ARTERIOPATÍA CORONARIA

Brian Cabral

CONTENIDOS

INTRODUCCIÓN

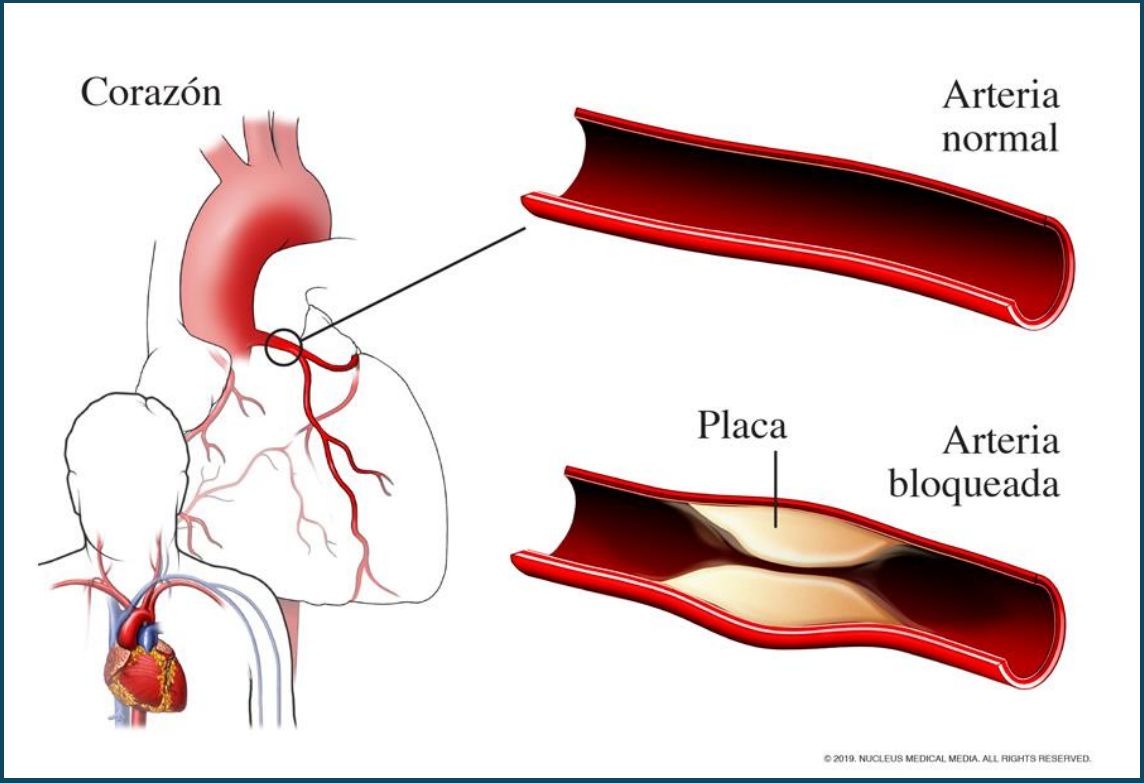
En un mundo donde el avance vertiginoso de la tecnología ha transformado nuestra realidad cotidiana, también ha traído consigo una alarmante tendencia hacia un estilo de vida sedentario y hábitos perjudiciales para la salud.

Las enfermedades cardiovasculares se han convertido en la principal causa de mortalidad en todo el mundo, representando aproximadamente el 30% de todas las defunciones.

Este proyecto personal tiene como objetivo capitalizar el poder de los datos y las técnicas de análisis para arrojar luz sobre los factores de riesgo y patrones de diagnóstico relacionados con la arteriopatía coronaria. Más allá de la mera exploración, aspira a generar conocimiento valioso y a contribuir a la concienciación pública y al perfeccionamiento de la atención médica en un momento crucial en que la salud cardiovascular se erige como una prioridad incuestionable en la sociedad.

¿Qué es la arteriopatía coronaria?

Es una enfermedad que afecta a las arterias que suministran sangre al corazón. Se caracteriza por la acumulación de placa en estas arterias, lo que puede reducir el flujo sanguíneo al corazón y causar problemas graves como dolor en el pecho, ataques cardíacos y, en casos avanzados, insuficiencia cardíaca.



Factores como la presión arterial alta, el colesterol elevado, la obesidad, la falta de ejercicio, una dieta poco saludable y el tabaquismo aumentan el riesgo de desarrollar esta enfermedad. El tratamiento puede incluir cambios en el estilo de vida, medicamentos y procedimientos médicos.

Un factor de riesgo es una característica o habito de una persona, que aumentan su probabilidad de enfermarse.

¿Qué factores aumentan el riesgo de sufrir arteriopatía coronaria?

Edad: el envejecimiento aumenta el riesgo de que las arterias se dañen y se estrechen.

Sexo: los hombres corren mayor riesgo de sufrir esta enfermedad.

Antecedentes familiares: los antecedentes familiares de enfermedad cardiaca te hacen más propenso a contraer arteriopatía coronaria.

Fumar: las personas que fuman corren un riesgo significativamente mayor de tener enfermedades cardiacas.

Hipertensión: la presión arterial alta puede endurecer las arterias, esta rigidez puede causar estrechamiento y disminución del flujo sanguíneo.

Colesterol alto: el exceso de colesterol malo en la sangre puede aumentar el riesgo. el colesterol malo se llama colesterol de lipoproteínas de baja densidad (LDL, por sus siglas en inglés). La falta de colesterol bueno (que se llama lipoproteína de alta densidad o HDL, por sus siglas en inglés) también puede aumentar el riesgo.

Diabetes: la diabetes aumenta el riesgo de tener enfermedad de las arterias coronarias.

Sobrepeso u obesidad: el sobrepeso es malo para la salud en general. La obesidad puede producir la diabetes tipo 2 y presión arterial alta.

Enfermedad renal crónica: la enfermedad renal crónica aumenta el riesgo de tener enfermedad de las arterias coronarias.

PREGUNTAS / PROBLEMA

En un momento en el que la salud cardiovascular se ha convertido en una prioridad global, esta investigación aspira a abordar de manera efectiva este problema de salud pública.

Para eso, a través del análisis de datos y con ayuda de visualizaciones intentaremos responder las siguientes preguntas:

1. ¿Cuál es la relación entre el sexo y la presencia de arteriopatía coronaria?

2. ¿Qué factores como la edad, el peso y el índice de masa corporal (BMI), están asociados con un mayor riesgo de arteriopatía coronaria?

3. ¿Cuál es la prevalencia de factores de riesgo como la diabetes, la hipertensión, el tabaquismo y la dislipidemia entre los pacientes con arteriopatía coronaria?

4. ¿Cuáles son los síntomas y las características de la enfermedad más comunes en los pacientes con arteriopatía coronaria?

5. ¿Existen patrones en los valores de laboratorio, como el nivel de glucosa en sangre, creatinina, lípidos y otros, que estén relacionados con la arteriopatía coronaria?

6. ¿Cómo afecta la enfermedad valvular a la arteriopatía coronaria, si es que lo hace?

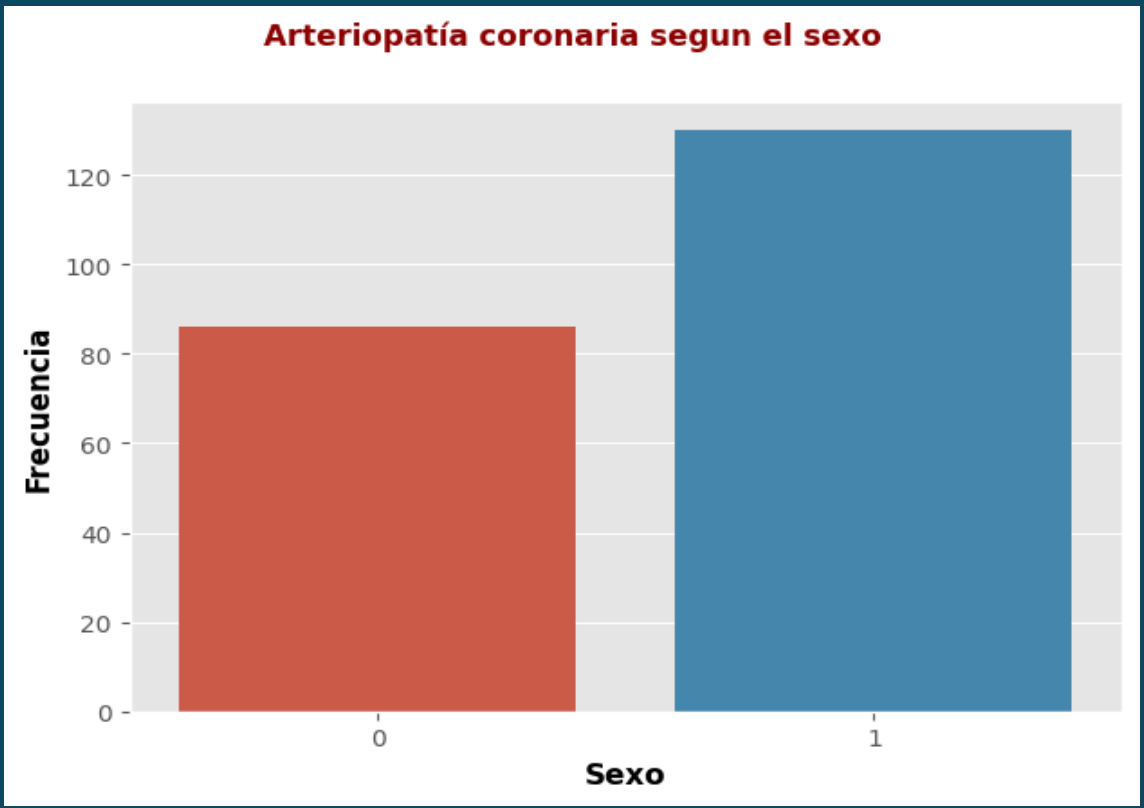
EDA (Análisis Exploratorio de Datos)

Breve descripción del conjunto de datos:

El conjunto de datos utilizado en este proyecto se obtuvo de Kaggle e incluye 303 muestras en total, de las cuales 216 corresponden a pacientes enfermos y 87 a pacientes normales. Estas muestras se caracterizan por un conjunto de 55 características que pueden ser divididas en cuatro categorías principales: Demográficas, Sintomáticas, Electrocardiograma y Valores de Laboratorio.

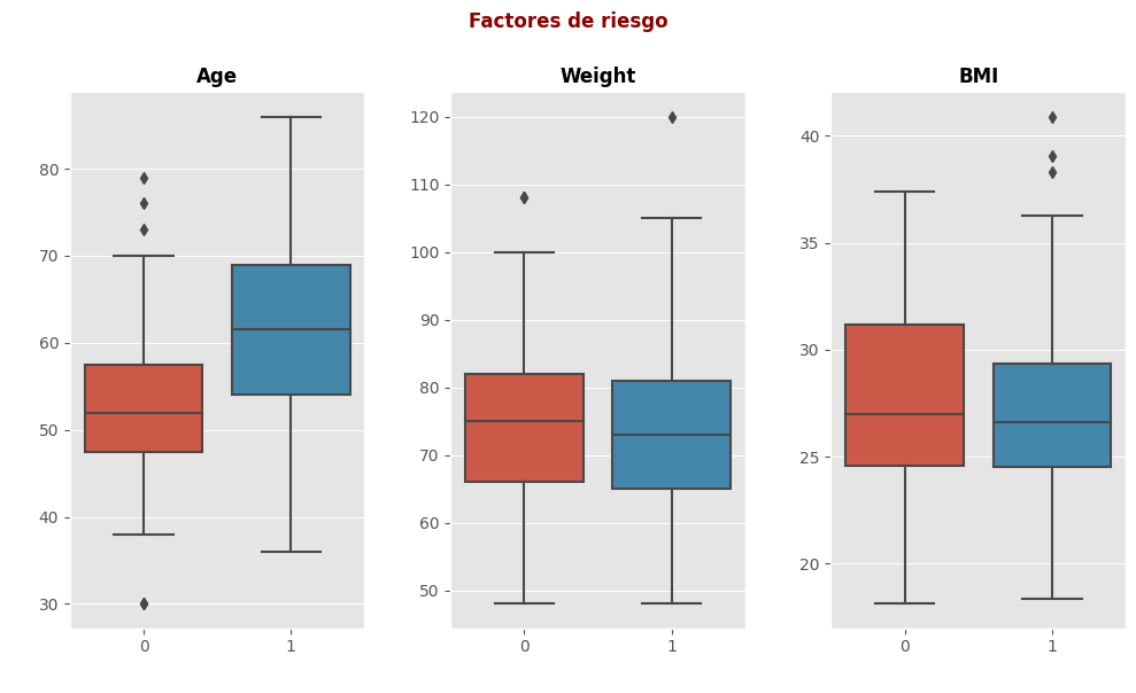
Además, no presenta valores faltantes ni duplicados.

1. ¿Cuál es la relación entre el sexo y la presencia de arteriopatía coronaria?



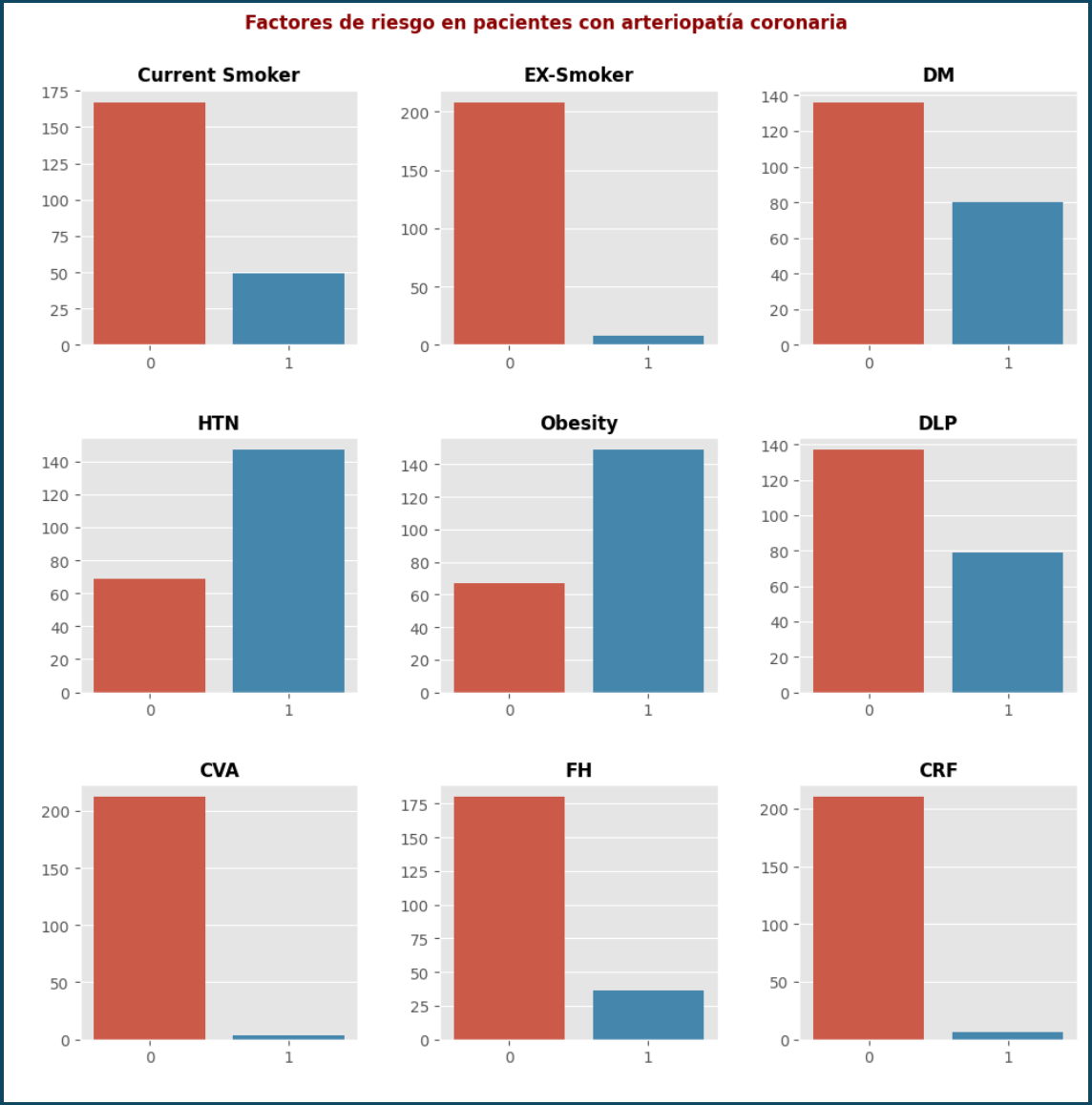
Los hombres tienen mayor tendencia a padecer la enfermedad.

2. ¿Qué factores como la edad, el peso y el índice de masa corporal (BMI), están asociados con un mayor riesgo de arteriopatía coronaria?



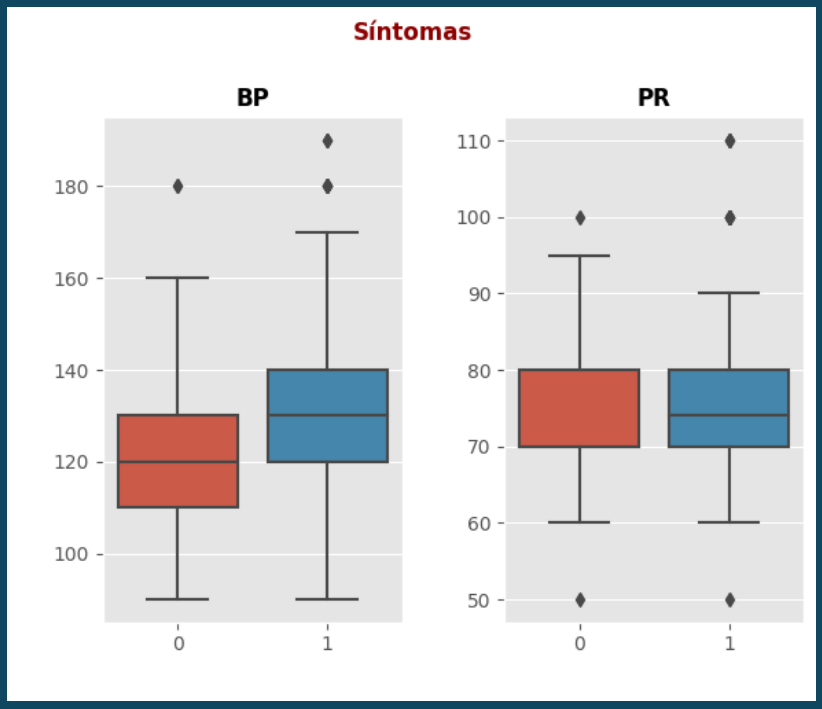
La edad promedio es mayor en pacientes que padecen la enfermedad. Esto indica que la edad está asociada a un mayor riesgo.

3. ¿Cuál es la prevalencia de factores de riesgo como la diabetes, la hipertensión, el tabaquismo y la dislipidemia entre los pacientes con arteriopatía coronaria?



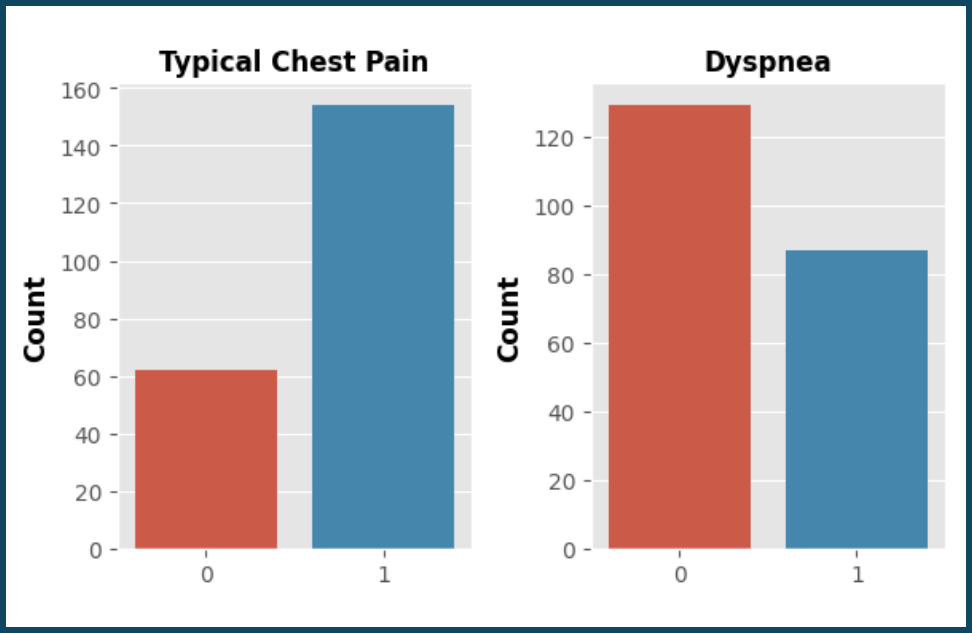
Los factores de riesgo que prevalecen entre los pacientes enfermos son la hipertensión, la obesidad y diabetes en menor proporción.

4. ¿Cuáles son los síntomas y las características de la enfermedad más comunes en los pacientes con arteriopatía coronaria?



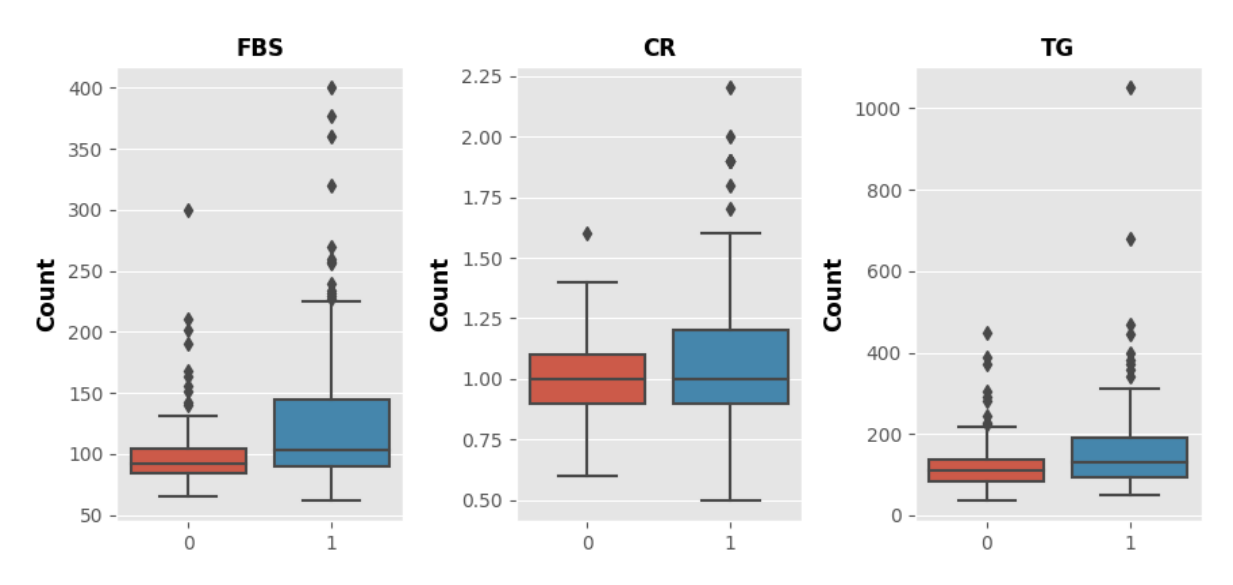
Los pacientes con enfermedad cardiaca presentan un valor promedio de presión sanguínea más alto que los pacientes sanos.

En cuanto a la frecuencia cardiaca mantienen un promedio similar



La mayoría de los pacientes afectados experimentan dolor de pecho típico de la arteriopatía coronaria y una gran cantidad de ellos padecen dificultad para respirar (disnea).

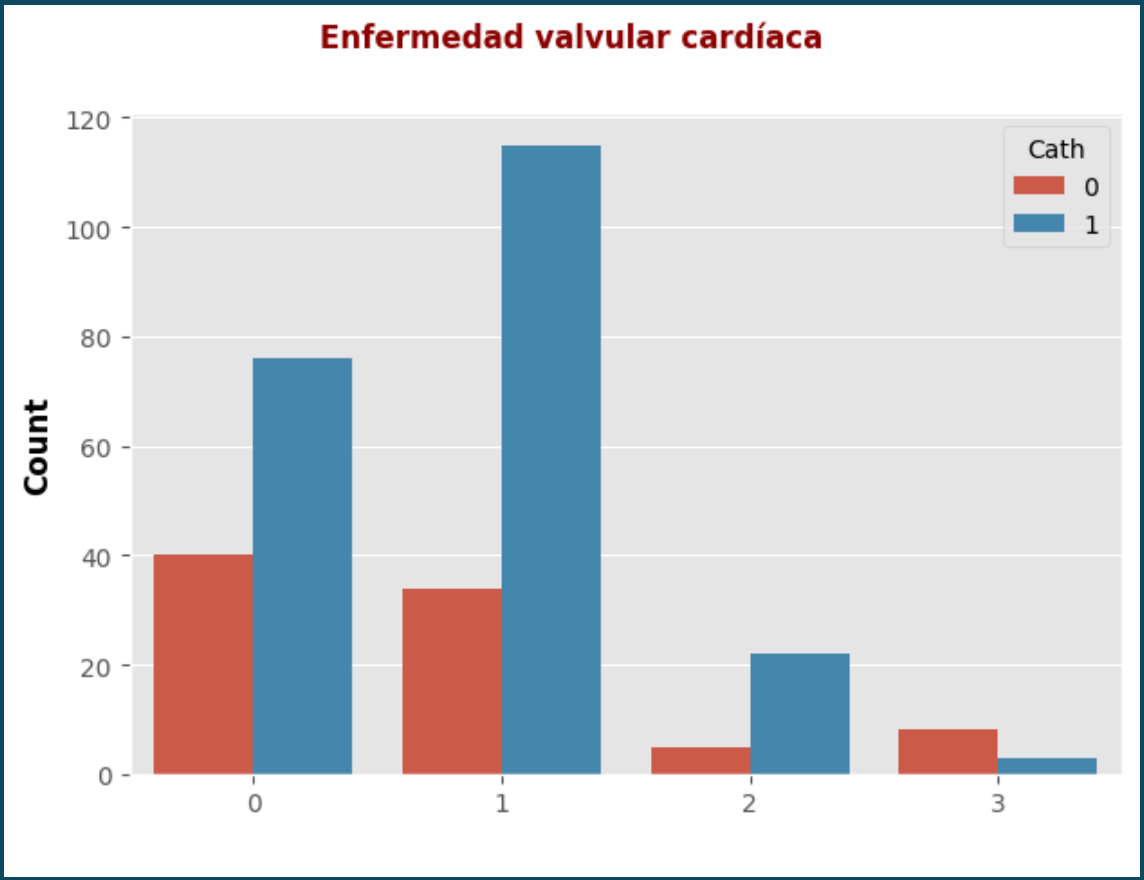
5. ¿Existen patrones en los valores de laboratorio, como el nivel de glucosa en sangre, creatinina, lípidos y otros, que estén relacionados con la arteriopatía coronaria?



Si bien algunos de los valores de laboratorio varían un poco entre pacientes enfermos y sanos, vamos a enfocarnos en aquellos que tienen relación con los factores de riesgo.

Gran parte de los pacientes enfermos presentan valores de glucosa en sangre (FBS) y triglicéridos (TG) más altos que los pacientes sanos. Al igual que una parte presenta valores altos de creatinina (CR), típico en personas con enfermedad renal.

6. ¿Cómo afecta la enfermedad valvular a la arteriopatía coronaria, si es que lo hace?



Siendo 0 la ausencia de enfermedad valvular cardíaca y 1, 2, 3 aquellos pacientes que están afectados de manera leve, moderada y severa respectivamente por esta enfermedad.

La mayoría de los pacientes que presentan arteriopatía coronaria están afectados de manera leve por la enfermedad valvular.

INGENIERÍA DE ATRIBUTOS

Codificación de variables categóricas:

Convertir las categorías o etiquetas de las variables categóricas de un conjunto de datos en una forma numérica, permite que los algoritmos de machine learning funcionen correctamente.

```
[7]  # Codificación de variables categóricas

     cath = {"Normal": 0, "Cad": 1}
     vhd = {"N": 0, "mild": 1, "Moderate": 2, "Severe": 3}
     sex = {"Male": 1, "Fmale": 0}

     df['Cath'] = df['Cath'].map(cath)
     df['VHD'] = df['VHD'].map(vhd)
     df['Sex'] = df['Sex'].map(sex)

     df.replace('N', 0, inplace=True)
     df.replace('Y', 1, inplace=True)
```

Balanceo de clases minoritarias:

El conjunto de datos presenta un desbalance entre las instancias que representan pacientes enfermos y sanos.
Con el fin de mejorar el rendimiento y la capacidad predictiva de los modelos de machine learning se utilizó RandomOverSampler para mitigar el desbalance.

```
[59] # Balanceo de clases minoritarias del dataset
     sm = RandomOverSampler()
     X,y = sm.fit_resample(X,y)
```

Selección de características:

Con el objetivo de mejorar el rendimiento de los modelos utilizados y eliminar aquellas características irrelevantes, redundantes o ruidosas, se utilizó el método de selección de características (SelectKBest) para extraer un subconjunto relevante y significativo de variables o atributos del conjunto de datos original.

```
[58] # Inicializa el selector para que elija las mejores 'k' características utilizando chi-cuadrado
     selector = SelectKBest(chi2, k=10)

     # Ajuste del selector a los datos
     X = selector.fit_transform(X, y)

     # Obtén las características seleccionadas
     selected_features = selector.get_support(indices=True)

     # Visualiza las características seleccionadas
     selected_feature_names = [df.columns[i] for i in selected_features]
     print("Características seleccionadas:", selected_feature_names)
```

Estandarizado de los datos:

La estandarización de datos en machine learning permite lograr que todas las características tengan una escala y una distribución comparables. Esto es importante porque muchos algoritmos funcionan mejor cuando las características están en la misma escala.

```
[61] # Estandarizado de los datos
     scaler = StandardScaler()
     X_train = scaler.fit_transform(X_train)
     X_test = scaler.transform(X_test)
```

ENTRENAMIENTO Y TESTEO

Para este trabajo se entrenaron dos modelos de machine learning distintos:

- RandomForestClassifier

```
[136] from sklearn.ensemble import RandomForestClassifier

      # Creación de modelo
      RFC = RandomForestClassifier()
```

- KNeighborsClassifier

```
[143] from sklearn.neighbors import KNeighborsClassifier

      # Creacion del modelo
      KNN = KNeighborsClassifier()
```

OPTIMIZACIÓN DE HIPERPARÁMETROS

El objetivo de la optimización de hiperparámetros en machine learning es encontrar la combinación de hiperparámetros que permita que un modelo de machine learning alcance su mejor rendimiento posible frente a un problema específico.

Existen diversas técnicas para la optimización de hiperparámetros, como GridSearchCV y RandomizedSearchCV, utilizados en el presente trabajo.

- Optimización de hiperparámetros (RandomForestClassifier):

```
[211] # Definir la cuadrícula de hiperparámetros para la búsqueda aleatoria
      param_dist = {
          'n_estimators': [50, 100, 150, 200],
          'max_depth': [5, 10, 15, 20],
          'min_samples_split': [2, 5, 10, 15],
          'min_samples_leaf': [1, 2, 4, 8],
          'max_features': [4, 6, 8, 10]
      }

      # Inicializar RandomizedSearchCV
      random_search = RandomizedSearchCV(RFC, param_distributions=param_dist, n_jobs=-1, cv=10, scoring='recall')

      # Realizar la búsqueda aleatoria en los datos de entrenamiento
      random_search.fit(X_train, y_train)

      # Obtener el modelo con los mejores hiperparámetros
      best_RFC = random_search.best_estimator_

      # Imprimir modelo con los mejores hiperparametros
      print("Mejores hiperparametros:", random_search.best_params_)
```

- Optimización de hiperparámetros (KNeighborsClassifier):

```
[218] # Definir los hiperparámetros que deseas ajustar
      param_grid = {
          'n_neighbors': [2, 3, 4, 5, 10, 15, 20, 25, 30, 35, 40, 45, 50],
          'weights': ['uniform', 'distance'],
          'p': [1, 2]
      }

      # Crear un objeto GridSearchCV con KNN y los hiperparámetros definidos
      grid_search = GridSearchCV(KNN, param_grid, cv=10, scoring='recall')
      grid_search.fit(X_train, y_train)

      # Obtener el modelo con los mejores hiperparámetros
      best_KNN = grid_search.best_estimator_

      # Imprimir modelo con los mejores hiperparámetros
      print("Mejores hiperparametros:", grid_search.best_params_)
```

EVALUACIÓN Y SELECCIÓN DE MODELOS

Ambos modelos fueron evaluados mediante la métrica Recall (también conocida como sensibilidad o tasa de verdaderos positivos). Esto nos permite medir la capacidad del modelo para identificar de manera efectiva todos los ejemplos positivos en un conjunto de datos.

Con esto logramos minimizar los falsos negativos, es decir, los casos positivos que el modelo predice incorrectamente como negativos.

- Evaluación del modelo RandomForestClassifier:

```
[213] # Evaluación de precisión del modelo
      print(classification_report(y_test,y_pred))
```

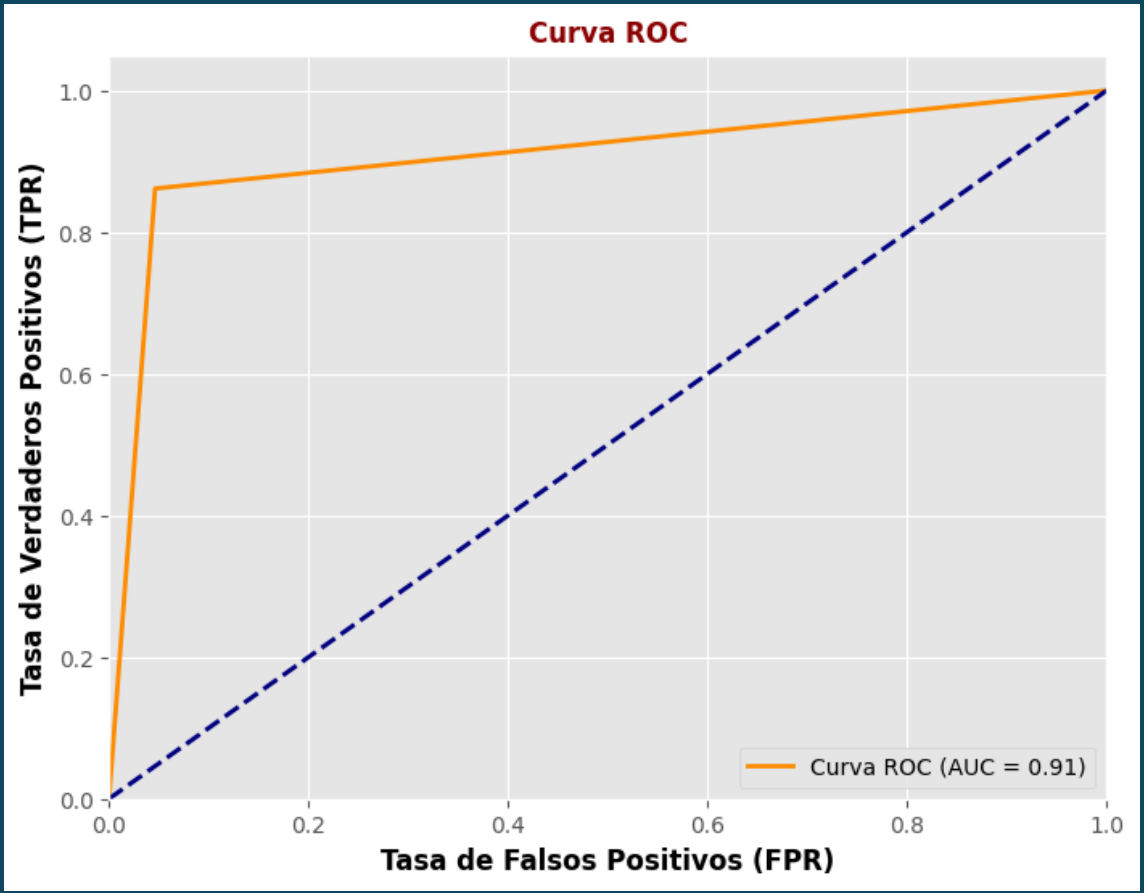
```
                  precision    recall  f1-score   support

               0       0.87      0.95      0.91        65
               1       0.95      0.86      0.90        65

        accuracy                           0.91       130
       macro avg       0.91      0.91      0.91       130
    weighted avg       0.91      0.91      0.91       130
```

```
[214] # Matriz de confusión
      print(confusion_matrix(y_test, y_pred))
```

```
[[62  3]
 [ 9 56]]
```



- Evaluación del modelo KNeighborsClassifier:

```
[220] # Evaluación de precisión del modelo
      print(classification_report(y_test,y_pred))
```

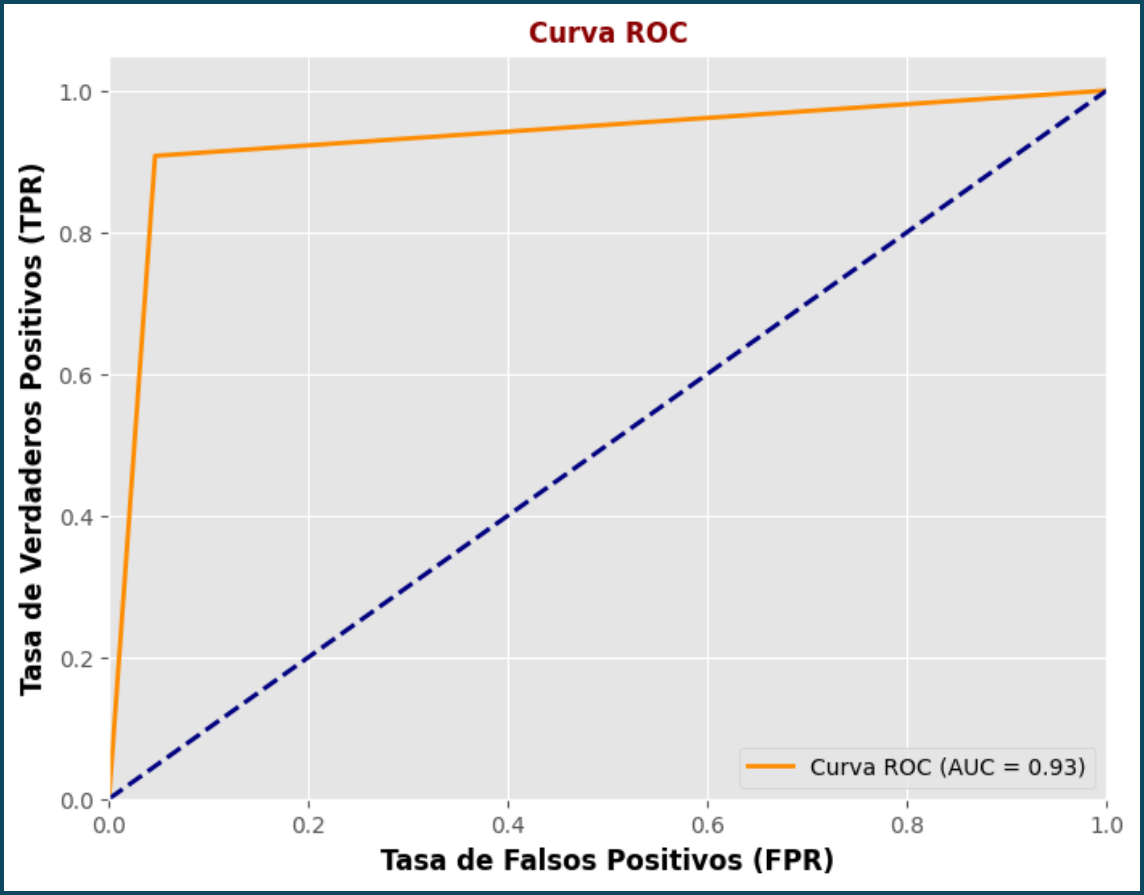
```
                  precision    recall  f1-score   support

               0       0.91      0.95      0.93        65
               1       0.95      0.91      0.93        65

        accuracy                           0.93       130
       macro avg       0.93      0.93      0.93       130
    weighted avg       0.93      0.93      0.93       130
```

```
[221] # Matriz de confusión
      matrix = confusion_matrix(y_test, y_pred)
      print(matrix)
```

```
[[62  3]
 [ 6 59]]
```



Ambos modelos presentan métricas similares, por lo que utilizaremos el método de validación cruzada para poder evaluar de manera más confiable y precisa el rendimiento de los dos modelos.

```python
# Crea una lista para almacenar los modelos que se van a evaluar
models = []
models.append(('RFC', best_RFC))  # Agrega el mejor modelo de RandomForestClassifier al conjunto
models.append(('KNN', best_KNN))  # Agrega el mejor modelo de KNeighborsClassifier al conjunto

# Inicializar un diccionario para almacenar los resultados
results = dict()

# Iteración de los modelos y realizar validación cruzada
for name, model in models:
  kfold = KFold(n_splits=10)
  cv_results = cross_val_score(model, X_train, y_train, cv=kfold, scoring='recall')
  cv_results1 = cross_val_score(model, X_train, y_train, cv=kfold, scoring='accuracy')
  results[name]=(cv_results.mean(), cv_results.std(), cv_results1.mean(), cv_results1.std())

# Imprime los resultados
print("name   recall.mean          recall.std          accuracy.mean          accuracy.std")
for key, value in results.items():
    print(key, value)
```

```
name   recall.mean          recall.std          accuracy.mean          accuracy.std
RFC (0.8371982841719683, 0.09075618619873445, 0.8704301075268818, 0.06091613753540701)
KNN (0.7960097358781569, 0.08936647165795379, 0.8807526881720431, 0.053934030437791804)
```

CONCLUSIONES

En función de las métricas proporcionadas, el modelo KNN parece tener un rendimiento ligeramente mejor en términos de accuracy, mientras que el modelo RFC tiene un recall ligeramente mejor. La elección entre los dos modelos depende de los objetivos y de la importancia relativa que asignes a estas métricas en un problema específico.

En este caso, la detección de casos positivos es la prioridad, dejando en segundo plano la precisión global, bajo estas consideraciones el modelo RFC es el más optimo. En cambio, si la precisión global es más importante y los falsos negativos no son tan críticos, el modelo KNN podría ser la elección preferida.