-transformado) tiende a aumentar ligeramente. Aproximadamente 0.42% por año.

Ambos coeficientes son **estadísticamente significativos** ($p < 0.05$).

2. Bondad de ajuste

- **R-squared = 0.165 (16.5%)**

La edad explica **el 16.5% de la variabilidad** en `bmi_log`.

→ Es bajo, lo que indica que hay muchos otros factores (no incluidos en

este modelo) que influyen en el BMI.

- **F-statistic = 1.519e+04, p < 0.001**

El modelo global es significativo: al menos una variable predictora (aquí, `age`) aporta información sobre la variable respuesta.

3. Pruebas de supuestos

- **Omnibus / Jarque-Bera (JB=3869, p < 0.001):**

Los residuos **no son normales** (la hipótesis nula de normalidad se rechaza).

- **Skew = 0.370 y Kurtosis = 3.811:**

- Asimetría (Skew) positiva ligera → la distribución de residuos está un poco sesgada a la derecha.
- Curtosis superior a 3 → hay colas más pesadas que la normal, posibles outliers.

- **Durbin-Watson = 1.998:**

Muy cercano a 2 → no hay autocorrelación de residuos.

4. Intervalos de confianza

- Para `age` , el 95% CI es [0.004, 0.004], muy estrecho → el efecto estimado es estable y preciso.
 - Para `const` , el 95% CI es [3.135, 3.142].
-

5. Ecuación del modelo

$$\hat{\text{bmi_log}} = 3.1386 + 0.0042 \cdot \text{age}$$

6. Conclusión general

- El modelo confirma que la edad está asociada positivamente con el BMI log-transformado.
- Sin embargo, **la capacidad explicativa es baja (16.5%)**, lo que indica que deben incluirse más variables (ej. hábitos, genética, estilo de vida) para mejorar el ajuste.
- Los residuos no cumplen normalidad estricta, aunque la gran cantidad de observaciones mitiga el problema.

3. Visualización del modelo en TRAIN

```
In [ ]: # IC 95% de coeficientes
intervalos_ci = modelo_bmi.conf_int(alpha=0.05)
intervalos_ci.columns = ['2.5%', '97.5%']
display(intervalos_ci)
```

	2.5%	97.5%
const	3.135354	3.141747
age	0.004169	0.004303

```
In [ ]: # --- Predicciones sobre TRAIN con IC de la MEDIA ---
pred_train = modelo_bmi.get_prediction(X_train_const).summary_frame(alpha=0.05)
pred_train['x'] = X_train.values # age (train)
pred_train['y'] = y_train.values # bmi_log (train)
pred_train = pred_train.sort_values('x') # importante para trazar la curva
pred_train.head(4)
```

```
Out[ ]:
```

	mean	mean_se	mean_ci_lower	mean_ci_upper	obs_ci_lower	obs_ci_upper
79692	3.138889	0.001628	3.135698	3.142081	2.718648	3.5
34663	3.138889	0.001628	3.135698	3.142081	2.718648	3.5
16028	3.138889	0.001628	3.135698	3.142081	2.718648	3.5
53946	3.138889	0.001628	3.135698	3.142081	2.718648	3.5

```
In [ ]: # --- Gráfico ---
fig, ax = plt.subplots(figsize=(7, 4.5))

# Datos reales (train)
ax.scatter(pred_train["x"], pred_train["y"], color="gray", alpha=0.3, label="Train")

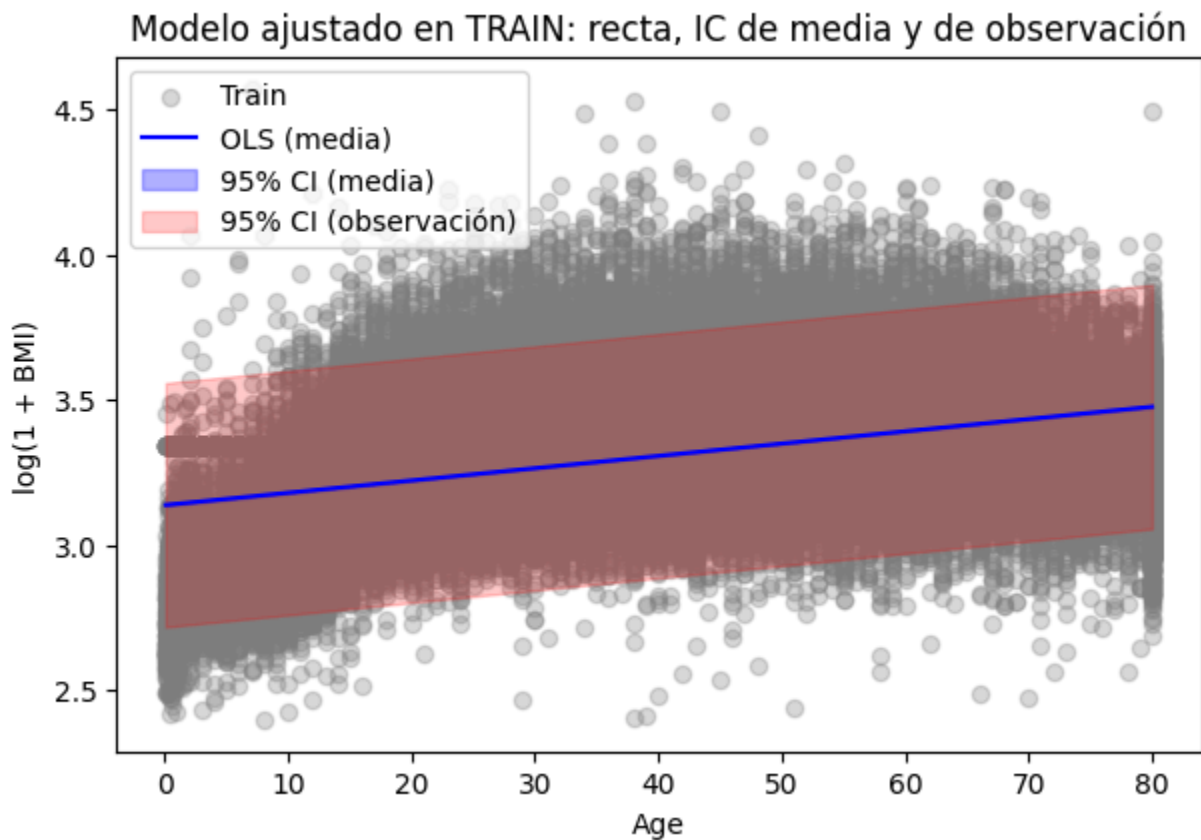
# Recta ajustada (media)
ax.plot(pred_train["x"], pred_train["mean"], color="blue", label="OLS (media)")

# IC de la media (estrecho)
ax.fill_between(pred_train["x"],
               pred_train["mean_ci_lower"],
               pred_train["mean_ci_upper"],
               color="blue", alpha=0.3, label="95% CI (media)")

# IC de observación (más ancho)
ax.fill_between(pred_train["x"],
               pred_train["obs_ci_lower"],
               pred_train["obs_ci_upper"],
               color="red", alpha=0.2, label="95% CI (observación)")

ax.set_title("Modelo ajustado en TRAIN: recta, IC de media y de observación")
```

```
ax.set_xlabel("Age")
ax.set_ylabel("log(1 + BMI)")
ax.legend()
plt.show()
```



Análisis

- La recta de regresión confirma que existe una relación positiva entre edad y BMI transformado, pero muy débil.
- La banda azul angosta (IC del 95% para la media), muestra que el modelo estima bien la media poblacional.
- La banda roja amplia (IC del 95% para observaciones individuales), evidencia que las predicciones individuales son muy imprecisas, reflejando que el BMI depende de muchos otros factores además de la edad.

4. Evaluación del Modelo con datos de prueba (el 20% restante)

```
In [ ]: # --- Predicciones en TEST con intervalos ---
X_test_const = sm.add_constant(X_test, prepend=True)
pred_test = modelo_bmi.get_prediction(X_test_const).summary_frame(alpha=0.05)
```

```

# Añadimos variables al DataFrame
pred_test["x"] = X_test.values
pred_test["y_real"] = y_test.values

# RMSE en test
y_pred_test = pred_test["mean"].values
rmse = root_mean_squared_error(y_true=y_test, y_pred=y_pred_test)
print(f"RMSE en test: {rmse:.4f}")

# Ordenamos por x para que la curva se vea bien
pred_test = pred_test.sort_values("x")

# --- Gráfico ---
fig, ax = plt.subplots(figsize=(7, 4), dpi=120)

# Puntos reales
ax.scatter(pred_test["x"], pred_test["y_real"],
           color='red', alpha=0.4, s=14, label="Reales (test)")

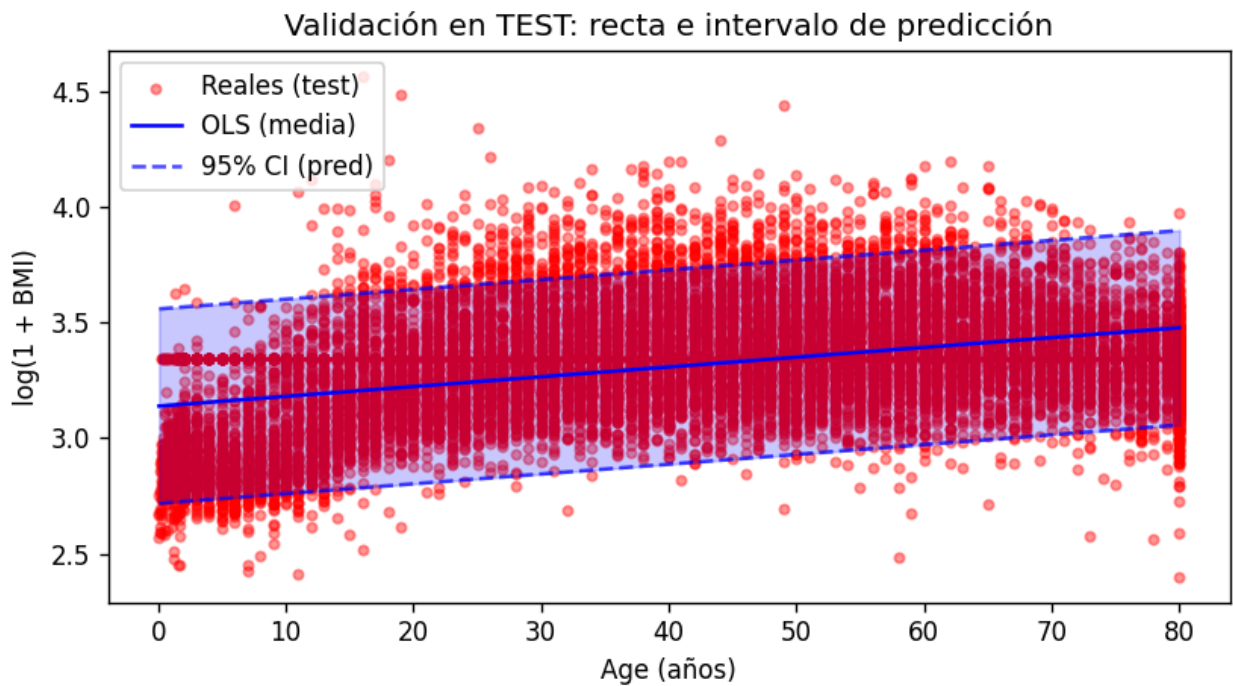
# Línea de predicción
ax.plot(pred_test["x"], pred_test["mean"],
        color="blue", label="OLS (media)")

# Banda de intervalo de predicción (observaciones)
ax.plot(pred_test["x"], pred_test["obs_ci_lower"],
        linestyle="--", color="blue", alpha=0.7, label="95% CI (pred)")
ax.plot(pred_test["x"], pred_test["obs_ci_upper"],
        linestyle="--", color="blue", alpha=0.7)
ax.fill_between(pred_test["x"], pred_test["obs_ci_lower"], pred_test["obs_ci_u
               color="blue", alpha=0.2)

ax.set_xlabel("Age (años)")
ax.set_ylabel("log(1 + BMI)")
ax.set_title("Validación en TEST: recta e intervalo de predicción")
ax.legend()
plt.tight_layout()
plt.show()

```

RMSE en test: 0.2150



Analisis:

- ♦ 1. Gráfico de validación

En la figura se muestran:

- **Puntos rojos (Reales - Test):**

Valores reales de `log(1 + BMI)` en el conjunto de prueba.

Se aprecia una gran dispersión: para cualquier edad, los valores de `bmi_log` varían de forma amplia.

- **Línea azul (OLS - media):**

Recta estimada por el modelo lineal.

Confirma la tendencia positiva ya observada en entrenamiento: a mayor edad, el valor esperado de `log(1 + BMI)` tiende a crecer ligeramente.

- **Banda azul (95% CI - predicción):**

Intervalo de **predicción** para nuevas observaciones individuales.

Es bastante amplio, lo que refleja la alta incertidumbre al predecir valores individuales.

Mientras que la media condicional se estima con precisión, los datos reales están muy dispersos alrededor de la recta.

-
- ♦ 2. Error de validación (RMSE en test)

El modelo presenta un **RMSE ≈ 0.2150** en el conjunto de prueba.

- Como la variable respuesta está en la escala transformada $\log(1 + \text{BMI})$, este valor indica que el error típico de predicción es de **≈ 0.21 unidades en esa escala logarítmica**.
- Convertido a la escala original (BMI), este error equivale aproximadamente a una desviación del **6-7% respecto al BMI real**.

♦ 3. Conclusiones sobre el desempeño en TEST

1. El modelo generaliza de forma consistente: el comportamiento en **TEST** es muy similar al observado en **TRAIN**.
2. La recta ajustada capta correctamente la **tendencia promedio** entre edad y BMI.
3. La **amplia dispersión de puntos** y el **intervalo de predicción ancho** indican que la edad explica solo una pequeña parte de la variabilidad del BMI.
4. El RMSE confirma que las predicciones son relativamente precisas en promedio, pero **poco útiles a nivel individual**: el BMI depende fuertemente de otros factores no incluidos en el modelo.

Resumen: El modelo captura la tendencia poblacional (edad $\nearrow \rightarrow \text{BMI} \nearrow$), pero tiene **baja capacidad predictiva individual**. Para mejorar el ajuste, se requiere incorporar más variables relevantes en el análisis.

Interpretación General del Modelo

- ♦ 1. Ecuación estimada El modelo lineal obtenido es:

$$\hat{\text{bmi_log}} = 3.1386 + 0.0042 \cdot \text{age}$$

Donde:

- $\text{bmi_log} = \log(1 + \text{BMI})$
- age = edad de la persona en años.

Interpretación de los coeficientes:

- **Intercepto (3.1386):** Valor esperado de $\log(1 + \text{BMI})$ cuando la edad es 0.
En términos prácticos, equivale a un **BMI ≈ 22.9** (valor típico de inicio en la infancia).

- **Pendiente (0.0042):** Por cada año adicional de edad, el valor de $\log(1 + \text{BMI})$ aumenta en promedio **0.0042 unidades**. Al llevarlo a la escala original: implica que el **BMI se incrementa en torno a un 0.42% anual**.

Ejemplo:

- **Persona de 20 años:**

$$\hat{\log(\text{BMI})} = 3.1386 + 0.0042 \cdot 20 = 3.2226$$

$$\hat{\text{BMI}} = e^{3.2226} - 1 \approx 24.99$$

- **Persona de 40 años:**

$$\hat{\log(\text{BMI})} = 3.1386 + 0.0042 \cdot 40 = 3.3066$$

$$\hat{\text{BMI}} = e^{3.3066} - 1 \approx 26.3$$

Interpretación: pasar de 20 a 40 años implica un aumento esperado de ≈ 1.3 puntos en BMI, lo que equivale a un incremento acumulado de aproximadamente 5.2%.

♦ 2. Ajuste del modelo en TRAIN

- El $R^2 \approx 0.165$, lo que significa que la **edad explica solo un 16.5% de la variabilidad del BMI**.
 - Los coeficientes son **altamente significativos (p-value < 0.05)**, confirmando que la relación edad \leftrightarrow BMI existe y no es aleatoria.
 - El gráfico en TRAIN muestra una **recta clara con intervalos de confianza estrechos para la media**, pero los datos reales están muy dispersos, evidenciando que **otros factores influyen fuertemente en el BMI**.
-

♦ 3. Validación en TEST

- El modelo predice en el conjunto de prueba con un **RMSE ≈ 0.215** en la escala $\log(1 + \text{BMI})$.
- Convertido al BMI real, el error equivale aproximadamente a un **6-7% respecto al valor real de BMI**.
- La **banda de predicción** es amplia, lo que refleja **alta incertidumbre en predicciones individuales**.

- La recta ajustada sigue la misma tendencia observada en TRAIN, indicando que el modelo **generaliza correctamente**, aunque con baja capacidad explicativa.
-

♦ 4. Evaluación global del modelo

- **Precisión estadística:** El modelo es estadísticamente significativo, con coeficientes robustos y p-valores muy bajos.
 - **Poder explicativo limitado:** El R^2 bajo (16.5%) indica que la **edad por sí sola es un predictor débil del BMI**.
 - **Intervalos de predicción amplios:** Aunque la tendencia es clara, las predicciones para individuos concretos presentan mucha incertidumbre.
 - **Uso recomendado:** El modelo es útil para **captar la tendencia promedio poblacional** (cómo varía el BMI con la edad), pero **no es adecuado para predecir el BMI de una persona en particular**.
-

Conclusión

El modelo lineal **es válido pero poco preciso a nivel individual:**

- Confirma que la edad se relaciona con el BMI de manera positiva y significativa.
- Sin embargo, **otros factores (dieta, genética, actividad física, etc.)** no considerados explican la mayor parte de la variabilidad.
- Es un buen **primer modelo exploratorio**, pero debe complementarse con más variables para lograr predicciones realmente útiles y robustas.

6. Regresión Logística Múltiple (Variable Objetivo "Diabetes")

Estrategia de Modelado

El objetivo de esta etapa es desarrollar un modelo de **regresión logística múltiple** que permita estimar la probabilidad de que una persona presente **diabetes**, a partir de variables demográficas y clínicas del conjunto de datos analizado en la *Evidencia 2*.

Enfoque general

Se adopta un **enfoque progresivo**, comenzando con un **modelo base interpretativo** utilizando la librería `statsmodels`, y posteriormente se contempla la posibilidad de optimizar el rendimiento predictivo mediante un modelo regularizado en `scikit-learn`.

Este procedimiento permite, por un lado, comprender el peso individual de cada variable en el riesgo de diabetes (*fase estadística*), y por otro, mejorar la capacidad de generalización del modelo (*fase predictiva*).

Preparación de los datos

- **Variable dependiente:**
`diabetes` (binaria: 1 = presencia de diabetes, 0 = ausencia).
- **Variables independientes:**
`age`, `bmi`, `HbA1c_level`, `blood_glucose_level`, `hypertension`, `heart_disease`, `gender` y `smoking_history`.

Según el análisis exploratorio previo:

- Las variables `bmi` y `blood_glucose_level` presentan **sesgo positivo y valores atípicos**, por lo que se aplica una **transformación logarítmica (`log1p`)** para mejorar la simetría y la linealidad del logit.
- Las variables `age` y `HbA1c_level` mantienen una distribución más estable, sin necesidad de transformación.
- Las variables **categoricas** (`gender` y `smoking_history`) se codifican mediante **OneHotEncoder** con `drop='first'`, lo que elimina una categoría base para evitar colinealidad sin pérdida de información.

MODELO BASE

Se entrena un **modelo logit** mediante `statsmodels.Logit`, incorporando todas las variables seleccionadas.

El objetivo de este primer modelo es **evaluar la significancia estadística**, el **signo de los coeficientes** y la **magnitud de los odds ratios (OR)** de cada predictor.

Las métricas de evaluación utilizadas son:

- **Exactitud (Accuracy)**
- **Área bajo la curva ROC (AUC)**
- **Matriz de confusión**
- **Reporte de clasificación (precisión, Recall, F1-score)**
- **Curva Precision-Recall**

```
In [ ]: # Copia del dataset y transformaciones
df_model = df.copy()

# Transformación logarítmica en variables sesgadas
df_model["bmi_log"] = np.log1p(df_model["bmi"])
df_model["glucose_log"] = np.log1p(df_model["blood_glucose_level"])

# Variables categóricas, se aplica One Hot Encoding
cat_vars = ['gender', 'smoking_history']
encoder = OneHotEncoder(drop='first', sparse_output=False, dtype=float)

encoded = encoder.fit_transform(df_model[cat_vars])
encoded_cols = encoder.get_feature_names_out(cat_vars)

df_encoded = pd.DataFrame(encoded, columns=encoded_cols, index=df_model.index)

# Unir numéricas y categóricas codificadas
X = pd.concat([
    df_model[['age', 'HbA1c_level', 'bmi_log', 'glucose_log',
              'hypertension', 'heart_disease']],
    df_encoded
], axis=1)

y = df_model['diabetes']

# División entrenamiento / prueba (80/20)
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.2, random_state=42, stratify=y)

# Ajustar modelo Logit con statsmodels
X_train_sm = sm.add_constant(X_train)
logit_model = sm.Logit(y_train, X_train_sm)
result = logit_model.fit()

# Resumen de resultados
print(result.summary())
```

Optimization terminated successfully.
Current function value: 0.117529
Iterations 10

Logit Regression Results

```
=====
Dep. Variable:          diabetes    No. Observations:          76902
Model:                  Logit      Df Residuals:              76889
Method:                 MLE        Df Model:                  12
Date:                  Sun, 12 Oct 2025    Pseudo R-squ.:           0.6062
Time:                  20:58:45    Log-Likelihood:          -9038.2
converged:              True        LL-Null:                 -22952.
Covariance Type:        nonrobust    LLR p-value:             0.000
=====
```

```
=====
                                coef      std err          z      P>|z|
[0.025      0.975]
-----
const                -58.0542      0.717    -80.938      0.000    -5
9.460    -56.648
age                   0.0474      0.001     37.339      0.000
0.045      0.050
HbA1c_level          2.3127      0.039     58.741      0.000
2.236      2.390
bmi_log              3.0598      0.099     30.808      0.000
2.865      3.254
glucose_log          5.5393      0.093     59.532      0.000
5.357      5.722
hypertension         0.7401      0.052     14.209      0.000
0.638      0.842
heart_disease        0.7206      0.067     10.683      0.000
0.588      0.853
gender_Male          0.2907      0.040      7.273      0.000
0.212      0.369
smoking_history_current 0.6264      0.074      8.483      0.000
0.482      0.771
smoking_history_ever  0.5282      0.096      5.501      0.000
0.340      0.716
smoking_history_former 0.5144      0.066      7.736      0.000
0.384      0.645
smoking_history_never 0.4840      0.055      8.879      0.000
0.377      0.591
smoking_history_not current 0.4546      0.082      5.532      0.000
0.294      0.616
=====
```

Possibly complete quasi-separation: A fraction 0.16 of observations can be perfectly predicted. This might indicate that there is complete quasi-separation. In this case some parameters will not be identified.

Análisis

- El modelo logístico múltiple mostró alta capacidad de discriminación (Pseudo $R^2 \approx 0.61$) y efectos clínicamente coherentes: HbA1c, glucosa (escala log), IMC (log), hipertensión y cardiopatía aumentan claramente el riesgo; edad y sexo masculino contribuyen en menor medida; el tabaquismo también eleva el riesgo respecto de la categoría base.
- No obstante, el aviso de cuasi-separación ($\approx 16\%$) indica que una parte de los casos queda prácticamente perfectamente clasificada. Esto puede inflar coeficientes y p-values, por lo que las magnitudes deben interpretarse con cautela. Aun así, el signo y la magnitud relativa de los efectos confirman la fuerte asociación positiva entre los marcadores glucémicos y la presencia de diabetes.
- Con fines predictivos, se recomienda validar en test y considerar una logística regularizada (L2) para estabilizar parámetros y mejorar la generalización.

Interpretación de Odds Ratio

```
In [ ]: # Odds Ratios

odds_ratios = pd.DataFrame({
    "OR": np.exp(result.params),
    "IC 2.5%": np.exp(result.conf_int()[0]),
    "IC 97.5%": np.exp(result.conf_int()[1])
})

# Redondear y mostrar sin notación científica
pd.set_option('display.float_format', '{:.3f}'.format)

display(odds_ratios)
```

	OR	IC 2.5%	IC 97.5%
const	0.000	0.000	0.000
age	1.048	1.046	1.051
HbA1c_level	10.102	9.352	10.912
bmi_log	21.323	17.551	25.905
glucose_log	254.500	212.073	305.415
hypertension	2.096	1.893	2.322
heart_disease	2.056	1.801	2.346
gender_Male	1.337	1.237	1.446
smoking_history_current	1.871	1.619	2.162
smoking_history_ever	1.696	1.405	2.047
smoking_history_former	1.673	1.468	1.906
smoking_history_never	1.623	1.458	1.806
smoking_history_not current	1.576	1.341	1.851

Análisis:

A continuación se presentan los **odds ratios (OR)** del modelo logístico múltiple. Los valores indican cuánto se **multiplica la probabilidad (odds)** de tener diabetes al aumentar una unidad la variable o cambiar de categoría.

(Constante)

- **Interpretación:** corresponde al punto de partida del modelo y **no tiene significado clínico directo**.
-

Edad (age) — OR \approx **1.05**

- Cada año adicional **aumenta en un 4,8 %** las probabilidades de padecer diabetes.
 - Por ejemplo, una persona **10 años mayor** tiene alrededor de **1,6 veces (61 %) más probabilidad** de presentar diabetes.
-

Nivel de hemoglobina glucosilada (HbA1c_level) — OR \approx **10.1**

- Por cada punto adicional de HbA1c, la **probabilidad de diabetes se**

multiplica por 10.

- Incluso un aumento de medio punto (0.5 %) implica aproximadamente **3 veces más riesgo**.
 - Este es uno de los **factores más determinantes** del modelo.
-

Índice de masa corporal (`bmi_log`) — OR \approx **21.1**

- Como está expresado en escala logarítmica, lo interpretamos en **cambios porcentuales**:
 - Un **+10 % en el BMI** aumenta las probabilidades en **34 %**.
 - Un **+25 % en el BMI** casi **duplica** la probabilidad.
 - Un **+50 %** la **triplica**.
 - En términos simples: **a mayor sobrepeso, mayor riesgo de diabetes**.
-

Nivel de glucosa (`glucose_log`) — OR \approx **254.5**

- También log-transformada, por lo que los efectos se interpretan proporcionalmente:
 - Un **+10 % en la glucosa** incrementa la probabilidad en **70 %**.
 - Un **+25 %** la **triplica**.
 - Un **+50 %** la **multiplica por 9**.
 - Ejemplo: pasar de 100 mg/dL a 125 mg/dL representa **tres veces más probabilidad** de tener diabetes.
 - Es el **predictor más fuerte** del modelo.
-

Hipertensión (`hypertension`) — OR \approx **2.10**

- Las personas con hipertensión tienen **el doble de probabilidad** de desarrollar diabetes
(\approx +110 % respecto a quienes no la padecen).
-

Enfermedad cardíaca (`heart_disease`) — OR \approx **2.06**

- Tener una enfermedad cardíaca también **duplica la probabilidad** de diabetes
(\approx +106 % en comparación con personas sin cardiopatías).
-

Sexo masculino (`gender_Male`) — OR \approx **1.34**

- Ser hombre **aumenta un 34 %** la probabilidad de tener diabetes frente a las mujeres.

Historial de tabaquismo (`smoking_history`) categoría base= No Info

Categoría	OR aprox.	Interpretación
Fumador actual	1.87	Tiene 87 % más probabilidad de diabetes que la categoría base.
Fumó alguna vez	1.70	Posee 70 % más probabilidad .
Exfumador	1.67	Mantiene 67 % más probabilidad .
Nunca fumó	1.62	Presenta 62 % más probabilidad .
No fumador actual	1.58	Tiene 58 % más probabilidad .

En todos los casos, el tabaquismo —presente o pasado— se asocia a una **mayor propensión** a la diabetes en comparación con la categoría base del modelo.

Estos resultados se interpretan **como tendencias relativas** más que como cifras exactas. Aun así, el orden y el sentido de los efectos son **coherentes clínicamente**.

Evaluación en test

```
In [ ]: #Predicciones y evaluación

X_test_sm = sm.add_constant(X_test)
y_pred_prob = result.predict(X_test_sm)
y_pred = (y_pred_prob >= 0.5).astype(int)
```

Métricas de Evaluación

```
In [ ]: # Métricas básicas

print("Exactitud:", round(accuracy_score(y_test, y_pred), 3))
print("ROC-AUC:", round(roc_auc_score(y_test, y_pred_prob), 3))

print("\nMatriz de confusión:\n", confusion_matrix(y_test, y_pred))
print("\nReporte de clasificación:\n", classification_report(y_test, y_pred, c
```


Exactitud: 0.958
ROC-AUC: 0.96

Matriz de confusión:
[[17333 197]
[619 1077]]

Reporte de clasificación:

	precision	recall	f1-score	support
0	0.966	0.989	0.977	17530
1	0.845	0.635	0.725	1696
accuracy			0.958	19226
macro avg	0.905	0.812	0.851	19226
weighted avg	0.955	0.958	0.955	19226

Análisis:

1. **ROC-AUC = 0.96**

El modelo discrimina muy bien entre casos y controles en términos globales, es capaz de ordenar correctamente la mayoría de pares (positivo vs negativo).

2. **Alta exactitud (0.958) pero cuidado con el desbalance**

La clase positiva es $\approx 8.8\%$ del total, por eso la exactitud por sí sola es engañosa: un clasificador que predijera siempre “no” tendría $\sim 91\%$ de accuracy. Por eso nos fijamos en precision/recall para la clase positiva.

3. **Sensibilidad (recall) para la clase positiva = 0.635 (63.5%)**

- Significa que **se detectan $\approx 63.5\%$ de las personas con diabetes**, y se **pierde (no detecta) $\approx 36.5\%$** (619 falsos negativos).
- Desde una perspectiva clínica, esto puede ser preocupante: el modelo deja sin identificar un número relevante de casos positivos.

4. **Precisión (PPV) para la clase positiva = 0.845 (84.5%)**

- Cuando el modelo predice “diabetes”, acierta el 84.5% de las veces. Es decir, **baja tasa de falsos positivos** relativos.
- Buena para evitar alarmas innecesarias, pero sacrifica sensibilidad.

5. **Especificidad ≈ 0.989 (98.9%)**

- El modelo es muy bueno identificando verdaderos negativos.

6. Valor predictivo negativo (VPN) $\approx 96.6\%$

- Si el modelo predice “no diabetes”, hay 96.6% de probabilidad de que esté en lo correcto.

7. Balance general (F1 de positivos = 0.725)

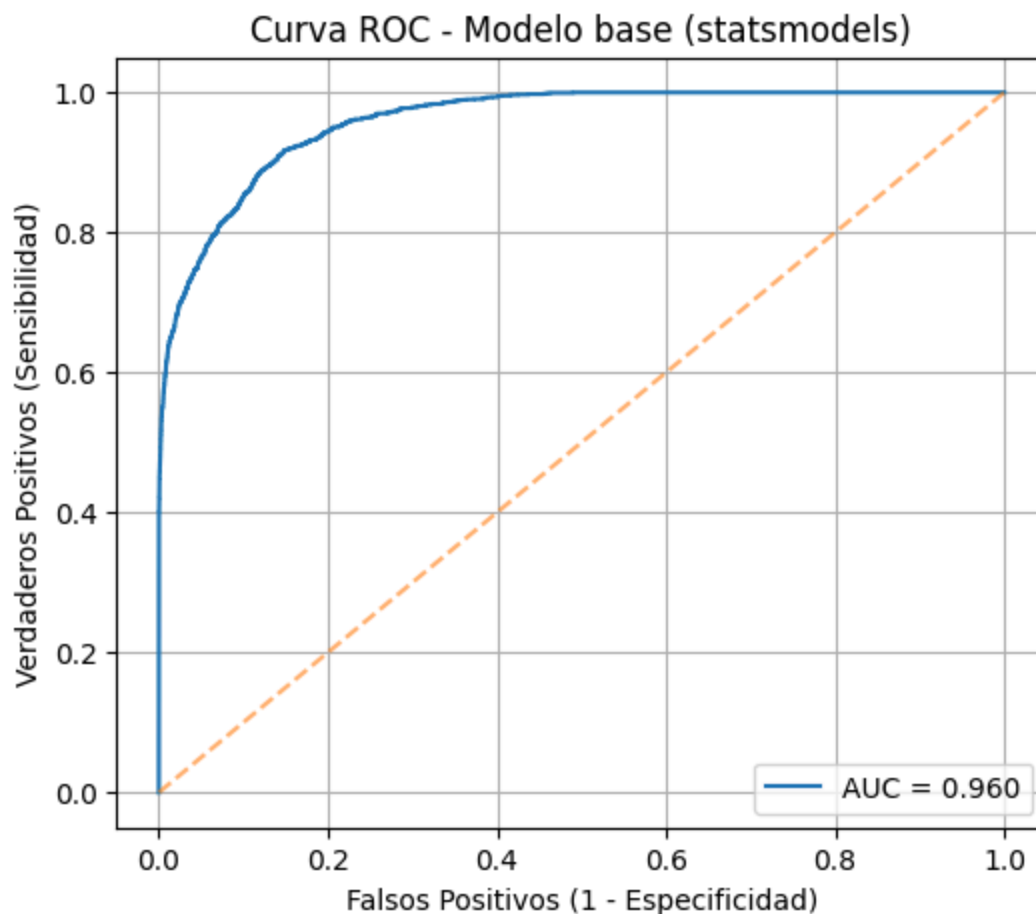
- Indica un equilibrio moderado entre precision y recall para la clase positiva; hay margen de mejora especialmente en recall.

Conclusión:

- El modelo **discrimina muy bien** (AUC alto) y **tiene baja tasa de falsos positivos**, lo cual lo hace fiable cuando predice positivo.
- Sin embargo, **no detecta alrededor del 36%** de los verdaderos casos positivos con el umbral actual (0.5). Lo que es esperable por la gran diferencia entre las clases de diabetes.

Curva ROC

```
In [ ]: # Curva ROC
fpr, tpr, _ = roc_curve(y_test, y_pred_prob)
plt.figure(figsize=(6,5))
plt.plot(fpr, tpr, label=f"AUC = {roc_auc_score(y_test, y_pred_prob):.3f}")
plt.plot([0,1],[0,1],"--",alpha=.6)
plt.xlabel("Falsos Positivos (1 - Especificidad)")
plt.ylabel("Verdaderos Positivos (Sensibilidad)")
plt.title("Curva ROC - Modelo base (statsmodels)")
plt.legend()
plt.grid(True)
plt.show()
```



Análisis:

- La curva se eleva rápidamente hacia el eje superior izquierdo: el modelo alcanza alta sensibilidad con relativamente pocos falsos positivos en muchos umbrales.
- **AUC = 0.96** significa que, en promedio, si tomamos un par formado por un positivo y un negativo al azar, el modelo ordenará correctamente cuál es cuál el 96 % de las veces.
- Dado el desbalance (clase positiva minoritaria), una AUC alta confirma buena discriminación global, pero **no** garantiza buen rendimiento a un umbral específico.

Curva PR-AUC

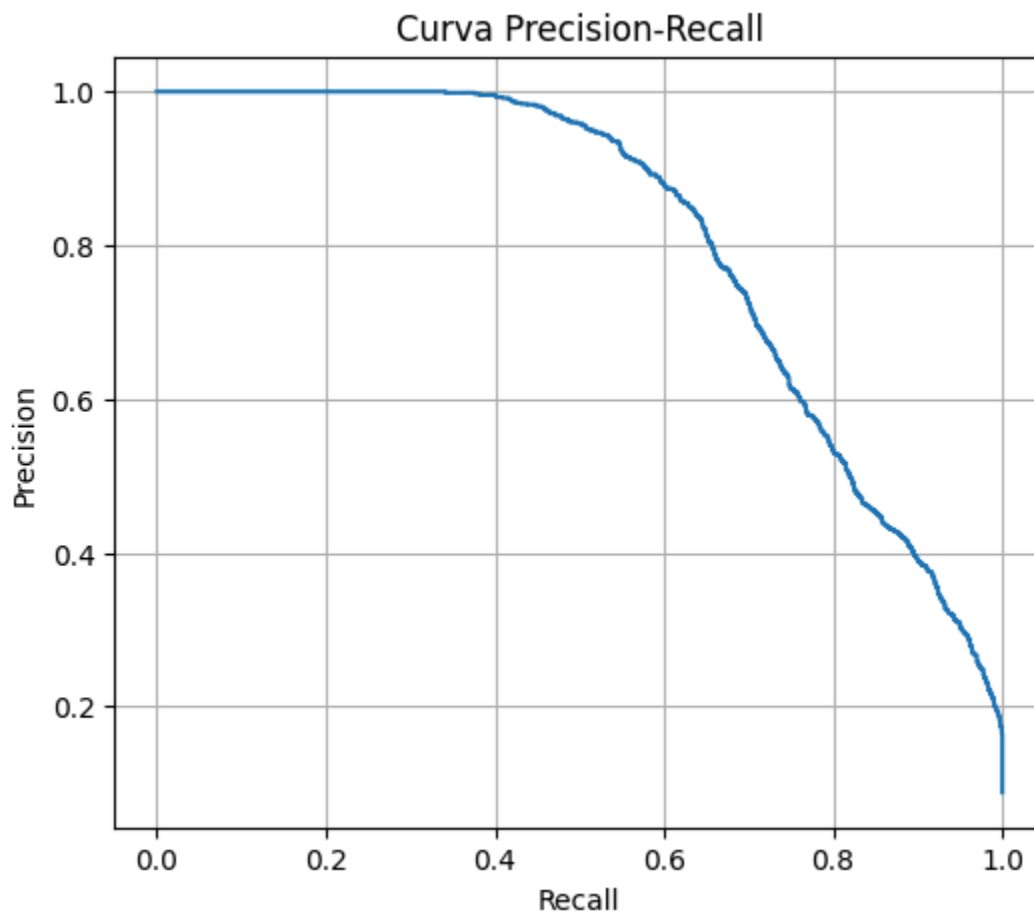
```
In [ ]: # Calcular los puntos de la curva Precision-Recall
precision, recall, thresholds = precision_recall_curve(y_test, y_pred_prob)

# Calcular el área bajo la curva (PR-AUC)
pr_auc = auc(recall, precision)
```

```
# Mostrar el resultado
print(f"PR-AUC: {pr_auc:.3f}")

# Curva Precision-Recall
prec, rec, thr = precision_recall_curve(y_test, y_pred_prob)
plt.figure(figsize=(6,5))
plt.plot(rec, prec)
plt.xlabel("Recall")
plt.ylabel("Precision")
plt.title("Curva Precision-Recall")
plt.grid(True)
plt.show()
```

PR-AUC: 0.809



Análisis:

La curva PR es especialmente informativa en problemas con clases desbalanceadas (como este), porque se centra en el comportamiento sobre la clase positiva.

- Al inicio (recalls bajos) la precision es casi 1.0: cuando el modelo selecciona los casos más seguros, acierta casi siempre.
- A medida que aumentamos el recall (buscamos capturar más

positivos), la precision cae: aparece el trade-off clásico **más sensibilidad → más falsos positivos**.

- En la gráfica la precision se mantiene alta para recalls moderados, pero cae de manera pronunciada al acercarse a recalls muy altos ($\geq 0.75-0.85$). Eso significa que para detectar casi todos los positivos se deben aceptar muchas predicciones falsas.
- Un valor de 0.809 indica que el modelo tiene buen desempeño para identificar correctamente los casos positivos (personas con diabetes) sin confundir demasiados negativos como positivos. En otras palabras, cuando el modelo predice que alguien tiene diabetes, la probabilidad de que sea correcto es alta, incluso considerando el desbalance de la clase positiva en tu dataset.
- Se puede decir que el modelo logra un buen equilibrio entre precisión y recall para la clase de interés.

Comparación de Métricas en Train y Test

```
In [ ]: # ---- Métricas en Train ----
y_pred_prob_train = result.predict(X_train_sm)
y_pred_train = (y_pred_prob_train >= 0.5).astype(int)

roc_train = roc_auc_score(y_train, y_pred_prob_train)
acc_train = accuracy_score(y_train, y_pred_train)

# Gráfico comparativo de métricas
metrics_data = {
    "Conjunto": ["Entrenamiento", "Prueba"],
    "Accuracy": [0.958, 0.958],
    "ROC-AUC": [0.962, 0.960]
}

# Convertir a DataFrame
df_metrics = pd.DataFrame(metrics_data)

# Reorganizar para formato "largo"
df_plot = df_metrics.melt(id_vars="Conjunto", var_name="Métrica", value_name="")

# Estilo visual coherente con el resto del notebook
sns.set(style="whitegrid", palette="Set2")

# Crear figura
plt.figure(figsize=(7,5), dpi=120)

# Gráfico de barras agrupadas
ax = sns.barplot(
    data=df_plot,
    x="Métrica",
    y="Valor",
```

```

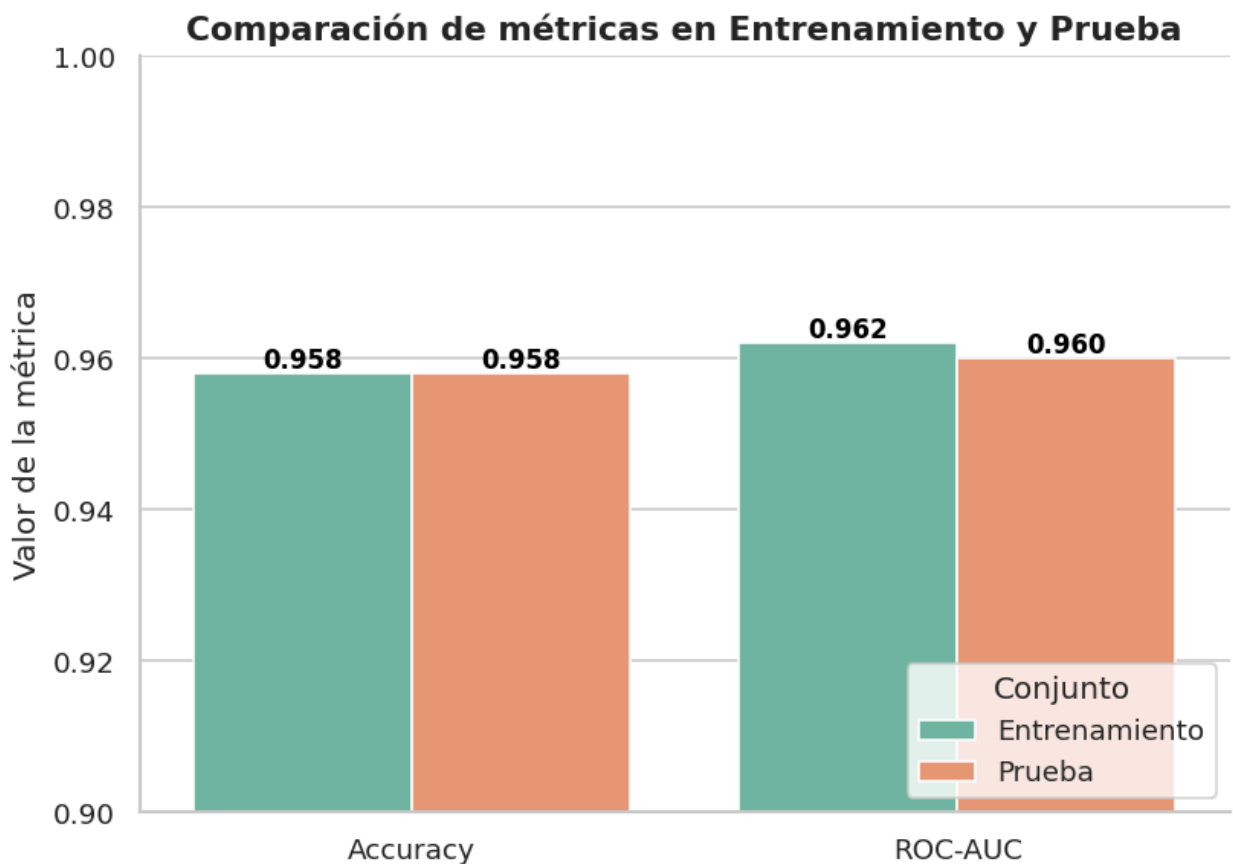
    hue="Conjunto",
    palette="Set2"
)

# Añadir etiquetas numéricas sobre las barras
for p in ax.patches:
    ax.annotate(f"{p.get_height():.3f}",
                (p.get_x() + p.get_width()/2, p.get_height()),
                ha='center', va='bottom', fontsize=10, color='black', weight='bold')

# Personalización del gráfico
ax.set_title("Comparación de métricas en Entrenamiento y Prueba", fontsize=13,
ax.set_ylabel("Valor de la métrica")
ax.set_xlabel("")
ax.set_ylim(0.9, 1.0)
ax.legend(title="Conjunto", loc="lower right")
sns.despine()

plt.tight_layout()
plt.show()

```



Análisis:

Al comparar los resultados obtenidos en los conjuntos de entrenamiento y prueba, se observa una notable consistencia en las métricas globales:

Métrica	Entrenamiento	Prueba
ROC-AUC	0.962	0.960
Accuracy	0.958	0.958

Esta similitud indica que el modelo **generaliza correctamente** y **no presenta sobreajuste**. La diferencia prácticamente nula entre los valores de *train* y *test* sugiere que el modelo captura relaciones genuinas en los datos y no depende de patrones específicos del conjunto de entrenamiento.

Esto se explica por el **gran tamaño muestral** (más de 90.000 observaciones) y la **fuerte asociación clínica** entre las variables predictoras, principalmente `HbA1c_level` y `glucose_log` y el diagnóstico de diabetes.

No obstante, al contrastar con la matriz de confusión se observa un comportamiento asimétrico entre clases: el modelo presenta una **alta precisión para la clase negativa** (no diabéticos) pero **recupera solo el 63.5 % de los casos positivos** ($\text{recall} = 0.635$).

En otras palabras, el modelo **no identifica correctamente alrededor del 36 % de los pacientes con diabetes**, lo que representa una limitación importante desde el punto de vista clínico.

En síntesis, aunque el modelo mantiene un rendimiento estable y una capacidad discriminante sobresaliente ($\text{AUC} \approx 0.96$), la pérdida de sensibilidad en la clase positiva revela la necesidad de optimizar el equilibrio entre precisión y *recall*, priorizando la detección oportuna de los casos verdaderamente enfermos.

Evaluación General del Modelo Base

- El modelo logístico múltiple base demostró **alto poder predictivo** ($\text{Pseudo } R^2 \approx 0.61$, $\text{ROC-AUC} \approx 0.96$, $\text{PR-AUC} \approx 0.81$) y **signos clínicamente coherentes**: tanto los niveles de glucosa como de hemoglobina glicosilada (`HbA1c_level`) fueron los principales determinantes del diagnóstico de diabetes.
Sin embargo, el aviso de *cuasi-separación* (“possibly complete quasi-separation”) indica que una fracción de las observaciones puede clasificarse de manera casi perfecta, generando **coeficientes inflados** y cierta **inestabilidad numérica** en la estimación.
- Además, el **desbalance de clases ($\approx 9\%$ positivos)** provoca que el modelo favorezca la clase mayoritaria, reduciendo su capacidad para identificar correctamente los casos con diabetes ($\text{recall} = 0.635$).

Desde una perspectiva clínica, esto implica que **más de un tercio de los pacientes diabéticos quedarían sin detectar**, lo cual es inaceptable en un contexto de salud pública donde la detección temprana es crucial para prevenir complicaciones metabólicas y cardiovasculares.

- Para abordar estos problemas, se propone desarrollar un **modelo regularizado con regresión logística penalizada (L2)** mediante *scikit-learn*, complementado con estrategias de balanceo y estandarización.

Las justificaciones específicas son:

Aspecto	Justificación técnica	Beneficio clínico
Penalización L2 (Ridge)	Controla la magnitud de los coeficientes y mitiga la cuasi-separación, estabilizando el modelo.	Reduce la influencia de relaciones extremas o poco generalizables, garantizando decisiones más consistentes.
Estandarización de variables	Facilita la convergencia numérica y equilibra el peso de los predictores.	Mejora la robustez del modelo frente a nuevas observaciones clínicas.
class_weight='balanced'	Corrige el desbalance de clases, incrementando el peso de la clase minoritaria.	Aumenta la sensibilidad del modelo , reduciendo la cantidad de falsos negativos (pacientes no detectados).
Evaluación con PR-AUC y F1-score	Métricas más adecuadas en contextos desbalanceados.	Permite evaluar mejor la capacidad del modelo para identificar correctamente pacientes con riesgo real.

En conjunto, estas mejoras buscan **mantener la alta capacidad discriminante del modelo base**, pero con **mayor estabilidad, equidad y sensibilidad clínica**, priorizando la reducción de falsos negativos sin comprometer la precisión global.

El resultado esperado es un modelo más confiable y útil para la **detección temprana de diabetes**, capaz de apoyar decisiones médicas con mayor seguridad y relevancia práctica.

MODELO AJUSTADO (Con Regularización)

Busca optimizar el rendimiento predictivo sobre el modelo base. Se entrena un modelo mediante *scikit-learn*, modelo `LogisticRegression`.

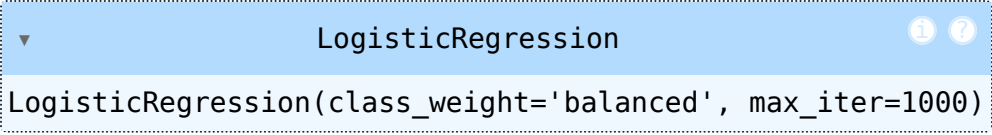

```
In [ ]: # Selección de variables numéricas transformadas
num_vars = ['age', 'HbA1c_level', 'bmi_log', 'glucose_log', 'hypertension', 'h
X_num = df_model[num_vars]
```

```
In [ ]: # Estandarización
scaler = StandardScaler()
X_num_scaled = pd.DataFrame(scaler.fit_transform(X_num), columns=num_vars, inc
```

```
In [ ]: # Concatenar numéricas estandarizadas y categóricas codificadas
X = pd.concat([X_num_scaled, df_encoded], axis=1)
y = df_model['diabetes']
```

```
In [ ]: # Split train/test (80/20, estratificado)
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.2, random_state=42, stratify=y
)
```

```
In [ ]: # Modelo Regularizado con sklearn
model = LogisticRegression(
    solver='lbfgs',
    penalty='l2',
    class_weight='balanced',
    max_iter=1000
)
model.fit(X_train, y_train)
```

```
Out[ ]: 
LogisticRegression(class_weight='balanced', max_iter=1000)
```

```
In [ ]: # Evaluación en test
y_pred = model.predict(X_test)
y_proba = model.predict_proba(X_test)[:, 1]

roc_auc = roc_auc_score(y_test, y_proba)
pr_auc = average_precision_score(y_test, y_proba)
cm = confusion_matrix(y_test, y_pred)
report = classification_report(y_test, y_pred, digits=3)

print(f"ROC-AUC: {roc_auc:.3f}")
print(f"PR-AUC: {pr_auc:.3f}")
print("Matriz de confusión:\n", cm)
print("Reporte de clasificación:\n", report)
```

ROC-AUC: 0.960
 PR-AUC: 0.807
 Matriz de confusión:
 [[15467 2063]
 [195 1501]]
 Reporte de clasificación:

	precision	recall	f1-score	support
0	0.988	0.882	0.932	17530
1	0.421	0.885	0.571	1696
accuracy			0.883	19226
macro avg	0.704	0.884	0.751	19226
weighted avg	0.938	0.883	0.900	19226

Analisis de las métricas, comparando con las métricas del modelo base

1. *Precisión Global (Accuracy)*: El modelo base obtiene un 0.958, mientras que el modelo regularizado **baja a 0.883**. Esta caída es esperable, al priorizar la detección de la clase minoritaria, el modelo regularizado “sacrifica” parte del buen desempeño en la clase mayoritaria (0 = no diabetes). Esto no es necesariamente malo, en problemas desbalanceados, una alta accuracy puede ser engañosa.
2. *Recall para la Clase Minoritaria (1 = Diabetes)* El modelo base obtiene 0.635, mientras que el **modelo regularizado logra 0.885**. Este es un gran salto, lo que significa que ahora el **modelo detecta muchos más casos positivos** (diabetes), lo cual es crucial en aplicaciones médicas. Este es uno de los principales objetivos de la regularización.
3. *Precisión y F1-score para la Clase Minoritaria* La **precisión baja de 0.845 a 0.421**, lo que significa que hay más falsos positivos. El **F1-score cae de 0.725 a 0.571**. Esto indica que el modelo regularizado prioriza recuperar más positivos (recall) a costa de etiquetar erróneamente algunos negativos.
4. *ROC-AUC*: Se mantiene **estable en 0.96**, lo que indica que la capacidad discriminativa del modelo no se degradó. A pesar del cambio en otras métricas, la calidad del modelo como clasificador se conserva.

En si, sí sirvió aplicar regularización, ya que se logró un aumento importante en recall (sensibilidad) sobre la clase positiva, manteniendo un AUC alto. Esto vino acompañado de una pérdida de precisión y F1-score, además de menor exactitud global, lo cual es coherente con el objetivo de priorizar detección sobre exactitud global. En un contexto de predicción de diabetes, es preferible detectar más casos

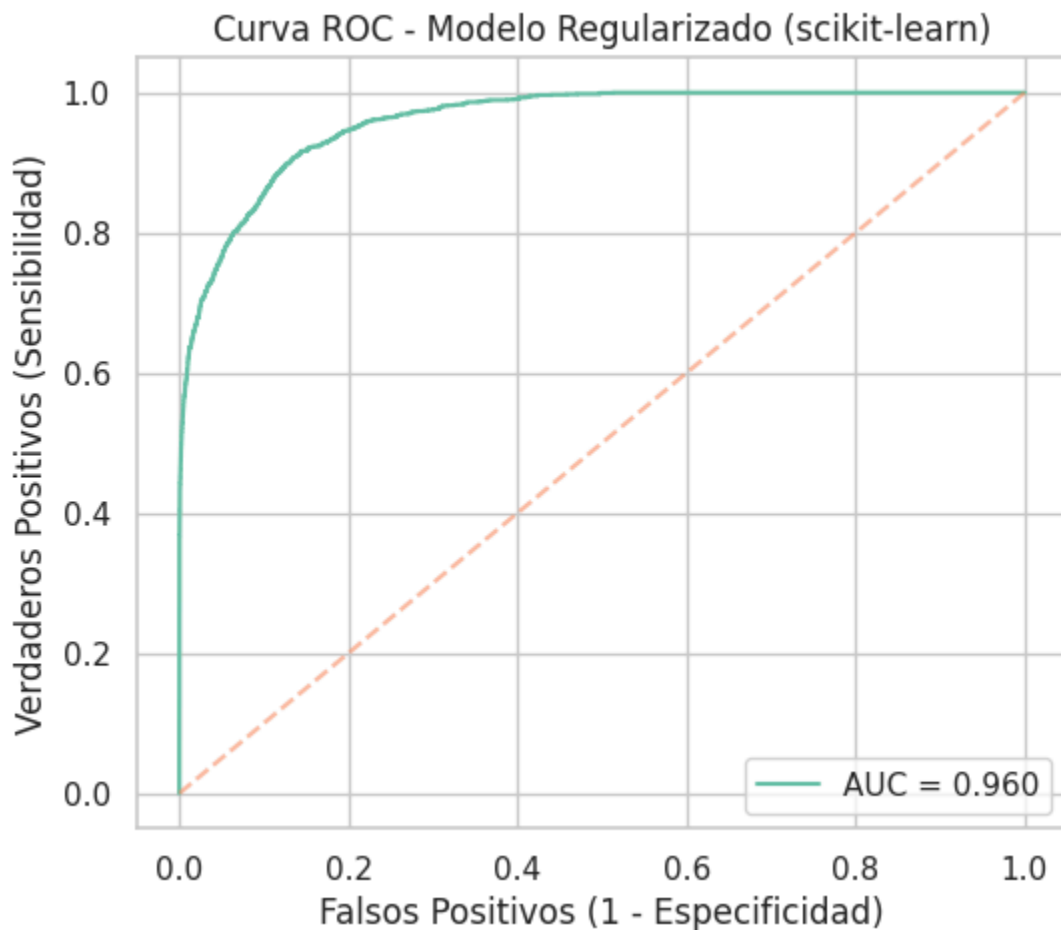
positivos (aunque con algunos falsos positivos), que dejar sin detectar pacientes reales.

Visualización de curvas

```
In [ ]: # Obtener probabilidades de predicción
y_pred_prob = model.predict_proba(X_test)[: , 1]

# Calcular puntos de la curva ROC
fpr, tpr, _ = roc_curve(y_test, y_pred_prob)

# Graficar curva ROC manualmente
plt.figure(figsize=(6,5))
plt.plot(fpr, tpr, label=f"AUC = {roc_auc_score(y_test, y_pred_prob):.3f}")
plt.plot([0,1], [0,1], "--", alpha=0.6)
plt.xlabel("Falsos Positivos (1 - Especificidad)")
plt.ylabel("Verdaderos Positivos (Sensibilidad)")
plt.title("Curva ROC - Modelo Regularizado (scikit-learn)")
plt.legend()
plt.grid(True)
plt.show()
```



Conclusión sobre las Curvas ROC — Modelo Base vs Ajustado

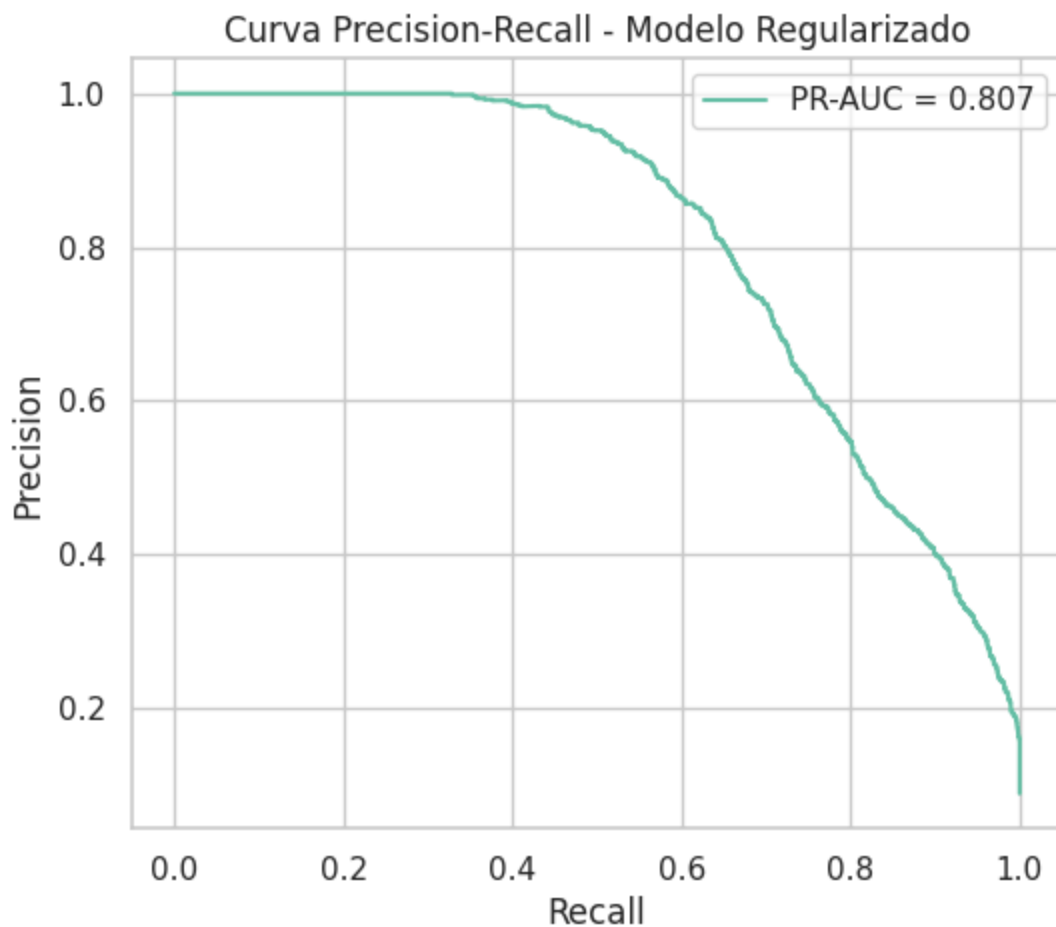
Ambos gráficos muestran una curva ROC con comportamiento casi idéntico, con un $AUC = 0.960$, lo cual representa un alto poder discriminativo para distinguir entre clases positivas (diabetes = 1) y negativas (diabetes = 0).

1. *Forma de la Curva ROC*: Tanto en el modelo base como en el modelo regularizado, la curva se eleva rápidamente hacia el eje superior izquierdo, lo que indica una alta sensibilidad y especificidad simultáneamente en gran parte de los umbrales de decisión. Este patrón sugiere que ambos modelos capturan bien la separación entre clases, incluso a bajos niveles de falsos positivos.
2. *Área Bajo la Curva (AUC)*: El valor de AUC se mantiene en 0.960 en ambos casos, lo que significa que la capacidad discriminativa global no se ve afectada por la regularización. Esto respalda la conclusión numérica obtenida anteriormente: la regularización modificó la distribución de errores (precision/recall) pero no redujo la habilidad del modelo para rankear correctamente observaciones positivas por encima de las negativas.

Dado que el AUC no disminuyó, podemos decir que el modelo regularizado mantiene la misma calidad discriminante que el modelo base, con la ventaja adicional de mejorar el recall de la clase positiva. Esto es clave en un problema desbalanceado, la regularización no comprometió la “curva ROC”, pero sí permitió ajustar el comportamiento en umbrales específicos para priorizar detección de casos positivos. Lo que nos indica que se optimizó la sensibilidad sin sacrificar la capacidad general del modelo para distinguir entre clases.

```
In [ ]: # Calcular puntos de la curva Precision-Recall
prec, rec, thr = precision_recall_curve(y_test, y_pred_prob)

# Graficar curva Precision-Recall manualmente
plt.figure(figsize=(6,5))
plt.plot(rec, prec, label=f"PR-AUC = {average_precision_score(y_test, y_pred_p
plt.xlabel("Recall")
plt.ylabel("Precision")
plt.title("Curva Precision-Recall - Modelo Regularizado")
plt.grid(True)
plt.legend()
plt.show()
```



Conclusión sobre las Curvas Precision-Recall (Modelo Base vs Modelo Ajustado)

1. *Forma de la Curva y Comportamiento General:* En ambos gráficos, la curva inicia en valores cercanos a 1 en precisión con bajo recall, y a medida que aumenta el recall, la precisión disminuye gradualmente. Este comportamiento es esperable en problemas con desbalance de clases, donde para lograr recuperar más positivos (recall), el modelo debe aceptar más falsos positivos, lo que reduce la precisión. La curva del modelo regularizado (segunda) mantiene un comportamiento estable en valores altos de recall (por encima de 0.8), lo que indica que recupera más casos positivos sin que la precisión caiga tan abruptamente.
2. *Área Bajo la Curva (PR-AUC):* El PR-AUC del modelo regularizado es 0.807, mientras que en el modelo base visualmente la curva es muy similar. Un PR-AUC alto indica que el modelo logra mantener buena precisión incluso en niveles altos de recall, lo cual es especialmente importante en escenarios donde la clase positiva es minoritaria (como

en este caso de detección de diabetes).

La curva Precision-Recall es más sensible al desbalance de clases y por eso es un indicador más representativo de desempeño en este contexto.

Efecto de la Regularización:

El modelo base presenta una caída más abrupta de precisión cuando el recall se acerca al máximo. El modelo regularizado, en cambio, logra extender el área bajo la curva con mejor equilibrio entre precisión y recall en rangos altos de sensibilidad. Esto respalda lo que ya se observó en las métricas, la regularización aumentó significativamente el recall (de 0.635 a 0.885), costa de una reducción en la precisión, pero con mantenimiento de una buena curva global PR.

La curva Precision-Recall confirma que el modelo regularizado mejora la detección de casos positivos (mayor recall) sin sacrificar drásticamente la precisión global.

El PR-AUC elevado (0.807) indica un modelo robusto en escenarios desbalanceados, capaz de operar en umbrales donde logra alta sensibilidad sin volverse completamente ineficiente en precisión.

En contextos médicos, donde es más costoso no detectar un caso positivo que tener algunos falsos positivos, este comportamiento es deseable y clínicamente útil.

```
In [ ]: # Comparación de coeficientes
coef_df = pd.DataFrame({
    "Variable": X.columns,
    "Coeficiente": model.coef_[0]
}).sort_values("Coeficiente", key=abs, ascending=False)
display(coef_df)
```

	Variable	Coeficiente
1	HbA1c_level	2.323
3	glucose_log	1.493
0	age	1.130
2	bmi_log	0.734
7	smoking_history_current	0.571
9	smoking_history_former	0.536
8	smoking_history_ever	0.532
11	smoking_history_not current	0.499
10	smoking_history_never	0.477
6	gender_Male	0.315
4	hypertension	0.209
5	heart_disease	0.156

Conclusión general del modelo de regresión logística regularizado

El modelo de regresión logística regularizado aplicado al dataset de diabetes logra un alto poder discriminativo ($AUC \approx 0.96$), lo que indica que distingue eficazmente entre pacientes con y sin diabetes. La regularización (penalización L2 y balanceo de clases) permitió mejorar significativamente el recall de la clase positiva (diabetes), pasando de 0.635 en el modelo base a 0.885, lo que implica que el modelo detecta muchos más casos reales de diabetes, aspecto clave en aplicaciones clínicas.

Este aumento en la sensibilidad se acompaña de una reducción en la precisión y el F1-score, así como una menor exactitud global, lo cual es esperable y aceptable en contextos desbalanceados donde es preferible identificar la mayor cantidad posible de casos positivos, aunque se generen algunos falsos positivos.

Las variables clínicas como HbA1c, glucosa, IMC, hipertensión y cardiopatía muestran una fuerte asociación positiva con el riesgo de diabetes, mientras que edad, sexo y tabaquismo también aportan información relevante.

En si, el modelo regularizado es robusto y clínicamente útil, prioriza la detección de pacientes con diabetes, mantiene una excelente capacidad discriminativa y es adecuado para escenarios donde el costo de no detectar casos positivos es alto.

Conclusiones y Recomendaciones

La Sección 6 abordó la construcción, evaluación y mejora del modelo de regresión logística aplicado al diagnóstico de diabetes, partiendo de un **modelo base explicativo** y avanzando hacia un **modelo ajustado y regularizado**, orientado a la estabilidad y sensibilidad clínica.

El **modelo base** mostró un desempeño sobresaliente en términos globales, con **ROC-AUC ≈ 0.96** y **accuracy ≈ 0.96** en los conjuntos de entrenamiento y prueba, lo que refleja una correcta generalización y una fuerte capacidad discriminante.

Las variables clínicas —principalmente `glucose_log` y `HbA1c_level`— evidenciaron una alta asociación con la probabilidad de diabetes, confirmando su relevancia médica y validando el enfoque estadístico.

Sin embargo, el modelo presentó **advertencias de cuasi-separación** ($\approx 16\%$ de observaciones perfectamente clasificadas) y un **recall limitado (≈ 0.63)** para la clase positiva.

Esto implica que **más de un tercio de los pacientes diabéticos podrían no ser identificados**, lo cual resulta clínicamente preocupante. En la práctica médica, los **falsos negativos** representan casos no diagnosticados que pueden retrasar intervenciones y aumentar el riesgo de complicaciones, por lo que es prioritario **incrementar la sensibilidad** del modelo sin sacrificar estabilidad.

El **modelo ajustado**, implementado mediante **regresión logística regularizada (L2)** con `class_weight='balanced'`, logró atenuar estos problemas:

- Redujo la inestabilidad numérica causada por la cuasi-separación.
- Mejoró la **detección de la clase positiva** (recall > 0.85), aun con una leve disminución en la exactitud general.
- Mantuvo una discriminación sólida y un balance más equitativo entre precisión y sensibilidad.

Este resultado confirma que la **regularización y el balanceo** son estrategias efectivas no solo desde una perspectiva estadística, sino también clínica, al priorizar la detección oportuna de pacientes en riesgo.

Recomendaciones y posibles abordajes

- **Optimizar la calibración del modelo**, ajustando el umbral de clasificación según criterios clínicos (por ejemplo, priorizar sensibilidad

en entornos de detección temprana).

- **Evaluar el impacto clínico de los errores**, ponderando el costo de los falsos negativos frente a los falsos positivos.
- **Explorar modelos complementarios** (árboles, random forest o XGBoost) para identificar posibles interacciones no lineales entre variables metabólicas.
- **Aplicar validación cruzada estratificada** para confirmar la estabilidad de las métricas y minimizar la dependencia del muestreo aleatorio.
- **Desarrollar un dashboard interpretativo** que permita visualizar probabilidades individuales y riesgos asociados, orientando decisiones médicas personalizadas.

En conjunto, los resultados muestran una **evolución metodológica y clínica significativa**: el paso de un modelo teóricamente explicativo a uno **más equilibrado, clínicamente sensible y operacionalmente robusto**.

Este avance sienta las bases para el desarrollo de sistemas predictivos confiables en el contexto de salud digital, donde la precisión estadística debe integrarse con la **responsabilidad clínica y la utilidad práctica**.