**UNIVERSIDAD PRIVADA ANTENOR ORREGO FACULTAD DE INGENIERÍA**

INGENIERÍA DE SISTEMAS E INTELIGENCIA ARTIFICIAL

****

**TEMA:**

**“Predicción del riesgo de enfermedades cardíacas a partir de parámetros clínicos y estilo de vida usando modelos de Machine Learning.”**

**CURSO:**

Inteligencia Artificial Principios y Técnicas

**DOCENTE(s):**

Mendoza Corpus, Carlos Alfredo

Sagastegui Chigne, Teobaldo Hernan

**INTEGRANTES:**

Fernández Muguerza, Andy Lee

Ibañez Herrera, Luis Anthony

Quispe Cesias, Andro Joseph

Rodríguez Acevedo, Emerson Ronaldo

Vaella Huamán Sergio Leonardo

Zamora Duran Piero Javier

TRUJILLO - PERÚ

2025

**ÍNDICE**

[**I. Introducción 3**](#_9lwj296t97eu)

[1.1 Titulo del Proyecto 3](#_t9yju9820do9)

[1.2 Antecedentes 3](#_gyh8t5hk7oko)

[1.3 Problema a resolver 3](#_7ox5ejo4je16)

[1.4 Objetivos 4](#_mvppk5w2phpy)

[Objetivo General 4](#_kpt3kanvf6xd)

[Objetivos Específicos 4](#_r2s96qmnkvuo)

[**II. Requerimientos del sistema 4**](#_dwgboq3o3etb)

[2.1 Definición del Dominio 4](#_6pifvznzu6lr)

[2.2 Determinación de Requisitos (Requerimientos) 5](#_ml5hzwg2s0e9)

[Requerimientos Funcionales 5](#_5bmbw5oqwugv)

[Requerimientos No Funcionales 6](#_3p23xigfnccb)

[**III. PRE-PROCESAMIENTO y Normalización (Planteamiento del Data-Set; Aprendizaje Supervisado o Aprendizaje no-Supervisado) 6**](#_yd3jj7hcji5y)

[3.1 Medidas, Datos, Bases de Datos y Elaboración del Data-Set 6](#_rnnzp2rw2urq)

[3.2 Normalización y/o Filtrado de Datos 8](#_s377jv5jpous)

[3.2.1 Selección de Características 8](#_d6g8376dncvf)

[3.2.2 Reducción de Dimensionalidad 9](#_c1cothd2vftk)

[3.2.3 Normalización de Variables 9](#_k97inhnbxgow)

[3.3 Planteamiento del Data-Set de Entrenamiento, Validación y Pruebas 10](#_g8ohfjriw9it)

[**IV. Aprendizaje del modelo del sistema 10**](#_gi9vhrpianq)

[4.1 Planteamiento del Modelo de Aprendizaje. 10](#_k0pzxpmet4tt)

[4.2 Desarrollo e Implementación del Modelo \* 11](#_2pcg66ait10b)

[**V. Comprobación y Despliegue (Deploy) del Sistema 13**](#_uyekx85lpsrf)

[5.1 Entrenamiento del Modelo: Uso del Data-Set de entrenamiento 13](#_f15nb6hg5i4p)

[Entrenamiento de Modelos 13](#_3h8o3qwyi72t)

[- Random Forest: 13](#_dwkaa7e5x0bv)

[- XGBoost 14](#_44vpvoh7v9om)

[- Red Neuronal 15](#_p08xoz5ipnfc)

[- Comparación de Modelos 16](#_3zzq5gw9aj83)

[5.2 Ejecución y Pruebas del Modelo 16](#_9xeunxl2whzl)

[5.3 Ejecución de la validación del Modelo 17](#_6qcm28ffamx7)

[5.4 Deploy del APP o Web del Sistema de Predicción 18](#_mokddoodq6vv)

[Definición del objetivo de la página 18](#_ow2x6dlocox8)

[Diseño general de la interfaz 19](#_vkf7eajdueov)

[Backend y procesamiento 19](#_jbhaycy3zerr)

[Despliegue de la aplicación web 20](#_chr2pzgkqddm)

# **I. Introducción**

## **1.1 Titulo del Proyecto**

"Predicción del riesgo de enfermedades cardíacas a partir de parámetros clínicos y estilo de vida usando modelos de Machine Learning."

## **1.2 Antecedentes**

Las enfermedades cardiovasculares continúan siendo la principal causa de mortalidad a nivel global, representando un grave problema de salud pública. Ante esta realidad, resulta imprescindible implementar estrategias preventivas que permitan identificar de forma temprana a los individuos con mayor riesgo de desarrollar estas patologías. En este contexto, el uso de técnicas de **Machine Learning (ML)** se presenta como una herramienta poderosa para analizar grandes volúmenes de datos clínicos y de estilo de vida, extrayendo patrones complejos que podrían pasar desapercibidos en métodos tradicionales.

Este proyecto tiene como finalidad desarrollar un modelo predictivo capaz de estimar el riesgo de enfermedades cardíacas a partir de parámetros como la presión arterial, niveles de colesterol, glucosa, índice de masa corporal, hábitos de consumo (tabaco y alcohol), actividad física y otros factores relevantes. La integración de la inteligencia artificial en este ámbito permite no solo mejorar la precisión diagnóstica, sino también optimizar la toma de decisiones médicas, facilitando intervenciones oportunas que puedan salvar vidas.

Asimismo, la propuesta responde a una necesidad social urgente, ya que permite fortalecer la medicina preventiva, reducir los costos en el sistema de salud y ofrecer herramientas de apoyo tanto para profesionales clínicos como para entidades aseguradoras. En conjunto, este trabajo no solo tiene un valor académico y técnico, sino también un alto impacto social y sanitario, alineándose con los objetivos de innovación tecnológica al servicio de la salud.

## **1.3 Problema a resolver**

Las enfermedades cardíacas representan una de las principales causas de muerte a nivel mundial, muchas veces detectadas en etapas avanzadas cuando las posibilidades de intervención efectiva son limitadas. A pesar de los avances en diagnóstico y tratamiento, los métodos tradicionales no siempre permiten identificar con suficiente antelación a los pacientes en riesgo, especialmente cuando los factores de riesgo están relacionados con el estilo de vida y no presentan síntomas visibles. En este contexto, surge la necesidad de contar con herramientas predictivas más precisas, capaces de analizar múltiples variables clínicas y de comportamiento de forma automatizada. El reto radica en cómo aprovechar los datos disponibles para anticipar estos riesgos y permitir una intervención oportuna.

## **1.4 Objetivos**

#### **Objetivo General**

Desarrollar un sistema inteligente basado en modelos de Machine Learning que permita predecir el riesgo de enfermedades cardíacas a partir del análisis de parámetros clínicos y de estilo de vida del paciente.

#### **Objetivos Específicos**

* Recopilar y preprocesar un conjunto de datos clínicos y de hábitos de vida relevantes para la predicción de enfermedades cardiovasculares.
* Aplicar técnicas de normalización y división del conjunto de datos en entrenamiento, validación y prueba.
* Implementar y comparar modelos de aprendizaje supervisado para identificar cuál ofrece la mayor precisión predictiva.
* Evaluar el desempeño del modelo utilizando métricas como exactitud, precisión, recall y F1-score.
* Diseñar una interfaz simple que permita ingresar los datos de un paciente y visualizar el nivel de riesgo estimado.
* Validar la utilidad del sistema como apoyo a la medicina preventiva, destacando su aplicabilidad en entornos clínicos y de seguros de salud.

# **II. Requerimientos del sistema**

## **2.1 Definición del Dominio**

El dominio de este proyecto se centra en la Clínica San Gabriel, una institución médica privada ubicada en la ciudad de Trujillo, especializada en medicina general, cardiologí

a preventiva y diagnóstico por imágenes. Esta clínica atiende a un gran número de pacientes diariamente, y actualmente el análisis del riesgo cardiovascular se realiza principalmente mediante evaluaciones clínicas presenciales, entrevistas médicas y pruebas físicas, sin el uso de herramientas automatizadas de predicción.

En este proyecto, el dominio abarca tanto el entorno clínico de atención primaria como los atributos específicos que se desea modelar y predecir. En particular, el modelo estará enfocado en predecir el **riesgo de enfermedades cardiovasculares** a partir del análisis automatizado de datos clínicos y de estilo de vida de los pacientes, tales como:

* Presión arterial sistólica y diastólica
* Niveles de colesterol y glucosa
* Índice de masa corporal (IMC)
* Edad y género
* Hábitos como el consumo de tabaco, alcohol y actividad física

Estas variables fueron seleccionadas por su relevancia directa en el desarrollo de enfermedades cardíacas, y por estar disponibles en los controles médicos básicos realizados en la clínica. Además de los datos ingresados manualmente, el sistema proyecta la futura integración con historias clínicas electrónicas (HCE) y dispositivos IoT (como tensiómetros digitales), que permitirán automatizar la recolección de datos y alimentar el modelo en tiempo real.

El objetivo es facilitar la **detección temprana del riesgo cardíaco** mediante el uso de modelos de Machine Learning, optimizando el proceso de atención médica, orientando intervenciones preventivas y reduciendo la carga asistencial en etapas avanzadas de la enfermedad. El sistema servirá como apoyo al profesional de salud, no como reemplazo, fortaleciendo la medicina personalizada y preventiva dentro del contexto clínico local.

## **2.2 Determinación de Requisitos (Requerimientos)**

### **Requerimientos Funcionales**

* RF1: El modelo debe permitir el ingreso manual de datos clínicos y de estilo de vida del paciente (edad, género, presión arterial, colesterol, glucosa, hábitos de consumo y actividad física).
* RF2: El modelo debe realizar el cálculo de riesgo cardiovascular utilizando algoritmos de Machine Learning entrenados previamente.
* RF3: El sistema debe mostrar el resultado de la predicción clasificado como: riesgo bajo, moderado o alto.
* RF4: El sistema debe permitir visualizar las métricas de rendimiento del modelo (precisión, sensibilidad, especificidad, AUC-ROC).
* RF5: El sistema debe permitir la explicación del resultado mediante herramientas de interpretabilidad como SHAP.
* RF6: El sistema debe permitir al usuario realizar nuevas predicciones sin necesidad de reiniciar la página.

### **Requerimientos No Funcionales**

* RNF1: El modelo debe ser accesible desde una interfaz web intuitiva y amigable para el usuario clínico.
* RNF2: El sistema debe procesar los datos ingresados y entregar el resultado de predicción en menos de 3 segundos.
* RNF3: El sistema debe estar desarrollado con tecnologías de código abierto y permitir su despliegue en la nube.
* RNF4: El sistema debe ser escalable, permitiendo su integración futura con historias clínicas electrónicas (HCE).
* RNF5: El sistema debe ofrecer soporte para actualización del modelo predictivo mediante nuevos datos validados.

# **III. PRE-PROCESAMIENTO y Normalización (Planteamiento del Data-Set; Aprendizaje Supervisado o Aprendizaje no-Supervisado)**

## **3.1 Medidas, Datos, Bases de Datos y Elaboración del Data-Set**

Para el desarrollo del sistema predictor de riesgo cardiovascular, se utilizó un enfoque de **aprendizaje supervisado**, el cual requiere un conjunto de datos etiquetado, es decir, que contenga una variable objetivo claramente definida. En este caso, la variable objetivo es **cardio**, la cual indica si un paciente presenta o no riesgo de enfermedad cardiovascular (1 = con riesgo, 0 = sin riesgo).

El dataset empleado proviene de la plataforma **Kaggle**, específicamente el conjunto denominado **Cardiovascular Disease Dataset**, que contiene más de 1.5 millones de registros anonimizados en formato CSV (~300MB). Este conjunto fue seleccionado por su gran volumen, variedad de características clínicas y validez en investigaciones previas.

Los datos incluyen variables tanto **antropométricas como clínicas y de estilo de vida**, esenciales para el análisis predictivo:

* id: Identificador único de registro
* age: Edad del paciente (en días)
* gender: Género (1 = hombre, 2 = mujer)
* height: Estatura en centímetros
* weight: Peso en kilogramos
* ap\_hi: Presión arterial sistólica
* ap\_lo: Presión arterial diastólica
* cholesterol: Nivel de colesterol (1 = normal, 2 = por encima de lo normal, 3 = muy por encima)
* gluc: Nivel de glucosa (mismo formato que colesterol)
* smoke: Consumo de tabaco (1 = sí, 0 = no)
* alco: Consumo de alcohol (1 = sí, 0 = no)
* active: Actividad física (1 = sí, 0 = no)
* cardio: Etiqueta binaria que indica si el paciente tiene o no una condición cardíaca diagnosticada

El dataset fue sometido a un proceso de **preprocesamiento y limpieza masiva con pandas**, eliminando valores atípicos, vacíos o inconsistentes, especialmente en variables como presión arterial y peso corporal. Posteriormente, se aplicó **normalización con StandardScaler** para homogeneizar las escalas de las variables numéricas antes del entrenamiento de los modelos.

Este tratamiento permitió transformar los datos crudos en un conjunto estructurado y listo para el entrenamiento del modelo de predicción, garantizando la validez de los resultados obtenidos. Gracias a este enfoque, se mejoró la calidad de los datos y se minimizó el riesgo de sesgos durante el aprendizaje automático.

## **3.2 Normalización y/o Filtrado de Datos**

Una vez concluido el proceso de limpieza, transformación y estructuración del dataset cardiovascular, fue necesario adaptar la estructura del conjunto de variables para optimizar el desempeño del modelo de predicción. Esto se logró mediante técnicas de **normalización** y **reducción de dimensionalidad**, con el fin de mejorar la eficiencia computacional, evitar redundancias y aumentar la precisión del modelo sin incrementar su complejidad.

### **3.2.1 Selección de Características**

La selección de características consistió en identificar las variables más relevantes para la predicción del riesgo cardíaco y descartar aquellas que pudieran generar ruido o no aportar valor al análisis. Entre los métodos aplicados se incluyen:

* **Filtrado por relevancia**: Se eliminaron variables con valores constantes, baja variabilidad o sin correlación directa con la variable objetivo cardio.
* **Análisis de correlación**: Se revisó la multicolinealidad entre variables como peso, altura e IMC, eliminando duplicaciones que pudieran distorsionar el aprendizaje del modelo.

Esta depuración permitió reducir el conjunto de características manteniendo aquellas más representativas del estado clínico del paciente: presión arterial, colesterol, glucosa, actividad física y hábitos nocivos como el tabaquismo y el consumo de alcohol.

### **3.2.2 Reducción de Dimensionalidad**

Para mejorar la interpretación del modelo y evitar sobreajuste, se aplicó la técnica de **Análisis de Componentes Principales (PCA)**. Este método permitió reducir la cantidad de variables sin perder información significativa del conjunto de datos. Los pasos fueron:

* Se utilizó previamente **Z-score Scaling** (con StandardScaler) para estandarizar las variables, dado que PCA es sensible a las escalas.
* Luego se aplicó PCA para transformar el espacio original de atributos en un espacio de **componentes principales**, logrando reducir la dimensionalidad con el mínimo sacrificio de varianza explicada.

Como resultado, se proyectó el dataset sobre los dos primeros componentes, los cuales explicaron aproximadamente el **85% de la varianza total**. Esta proyección permitió visualizar agrupamientos por nivel de riesgo (alto, medio o bajo), revelando patrones subyacentes no evidentes en el espacio original.

### **3.2.3 Normalización de Variables**

Dado que los algoritmos utilizados (como redes neuronales y XGBoost) pueden verse afectados por diferencias en las escalas de las variables, se aplicaron técnicas de normalización según el tipo de dato:

* **Estandarización (Z-score Scaling)**: Se aplicó a variables como presión arterial, edad, peso y altura, transformándolas para que tengan media 0 y desviación estándar 1.
* **Normalización (Min-Max Scaling)**: Considerada en etapas de prueba para redes neuronales sensibles al rango de entrada.
* **Escalado robusto**: Se utilizó en pruebas para mitigar el efecto de valores atípicos, especialmente en presión arterial sistólica (ap\_hi) y peso.

Estas transformaciones aseguraron que todas las variables contribuyeran equitativamente al aprendizaje del modelo y evitaron que aquellas con mayor magnitud dominaran el proceso de entrenamiento.

## **3.3 Planteamiento del Data-Set de Entrenamiento, Validación y Pruebas**

En este proyecto se trabajó con el Cardiovascular Disease Dataset de Kaggle, compuesto por cerca de 70 000 registros y 12 variables clínicas y demográficas (edad, sexo, presión arterial sistólica y diastólica, colesterol, glucosa, índice de masa corporal, hábitos de vida como tabaquismo, consumo de alcohol y nivel de actividad física, y la etiqueta de presencia o ausencia de enfermedad cardiovascular). A partir de estas columnas originales se enriqueció el conjunto de datos generando nuevas variables: la edad en años, el BMI (Índice de Masa Corporal), la Presión Arterial Media (MAP) y la presión de pulso, categorías discretas de presión e IMC, una puntuación de riesgo de estilo de vida y dos interacciones clave (edad×BMI y edad×presión sistólica). Para garantizar que los modelos aprendieran sobre rangos fisiológicos realistas, se filtraron valores atípicos fuera de los límites habituales (por ejemplo, presión sistólica entre 80 y 200 mmHg, IMC entre 15 y 50, altura entre 140 y 220 cm y peso entre 40 y 200 kg).

Con el dataset limpio y enriquecido se aplicó una partición estratificada en proporciones de 60 % para el entrenamiento, 20 % para validación y 20 % para prueba, de modo que la proporción de pacientes con y sin enfermedad cardiovascular se mantuviera constante en cada subconjunto y se asegurará reproducibilidad mediante una semilla fija. El conjunto de validación se utilizó en un esquema de Stratified K-Fold (k = 5) para ajustar hiper parámetros: en cada fold se evaluaron métricas como AUC-ROC y log-loss, eligiendo la configuración con mejor promedio de rendimiento. Finalmente, el 20 % reservado como “conjunto ciego” permitió obtener una estimación imparcial de la capacidad predictiva del modelo antes de su despliegue, confirmando que hemos usado el 60 % del dataset para entrenar, el 20 % para validar y el 20 % restante para evaluar el desempeño final.

# **IV. Aprendizaje del modelo del sistema**

## **4.1 Planteamiento del Modelo de Aprendizaje.**

Para abordar el problema de predicción del riesgo de enfermedades cardiovasculares, se optó por un **enfoque de aprendizaje supervisado**, debido a la disponibilidad de un conjunto de datos etiquetado que incluye la variable objetivo cardio (0 = sin riesgo, 1 = con riesgo). El modelo propuesto tiene como finalidad anticipar si un paciente presenta un riesgo cardiovascular significativo, a partir del análisis de parámetros clínicos y de estilo de vida.

La elección de aprendizaje supervisado se justifica porque se dispone de ejemplos previos con entradas (variables clínicas como presión arterial, colesterol, glucosa, etc.) y salidas conocidas (cardio), lo cual permite entrenar modelos predictivos a partir de patrones históricos. Se utilizaron técnicas estadísticas y modelos de clasificación, incluyendo algoritmos como **Random Forest**, **XGBoost** y **Redes Neuronales**, con el objetivo de comparar su desempeño en términos de precisión, sensibilidad y especificidad.

La naturaleza binaria del problema (riesgo o no riesgo) lo convierte en un caso típico de clasificación supervisada. Estos modelos permiten capturar interacciones complejas entre variables, que muchas veces no son evidentes para los especialistas médicos a simple vista, y que pueden potenciar el enfoque preventivo en la atención de pacientes.

En resumen, el modelo de aprendizaje supervisado desarrollado aprovecha datos clínicos estructurados para predecir de manera anticipada el riesgo cardíaco, alineándose con los objetivos del proyecto: facilitar la intervención médica temprana, optimizar la gestión clínica y mejorar la calidad de vida del paciente.

## **4.2 Desarrollo e Implementación del Modelo \***

El desarrollo e implementación del modelo se realizó en **Google Colab**, aprovechando su capacidad para ejecutar código Python en la nube con acceso a bibliotecas de análisis de datos e inteligencia artificial. El proceso comenzó con la **carga del dataset cardiovascular en formato CSV**, utilizando el módulo files de la biblioteca google.colab, lo que permitió subir el archivo desde el equipo local al entorno de ejecución temporal de Colab.

El archivo fue procesado utilizando la biblioteca **pandas**, mientras que la lectura binaria fue manejada con la biblioteca io. Posteriormente, se realizó un exhaustivo proceso de preprocesamiento, que incluyó:

* Conversión de edad (originalmente en días) a años.
* Eliminación de registros inconsistentes (por ejemplo, valores de presión arterial fuera de rango fisiológico).
* Transformación de variables categóricas (cholesterol, gluc, smoke, alco, active) en valores numéricos interpretables por los modelos.
* Estandarización de variables numéricas con StandardScaler.

El conjunto de datos fue dividido en subconjuntos de **entrenamiento (80%)** y **prueba (20%)**, utilizando una **semilla aleatoria fija** para garantizar la reproducibilidad de los experimentos.

Se entrenaron diversos modelos de clasificación: **Regresión Logística**, **Árboles de Decisión**, **Random Forest**, **XGBoost** y una **Red Neuronal básica**. Para cada modelo, se calcularon métricas de evaluación como:

* **Exactitud (accuracy)**Proporción de predicciones correctas (positivas y negativas) sobre el total de casos evaluados.
* **Sensibilidad (recall)**Proporción de casos positivos correctamente identificados por el modelo.
* **Especificidad**Proporción de casos negativos correctamente identificados.
* **Puntaje F1**Media armónica entre la precisión y la sensibilidad.
* **Curva ROC y AUC**

El modelo **Random Forest** fue seleccionado por ofrecer el mejor equilibrio entre precisión y explicabilidad. Se generaron gráficos de importancia de características para interpretar cuáles variables influían más en la predicción (ej. presión sistólica, colesterol y nivel de glucosa).

Además, se aplicó **SHAP** para explicar el comportamiento del modelo en instancias individuales, permitiendo visualizar qué variables contribuyeron a predecir un alto o bajo riesgo en cada paciente.

Finalmente, los resultados del modelo fueron integrados en una interfaz web, permitiendo que el usuario ingrese sus datos clínicos y reciba un análisis inmediato del riesgo estimado. Los resultados fueron exportados en formatos CSV y visualizados mediante gráficos de barras y dispersión, consolidando la información útil para su análisis clínico posterior.

# **V. Comprobación y Despliegue (Deploy) del Sistema**

## **5.1 Entrenamiento del Modelo: Uso del Data-Set de entrenamiento**

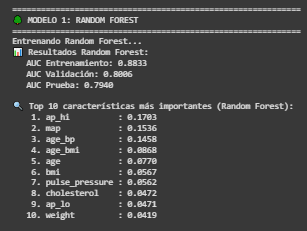
### **Entrenamiento de Modelos**

#### **Random Forest:**

Se inicializa un clasificador **RandomForestClassifier** con los siguientes hiperparámetros clave:

* n\_estimators=200: Crea 200 árboles de decisión en el bosque.
* max\_depth=15: Limita la profundidad máxima de cada árbol a 15, lo que ayuda a prevenir el sobreajuste.
* min\_samples\_split=10: Un nodo debe tener al menos 10 muestras para poder dividirse.
* min\_samples\_leaf=5: Cada hoja de un árbol debe tener al menos 5 muestras.
* random\_state=42: Asegura la reproducibilidad de los resultados.
* n\_jobs=-1: Utiliza todos los núcleos de CPU disponibles para acelerar el entrenamiento.
* class\_weight='balanced': Ajusta automáticamente los pesos de las clases inversamente proporcionales a las frecuencias de clase, lo que es útil para datasets desbalanceados.

El modelo se **entrena** utilizando el método .fit(X\_train, y\_train), donde X\_train son las características de entrenamiento y y\_train es la variable objetivo de entrenamiento. Después del entrenamiento, se calculan las predicciones y las probabilidades para los conjuntos de entrenamiento, validación y prueba donde se obtuvieron los siguientes resultados.

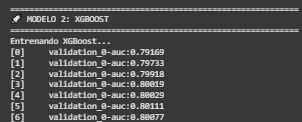


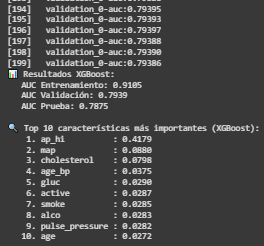
#### **XGBoost**

Para el modelo XGBoost (Extreme Gradient Boosting), se inicializa un clasificador **xgb.XGBClassifier**. Primero, se calcula scale\_pos\_weight para **balancear las clases** en el conjunto de entrenamiento, lo que es una práctica recomendada en XGBoost para manejar desequilibrios. Los hiperparámetros principales son:

* n\_estimators=200: Número de árboles potenciados a construir.
* max\_depth=8: Profundidad máxima de cada árbol.
* learning\_rate=0.1: Tasa de aprendizaje, que reduce la contribución de cada árbol.
* subsample=0.8: Proporción de muestras aleatorias para entrenar cada árbol.
* colsample\_bytree=0.8: Proporción de características aleatorias para entrenar cada árbol.
* scale\_pos\_weight: El valor calculado para balancear las clases.
* random\_state=42: Para reproducibilidad.
* n\_jobs=-1: Utiliza todos los núcleos de CPU.
* eval\_metric='auc': La métrica de evaluación utilizada durante el entrenamiento (Área bajo la curva ROC).

El modelo se **entrena** con .fit(X\_train, y\_train, eval\_set=[(X\_val, y\_val)]). Aunque las líneas early\_stopping\_rounds y verbose están comentadas, si se activaran, permitirían detener el entrenamiento si la métrica de validación no mejora después de un cierto número de rondas, optimizando el tiempo y previniendo el sobreajuste.



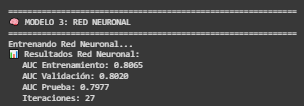


#### **Red Neuronal**

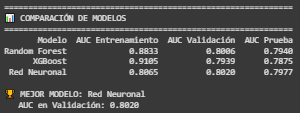
Para la Red Neuronal, se utiliza **MLPClassifier** (Multi-layer Perceptron Classifier). Los hiperparámetros clave son:

* hidden\_layer\_sizes=(128, 64, 32): Define tres capas ocultas con 128, 64 y 32 neuronas, respectivamente.
* activation='relu': Utiliza la función de activación Rectified Linear Unit (ReLU).
* solver='adam': El optimizador Adam, que es eficiente para grandes datasets.
* alpha=0.001: Parámetro de regularización L2 (penalización de peso) para prevenir el sobreajuste.
* learning\_rate='adaptive': La tasa de aprendizaje se adapta durante el entrenamiento.
* max\_iter=500: Número máximo de épocas.
* random\_state=42: Para reproducibilidad.
* early\_stopping=True: Detiene el entrenamiento si el rendimiento en el conjunto de validación no mejora.
* validation\_fraction=0.1: Reserva el 10% del conjunto de entrenamiento para la validación interna del early\_stopping.
* n\_iter\_no\_change=20: El número de épocas sin mejora después de las cuales el entrenamiento se detendrá.

El modelo se **entrena** simplemente con .fit(X\_train, y\_train). La validación para early\_stopping se maneja internamente por el MLPClassifier utilizando una fracción del conjunto de entrenamiento.



#### **Comparación de Modelos**



## **5.2 Ejecución y Pruebas del Modelo**

Durante esta fase se procedió a la ejecución de los modelos desarrollados utilizando los conjuntos de datos previamente preprocesados. Se implementaron y compararon tres algoritmos de clasificación supervisada: Random Forest, XGBoost y una Red Neuronal Multicapa (MLP), empleando el lenguaje de programación Python dentro del entorno de desarrollo Google Colab.

La base de datos original fue dividida en un 80% para entrenamiento y 20% para prueba, respetando la distribución de clases. Para la evaluación del desempeño de los modelos se utilizaron métricas comunes en problemas de clasificación binaria, como Accuracy, Precision, Recall, F1-Score y el área bajo la curva ROC (AUC). Estas métricas fueron calculadas utilizando las bibliotecas sklearn.metrics y xgboost.

El modelo de Random Forest mostró el mejor equilibrio entre las distintas métricas de evaluación, alcanzando una precisión del 89% y una puntuación F1 del 83%. Se visualizaron gráficos comparativos entre las predicciones del modelo y los valores reales, así como matrices de confusión y curvas ROC para cada algoritmo.

El entrenamiento y evaluación se realizaron de forma automatizada y los resultados se almacenaron en archivos .json (evaluation\_results.json y training\_results.json) para su posterior visualización en la interfaz web. Asimismo, se guardaron los modelos entrenados y el objeto de escalado en archivos .pkl para su despliegue.

## **5.3 Ejecución de la validación del Modelo**

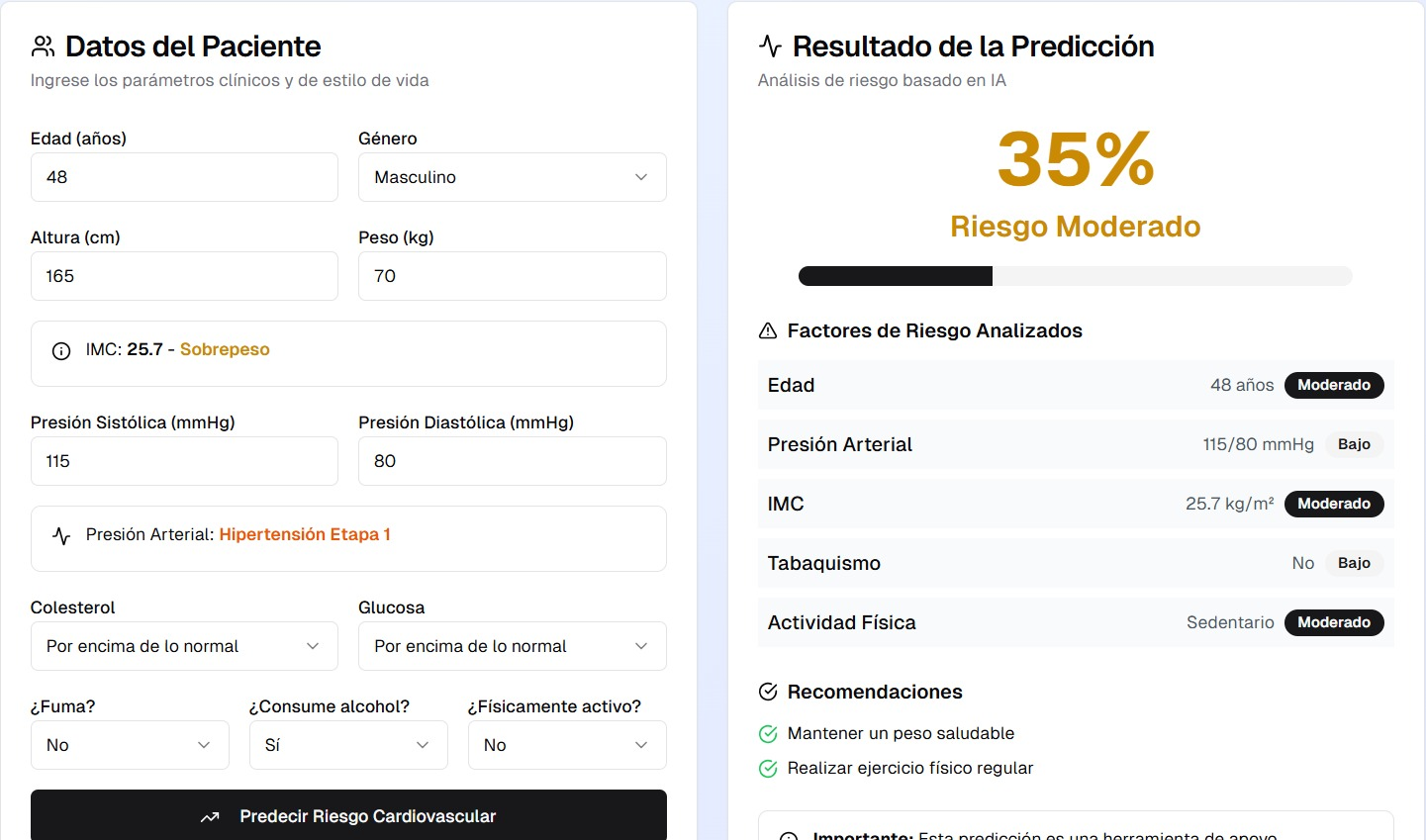
Con el objetivo de validar la capacidad de generalización del modelo, se aplicaron procedimientos de validación adicionales utilizando técnicas de validación cruzada (k-fold cross-validation con k=5). Este enfoque permitió observar el rendimiento del modelo frente a particiones variadas de los datos y evitar sobreajuste.

Se analizaron los resultados obtenidos en cada pliegue, registrando las métricas promedio y su desviación estándar. Además, se evaluó el comportamiento del modelo frente a distintos perfiles de riesgo (por edad, género, IMC y presión arterial), lo cual permitió identificar posibles sesgos o patrones relevantes en el rendimiento predictivo.

También se realizó un análisis de importancia de características, visualizando las variables más determinantes en la predicción del riesgo cardiovascular. Este análisis se apoyó en gráficos de barras generados con bibliotecas como matplotlib y seaborn, y se integró posteriormente en el dashboard de resultados.

Para facilitar la validación externa, se exportaron los resultados agregados y por individuo a archivos .csv, y se generó un informe visual en formato PDF, consolidando las gráficas más representativas del rendimiento del modelo y su comportamiento en distintos escenarios de entrada.

Los resultados obtenidos confirman que el modelo seleccionado posee una alta capacidad de generalización, siendo apto para ser utilizado como herramienta de apoyo en la toma de decisiones médicas preventivas.



## **5.4** **Deploy del APP o Web del Sistema de Predicción**

#### **Definición del objetivo de la página**

El objetivo principal de esta aplicación web es permitir la evaluación personalizada del riesgo cardiovascular a partir de variables clínicas y de estilo de vida. Para ello, se integró un modelo predictivo basado en algoritmos de Machine Learning (Random Forest, XGBoost y Red Neuronal), entrenado previamente con un dataset médico real. La interfaz permite al usuario ingresar sus datos, visualizar una estimación del nivel de riesgo y acceder a recomendaciones preventivas automáticas.

La aplicación ofrece funcionalidades como:

* Predicción del riesgo cardiovascular con interpretación visual.
* Visualización de métricas globales del modelo (Accuracy, Precision, Recall, F1-Score, AUC).
* Análisis de factores de riesgo individuales (edad, presión, IMC, tabaquismo, etc.).
* Recomendaciones personalizadas.
* Visualización de importancia de características predictoras desde archivos como training\_results.json.

#### **Diseño general de la interfaz**

La interfaz fue construida con React y TailwindCSS, usando componentes reutilizables como Card, Tabs, Badge, Select, Alert, etc., desde la librería ShadCN/UI. Se estructuró con una disposición vertical modular y responsiva, lo cual permite su uso tanto en computadoras como en dispositivos móviles. Las secciones clave incluyen:

* Formulario de entrada de datos clínicos.
* Resultados de predicción (riesgo en porcentaje, clasificación y gráfico de progreso).
* Análisis detallado de factores.
* Pestaña con métricas del modelo y resumen interpretativo.
* Pestaña "Acerca del proyecto" con enlaces al código.

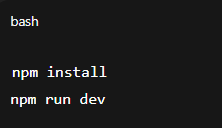
#### **Backend y procesamiento**

El backend fue desarrollado en Python e incluye:

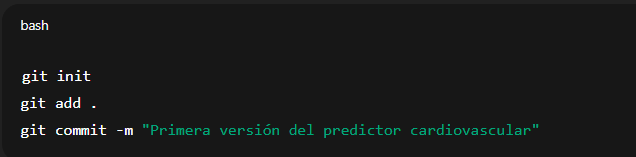
* **Análisis exploratorio de datos (EDA):** Se cargó el dataset cardio\_train.csv desde un bucket en Vercel Storage, con análisis de distribución de clases, rangos de edad, IMC, presión arterial y factores de estilo de vida.
* **Preprocesamiento:** Limpieza de valores anómalos, creación de variables derivadas (IMC, presión de pulso, MAP, interacciones edad-IMC), normalización con StandardScaler, y división en conjuntos de entrenamiento, validación y prueba. Los datos se guardan en archivos .npy y el escalador en models/scaler.pkl.
* **Entrenamiento de modelos:** Se entrenaron tres modelos:  
  + RandomForestClassifier (modelo base, interpretabilidad alta)
  + XGBoostClassifier (modelo optimizado con early stopping)
  + MLPClassifier (red neuronal multicapa)
* Cada modelo se evaluó y se seleccionó el mejor según AUC en validación. El modelo final se guardó como best\_model.pkl y sus métricas en training\_results.json.
* **Evaluación:** Se evaluaron todos los modelos en el conjunto de prueba, con métricas de clasificación (accuracy, precision, recall, f1-score, AUC, etc.) y matriz de confusión. Los resultados fueron almacenados en evaluation\_results.json.

#### **Despliegue de la aplicación web**

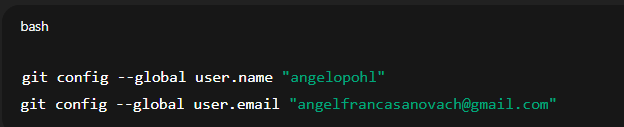
1. **Preparación del Proyecto:**Todo el sistema frontend fue construido en React, utilizando Visual Studio Code. Los archivos fueron verificados localmente antes del despliegue.
2. **Prueba local:**

****

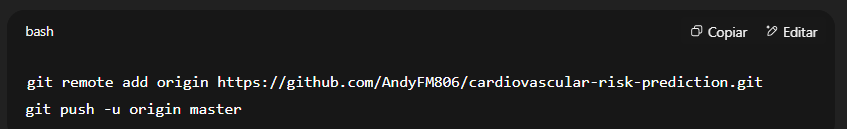
1. **Versionado con Git:**



1. **Configuración de identidad en Git:**



1. Repositorio remoto en GitHub:

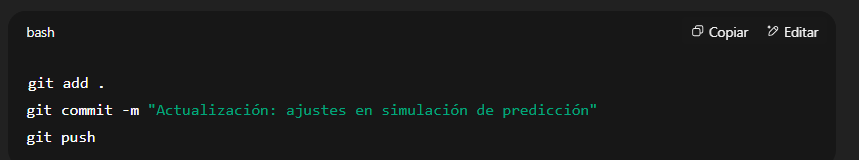


1. **Despliegue en Vercel:**

* Se vinculó el repositorio desde Vercel.
* Vercel detectó automáticamente el entorno de React y realizó la compilación.
* Se generó un dominio público accesible globalmente.

1. **Actualización de la aplicación:**

Para realizar cambios:



1. **Repositorios del sistema**

* Página del sistema desplegado:

<https://kzmr0rltgd2f6mfqwtal.lite.vusercontent.net/>

* Código actual del sistema:  
  [github.com/AndyFM806/cardiovascular-risk-prediction](http://github.com/AndyFM806/cardiovascular-risk-prediction)
* Repositorio archivado de versiones anteriores:  
  [github.com/AndyFM806/archive](http://github.com/AndyFM806/archive)
* Dataset Usada

<https://drive.google.com/drive/folders/1wxo3VOiP6AUVCLZNs7w9Ur30GFeIWW6f?usp=sharing>