Primero se realizó una inspección general de la estructura del conjunto de datos con las funciones df.info() y df.describe().  
Se clasificaron las variables en numéricas y en categóricas, usando las funciones select\_dtypes(), lo que permitió usar estrategias de limpieza diferenciadas según el tipo de variable.  
Esto permitió observar de manera detallada los datos y descubrir que había valores nulos y registros duplicados, también existían valores ausentes, más frecuentemente en variables numéricas como medidas fisiológicas y en ciertas variables categóricas relacionadas al diagnóstico.  
Se eliminaron las filas duplicadas para evitar la repetición de la información y reducir posibles sesgos en el análisis.

Para el manejo e imputación de datos su tratamiento se realizó de forma cuidadosa para no eliminar información útil, para esto se calcularon la cantidad absoluta como el porcentaje de valores nulos en cada columna, con el objetivo de priorizar las variables que tuvieran una proporción significativa de datos ausentes.  
La primera estrategia usada fue implementar mediante la librería scikit-learn el objeto SimpleImputer(strategy=”median”) para tener la mediana como valor de imputación, esto debido a que es una medida robusta frente a la presencia de outliers, es relevante en situaciones con variables clínicas donde los valores extremos pueden ser comunes, pero no necesariamente son erróneos.  
Este tratamiento es especialmente importante cuando los datos se destinan a modelos supervisados como la regresión logística o los árboles de decisión, donde los valores faltantes pueden distorsionar los coeficientes o las reglas de partición.

Luego, para las variables categóricas se usó la moda, donde la ausencia de datos puede interpretarse como una categoría válida y debido a esto se creó una nueva variable categórica llamada “Desconocido” o “No registrado”.  
Esto permitió mantener los registros del paciente sin descartar observaciones por falta de datos en una sola columna, con esta imputación se evitó la eliminación masiva de filas garantizando que el conjunto mantuviera la distribución original de las variables.  
Este enfoque favorece la estabilidad del modelo de clasificación al preservar la representatividad del conjunto y evitar el sesgo que generaría una reducción significativa de la muestra.

Posteriormente, para la detección y tratamiento de outliers se hizo un análisis detallado para encontrarlos mediante técnicas gráficas y estadísticos.  
Primero se generaron histogramas y diagramas de caja para las variables numéricas utilizando la librería seaborn, lo que permitió visualizar de manera intuitiva la dispersión de datos y presencia de valores fuera del rango esperado; si correspondían a un error de medición fueron eliminados, mientras que si eran valores poco frecuentes se conservaron para aportar variabilidad real necesaria para un buen comportamiento del modelo.  
Además, la justificación del por qué fue usado el IQR es por su mayor robustez ante distribuciones asimétricas que en este caso con datos biomédicos son comunes.  
El tratamiento de outliers de esta forma mejora la estabilidad y generalización de los modelos predictivos, evitando que los valores extremos dominen los parámetros de ajuste en algoritmos sensibles como regresión lineal o KNN.

Los algoritmos de aprendizaje automático son sensibles a las diferencias de magnitud entre variables, por esto se decidió normalizar los datos numéricos.  
Para esto se aplicó el RobustScaler de scikit-learn, el cual transforma los datos restando la mediana y dividiendo entre el rango intercuartílico.  
A diferencia de la estandarización tradicional (donde se usa la media y desviación estándar), esta técnica no se ve afectada por valores extremos; de esta forma las variables quedan centradas alrededor de cero y en una escala comparable, sin distorsionar la estructura de los datos.  
Esta normalización garantiza un mejor desempeño en modelos basados en distancia (como KNN o SVM) y en métodos que dependen de la magnitud de los coeficientes, como la regresión logística.

También se implementó el One-Hot Encoding para preparar las variables categóricas para los modelos de aprendizaje, ya que convierte cada categoría en una columna binaria para asegurar que cada categoría se representara de manera independiente.  
Esta transformación mejora la interpretabilidad de los modelos y evita la introducción de sesgos.  
Además, el uso de One-Hot Encoding es esencial para modelos lineales o de red neuronal, que no pueden procesar directamente variables no numéricas y requieren representaciones numéricas disjuntas para capturar adecuadamente la relación entre categorías y variable objetivo.

Finalmente, se calculó la matriz de correlación entre las variables numéricas mediante el método de Pearson, que se visualiza en el mapa de calor (heatmap).  
El objetivo de este análisis fue identificar pares de variables altamente correlacionadas (|r| > 0.6), que pudieran introducir redundancia o colinealidad en modelos lineales.  
Esto permite anticipar posibles problemas de multicolinealidad en modelos explicativos como la regresión logística, garantizando que los coeficientes mantengan interpretabilidad y estabilidad en la estimación.